Guaranteeing Feasible Outcomes in Judgment Aggregation

MSc Thesis *(Afstudeerscriptie)*

written by

Rachael H. Colley
(born October 21st, 1994 in Grantham, United Kingdom)

under the supervision of **Dr Ulle Endriss**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:	**Members of the Thesis Committee:**
September 26th, 2019	Dr Ekaterina Shutova (Chair)
	Dr Ulle Endriss (Supervisor)
	Prof Dr Davide Grossi
	Dr Ronald de Haan

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Abstract

In this thesis, we identify properties which guarantee consistent outcomes in a model of judgment aggregation, called *binary aggregation with rationality and feasibility constraints*. We consider an outcome to be consistent when we can guarantee that the outcome will abide by with the feasibility constraint when all voters provide a judgment that is consistent with the rationality constraint. In order to guarantee feasible outcomes, we take inspiration from the formula-based model of judgment aggregation and translate both properties and the consistency results which follow from them, to our model. We translate types of *agenda properties* and *domain restrictions* to our setting, in particular the (k-)median property and value restriction, respectively. Following this, we recreate the corresponding consistency results, guaranteeing feasible outcomes on rational profiles.

In turn, we study the computational complexity of problems related to the median property and value restriction, as well as their binary aggregation counterparts. Our results support the claim that they are complete for a class at least as hard as $\mathbf{coNP}$, and no harder than Π_2^p.

Contents

Acknowledgements

The first person I must thank for this thesis is my supervisor, Ulle. You are so incredibly patient, and it is this patience that has allowed me to grow as a researcher. Your attention to detail made this thesis something that I am proud of. Thank you.

I must also thank the ILLC's COMSOC group. Ronald, your enthusiasm for computational complexity rubbed off on me, and your generosity with your time allowed me to grow in an area I previously had not encountered. I have to thank Sirin and Zoi for introducing me to the field of computational social choice, and since then you have always supported me. Sirin, thank you for your constant guidance and ability to make me laugh.

To all of the friends I have made through the Master of Logic. I cannot name them all here, but I will give some honourable mentions.

To my housemates (including Charles the cat), you all made our space so comfortable and special. I will miss our logic family dinners the most.

Jack, I am so grateful that we were writing our theses on related topics. Your sound judgment and clarity helped me so much.

The women of the MoL had a special role in these past two years. Mina, thank you for your continual encouragement. Jelly, thank you for always listening to me in the hall way. Lisa, thank you for your constant kindness. Maria, thank you for always making me laugh (and not being a MoL student). Robin, thank you for inventing the Square Horse. Yvette, thank you for being my home away from home.

To my mother, brother and grandfather, thank you for believing in me, despite only roughly knowing what I do. Your unconditional support makes this all a lot easier.

Jamie, thank you. No words can describe how much you support me and any life choice I make. Thank you for helping me succeed.

Finally, the person I have to thank the most, is my father. He gave me that final push to start this amazing journey in Amsterdam.

Chapter 1

Introduction

This thesis is concerned with guarantee feasible outcomes in binary aggregation with rationality and feasibility constraints. This is a model of judgement aggregation, a field that has its roots in social choice theory. Social choice theory aims to model collective decision making, such as groups of friends deciding where to go for dinner, a national referendum, or a group of experts deciding on the best course of action. With social choice theory, problems of consistency can arise and one of its aims is to avoid inconsistent outcomes by restricting the input in some way. Judgment aggregation formally models the aggregation of a group of agents' individual opinion on a given situation. It usually manifests in such a way that we can see the collective decision-making process as a voting procedure between agents.

In the next section, we motivate this thesis. Following this, we will go on to describe the focus of the remaining chapters of this thesis.

1.1 Motivation for this Thesis

Consistent outcomes in judgment aggregation do not come about organically. There are situations that occur where, even though each agent's judgment is consistent, the outcome is not. An example of this is the *doctrinal paradox* (Pettit, 2001), leading to situations depicted in Table 1.1.

	a	b	$a \wedge b$
Voter 1	✓	✓	✓
Voter 2	✓	×	×
Voter 3	×	✓	×
Majority	✓	✓	×

Table 1.1: The Doctrinal Paradox (Pettit, 2001)

In Table 1.1, we see that although each voter gives a consistent judgment, the outcome is inconsistent. Specifically, each agent either rejects $a \wedge b$ or accepts both a and b. However, the majority outcome accepts a and b, yet rejects $a \wedge b$.

One aim of judgment aggregation is trying to avoid these inconsistent outcomes. This can be done by restricting the types of profiles given to the aggregation rule, or the types of agendas which we allow.

We will be using a different model of judgment aggregation called binary aggregation with rationality and feasibility constraints (Endriss, 2018). In this model, agents vote on independent issues. Therefore, issues are no longer interconnected, and no problems of consistency can arise. With the addition of constraints to the model, we can impose a logical connect between the issues in a given situation (Grandi, 2012a).

Our first constraint is the rationality constraint, this relates to the rationality conditions in judgment aggregation, i.e. that their judgments have to be complete and consistent. The rationality constraint reflects the rationality of the agents in terms of their capacity to make consistent choices. For example, choosing a single food item and a single drink item at a meal, or supporting all of the policies of your political party.

The second constraint is the feasibility constraint. It relates to the expectations of the collective decision. In both models judgment aggregation, we have expectations of the outcome. In the formula-based model of judgment aggregation, we want the outcome to be consistent. In our model, we want the outcome to abide by this feasibility constraint. We can think of this constraint as expressing the practical necessities of the situation, such as abiding by the local council's budget, or only admitting the correct number of students to a course.

Binary aggregation with rationality and feasibility constraints allows us to have different conditions on the agents' judgments and what we expect on the outcome of the rule. This makes it more expressive than the formula-based model of judgment aggregation.

Our focus will be finding ways in which we can *guarantee feasible outcomes*, given that every agent's judgment is rational. We will draw on the consistency results from the formula-based model of judgment aggregation to guarantee consistent outcomes. We translate these ways of guaranteeing consistent outcomes to our model, in order to guarantee feasible outcomes. Furthermore, we will be looking at the relationship between these two constraints, and from this find when a rule can guarantee feasible outcomes on rational profiles. In the next section, we will outline how this will be achieved in each chapter.

1.2 About this Thesis

In what follows, we will give an overview of the chapters of this thesis.

Chapter 2

In this chapter, we will introduce binary aggregation with rationality and feasibility constraints. Here we will also give a small introduction to the more widely used model of judgment aggregation where the agents vote on an interconnected list of formulas. Furthermore, we will show a translation from judgment aggregation to our binary aggregation setting with constraints (Grandi and Endriss, 2013). Then in Sections 2.3 and 2.5, we begin to introduce the work already carried out in binary aggregation with either a single constraint or rationality and feasibility constraints, respectively.

Chapter 3

In this chapter, we focus on which quota rules can give feasible outcomes on rational profiles for a given pair of constraints. This chapter follows on from the work of Grandi and Endriss (2013), where the rationality and feasibility constraints are logically equivalent. Furthermore, this single constraint is restricted to be a single clause. The aim of this work was to find when quota rules can guarantee feasible outcomes with respect to a single clause, assuming that the profile was rational with respect to the same single clause as well. We extend this to look at two single clauses, one for deciding a rational input and the second determining a feasible outcome. We build upon this, and look at quota rules which can guarantee feasible outcomes on rational profiles when the constraints can have any finite number of clauses.

Chapter 4

In this chapter, we look to judgment aggregation for ways we can guarantee consistent outcomes by restricting the domain of inputs. We then translate them to our setting. In judgment aggregation, one way this is done is through *domain restriction*, only allowing profiles with some specific characteristic. We introduce domain restrictions in judgment aggregation, along with the motivation of focusing on value restriction from the remainder of chapter. We translate the property of value restriction to both the single-constraint and two-constraint settings of binary aggregation. Then, we go on to show that this ensures feasible outcomes under the majority rule. However, this result comes with the extra assumption that there has to be an odd number of agents. In light of this, we extend the notion of a profile being value-restricted, to being negatively value-restricted. Then we show that this guarantees feasible outcomes for any number of agents.

Chapter 5

In this chapter, we will look at the computational complexity of three problems in depth which relate to the properties explored in Chapters 3 and 4. These problems are: VALUERESTRICTED, BINVALUERESTRICTED and PAIRSIMPLE. VALUERESTRICTED, checks if a formula-based profile being value restricted with respect to the agenda. BINVALUERESTRICTED is the binary aggregation analogue of VALUERESTRICTED and it checks if a binary profile is value-restricted with respect to a pair of constraints. The final problem we will look at is PAIRSIMPLE, which checks if a pair of constraints has the property of being simple. The judgment aggregation analogue of this problem, MP, is a Π_2^p-complete problem (Endriss et al., 2012). For each of the three problems that we will inspect, we will see that they are all **coNP**-hard and that they have membership in Π_2^p. From these results, we conclude that the problems must be complete for a class at least as hard as **coNP**, however, no harder than the class Π_2^p.

Chapter 2

Binary Aggregation with Constraints

In this chapter, we outline the model of binary aggregation which will be used throughout this thesis. We give definitions and the notation required for the results and proofs in this thesis. Following this, we will outline the existing work in binary aggregation with constraints.

In Section 2.1, we will lay out the model of binary aggregation. A model which finds a collective outcome of a group of voters, where each voter gives a yes or no answer to a series of binary issues. In Section 2.2, we will introduce the more commonly used formula-based model of judgment aggregation. In this model voters select a complete and consistent subset of the items of the agenda, it was originally laid out by List and Pettit (2002). Here we will also introduce some of the central problems and results of judgment aggregation that we will be touching upon. Following this introduction, we will show the connection between the two models by giving a translation from judgment aggregation to binary aggregation (Grandi, 2012b).

In Section 2.3, we will lay out the work of Grandi and Endriss (2013). This focuses on *integrity constraints*, which dictates what is rational for the agents to vote in accordance with. If the aggregation rule is *collectively rational* with respect to the constraint, then the outcome abides by the integrity constraint as well. These integrity constraints mirror the assumption that judgments and outcomes are complete and consistent. Grandi and Endriss translate some of the solutions to guarantee consistent outcomes in judgment aggregation to the binary aggregation setting. For example, they translate the median property, which guarantees consistent results under the majority rule (Nehring and Puppe, 2007), and the k-median property, which guarantees consistent results under some quota rules, dependent on the value of k (Dietrich and List, 2007).

In Section 2.4, we will introduce prime implicates. We use prime implicates to relate pairs of constraints. Their use allows us to examine the relationship between constraints in a more fine grained way than only semantic entailment. Prime implicates will be used mainly in the rationality and feasibility model for this purpose.

In Section 2.5, we see an extension of the work described in Section 2.3. Now we consider two constraints instead of a single integrity constraint, namely the rationality and feasibility constraints (Endriss, 2018). The rationality constraint dictates what is individually rational in a given context, whereas, the feasibility constraint reflects what we expect from the outcome of the rule. We see that using rationality and feasibility constraints is more expressive than using a single integrity constraint. As what is expected of the agents is not necessarily the same as what we expect of the outcome. We will introduce some of the terminology in order to understand one of the main results by Endriss (2018). This result is an analogue of the result by Nehring and Puppe (2007) regarding the median property in judgment aggregation. Furthermore, it also extends the Grandi and Endriss (2013) result from the single-constraint case, to the two-constraint case.

2.1 The Model

In this section, we will describe the model of binary aggregation. In the formula-based model of judgment aggregation our agenda contains well-formed formulas. Whereas, in binary aggregation our agenda contains independent issues, which can be thought of as non-negated propositional variables. Therefore, in binary aggregation, agents vote either for or against a proposition, this can be thought of as a yes/no response to a certain issue. Our notation of binary judgement aggregation will follow that of Endriss (2018); Grandi (2012a); Grandi and Endriss (2013).

2.1.1 Judgments and Binary Aggregation

We let $\mathcal{N}$ be the set of n voters, such that $\mathcal{N} = \{1, \ldots, n\}$, and we assume that $n > 1$. Unless specifically stated. we do not make an assumption on whether n is odd or even. Moreover, we will use voter and agent interchangeably.

An agenda Φ is a set of independent issues. We usually denote the issues of the agenda as letters of the Greek alphabet, such as φ or ψ. We define a voter's *judgment*, B_i as such: for a voter $i \in \mathcal{N}$, $B_i : \Phi \to \{0, 1\}$. Here we see that 0 corresponds to answering 'no' or a rejection, whereas, 1 represents answering 'yes' or an acceptance of an issue. We call the collection of all the voters' judgments a *profile*, denoted by $\boldsymbol{B} \in (\{0,1\}^{\Phi})^n$.

To be able to speak about certain subsets of $\mathcal{N}$, we introduce the following notation. We let $\mathcal{N}_{\varphi}^{\boldsymbol{B}}$ denote the set of voters who vote for φ in the profile $\boldsymbol{B}$. Therefore, $|\mathcal{N}_{\varphi}^{\boldsymbol{B}}|$ will be used to denote the number of voters who voted for φ in the profile $\boldsymbol{B}$.

Moreover, we need to address how we aggregate the voters' judgments in order to gain a collective judgment. An aggregation rule, F, is a function which takes in profiles, and outputs a judgment. We define it formally as such:

$$F : (\{0,1\}^{\Phi})^n \to \mathcal{P}(\{0,1\}^{\Phi}) \backslash \{\emptyset\}$$

Note that the rule will return some non-empty set of judgments. We now go on to define some aggregation rules in binary aggregation which we will be focusing on in this thesis.

A quota is a function that takes an issue of the agenda and gives a real number. This number dictates how many votes are required for the issue to be accepted by the quota rule.

Definition 2.1 (Quota)**.** A *quota* is a function such that:

$$q : \Phi \to [0, n+1]$$

Thus, we denote the quota of an issue φ as $q(\varphi)$. Using Definition 2.1, we now define a quota rule.

Definition 2.2 (Quota Rule, Grandi, 2012a)**.** A *quota rule* accepts an issue $\varphi \in \Phi$ if and only if $|\mathcal{N}_{\varphi}^{\boldsymbol{B}}| \geq q(\varphi)$, otherwise φ is rejected by the rule.

Observe that quota rules in binary aggregation do not treat the acceptance or rejection of an issue in the same way. In essence, we can think of there being different quotas for issues being accepted or rejected, with an issue requiring $q(\varphi)$ votes of for the issues or $n - q(\varphi)$ votes against it, respectively. A special case of the quota rules are uniform quota rules, where every issue of the agenda is assigned the same quota.

Definition 2.3 (Uniform Quota Rule)**.** A quota rule is a *uniform quota rule* if and only if q is a constant function, for all $\varphi \in \Phi$ we have that $q(\varphi) = c$.

The final aggregation rule we will introduce in this section is the strict majority rule.

Definition 2.4 (Strict Majority Rule)**.** The *strict majority rule* accepts an issue $\varphi \in \Phi$ if and only if $|\mathcal{N}_{\varphi}^{\boldsymbol{B}}| > \frac{n}{2}$.

We will refer to *the strict majority rule* as simply *the majority rule* in this thesis. Observe that when n is odd, the support required for an issue and its negation to be consistent with the outcome is the same. However, when n is even, an issue requires $\frac{n}{2} + 1$ votes to be accepted. Whereas, its negation requires $\frac{n}{2}$ votes against the issue for it to be rejected.

2.1.2 Constraints

In this subsection, we will add constraints to our model. With this we create connections between the independent issues of the agenda.

The propositional language used to create the constraints is $\mathcal{L}(\Phi)$. This is the set of well-formed formulas built using the usual connectives: $\wedge$, $\vee$, $\neg$, $\rightarrow$ and the propositional variables from Φ. We will usually denote the constraints as Γ, or Γ'. We will think of the constraints as being in *conjunctive normal form* (CNF), a conjunction of clauses. We define a clause as such:

Definition 2.5 (Clause)**.** A *clause*, π, is a disjunction of literals.

We call a clause *empty* if it does not contain any propositional variables.

As all formulas can be rewritten in CNF, we will assume throughout this thesis that our constraints are in CNF. Therefore, a formula in CNF with s clauses can be written as $\Gamma = \bigwedge_{i=1}^{s} \pi_i$. Next, we move onto being able to speak about the issues which appear in the constraints, namely, the *variable function*.

Definition 2.6 (Variable Function)**.** Let $\Gamma \in \mathcal{L}(\Phi)$. We define the following function which returns a set of the issues which appear in Γ.

$$\text{Var} : \mathcal{L}(\Phi) \rightarrow \mathcal{P}(\Phi)$$

A voter's judgment, B_i, will satisfy a constraint $\Gamma \in \mathcal{L}(\Phi)$, when the constraint Γ evaluates to true under the same assignment entailed by the voter's judgment B_i. We denote this by $B_i \models \Gamma$. Following from the work of Endriss (2018), we will define this notion of satisfaction recursively as such:

- $B_i \models \varphi$ for an issue $\varphi \in \Phi$ if and only if $i \in \mathcal{N}_{\varphi}^{\boldsymbol{B}}$.
- $B_i \models \neg\Gamma$ if and only if it is not that case that $B_i \models \Gamma$
- $B_i \models \Gamma \wedge \Gamma'$ if and only if $B_i \models \Gamma$ and $B_i \models \Gamma'$
- $B_i \models \Gamma \vee \Gamma'$ if and only if $B_i \models \Gamma$ or $B_i \models \Gamma'$
- $B_i \models \Gamma \rightarrow \Gamma'$ if and only if $B_i \not\models \Gamma$ or $B_i \models \Gamma'$

When looking at the relationship between pairs of constraints, we usually to see if a semantic entailment holds between the pair. We say that $\Gamma \models \Gamma'$, if for all B_i such that $B_i \models \Gamma$, then it is also the case that $B_i \models \Gamma'$. However, a more concrete way of thinking about this is by introducing the following notation.

Definition 2.7 (Models of a Constraint). Let Γ be a constraint such that $\Gamma \in \mathcal{L}(\Phi)$. We denote the models which satisfy a constraint as $\text{Mod}(\Gamma)$. We define this formally as such:

$$\text{Mod}(\Gamma) = \{B \in \{0,1\}^{\Phi} \mid B \models \Gamma\}$$

We can denote $\Gamma \models \Gamma'$ as $\text{Mod}(\Gamma) \subseteq \text{Mod}(\Gamma')$, that is that all models of Γ are also models of Γ'.

Lastly, we need to define the notion of consistency in this setting. In the existing literature, this relies on two notions relating to the input and output of the aggregation rule (Endriss, 2018; Grandi and Endriss, 2013; Grandi, 2012a). Although, only Endriss (2018) uses the terminology of *rationality* and *feasibility* constraints, we will stick to this terminology to avoid confusion, as all of the existing work can be thought of in these terms.

Definition 2.8 (Γ-rational Profile, Endriss, 2018). A profile $\boldsymbol{B} \in (\{0,1\}^{\Phi})^n$ is Γ-rational if for all $i \in \mathcal{N}$ $B_i \models \Gamma$ (alternatively, if $\boldsymbol{B} \in \text{Mod}(\Gamma)^n$).

Next we want to define what is a feasible outcome.

Definition 2.9 (Γ'-feasible outcome, Endriss, 2018). An outcome $F(\boldsymbol{B})$ is Γ'-feasible if $F(\boldsymbol{B}) \models \Gamma'$ (alternatively $F(\boldsymbol{B}) \in \text{Mod}(\Gamma')$).

Finally, our notion of consistency is when the two previous definitions coincide. We refer to this as *guaranteeing a feasible outcome on rational profiles*, which we define it as such:

Definition 2.10 (Guaranteeing Γ'-feasible outcomes, Endriss, 2018)**.** An aggregation rule F is said to *guarantee* Γ'-feasible outcomes on Γ-rational profiles, if for every profile $\boldsymbol{B} \in \text{Mod}(\Gamma)^n$ it is the case that $F(\boldsymbol{B}) \in \text{Mod}(\Gamma')$.

2.2 From Judgment Aggregation to Binary Aggregation

In this section, we will take the more commonly used model of formula-based judgement aggregation and show that it can be translated to binary aggregation setting as described in Section 2.1. However, in order to do this, we will first introduce the formula-based model of judgment aggregation and some key results.

2.2.1 An Introduction to Judgment Aggregation

We will give the formula-based model of judgment aggregation introduced by List and Puppe (2009), which we will refer to as just *judgment aggregation*. We have a set of propositions which represent different items in a certain context. We will denote this set of propositions as $P = \{p_1, ..., p_t\}$.[1] We assume that this set of propositions is finite. From this set of propositions we can build well-formed formulas with the standard logical connectives: $\neg, \wedge, \vee$ and $\rightarrow$. We denote the upward closure of the set of propositions with the logical connectives as $\mathcal{L}(P)$, and it is from here that the judgment aggregation agenda is built.

Definition 2.11 (A formula-based agenda)**.** The agenda in formula-based model judgment aggregation, X is a set of issues, which is closed under complementation, such that $X \subseteq \mathcal{L}(P)$.

Npte that we will use X^+ to refer to just the positive (non-negated) propositions in X.

We will denote a judgment in the formula-based model with J (in binary aggregation we will denote the judgment as B). Therefore, a profile in judgment aggregation (which we will denote as $\boldsymbol{J}$) is a subset set of the agenda, $\boldsymbol{J} \subseteq X^n$. However, allowing for $\boldsymbol{J} \subseteq X^n$, without any further restrictions, means that the profiles could contain inconsistent or incomplete judgments. Therefore, in judgment aggregation we have conditions imposed on an agent's judgment, such as that they have to be:

- complete, i.e. for all $\alpha \in X^+$, either $\alpha \in J_i$ or $\neg\alpha \in J_i$, and;
- consistent, i.e. there exists a truth assignment that satisfies all $\alpha \in J_i$.

[1]Note that these are atomic propositions excluding the symbols of $\top$ and $\bot$. Thus, the propositions are not contradictions nor tautologies themselves.

We denote the set of judgments which abide by these rationality conditions by $\mathcal{J}(X)$. We define aggregation rules in judgment aggregation as such:

Definition 2.12 (Aggregation rules)**.** An aggregation rule is a function which takes a profile, where each judgment abides by the rationality conditions and gives a result which is a subset of the agenda:

$$F : \mathcal{J}(X)^n \to 2^X$$

We will not formally define specific aggregation rules, such as the majority rule and quota rules, in judgment aggregation. However, they work in an analogous way as in binary aggregation.

Now that we have a model of judgment aggregation we will look at some central results.

As we saw in the introduction, although all agents give a consistent judgment, there can be inconsistent outcomes, as in the doctrinal paradox (see Table 1.1). In the field of judgment aggregation, there have been strides to find conditions when the majority rule will provide consistent outcomes, one way in which this is done is by only allowing agendas which have the *median property*.

Definition 2.13 (Median Property, Nehring and Puppe, 2007)**.** An agenda X has the *median property* if and only if every minimally inconsistent subset of Φ has a size of at most two.

Nehring and Puppe (2007) found that the agenda having the median property is a necessary and sufficient condition for the the majority rule giving consistent outcomes.

Theorem 2.1 (Nehring and Puppe, 2007)**.** *The majority rule guarantees consistent outcomes if and only if the agenda has the median property.*

A similar problem concerning guaranteeing consistency, as in the doctrinal paradox, can occur when using quota rules. Therefore, there is a variation of the median property to avoid inconsistencies under quota rules, namely the k-median property.

Definition 2.14 (k-Median Property, Dietrich and List, 2007)**.** An agenda X has the *k-median property* if and only if every minimally inconsistent subset of X has a size of at most k.

This definition leads to the following result by Dietrich and List (2007), as the necessary and sufficient conditions for consistent outcomes under the a certain class of quota rules.

Proposition 2.2 (Theorem $2a$, Dietrich and List, 2007)**.** A quota rule F gives consistent outcomes if and only if $\sum_{p \in Z} q(p) > n(|Z| - 1)$, for every minimally inconsistent subset Z of the agenda X.

There are other types of conditions which are used to avoid inconsistent outcomes in judgment aggregation. However, the existing work on binary aggregation with constraints has currently only focussed on agenda properties. Hence, we will introduce any further background knowledge when necessary.

2.2.2 Translation from Judgment Aggregation to Binary Aggregation

In this subsection, we will see the translation between judgment aggregation and binary aggregation with constraints (Grandi and Endriss, 2011; Grandi, 2012b). From the previous subsection, we can now show the translation from judgment aggregation to binary aggregation with constraints.

We start with the agenda. Take an agenda X from judgment aggregation. Then we can define an agenda of binary aggregation as $\Phi_X = \{\varphi_\alpha \mid \alpha \in X\}$, giving us a binary translation of the agenda X. Thus, if we have the formula-based agenda $X = \{p_1, \neg p_1, p_2, \neg p_2, p_1 \wedge p_2, \neg(p_1 \wedge p_2)\}$, then its translation to binary aggregation will be $\Phi = \{\varphi_{p_1}, \varphi_{\neg p_1}, \varphi_{p_2}, \varphi_{\neg p_2}, \varphi_{p_1 \wedge p_2}, \varphi_{\neg(p_1 \wedge p_2)}\}$.

Next we have to translate the domain of profiles from judgment aggregation to binary aggregation. For this we will introduce some notation. We let $Y \overset{\text{m.i.}}{\subseteq} Z$ denote that Y is a minimally inconsistent subset of Z. To find the translation of $\mathcal{J}(X)$, we need a translation of the rationality conditions, which we saw in the previous subsection. As formulated by Grandi (2012a, Section 3.2.2) we restrict the judgments in the profiles such that they abided by the following constraints:

- Completeness: $\varphi_\alpha \vee \varphi_{\neg\alpha}$ for all $\alpha \in X^+$
- Consistency: $\neg(\bigwedge_{\alpha \in S} \varphi_\alpha)$ for every $S \overset{\text{m.i.}}{\subseteq} X$.

The translation of completeness is clear, every agent has to vote for either φ_α or $\varphi_{\neg\alpha}$. This relates to the formula-based model, where the voter has to accept either α or $\neg\alpha$. The constraint hat represents consistency tells us that an agent cannot vote for all of the formulas of a minimally inconsistent subset of the agenda X, making their judgments consistent with respect to X.[2]

In our model of binary aggregation, the domain of profiles must abide by the constraint Γ_X, where:

$$\Gamma_X = (\bigwedge_{\alpha \in X^+} (\varphi_\alpha \vee \varphi_{\neg\alpha})) \wedge (\bigwedge_{\substack{\text{m.i.} \\ S \subseteq X}} \neg(\bigwedge_{\alpha \in S} \varphi_\alpha))$$

It is clear that a translation from judgment aggregation to binary aggregation holds, as the Γ_X-rational profiles will equate exactly to the judgments in $\mathcal{J}(X)$. Furthermore, it is possible to translate from 'binary ballots' to judgments (see Grandi (2012a, Section 6.2.2) for details). However, this will not be expanded upon here as we will only be concerned with the translation from judgment aggregation to binary aggregation.

2.3 Integrity Constraints

In this section, we will expand upon the work of Grandi and Endriss (2013). The aim of this work was to find consistent outcomes with respect to an integrity constraint which the voters all abide by. They call rules *collectively rational* when all voters abide by the constraint, the outcome abides by it as well (Grandi and Endriss, 2013). This mirrors trying to avoid inconsistent outcomes in judgment aggregation. The following example shows the connection between the two using the translation spelled out in the previous section.

Example 1. Three agents $\mathcal{N} = \{a_1, a_2, a_3\}$ vote on the following binary agenda, $\Phi = \{\varphi_\alpha, \varphi_{\neg\alpha}, \varphi_\beta, \varphi_{\neg\beta}, \varphi_{\alpha\wedge\beta}, \varphi_{\neg(\alpha\wedge\beta)}\}$, translated from the formula-based agenda $X^+ = \{\alpha, \beta, \alpha \wedge \beta\}$. Our constraint says that all judgments have to be complete and consistent with respect to the agenda X (the subscripts of our binary issues). Therefore, we have the following integrity constraint, $\Gamma = (\bigwedge_{\alpha \in X^+}(\varphi_\alpha \vee \varphi_{\neg\alpha})) \wedge (\bigwedge_{\substack{\text{m.i.} \\ S \subseteq X}} \neg(\bigwedge_{\alpha \in S} \varphi_\alpha))$. Now consider the profile depicted in Table 2.1.

We see that every voter abides by the integrity constraint Γ, as they give complete and consistent judgements with respect to X. Therefore, for all $i \in \mathcal{N}$, $B_i \vDash \Gamma$. However, the outcome does not abide by the integrity constraint Γ, as the outcome fails to be consistent with respect to X, therefore, $F(\boldsymbol{B}) \nvDash \Gamma$. Thus, the majority is

[2]As we are looking at minimally inconsistent subsets of the agenda, taking one item away from this set will mean that the set is consistent. Therefore, not voting for all of the items in the minimally inconsistent subset means that there must be at least one item that you didn't vote for. Thus, the judgment is consistent.

	φ_α	$\varphi_{\neg\alpha}$	φ_β	$\varphi_{\neg\beta}$	$\varphi_{\alpha\wedge\beta}$	$\varphi_{\neg(\alpha\wedge\beta)}$
a_1	✓	×	×	✓	×	✓
a_2	✓	×	✓	×	✓	×
a_3	×	✓	✓	×	×	✓
Majority	✓	×	✓	×	×	✓

Table 2.1: Translation of the doctrinal paradox

not collectively rational with respect to Γ. △

From this example, we see that in binary aggregation the same problems of inconsistent outcomes can arise. Grandi and Endriss appeal to some of the solutions in judgment aggregation in order to solve the analogue problems in binary aggregation with integrity constraints. One of their solutions is translating the median property to binary aggregation. In the previous example, we see that if all of the minimally inconsistent subsets of the agenda were of size at most two, then the clauses of Γ would also be of size at most two. Thus, Grandi and Endriss (2013, Proposition 1) prove that in binary aggregation the median property corresponds to the clauses of the integrity constraint having at most two literals each, leading us to the following proposition.

Proposition 2.3 (Proposition 1 Grandi, 2012b)**.** The majority rule is collectively rational with respect to clause with two literals.

Grandi and Endriss (2013) also translate the k-median property. They recreate the result by Dietrich and List (2007) (see Theorem 2.2 in this thesis) in their single-constraint setting. Grandi and Endriss translate this through a series of results leading to the following result.

Theorem 2.4 (Theorem 30, Grandi and Endriss, 2013)**.** *A quota rule is collectively rational with respect to a clause Γ with k literals if and only if the quotas of the issues which appear in Γ abide by*

$$\sum_{\varphi_j \in \mathit{Var}(\Gamma^-)} q(\varphi_j) + \sum_{\varphi_j \in \mathit{Var}(\Gamma^+)} (n - q(\varphi_j) + 1) > n(k-1),$$

or an issue in Γ either has a quota of 0 or $n+1$ if it appears positively or negatively, respectively.

This theorem is with respect to a single clause, instead of formulas with many clauses we described in Section 2.1. This is something that will be touched upon later in Section 3.2.

Note that we will refer to integrity constraints as the single-constraint case of binary aggregation. Furthermore, we will not use the notation of the single-constraint model, but that of rationality and feasibility model, as the former can be thought of in terms of the latter.

2.4 Prime Implicates

In this section, we will gain some understanding of prime implicates. We use this to relate constraints with multiple clauses. They allow us to inspect the constraints in a more in-depth way than just comparing their clauses when in CNF form or if there is a semantic entailment between the formulas.

The prime implicates of a formula are the formula's logically strongest clauses, and therefore, do not contain redundancies. The study of implicates and prime implicates can be found in the work of Inoue (1992) and Marquis (2000). A more detailed account of prime implicates can be found there. However, we shall be following the same notation used by Endriss (2018). First, we formally define an implicate.

Definition 2.15 (Implicate). Let $\Gamma \in \mathcal{L}(\Phi)$. A clause $\pi \in \mathcal{L}(\Phi)$ is an *implicate* of Γ if and only if $\Gamma \models \pi$ and π is not a tautology.

We see here that an implicate of a CNF formula can be a clause of the formula, describing part of the formula. Note that, as described by Tourret (2012), we exclude tautologies from being implicates. The definition of an implicate has no notion of strength nor a definite relationship between the clause and the formula. For example, consider the formula $\Gamma = (p_1 \vee p_2)$. An implicate of Γ is $\pi = (p_1 \vee p_2 \vee p_3)$. Here we see that although π is an implicate of Γ, it is not very informative with respect to Γ. Therefore, we define a stronger notion of an implicate to describe a given formula, we call this a prime implicate.

Definition 2.16 (Prime Implicate). Let $\Gamma \in \mathcal{L}(\Phi)$. A clause $\pi \in \mathcal{L}(\Phi)$ is a *prime implicate* of Γ if and only if:

- π is an implicate of Γ;
- and for every implicate π' of Γ, if $\pi' \models \pi$, then $\pi \models \pi'$ holds.

Definitions 2.15 and 2.16 are a reformulation of Definition 3.3 from Marquis (2000). To understand what a prime implicate is we will see an example.

Example 2. Consider the following formula, $\Gamma = (\varphi_1 \vee \varphi_2) \wedge (\neg\varphi_2 \vee \neg\varphi_3)$.

φ_1	φ_2	φ_3
✓	×	×
✓	×	✓
✓	✓	×
×	✓	×

Table 2.2: Satisfying truth assignments of $\Gamma = (\varphi_1 \vee \varphi_2) \wedge (\neg\varphi_2 \vee \neg\varphi_3)$

We can see that the prime implicates of Γ are $(\varphi_1 \vee \varphi_2)$, $(\neg\varphi_2 \vee \neg\varphi_3)$ and $(\varphi_1 \vee \neg\varphi_3)$. It is worth noting that each of these truth assignments also make each of the prime implicates true. Furthermore, if we remove an literal of a prime implicate, then one of these truth assignments would make it false. △

We can now use prime implicates to describe the strongest clauses of the rationality and feasibility constraints. Furthermore, with the following lemma, we can now use prime implicates to inspect the relationship between the constraints.

Lemma 2.5 (Marquis, 2000). *If* $\Gamma \models \Gamma'$ *is the case, then for every prime implicate* π' *of* Γ' *there exists a prime implicate* π *of* Γ *such that* $\pi \models \pi'$.

The proof of Lemma 2.5 can be found in the work of Inoue (1992, Theorem 4.7). The use of Lemma 2.5 allows us to make a connection between prime implicates of the two constraints.

2.5 Rationality and Feasibility Constraints

In this section, we will focus on binary aggregation with rationality and feasibility constraints, introduced by Endriss (2018). The integrity constrain model, explored in the previous section is a special case of this model[3]. The rationality constraint relates to the input, while the feasibility constraint relates to what we expect of the output, denoted by Γ and Γ', respectively.

In order to see how these constraints could be used, we will consider an example where agents decide what a budget should be spent on.

Example 3. Three agents ($\mathcal{N} = \{a_1, a_2, a_3\}$) are deciding which of the three projects should be funded. We denote $\Phi = \{\varphi_1, \varphi_2, \varphi_3\}$ as the set containing the three projects. The budget cannot fund all of the projects. Therefore, $\Gamma' = \bigvee_{\varphi_i \in \Phi} \neg\varphi_i$.

[3]Note that the single-constraint case is a special case of the rationality and feasibility model, where the two constraints are logically equivalent ($\models \Gamma \leftrightarrow \Gamma'$).

However, it is rational for voters to support at least one of the projects. Thus, $\Gamma = \varphi_1 \vee \varphi_2 \vee \varphi_3$. Using the majority rule, we can have a situation as in Table 2.3.

	φ_1	φ_2	φ_3
a_1	✓	✓	×
a_2	×	✓	✓
a_3	✓	×	✓
Majority	✓	✓	✓

Table 2.3: A Γ-rational profile without a Γ'-feasible outcome

In Table 2.3, we see that all agents have accepted one of the projects. Therefore, it is a Γ-rational profile. However, all three of the projects have been selected. Therefore, this is not a Γ'-feasible outcome. △

From this previous example we can observe two things:

First, we see that the rationality and feasibility model of binary aggregation is much more expressive than the integrity constraint model. In the previous example, it might not be rational for voters to think about the strict monetary outcome when deciding which items of Φ should be supported. Whereas, this is not possible in the single-constraint model.

Second, we can see in this example that we are not guaranteed Γ'-feasible outcomes. Thus, the majority rule cannot guarantee consistent outcome with respect to Γ and Γ'. One way to be able to guarantee feasible outcomes on rational profiles is via *simple* clauses.

Definition 2.17 (Simple Clause)**.** A clause is *simple* if and only if it is logically equivalent to a clause with at most two literals.

Endriss (2018) extends this definition of a simple clause to a pair of formulas being simple.

Definition 2.18 (A simple pair of formulas Endriss, 2018)**.** A pair of formulas $(\Gamma, \Gamma') \in \mathcal{L}(\Phi)^2$ is *simple*, if for every non-simple prime implicate π' of Γ' there exists a simple prime implicate π of Γ such that $\pi \models \pi'$ holds.

To unpack this definition, we will now look at an example of a simple pair of formulas, where both formulas themselves are not simple.

Example 4. Consider the following pair of formulas: $\Gamma = (\varphi_1 \vee \varphi_2) \wedge (\varphi_3 \vee \varphi_4 \vee \varphi_5)$ and $\Gamma' = (\varphi_1 \vee \varphi_2 \vee \varphi_3)$. Here we see that neither Γ nor Γ' are simple formulas as they contain clauses which have more than two literals. However, the pair is simple. Take the only non-simple prime implicate of Γ', namely $(\varphi_1 \vee \varphi_2 \vee \varphi_3)$. We see that there exists a simple prime implicate of Γ, namely $(\varphi_1 \vee \varphi_2)$, which entails $(\varphi_1 \vee \varphi_2 \vee \varphi_3)$. Therefore, the pair of formulas (Γ, Γ') is simple. △

Next, we see the result from Endriss (2018, Theorem 2), giving the conditions to guarantees feasible outcomes under the majority rule.

Theorem 2.6 (Endriss, 2018)**.** *The Majority Rule guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if $\Gamma \vDash \Gamma'$ and (Γ, Γ') is simple.*

The crux of the application of this theorem is that the prime implicates of the feasibility constraint will always be entailed by a simple prime implicate of the rationality constraint. As this prime implicate is simple and the profile is rational, every agent votes for at least one of these two literals in the simple prime implicate. This entails that the outcome will support one of these literals. As the simple prime implicate is accepted, the corresponding prime implicate of the feasibility constraint will also be accepted, giving a feasible outcome. Next, we see an application of Theorem 2.6.

Example 5. Consider three agents, $\mathcal{N} = \{a_1, a_2, a_3\}$, who are voting on an agenda of five issues, $\Phi = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5\}$. The agents have a restriction on their judgments, given by $\Gamma = (\varphi_1 \vee \varphi_2) \wedge (\varphi_3 \vee \varphi_4 \vee \varphi_5)$. Furthermore, the agents expect the outcome to abide by the feasibility constraint $\Gamma' = (\varphi_1 \vee \varphi_2 \vee \varphi_3)$. The profile of the three agents' judgments are depicted in Table 2.4.

	φ_1	φ_2	φ_3	φ_4	φ_5
a_1	✓	×	✓	×	×
a_2	×	✓	×	✓	×
a_3	✓	×	×	×	✓
Majority	✓	×	×	×	×

Table 2.4: A Γ-rational profile with a Γ'-feasible outcome.

Here we see that all of the agents have voted in accordance with the rationality constraint Γ; therefore, it is a Γ-rational profile. From the previous example, we see that the pair of formulas is simple. Consequently, the outcome is Γ'-feasible. △

This result corresponds to a theorem from Nehring and Puppe (2007) in the formula-based model of judgement aggregation. This is Theorem 2.1 in this thesis, using the median property (Definition 2.13). We see the link between the size of a minimally inconsistent subset and the size of of the prime implicates in the rationality constraint.

Chapter 3

Guaranteeing Feasible Outcomes Using Quota Rules

This chapter will extend upon the work of Grandi and Endriss (2013). They take agenda properties of judgment aggregation, such as the Median property, and translates them into the single-constraint setting (described in Section 2.3). In particular, they translate the k-median property (Definition 2.14) to binary aggregation with a single constraint. This allows them to recreate Dietrich and List's consistency result (see Theorem 2.2) in this new setting. The chapter will be split into two sections. In Section 3.1, we recreate the translation of the k-median property, however, now in the two-constraint setting (Grandi and Endriss, 2013). Then in Section 3.2, we look at guaranteeing feasible outcomes for any pair of constraints. We do this using prime implicates and the results from the previously mentioned section.

The first result in this chapter motivates an assumption made throughout this chapter. Furthermore, it is one of the conditions in Theorem 2.6 by Endriss (2018). This assumption is that the rationality constraint must entail the feasibility constraint for our judgment aggregation rule to guarantee feasible outcomes. The contrapositive of this result shows that if this entailment is not the case, then there is no guarantee that the outcome will be feasible when given rational profiles.

Lemma 3.1. *Suppose that F is an aggregation rule that satisfies the axiom of Unanimity. If F guarantees Γ'-feasible outcomes on Γ-rational profiles, then $\Gamma \models \Gamma'$.*

Proof. We shall prove the contraposition of the statement in Lemma 3.1. When assuming that $\Gamma \not\models \Gamma'$ holds, there must be a truth assignment which satisfies Γ, but does not satisfy Γ'. Equivalently there exists a $B \in \text{Mod}(\Gamma)\backslash\text{Mod}(\Gamma')$. Consider a voting profile, $\boldsymbol{B}$ for which the agents vote unanimously for this judgment, B. As the aggregation rule satisfies unanimity, we have that $F(\boldsymbol{B}) = B$. Thus, $F(\boldsymbol{B}) \not\models \Gamma'$. Therefore, there is a Γ-rational profile which does not guarantee a Γ'-feasible outcome. □

In Section 3.1, we restrict rationality and feasibility constraints to be single clauses. Note that when we make this assumption, the literals of Γ are also literals of Γ' (Var(Γ) $\subseteq$ Var(Γ')). Due to this observation, we define a recurring feature of the feasibility constraint Γ' will arise throughout this chapter.

Definition 3.1 (Trivial Quota with respect to a constraint Γ)**.** A quota $q(\varphi)$ is a *trivial quota with respect to* Γ if and only if either:

- $q(\varphi) = 0$ when $\varphi \models \Gamma$,
- or $q(\varphi) = n + 1$ when $\neg\varphi \models \Gamma$.

If the feasibility constraint is a single clause and one of its literals has a trivial quota, then we are guaranteed that the outcome will be consistent with that literal. Therefore, the outcome will be consistent with the feasibility constraint. We show this in the following Lemma.

Lemma 3.2. *Let Γ' be a single clause. If a quota rule has a trivial quota with respect to Γ', then we are guaranteed Γ'-feasible outcomes.*

Proof. We assume that Γ' has a trivial quota. Therefore, there exists a $\varphi \in$ Var(Γ') such that it has a quota of $q(\varphi) = 0$ ($q(\varphi) = n + 1$) if φ ($\neg\varphi$) is a literal of Γ'.

If φ is a literal of Γ' and $q(\varphi) = 0$, then no matter the number of agent who vote against φ, φ will always be accepted by the rule. Therefore, as Γ' is a single clause, the outcome will be Γ'-feasible.

If $\neg\varphi$ is a literal of Γ' and $q(\varphi) = n + 1$, then no matter the number of agents who voted for φ, we see that φ would be reject by the rule. Therefore, as Γ' is a single clause, the outcome will be Γ'-feasible. □

Next we give an outline of this chapter. In Section 3.1, we will first recreate the single-constraint case results corresponding to the k-median property in the rationality and feasibility setting. In Subsection 3.1.1, we shall look at the existing results by Grandi and Endriss (2013). In Subsections 3.1.2, 3.1.3 and 3.1.4, we follow the steps of Grandi and Endriss (2013) to recreate the analogue results in our setting.

In Section 3.2, we move from constraints which are single clauses to constraints with any finite number of clauses. In Subsection 3.2.1, we will look at the steps already taken by Grandi and Endriss (2013) to extend the results from a clause to a constraint of any length in the single-constraint setting. We strengthen this result using prime implicates; however, by restricting the aggregation rule to be any quota rule. In Section 3.2.2, we extend both of these results to the rationality and feasibility setting.

3.1 Constraints which are a Single Clause

A lot of the groundwork of investigating consistent outcomes with respect to constraints has been carried out by Grandi and Endriss (2013). This work uses one constraint, which in our model means that the rationality and feasibility constraints are logically equivalent. In this section, we will introduce the existing quota rule results by Grandi and Endriss (2013) and then prove the analogue results in the rationality and feasibility setting. These results will be split into three cases where the rationality constraint is: a positive clause, a negative clause and finally any clause.

3.1.1 Integrity Constraints and Quota Rules

In Section 2.3, we saw an overview of the work carried out by Grandi and Endriss (2013). Many of their results restrict the integrity constraints to be single clauses. They go on the find the types of quotas required to guarantee outcomes consistent with the constraints. They do this by focussing on constraints that contain literals which are either all positive, all negative, or a mix of both positive and negative literals. As we will follow the same steps as Grandi and Endriss (2013), next we will introduce some of their terminology which will be useful in the remainder of this chapter.

A technique used in the proofs of the single-constraint results, is that we want to label votes as either 'wrong' or 'correct'. We want to label votes in this way when we want to say if a vote supports a clause or not, without making any assumptions on the literals in the clause itself.

Definition 3.2 (Wrong vote with respect to a clause π, Grandi and Endriss, 2013)**.** A clause π has received the *wrong vote* from an agent if and only if all of the literals which appear positively (negatively) in the clause π have been rejected (accepted). We will denote this with 'W'.

Next, we will define a 'correct' vote with respect to a clause π.

Definition 3.3 (Correct vote with respect to a clause π, Grandi and Endriss, 2013)**.** A clause π has received a *correct vote* from an agent if and only if one of the literals which appear positively (negatively) in the clause π has been accepted (rejected). We will denote this with 'C'.

As in Grandi and Endriss (2013) we will assume a particular form of the clauses we speak about in this thesis. We make the assumption that the clauses are not trivial. Where a *trivial clause* either has a repeated literal, or the clause contains both an issue and its negation. If a clause is trivial, then it can be reduced to a non-trivial clause or an empty clause. For example, the clause $p \vee \neg p$ would be reduced to an empty clause, and the clause $\neg p \vee q \vee \neg p$ would be reduced to $\neg p \vee q$. From now on, we shall assume that all clauses are in their non-trivial form.

3.1.2 Positive Rationality Constraints

In this subsection, we will inspect which classes of quota rules can guarantee feasible outcomes, where our rationality constraint is some positive clause. First, we will define a positive clause.

Definition 3.4 (Positive Clause)**.** A *positive clause* is a clause which only contains positive literals.

The following proposition generalises the result from Grandi and Endriss (2013, Proposition 21).

Proposition 3.3. Let Γ and Γ' be single clauses such that $\Gamma \models \Gamma'$ holds and let Γ be a positive clause with k literals. A quota rule guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if there is a trivial quota with respect to Γ' or the quotas of the issues in Γ satisfy:

$$\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) < n + k.$$

Proof. We shall prove the left-to-right direction via contraposition. Assume that there are no trivial quotas with respect to Γ', as well as that $\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) \geq n + k$. We want to show that there exists a Γ-rational profile which does not result in a Γ'-feasible outcome.

Due to the conditions on the two constraints in the Proposition, we can number the k issues of Γ from one to k. thus, we can formulate the constraints as such: $\Gamma = \bigvee_{j=1}^{k} \varphi_j$ and $\Gamma' = (\bigvee_{j=1}^{k} \varphi_j) \vee \psi$, where ψ is a clause which could be empty.

Consider the Γ-rational profile depicted in Table 3.1, where as defined in Definition 3.2 'W' represents the 'wrong' vote for the clause ψ. This means that each of the issues in ψ is rejected (accepted) if they are a positive (negative) literal in ψ.

# of agents	φ_1	φ_2	φ_3	...	φ_k	ψ
$\ell_1 \leqslant q(\varphi_1) - 1$	✓	×	×	...	×	W
$\ell_2 \leqslant q(\varphi_2) - 1$	×	✓	×	...	×	W
$\ell_3 \leqslant q(\varphi_3) - 1$	×	×	✓	...	×	W
...						
$\ell_k \leqslant q(\varphi_k) - 1$	×	×	×	...	✓	W
Outcome	×	×	×	...	×	W

Table 3.1: Profile $\boldsymbol{B}$ for the proof Proposition 3.3

Observe that $n = \sum_{\varphi_j \in \text{Var}(\Gamma)} \ell_j \leqslant \sum_{\varphi_j \in \text{Var}(\Gamma)} (q(\varphi_j) - 1)$, which is consistent with our assumption that $\sum_{\varphi_j \in \text{Var}(\Gamma)} q(\varphi_j) \geq n + k$, as Γ has k literals. Since there are no trivial quotas with respect to Γ' (and therefore no trivial quotas with respect to Γ), our claim that $\ell_j \leqslant q(\varphi_j) - 1$ holds.

$\boldsymbol{B}$ is Γ-rational; however, the outcome is not Γ'-feasible. This is in part due to there being no trivial quota with respect to Γ'. Thus, none of the literals in Γ' will agree with the issues in the outcome, regardless of the profile. Moreover, as each issue φ_j receives strictly less than $q(\varphi_j)$ votes, none of the issues of Γ will be accepted. Furthermore, as ψ receives the 'wrong' vote from all n agents, all positive (negative) literals in ψ have receive zero $(n + 1)$ votes. Therefore, we see that the outcome will not agree with ψ, as there are no trivial quotas with respect to ψ. Therefore, we have found a Γ-rational profile without a Γ'-feasible outcome.

For the right-to-left direction, suppose that either Γ' has a trivial quota or that the issues which appear in Γ are such that $\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) < n + k$. In the first case, we assume that Γ' has a trivial quota. It follows from Lemma 3.2 that we are guaranteed Γ'-feasible outcomes.

For the second case, it is left to show is that if $\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) < n + k$ holds then all outcomes are Γ'-feasible. However, for the sake of a contradiction, suppose that there is a Γ-rational profile which does not result in a Γ'-feasible outcome. Then no $\varphi \in \text{Var}(\Gamma)$ will have reached its quota, thus the cumulative number of votes for these issues is at most $\sum_{\varphi \in \text{Var}(\Gamma)} (q(\varphi) - 1)$. By our assumption $\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) < n + k$, then $\sum_{\varphi \in \text{Var}(\Gamma)} (q(\varphi) - 1) < n$. However, as the profile is Γ-rational, each agent must vote for a $\varphi \in \text{Var}(\Gamma)$, so the cumulative number of votes for the issues in Γ must be such that $\sum_{\varphi \in \text{Var}(\Gamma)} (q(\varphi) - 1) \geq n$. Thus, we have derived a contradiction. □

We can see that this proof is a generalisation of Proposition 21 from Grandi and Endriss (2013). If Γ' has a trivial quota, then this is equivalent to the positive constraint having a quota equal to zero. Furthermore, when Γ and Γ' are logically equivalent, then it is clear that our condition $\Gamma \models \Gamma'$ holds. Thus, Proposition 21 from Grandi and Endriss (2013) is a special case of Proposition 3.3. Next, we will see an example of Proposition 3.3 being used.

Example 6. Consider a group of five agents ($\mathcal{N} = \{a_1, a_2, a_3, a_4, a_5\}$), who are voting on the following agenda: $\Phi = \{\varphi_a, \varphi_b, \varphi_c, \varphi_d, \varphi_e\}$. The rationality constraint is $\Gamma = \varphi_a \vee \varphi_b \vee \varphi_c$, a clause with three positive literals, thus $k = 3$. We take the feasibility constraint to be $\Gamma' = \varphi_a \vee \varphi_b \vee \varphi_c \vee \neg\varphi_d \vee \varphi_e$. It is clear here that $\Gamma \models \Gamma'$.

By Proposition 3.3, we see that the sum of the quotas of issues φ_a, φ_b and φ_c have to be strictly less than $n + k = 8$. Say we have the following quotas: $q(\varphi_a) = 3$, $q(\varphi_b) = 2$, $q(\varphi_c) = 2$, $q(\varphi_d) = 4$ and $q(\varphi_e) = 5$, it is clear that this abides by the restriction from the proposition. Now consider the profile depicted in Table 3.2.

	φ_a	φ_b	φ_c	φ_d	φ_e
a_1	✓	×	×	×	×
a_2	✓	×	×	✓	✓
a_3	×	✓	×	✓	×
a_4	×	×	✓	✓	×
a_5	×	×	✓	✓	×
Total	2	1	2	4	1

Table 3.2: A Γ-rational profile with a Γ'-feasible outcome

We see that this is a Γ-rational profile as every agent has voted for φ_a, φ_b, or φ_c. Furthermore, this is a Γ'-feasible outcome as φ_c has reached its quota of $q(\varphi_c) = 2$.

However, without a_5's vote, the issues φ_a, φ_b, and φ_c have each receive one vote less than their quotas. Thus, no literal in the rationality constraint, nor the feasibility constraint would be supported by the outcome. For the profile to be Γ-rational, a_5 has to vote for one of the three issues in Γ, meaning that the quota rule would accept one of the issues. Thus, ensuring a Γ'-feasible outcome. △

From Proposition 3.3 follows a corollary regarding uniform quota rules, analogous to how Grandi and Endriss (2013) proceed from Proposition 21 to Corollary $24i$.

Corollary 3.4. *Let Γ and Γ' be single clauses such that $\Gamma \models \Gamma'$ and Γ be a positive clause with k literals. A non-uniform quota rule guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if the uniform quota u satisfies:*

$$u \leqslant \lceil \frac{n}{k} \rceil.$$

The proof of Corollary 3.4 follows directly from Proposition 3.3 where we use the facts that $q(\varphi) = u$ for all $\varphi \in \text{Var}(\Gamma)$, as well as that $\lceil \frac{n}{k} \rceil$ is the largest integer smaller than $\frac{n}{k} + 1$.

Hence, we have the necessary and sufficient conditions for obtaining Γ'-feasible outcomes on Γ-rational profiles for quota rules when Γ is a positive clause. The next step is to repeat this process when Γ is a negative clause.

3.1.3 Negative Rationality Constraints

In this subsection, we will again look at when we can guarantee feasible outcome on rational profiles. However, now we are restricting the rationality constraint to be a *negative clause*.

Definition 3.5 (Negative clause)**.** A *negative clause* is a clause which only contains negated propositions as its literals.

Next, we generalisation Proposition 22 from Grandi and Endriss (2013).

Proposition 3.5. Let Γ and Γ' be single clauses such that $\Gamma \models \Gamma'$ holds and let Γ be a negative clause with k literals. A quota rule guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if there is a trivial quota with respect to Γ' or the quotas of the issues in Γ satisfy:

$$\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) > (k-1)n.$$

Proof. We shall prove the left-to-right direction via contraposition. Assume that there are no trivial quotas with respect to Γ', as well as that $\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) \leqslant (k-1)n$ holds. We want to show that there exists a Γ-rational profile which does not result in a Γ'-feasible outcome.

Due to the conditions on the two constraints in the proposition, we can number the k issues of Γ from one to k and denote the constraints as $\Gamma = \bigvee_{j=1}^{k} \neg\varphi_j$ and $\Gamma' = (\bigvee_{j=1}^{k} \neg\varphi_j) \vee \psi$, where ψ is a clause which could be empty.

Consider the Γ-rational profile depicted in Table 3.3, where, as before 'W' represents the 'wrong vote' for the clause ψ as defined in Definition 3.2.

# of agents	φ_1	φ_2	φ_3	...	φ_k	ψ
$\ell_1 \leqslant n - q(\varphi_1)$	×	✓	✓	...	✓	W
$\ell_2 \leqslant n - q(\varphi_2)$	✓	×	✓	...	✓	W
$\ell_3 \leqslant n - q(\varphi_3)$	✓	✓	×	...	✓	W
...						
$\ell_k \leqslant n - q(\varphi_k)$	✓	✓	✓	...	×	W
Outcome	✓	✓	✓	...	✓	W

Table 3.3: Profile $\boldsymbol{B}$ in the proof of Proposition 3.5

Observe that $n = \sum_{\varphi_j \in \text{Var}(\Gamma)} \ell_j \leqslant \sum_{\varphi_j \in \text{Var}(\Gamma)} n - q(\varphi_j)$ holds, which is consistent with our assumption that $\sum_{\varphi_j \in \text{Var}(\Gamma)} q(\varphi_j) \leqslant (k-1)n$, as Γ has k literals. Furthermore, the outcome of our profile is consistent, as there are no trivial quotas with respect to Γ' (and therefore, there are no trivial quotas with respect to Γ), our claim that $\ell_j \leqslant n - q(\varphi_j)$ holds.

$\boldsymbol{B}$ is Γ-rational; however, the outcome is not Γ'-feasible. This is in part due there being no trivial quotas with respect to the issues of Γ'. Therefore, the outcome will not always be consistent with Γ', regardless of the profile submitted to the rule. Moreover, as each issue φ_j receives at most $n - q(\varphi_j)$ votes against the issue, none of the issues of Γ will be rejected. Furthermore, as ψ receives the 'wrong' vote from all n agents, and there are no trivial quotas with respect to ψ, the outcome won't agree with ψ. Therefore, we have found a Γ-rational profile without a Γ'-feasible outcome.

For the right-to-left direction, suppose that either Γ' has a trivial quota or that the issues which appear in Γ are such that $\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) > (k-1)n$ holds. In the first case, it follows from Lemma 3.2 that if there are no trivial quotas with respect to Γ', then we are guaranteed Γ'-feasible outcomes.

For the second case, it is left to show is that if $\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) > (k-1)n$ holds then all outcomes are Γ'-feasible. However, for the sake of a contradiction, suppose that there is a Γ-rational profile which does not result in a Γ'-feasible outcome. Then each $\varphi \in \text{Var}(\Gamma)$ will have reached its quota. Thus, the cumulative number of votes against these issues is at most $\sum_{\varphi \in \text{Var}(\Gamma)} (n - q(\varphi))$. By our assumption $\sum_{\varphi \in \text{Var}(\Gamma)} q(\varphi) > (k-1)n$, then $\sum_{\varphi \in \text{Var}(\Gamma)} (n - q(\varphi)) < n$. However, as the profile is Γ-rational, each agent must vote against at least one $\varphi \in \text{Var}(\Gamma)$ as it is a negative clause. Therefore, the cumulative number of votes against the issues must be such that $\sum_{\varphi \in \text{Var}(\Gamma)} (n - q(\varphi)) \geq n$ holds. Thus, we have derived a contradiction. $\square$

A corollary follows directly from Proposition 3.5, regarding the uniform quota rule. This links to Corollary $24ii$ by Grandi and Endriss (2013).

Corollary 3.6. *Let Γ and Γ' be single clauses such that $\Gamma \models \Gamma'$ and Γ be a negative clause with k literals. Any non-trivial uniform quota rule guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if the uniform quota u satisfies:*

$$u \geq n - \lceil \frac{n}{k} \rceil + 1.$$

This result follows from Proposition 3.5 with the assumption that $q(\varphi) = u$ for all $\varphi \in \text{Var}(\Gamma)$. This also uses that $\lceil \frac{n}{k} \rceil \geq \frac{n}{k}$.

3.1.4 Any Rationality Constraint

In the two previous subsections, we restrained the rationality constraints to be single clauses, with literals which are either only positive or only negative propositions. In this subsection, we look at the case where the rationality constraint can be any single clause.

Before we can do this, we need to set some notation, which will be used throughout the remainder of this chapter. As $\text{Var}(\Gamma)$ (see Definition 2.6) denotes all of the issues which appear in Γ, we will let $\text{Var}(\Gamma^+)$ denote the issue which appear positively in Γ. Similarly, we will let $\text{Var}(\Gamma^-)$ denote issues which appear negatively in Γ.

The first result in this subsection extends Theorem 30 by Grandi and Endriss (2013) to the rationality and feasibility setting.

Theorem 3.7. *Let Γ and Γ' be single clauses such that $\Gamma \models \Gamma'$ holds and Γ is a clause with k literals. A quota rule F guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if either F has a trivial quota with respect to Γ' or the quotas of the issues in Γ satisfy:*

$$\sum_{\varphi \in \mathit{Var}(\Gamma^-)} q(\varphi) + \sum_{\varphi \in \mathit{Var}(\Gamma^+)} (n - q(\varphi) + 1) > n(k-1).$$

Proof. The left-to-right direction will be shown via contraposition. Assume that there are no trivial quotas with respect to Γ', as well as that $\sum_{\varphi \in \text{Var}(\Gamma^-)} q(\varphi) + \sum_{\varphi \in \text{Var}(\Gamma^+)} (n-$

$q(\varphi) + 1) \leqslant n(k-1)$. We want to show that there is a Γ-rational profile without a Γ'-feasible outcome.

From the conditions on the constraints in the theorem, we see that they can be denoted as $\Gamma = \bigvee_{j=1}^{k}(L_j)$ and $\Gamma' = (\bigvee_{j=1}^{k}(L_j)) \vee \psi$. As before ψ is a clause which could be empty. We let L_j be a literal of Γ, which is either equal to φ_j or $\neg\varphi_j$ (note that here we number the k issues which appear in Γ from 1 to k).

Consider the Γ-rational profile depicted in Table 3.4. 'W', as before, represents the 'wrong' vote, and now a 'C' represents a 'correct' vote (see Definitions 3.2 and 3.3).

# of agents	φ_1	φ_2	...	φ_k	ψ
ℓ_1	C	W		W	W
ℓ_2	W	C		W	W
...					
ℓ_k	W	W		C	W
Outcome	W	W		W	W

Table 3.4: Profile $\boldsymbol{B}$ in the proof of Theorem 3.7

In Table 3.4, there are ℓ_j agents voting for a particular judgment (a row). For all $j \in [1,k]$ we have that ℓ_j is either equal to at most to $q(\varphi_j) - 1$ votes for φ_j, if L_j is a positive literal in Γ. Otherwise, ℓ_j is at most $n - q(\varphi_j)$ votes against φ_j if L_j is the negative literal. Observe that the number of 'wrong' votes for an issue φ_j ($j \in [1,k]$) in this profile is at least $n - q(\varphi_j) + 1$ if L_j is a positive literal in Γ, or at least $n - (n - q_j)$ if L_j is a negative literal in Γ.

From Table 3.4, we see that the total number of 'wrong' votes for issues in Γ is $\sum\limits_{\varphi_j \in \text{Var}(\Gamma)} n - \ell_j \leqslant \sum\limits_{\varphi_j \in \text{Var}(\Gamma^-)} q(\varphi_j) + \sum\limits_{\varphi_j \in \text{Var}(\Gamma^+)} (n - q(\varphi_j) + 1)$. As $\boldsymbol{B}$ is a Γ-rational profile this means that there are at least n 'correct' votes for issues in Γ from the total number of votes for the issues of Γ, nk. Hence, the total number of 'wrong' votes is at most $n(k-1)$. Thus, we have that $\sum\limits_{\varphi_j \in \text{Var}(\Gamma^-)} q(\varphi_j) + \sum\limits_{\varphi_j \in \text{Var}(\Gamma^+)} (n - q(\varphi_j) + 1) \leqslant n(k-1)$, which abides by our assumption.

$\boldsymbol{B}$ is a Γ-rational profile; however, the outcome is not Γ'-feasible. This is in part due to there being no trivial quota with respect to Γ'. Thus, none of the literals in Γ' will agree with the outcome, regardless of the profile submitted to the rule. Moreover, as each of the issues in Γ has received ℓ_j votes agreeing with L_j, it is clear that this is not enough for the outcome to agree with any of the literals in Γ. Furthermore, as ψ receives the 'wrong' vote from all n agents, the outcome will not agree with ψ as there are no trivial quotas with respect to Γ'. Therefore, we have found a Γ-rational profile without a Γ'-feasible outcome.

For the right-to-left direction, suppose that either Γ' has a trivial quota or the issues which appear in Γ are such that $\sum_{\varphi_j \in \text{Var}(\Gamma^-)} q(\varphi_j) + \sum_{\varphi_j \in \text{Var}(\Gamma^+)} (n - q(\varphi_j) + 1) > n(k-1)$ holds. In the first case, we assume that Γ' has a trivial quota, and we want to show that we are guaranteed Γ'-feasible outcomes. This was shown in Lemma 3.2.

For the second case, it is left to show that if $\sum_{\varphi_j \in \text{Var}(\Gamma^-)} q(\varphi_j) + \sum_{\varphi_j \in \text{Var}(\Gamma^+)} (n - q(\varphi_j) + 1) > n(k-1)$ holds, then all outcomes are Γ'-feasible. However, for the sake of a contradiction, suppose that there is a Γ-rational profile which does not result in a Γ'-feasible outcome.

As this profile is not Γ'-feasible, this entails that none of the literals in Γ agree with the outcome. All of the positive issues in Γ can receive at most $q(\varphi) - 1$ 'correct' votes, or at least $n - q(\varphi) + 1$ 'wrong' votes. Similarly, the negative issues in Γ can receive at most $n - q(\varphi)$ 'correct' votes, or at least $q(\varphi)$ 'wrong' votes. Cumulatively this means that there are at least $\sum_{\varphi_j \in \text{Var}(\Gamma^-)} q(\varphi_j) + \sum_{\varphi_j \in \text{Var}(\Gamma^+)} (n - q(\varphi_j) + 1)$ 'wrong' votes for the issues in Γ. By our assumption $\sum_{\varphi_j \in \text{Var}(\Gamma^-)} q(\varphi_j) + \sum_{\varphi_j \in \text{Var}(\Gamma^+)} (n - q(\varphi_j) + 1) > n(k-1)$, that is, the total number of 'wrong' votes should be greater than $n(k-1)$.

As this profile is Γ-rational, there must be at least n 'correct' votes for the issues in Γ. Conversely, there are at most $nk - n$ (or equivalently $n(k-1)$) 'wrong' votes for the issues in Γ. Thus, we have reached a contradiction, as the total number of 'wrong' votes for the issues in Γ has to be greater than and at most $n(k-1)$. □

We can see how Proposition 3.3 and Proposition 3.5 are special cases of Theorem 3.7. If either the first or the second summation in the inequality is empty, then we find the inequalities from these propositions.

Again, there is a corollary which follows from Theorem 3.7 regarding uniform quotas. This corollary corresponds to Corollary 31 in Grandi and Endriss (2013).

Corollary 3.8. *Let Γ and Γ' be single clauses such that $\Gamma \vDash \Gamma'$ holds and Γ be a clause with k literals. Any non-trivial uniform quota rule guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if the uniform quota u satisfies:*

$$u(k_- - k_+) > n(k_- - 1) - k_+.$$

Where k_+ is the number of positive literals in Γ ($k_+ = |Var(\Gamma^+)|$) and k_- is the number of negative literals in Γ ($k_- = |Var(\Gamma^-)|$).

The proof of Corollary 3.8 follows directly from Theorem 3.7. Observe that both Corollaries 3.4 and 3.6 can be recovered from Corollary 3.8, when we let either k_- or k_+ be equal to zero, respectively.

This result links to both the the formula-based model of judgment aggregation, as well as the single-constraint model of binary aggregation. We see from the result of Dietrich and List (2007, Corollary $2(a)$) (see Proposition 2.2), links directly to Corollary 31 from Grandi and Endriss (2013), and our result, Corollary 3.8.

3.2 Constraints with Multiple Clauses

In the previous section, we saw hoe to guarantee feasible outcomes when our rationality constraint is a single clause. In this section, we will use prime implicates (Section 2.4) to extend these results to constraints with any number of clauses.

In Section 3.2.1, we will introduce the existing work in the single-constraint case (Grandi and Endriss, 2013, Lemma 3). Using this result, we prove a theorem connecting quota rules that guarantee feasible outcomes for both a pair of constraints and their prime implicates, in the single-constraint setting. Following this, in Subsection 3.2.2, we recreate each on the results from the previous subsection in the rationality and feasibility setting.

3.2.1 Collectively Rational Outcomes on Any Constraint

In this subsection, we will follow on from the work of Grandi and Endriss (2013) using the integrity constraint setting of binary aggregation. This looks at classes of aggregation rules which guarantee feasible outcomes for every member of a set of constraints. We focus on their following lemma.

Lemma 3.9 (Lemma 3, Grandi and Endriss, 2013)**.** *Given a set of formulas $\mathcal{L} \subseteq \mathcal{L}(\Phi)$. If an aggregation rule F guarantees π-feasible outcomes on π-rational profiles for all $\pi \in \mathcal{L}$, then F guarantees $(\bigwedge_{\pi\in\mathcal{L}} \pi)$-feasible outcomes on $(\bigwedge_{\pi\in\mathcal{L}} \pi)$-rational profiles.*

In the following example from Chen and Endriss (2018) we see that the converse does not hold.

Example 7. Consider the following clauses as our rationality and feasibility constraints: $\pi_1 = \neg\varphi_a \vee \neg\varphi_b$ and $\pi_2 = \varphi_a$. We see from Grandi and Endriss (2013, Corollary 31) (the single-constraint case of Corollary 3.8), that taking π_1 as the constraint, we get that the class of uniform quota rules which guarantee π_1-feasible outcomes is when $u > \frac{n}{2}$. Similarly, when taking π_2 as the constraint, we get that the class of uniform quota rules which guarantee π_2-feasible outcomes is when $u < n+1$. From Lemma 3.9, we see that a uniform quota rule guarantee $(\pi_1 \wedge \pi_2)$-feasible outcomes on $(\pi_1 \wedge \pi_2)$-rational profiles when u abides by $n+1 > u > \frac{n}{2}$.

However, we see that $\pi_1 \wedge \pi_2 \equiv \varphi_a \wedge \neg\varphi_b$. It is clear that any non-trivial quota rule will give $(\pi_1 \wedge \pi_2)$-feasible outcomes on $(\pi_1 \wedge \pi_2)$-rational profiles. Therefore, the previous range $n+1 > u > \frac{n}{2}$ does not fully capture all of the aggregation rules which give $(\pi_1 \wedge \pi_2)$-feasible outcomes on $(\pi_1 \wedge \pi_2)$-rational profiles.

Moving away from Chen and Endriss's example, we can look to prime implicates fora solution to this problem. There are two prime implicates of $(\pi_1 \wedge \pi_2)$, namely $\neg\varphi_b$ and φ_a. Thus, by applying the result from Grandi and Endriss (2013, Corollary 31) on both prime implicates, we get that $n+1 > u > 0$. This captures all possible non-trivial quota rules that can lead to $(\pi_1 \wedge \pi_2)$-feasible outcomes. $\triangle$

The previous example motivates Theorem 3.11. However, first we will prove a lemma in order to prove this theorem. Note that this lemma appears to be similar to the left-to-right direction of Theorem 3.7. However, this lemma is not a special case of Theorem 3.7.

Lemma 3.10. *Let Γ be a formula, π be a prime implicate of Γ with k literals, and let F be a quota rule. If F guarantees π-feasible outcomes on Γ-rational profiles then F either has a trivial quota with respect to π, or the quotas of the issues in Var(π) abide by:*

$$\sum_{\varphi \in \mathit{Var}(\pi^-)} q(\varphi) + \sum_{\varphi \in \mathit{Var}(\pi^+)} (n - q(\varphi) + 1) > n(k-1).$$

Proof. First, it is worth noting that when Γ is inconsistent, then there are no Γ-rational profiles. Therefore, the lemma is vacuously true. Thus we will assume that Γ is consistent and that there exists some Γ-rational profiles.

We shall prove the lemma by contraposition. Therefore, we assume that F does not abide by the inequality for the quotas of the issues in π, as well as there being no trivial quotas with respect to π. We want to show that there exists a Γ-rational profile without a π-feasible outcome. Consider the following profile, $\boldsymbol{B}$, depicted in Table 3.5.

In Table 3.5, we let $\pi = \bigvee_{i=1}^{k} L_i$, where L_i is a literal which is equal to either φ_i or $\neg\varphi_i$. We see each judgment agrees with one of the literals of π and disagrees with the rest. In the 'remaining issues' column we see that every judgment in $\boldsymbol{B}$ gives a correct or consistent vote. We next need to check that this profile $\boldsymbol{B}$ can be

# of agents	φ_1	φ_2	φ_3	...	φ_k	Remaining Issues:Var(Γ)\Var(π)
ℓ_1	C	W	W	...	W	C
ℓ_2	W	C	W	...	W	C
ℓ_3	W	W	C	...	W	C
...						
ℓ_k	W	W	W	...	C	C
Outcome	W	W	W	...	W	N/A

Table 3.5: Profile $\boldsymbol{B}$ for the proof Lemma 3.10

a Γ-rational profile.

Suppose that a judgment supporting $L_i \wedge \bigwedge_{j\in[1,k]\setminus\{i\}} \neg L_j$ from Table 3.5 is inconsistent with Γ (with $i \in [1,k]$). Therefore, $\boldsymbol{B}$ cannot be a Γ-rational profile. Hence, we have that $\Gamma \wedge L_i \wedge \bigwedge_{j\in[1,k]\setminus\{i\}} \neg L_j \models \bot$, or $\Gamma \models \neg(L_i \wedge \bigwedge_{j\in[1,k]\setminus\{i\}} \neg L_j)$, which is equivalent to $\Gamma \models \neg L_i \vee \bigvee_{j\in[1,k]\setminus\{i\}} L_j$. Therefore, $\neg L_i \vee \bigvee_{j\in[1,k]\setminus\{i\}} L_j$ is an implicate of Γ. As both π and $\neg L_i \vee \bigvee_{j\in[1,k]\setminus\{i\}} L_j$ are implicates of Γ, it is clear that $\bigvee_{j\in[1,k]\setminus\{i\}} L_j$ must also be an implicate of Γ. [1] However, this entails that π is not a prime implicate of Γ, as the implicate $\bigvee_{j\in[1,k]\setminus\{i\}} L_j \models \pi$, and $\pi \not\models \bigvee_{j\in[1,k]\setminus\{i\}} L_j$. Therefore, we can conclude that $\boldsymbol{B}$ can be a Γ-rational profile.

Finally, we need to check that the profile depicted in Table 3.5 does not result in a π-feasible outcome. Under the earlier made assumptions, there are no trivial quotas with respect to π. Furthermore, the quotas of the issues in π do not abide by the inequality in the statement of the lemma. In the table, we let ℓ_j be equal to at most $q(\varphi_j) - 1$, if L_j is a positive literal in π. Otherwise, ℓ_j can be at most $n - q(\varphi_j)$, if L_j is the negative literal of π. Observe that the number of 'wrong' votes for an issue φ_j ($j \in [1,k]$) in this profile is at least $n - q(\varphi_j) + 1$ if L_j is a positive literal in π, or at least $n - (n - q(\varphi_j))$ if L_j is a negative literal in π.

From Table 3.5, we see that the total number of wrong votes for the issues in π is $\sum\limits_{\varphi_j\in\text{Var}(\pi)} n - \ell_j \leqslant \sum\limits_{\varphi_j\in\text{Var}(\pi^-)} q(\varphi_j) + \sum\limits_{\varphi_j\in\text{Var}(\pi^+)} (n - q(\varphi_j) + 1)$. As $\boldsymbol{B}$ is a Γ-rational profile this means that there are at least n correct votes for issues in π. Therefore, the total number of wrong votes for the issues in π can be at most $n(k-1)$ votes. Thus, we have that $\sum\limits_{\varphi_j\in\text{Var}(\pi^-)} q(\varphi_j) + \sum\limits_{\varphi_j\in\text{Var}(\pi^+)} (n - q(\varphi_j) + 1) \leqslant n(k-1)$, which is consistent with our assumption.

$\boldsymbol{B}$ is a Γ-rational profile; however, the outcome is not π-feasible. This is in part due to there being no trivial quota with respect to π. Therefore, none of the literals in π will agree with the outcome, regardless of the profile submitted to the rule.

[1] This is more commonly known as the resolution rule, used to find (prime) implicates of a formula. See Definition 8 from Tourret (2012).

Moreover, as each $\varphi_j \in \text{Var}(\pi)$ has received ℓ_j votes agreeing with L_j, it is clear that this is not enough for the outcome to agree with any of the literals of π. Therefore, we have found a Γ-rational profile without a Γ'-feasible outcome. □

Now that we have shown the previous lemma, we can prove the following theorem.

Theorem 3.11. *Let F be a quota rule, and Γ be a formula. F guarantees Γ-feasible outcomes on Γ-rational profiles if and only if for all prime implicates π of Γ, F guarantees π-feasible outcomes on π-rational profiles.*

Proof. In the right-to-left direction, we let P be the set of prime implicates of Γ. Thus Γ is logically equivalent to $\bigwedge_{\pi \in P} \pi$. Therefore, the right-to-left direction follows from Lemma 3.9, so left to prove is the left-to-right direction.

We assume that F guarantees Γ-feasible outcomes on Γ-rational profiles. We want to show that F also guarantees π-feasible outcomes on π-rational profiles, for all prime implicates of Γ. We pick a prime implicate π of Γ arbitrarily.

As F guarantees Γ-feasible outcomes on Γ-rational profiles, F also guarantees π-feasible outcomes on Γ-rational profiles. By Lemma 3.10 we see that this implies that F either has a trivial quota with respect to π, or the quotas of the issues of π abide by $\sum_{\varphi \in \text{Var}(\pi^-)} q(\varphi) + \sum_{\varphi \in \text{Var}(\pi^+)} (n - q(\varphi) + 1) > n(k-1)$.

We see that these are the conditions in Theorem 30 by Grandi and Endriss (2013), for F to guarantee π-feasible outcomes on π-rational profiles. As π was chosen arbitrarily, we have that F guarantees π-feasible outcomes on π-rational profiles for all prime implicates of Γ. □

3.2.2 Guarantee Feasible Outcomes on Any Pair of Constraints

In this subsection, we will look the rationality and feasibility analogues of the results which appeared in Subsection 3.2.1. Next, we will see a lemma which is similar to Lemma 3.9 by Grandi and Endriss (2013).

Lemma 3.12. *If F guarantees Γ_1'-feasible outcomes on Γ_1-rational profiles, and also guarantees Γ_2'-feasible outcomes on Γ_2-rational profiles, then F guarantees $(\Gamma_1' \wedge \Gamma_2')$-feasible outcomes on $(\Gamma_1 \wedge \Gamma_2)$-rational profiles.*

Proof. We assume that the aggregation rule F guarantees Γ_1'-feasible outcomes on Γ_1-rational profiles, and F also guarantees Γ_2'-feasible outcomes on Γ_2-rational profiles. For the sake of a contradiction, we assume that F does not guarantee $(\Gamma_1' \wedge \Gamma_2')$-feasible outcomes on $(\Gamma_1 \wedge \Gamma_2)$-rational profiles. Therefore, there exists a $\boldsymbol{B}$ such that $F(\boldsymbol{B}) \nvDash (\Gamma_1' \wedge \Gamma_2')$, but $\boldsymbol{B}$ is a $(\Gamma_1 \wedge \Gamma_2)$-rational profile.

As $\boldsymbol{B}$ is a $(\Gamma_1 \wedge \Gamma_2)$-rational profile, we have that for all $i \in \mathcal{N}$ that $B_i \in \text{Mod}(\Gamma_1 \wedge \Gamma_2)$. Therefore, for all $i \in \mathcal{N}$, $B_i \in \text{Mod}(\Gamma_1)$ and $B_i \in \text{Mod}(\Gamma_2)$. Hence, $\boldsymbol{B}$ is both a Γ_1-rational profile and a Γ_2-rational profile. By assumption, we can then say that $F(\boldsymbol{B})$ is both a Γ_1'-feasible outcome and a Γ_2'-feasible outcome. Thus, $F(\boldsymbol{B}) \in \text{Mod}(\Gamma_1')$ and $F(\boldsymbol{B}) \in \text{Mod}(\Gamma_2')$, which is equivalent to $F(\boldsymbol{B}) \in \text{Mod}(\Gamma_1') \cap \text{Mod}(\Gamma_2') = \text{Mod}(\Gamma_1' \wedge \Gamma_2')$. Therefore, $F(\boldsymbol{B}) \vDash (\Gamma_1' \wedge \Gamma_2')$ holds and we have reached a contradiction. □

It is clear that the above result can be extended to the conjunction of any finite number of constraints. In the following example, observe that converse of the above result does not hold.

Example 8. Consider the pair $\Gamma_1 = (\neg\varphi_a \vee \neg\varphi_b)$ and $\Gamma_1' = (\neg\varphi_a \vee \neg\varphi_b \vee \neg\varphi_c)$, as well as the pair $\Gamma_2 = \varphi_a$ and $\Gamma_2' = (\varphi_a \vee \neg\varphi_c)$. From Corollary 3.8, we see that we can guarantee Γ_1'-feasible outcomes on Γ_1-rational profiles when the uniform quota is such that $u > \frac{n}{2}$. Furthermore, from Corollary 3.8, we see that a uniform quota rule guarantees Γ_2'-feasible outcomes on Γ_2-rational profiles when $n+1 > u$. Therefore, from Lemma 3.12, we see that uniform quota rules can guarantee $(\Gamma_1' \wedge \Gamma_2')$-feasible outcomes on $(\Gamma_1 \wedge \Gamma_2)$-rational profiles when $n+1 > u > \frac{n}{2}$.

However, as $(\Gamma_1 \wedge \Gamma_2)$ is logically equivalent to $\varphi_a \wedge \neg\varphi_b$. It is clear that we can guarantee $(\Gamma_1' \wedge \Gamma_2')$-feasible outcomes on $(\Gamma_1 \wedge \Gamma_2)$-rational profiles for any non-trivial uniform quota rule.

Therefore, we can see that there are quota rules which guarantee $(\Gamma_1' \wedge \Gamma_2')$-feasible outcomes on $(\Gamma_1 \wedge \Gamma_2)$-rational profiles. However, they do not guarantee outcomes for both of the pairs (Γ_1, Γ_1') and (Γ_2, Γ_2'). △

Next, we will gain the other direction of Lemma 3.12 when considering only quota rules. However, first we will prove a lemma in order to prove our main result. As with Lemma 3.10, there are similarities between the following lemma and the left-to-right direction of Theorem 3.7. However, the following lemma is not a special case of Theorem 3.7.

Lemma 3.13. *Let F be a quota rule, Γ and Γ' be formulas such that $\Gamma \vDash \Gamma'$, and let π' be a prime implicate of Γ'. If F guarantees π'-feasible outcomes on Γ-rational profiles, then there exists a prime implicate π of Γ, such that $\pi \vDash \pi'$ and either F has a trivial* *quota with respect to π' or the quotas of the issues of π satisfy:*

$$\sum_{\varphi \in Var(\pi^-)} q(\varphi) + \sum_{\varphi \in Var(\pi^+)} (n - q(\varphi) + 1) > n(k-1).$$

Proof. We prove the following lemma by contraposition. Therefore, we assume that for any prime implicate π of Γ such that it entails π' and F does not satisfy the inequality with respect to π. Moreover, F has no trivial quotas with respect to π'. We want to show that there exists a Γ-rational profile without a π'-feasible outcome.

First, we take an arbitrary prime implicate π of Γ, such that $\pi \vDash \pi'$. We let $\pi = \bigvee_{i=1}^{k} L_i$, where L_i is either φ_i or $\neg\varphi_i$. As $\pi \vDash \pi'$, we denote π' as $\pi' = \psi \vee \bigvee_{i=1}^{k} L_i$, where ψ could be an empty clause. Consider the profile $\boldsymbol{B}$ depicted in Table 3.6.

# of agents	φ_1	φ_2	...	φ_k	ψ	Remaining Issues:Var(Γ)\Var(π')
ℓ_1	C	W	...	W	W	C
ℓ_2	W	C	...	W	W	C
...						
ℓ_k	W	W	...	C	W	C
Outcome	W	W	...	W	W	N/A

Table 3.6: Profile $\boldsymbol{B}$ for the proof Lemma 3.13

In Table 3.6, we have W denoting a wrong vote and C denoting a correct vote (see Definitions 3.2 and 3.3). The correct vote for the remaining issues, means that these agents give some consistent vote to make their judgment Γ-rational. Next we will show that $\boldsymbol{B}$ can be a Γ-rational profile.

Suppose that $\boldsymbol{B}$ is not a Γ-rational profile. Then there is a judgment from Table 3.6 which is incompatible with Γ. Say that judgment is $(L_i \wedge (\bigwedge_{j\in[1,k]\setminus\{i\}} \neg L_j) \wedge \neg\psi$ (for some $i \in [1,k]$). Thus, $\Gamma \wedge (L_i \wedge (\bigwedge_{j\in[1,k]\setminus\{i\}} \neg L_j) \wedge \neg\psi) \vDash \bot$. Equivalently, $\Gamma \vDash \neg(L_i \wedge (\bigwedge_{j\in[1,k]\setminus\{i\}} \neg L_j) \wedge \neg\psi)$. Therefore, $\neg L_i \vee \bigvee_{j\in[1,k]\setminus\{i\}} L_j \vee \psi$ is an implicate of Γ. By the resolution rule,[2] as we have that $\pi = \bigvee_{i=1}^{k} L_i$ and $\neg L_i \vee \bigvee_{j\in[1,k]\setminus\{i\}} L_j \vee \psi$ are both implicates of Γ, we have that $\bigvee_{j\in[1,k]\setminus\{i\}} L_j \vee \psi$ is also an implicate of Γ, $\text{Mod}(\Gamma) \subseteq \text{Mod}(\bigvee_{j\in[1,k]\setminus\{i\}} L_j \vee \psi)$.

It is clear that $\text{Mod}(\bigvee_{j\in[1,k]\setminus\{i\}} L_j \vee \psi) \subseteq \text{Mod}(\pi')$. However, although $\text{Mod}(\Gamma') \subseteq \text{Mod}(\pi')$, it may not be the case that $\text{Mod}(\Gamma') \subseteq \text{Mod}(\bigvee_{j\in[1,k]\setminus\{i\}} L_j \vee \psi)$. If $\text{Mod}(\Gamma') \subseteq \text{Mod}(\bigvee_{j\in[1,k]\setminus\{i\}} L_j \vee \psi)$ is the case, then π' is not a prime implicate of Γ', contradicting our assumption.

If $\text{Mod}(\Gamma') \nsubseteq \text{Mod}(\bigvee_{j\in[1,k]\setminus\{i\}} L_j \vee \psi)$, then $\Gamma' \vDash (\bigwedge_{j\in[1,k]\setminus\{i\}} \neg L_j) \wedge \neg\psi$. Conse-

[2]See Tourret (2012, Definition 8) for more details.

quently, $\text{Mod}(\Gamma') \subseteq \text{Mod}(\pi')$ and $\text{Mod}(\Gamma') \subseteq \text{Mod}((\bigwedge_{j\in[1,k]\setminus\{i\}} \neg L_j) \wedge \neg\psi)$, thus, $\text{Mod}(\Gamma') \subseteq \text{Mod}(\pi') \cap \text{Mod}((\bigwedge_{j\in[1,k]\setminus\{i\}} \neg L_j) \wedge \neg\psi) = \text{Mod}(L_i)$. Therefore, $\Gamma' \vDash L_i$, and we can conclude that π' is not a prime implicate of Γ' as $L_i \vDash \pi'$, but $\pi' \nvDash L_i$. So, $\boldsymbol{B}$ is a Γ-rational profile.

Therefore, we have checked that $\boldsymbol{B}$ is a Γ-rational profile. Next, we need to check that $F(\boldsymbol{B})$ is not a π'-feasible outcome. In Table 3.6, there are ℓ_j agents voting for a particular judgment. We let ℓ_j be at most $q(\varphi_j) - 1$, if L_j is a positive literal in π. Otherwise, we let ℓ_j be at most $n - q(\varphi_j)$ votes, when L_j is a negative literal of π. Observe that the number of wrong votes for an issue φ_j ($j \in [1,k]$) in this profile is at least $n - q(\varphi_j) + 1$ if L_j is a positive literal in π, or at least $n - (n - q(\varphi_j))$ if L_j is a negative literal in π.

From Table 3.6, we see that the total number of wrong votes for issues in π is $\sum\limits_{\varphi_j \in \text{Var}(\pi)} n - \ell_j \leqslant \sum\limits_{\varphi_j \in \text{Var}(\pi^-)} q(\varphi_j) + \sum\limits_{\varphi_j \in \text{Var}(\pi^+)} (n - q(\varphi_j) + 1)$. As $\boldsymbol{B}$ is a Γ-rational profile this means that there are at least n correct votes for issues in π. Therefore, the total number of 'wrong' votes is at most $n(k-1)$. Thus, we have that $\sum\limits_{\varphi_j \in \text{Var}(\pi^-)} q(\varphi_j) + \sum\limits_{\varphi_j \in \text{Var}(\pi^+)} (n - q(\varphi_j) + 1) \leqslant n(k-1)$, abiding by our assumption.

$\boldsymbol{B}$ is a Γ-rational profile; however, the outcome is not π'-feasible. This is in part due to there being no trivial quota with respect to π'. Thus, none of the literals in π' will agree with the outcome, regardless of the profile submitted to the rule. Moreover, as each $\varphi_j \in \text{Var}(\pi)$ has received ℓ_j votes agreeing with L_j, it is clear that this is not enough support for the outcome to agree with any of the literals in π. Furthermore, as ψ receives the wrong vote from all n agents, the outcome will not agree with ψ as there are no trivial quotas with respect to π'. Therefore, we have found a Γ-rational profile without a π'-feasible outcome.

As π was chosen arbitrarily, there will exist a Γ-rational profile without a π'-feasible outcome for each prime implicate π of Γ that entails π'. □

Now we can prove the following theorem.

Theorem 3.14. *Let Γ and Γ' be formulas such that $\Gamma \vDash \Gamma'$ and let F be a quota rule. F guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if for all prime implicates π' of Γ', there exists a prime implicate π of Γ such that $\pi \vDash \pi'$ and F guarantees π'-feasible outcomes on π-rational profiles.*

Proof. The right-to-left direction of this proof follows from Lemma 3.12. We let P' be the set of prime implicates of Γ'. Observe that Γ' is logically equivalent to

$\bigwedge_{\pi' \in P'} \pi'$. Furthermore, we let P be that set of prime implicates of Γ that are 'picked out' by the prime implicates of Γ'. We see that $\bigwedge_{\pi \in P} \pi$ will be such that $\text{Mod}(\Gamma) \subseteq \text{Mod}(\bigwedge_{\pi \in P} \pi)$. Therefore, we have that F guarantees Γ'-feasible outcomes on Γ-rational profiles.

For the left-to-right direction, we assume that F guarantees Γ'-feasible outcome on Γ-rational profiles. Therefore, for an arbitrary prime implicate π' of Γ', F guarantees π'-feasible outcomes on Γ-rational profiles. By Lemma 3.13, we can conclude that either F has a trivial quota with respect to π', or there is a prime implicate π of Γ, such that $\pi \vDash \pi'$ such that F satisfies $\sum\limits_{\varphi \in \text{Var}(\pi^-)} q(\varphi) + \sum\limits_{\varphi \in \text{Var}(\pi^+)} (n - q(\varphi) + 1) > n(k-1)$ with respect to the issues of π. These are the conditions of Theorem 3.7. Therefore, we can conclude that F guarantees π'-feasible outcomes on π-rational profile for some prime implicate π of Γ.

As π' was chosen arbitrarily, the statement holds for all prime implicates of Γ'. □

In order to shed light on the the previous result, we will now see an example of the theorem in practice, using a uniform quota rule.

Example 9. Consider five agents ($\mathcal{N} = \{a_1, a_2, a_3, a_4, a_5\}$) voting on the following agenda, $\Phi = \{\varphi_a, \varphi_b, \varphi_c, \varphi_d, \varphi_e, \varphi_f, \varphi_g, \varphi_h\}$. They have to vote in accordance with the rationality constraint, $\Gamma = (\varphi_a \vee \neg\varphi_b \vee \varphi_c) \wedge (\varphi_d \vee \neg\varphi_e \vee \varphi_f)$ and the outcome should be consistent with the feasibility constraint, $\Gamma' = (\varphi_a \vee \neg\varphi_b \vee \varphi_c \vee \varphi_g \vee \neg\varphi_h)$. Observe that $\Gamma \vDash \Gamma'$ holds. The only prime implicate of Γ' is itself. Furthermore, the only prime implicate of Γ which entails Γ' is $(\varphi_a \vee \neg\varphi_b \vee \varphi_c)$. We have that $n = 5$, $k_+ = 2$ and $k_- = 1$. Therefore, the uniform quota should be such that $u < 2$, for all $\varphi \in \text{Var}(\Phi)$. As we are only considering non-trivial quotas, the only quota for which we are guaranteed a Γ'-feasible outcome is when $u = 1$.

One can see that no Γ-rational profile will every produce an outcome which is not Γ'-feasible when the uniform quota is equal to 1. If 4 of the 5 agents vote against issue φ_b (thus it would not be rejected by the quota rule). For this to be a Γ-rational profile, the final agent would have to either vote for φ_a or φ_c or against φ_b. Thus, one of $\varphi_a, \neg\varphi_b$ or φ_c will have reached their quota and will be consistent with the outcome. Therefore, the outcome will be Γ'-feasible. △

3.3 Summary of Chapter 3

In this chapter, we looked at translating the k-median property from judgment aggregation to binary aggregation with constraints. The chapter splits into two sections, guaranteeing consistent outcome with respect to single clauses or any formulas. In the single-constraint case, Grandi and Endriss (2013) have translated the k-median property from judgment aggregation to the their setting. In Section 3.1, we followed the same steps as Grandi and Endriss, taking the k-median property from judgment aggregation to the rationality and feasibility constraint setting. We attained analogous results in this section as Grandi and Endriss did in their work.

In Section 3.2, we moved from considering single clause constraints to any formulas. We saw the existing results of Grandi and Endriss (2013) which links the ability to guarantee feasible outcomes when the constraints are single clauses to being able to guarantee feasible outcomes for the formula composed of the clauses. We extended this result, showing the link between guaranteeing feasible outcome with respect to a pair of formulas and their prime implicates. We recreated these two results in the rationality and feasibility setting. The main result being that is a quota rule guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if for each prime implicate π' of Γ', there exists a prime implicate π of Γ, such that the quota rule guarantees π'-feasible outcomes on π-rational profiles.

Chapter 4

Domain Restriction in Binary Aggregation

In this chapter, we will look at a property of profiles in judgment aggregation that guarantee give consistent outcomes. We will translate this property to our binary aggregation setting.

We introduce and motivate the translation domain restriction to binary aggregation. We approach two well-known domain restrictions, namely unidimensional alignment and value restriction. We motivate why our focus in the remainder is value restriction. We translate the corresponding Dietrich and List (2010) result from judgment aggregation to both binary aggregation settings.

4.1 An Introduction to Domain Restriction

As we've seen throughout this thesis, the majority rule cannot guarantee consistent results. In judgment aggregation, we try to get around outcomes which are inconsistent by finding properties of the agenda or the profiles. One way this is done is called *domain restriction*. The idea of domain restriction is that we limit the profiles that the majority rule can take as an input in order to be able to guarantee consistent outcomes. This is a more specific set of profiles than the set of consistent and complete profiles, $\mathcal{J}(X)$.

In the next subsection, we will address unidimensional alignment and why we are not translating this to our binary aggregation settings.

4.1.1 Unidimensional Alignment

Introduced by List (2003), one widely known way of restricting the domain of profiles is called *unidimensional alignment*. In judgment aggregation, this concerns only allowing the aggregation rule to receive profiles which are unidimensionally aligned. This can be thought of as the items of the agenda being ordered on a scale. For example, we could order political policies from extreme left-wing policies to extreme right-wing policies.

The profile is unidimensionally aligned if all agents' judgments agree with a subset of items that are adjacent in the scale. For example, if someone agrees with the most extreme left-wing policy, then they will only agree with the items adjacent to it on the scale, until policies becomes 'too right wing'. Then the agent will disagree with all remaining items of the agenda. On the other hand, a profile would not unidimensionally aligned if an agent agrees with items of the agenda which are not adjacent in the scale. For example, if an agent votes for both the most extreme left-wing and right-wing policies. The outcome of having profiles like this is that as all agent's submit consistent judgments. Then the median voter(s) judgment(s) will coincide with the outcome. This also links to notion of single-peaked preferences in preference aggregation, introduced by Black (1948).

In this thesis, we are nottranslating the notion of unidimensional alignment to binary aggregation with constraints. This is due to the fact that the rationality and feasibility constraints reflect a logical property of the profile. Whereas, unidimensional alignment does not. In the single-constraint case of binary aggregation, a translation of the corresponding consistency result may hold. This would be due to the median voter's judgment, which is rational with respect to the constraint, being the same as outcome. Therefore, the outcome will also abide by the constraint. A similar result could be attained in the two-constraint case with the extra condition that the rationality constraint entails the feasibility constraint. Therefore, the translation of unidimensional alignment to binary aggregation will not provide any further insight as to how the constraints can guaranteeing feasible outcomes.

4.1.2 Value Restriction

Next, we introduce the type of domain restriction which will be the main focus of the remainder of the chapter, namely value restriction. The idea of value restriction was introduced by Sen (1966) in preference aggregation. Voters are able to agree that one of the candidates is either the 'best' candidate, the 'worst' candidate, or the 'medium' (or middle) candidate. However, we are concerned with the translation of this value restriction in the context of judgment aggregation. Therefore, we will be focussing on the definition given by Dietrich and List (2010). For details on the notation of judgment aggregation we refer to Section 2.2.1.

Definition 4.1 (Value restriction, Dietrich and List, 2010)**.** A profile $\boldsymbol{J}$ is *value-restricted* if every minimally inconsistent subset $Y \overset{\text{m.i.}}{\subseteq} X$ of the agenda X, has a two-element subset $Z \subseteq Y$ that is not a subset of any $J_i \in \boldsymbol{J}$.

Using this definition in the formula-based model of judgment aggregation, there is the following result by Dietrich and List (2010) (Proposition $7(a)$) regarding consistent results from the majority rule with respect to the agenda X.

Proposition 4.1 (Dietrich and List, 2010)**.** For any profile $\boldsymbol{J}$ of consistent judgment sets, if $\boldsymbol{J}$ is value-restricted, then the majority outcome is consistent.

Not only do Dietrich and List (2010) introduce this definition of value-restricted profiles. They also extend the idea of unidimensional alignment, with the notions of single-canyonedness and single-plateauedness. They are similar to the notion of unidimensional alignment. However, instead of insisting that the items can be permuted in such a way that for all voters their judgements are collected into those accepted and those rejected, they are collected into three sections. Thus, creating a 'plateau' or a 'canyon' of accepted or rejected issues. Dietrich and List show the logical connections between these different types of domain restriction. They find that the class of value-restricted profiles contains all of the other types of domain restrictions mentioned. This further motivates our focus on translating value restriction to binary aggregation with constraints.

4.2 Binary Value Restriction with a Single Constraint

In this section, we will take the approach of value restriction in judgment aggregation and translate this into the single-constraint setting to recreate Theorem 4.1 from Dietrich and List. This gives sufficient conditions for consistent results in judgment aggregation under the majority rule. In the next section, we look at extending this to the rationality and feasibility model of binary aggregation.

We will now give the single-constraint binary aggregation analogues for both the definition of value restriction and the corresponding consistency result.

Definition 4.2 (Binary value-restricted profile with respect to a constraint)**.** A binary profile $\boldsymbol{B}$ is *value-restricted with respect to a constraint* Γ if and only if for all prime implicates π of Γ, there exist two distinct literals ℓ_i and ℓ_j of π such that no voter disagrees with both.

While we consider the analogue to value restriction with respect to a single constraint in binary aggregation, we will continue to use the rationality and feasibility notation and terminology. However, we are restricted to the rationality and feasibility constraints being the same.

Theorem 4.2. *Let n be odd and $\boldsymbol{B}$ be a Γ-rational profile. If $\boldsymbol{B}$ is a value-restricted profile with respect to Γ, then the majority rule guarantees Γ-feasible outcomes.*

Proof. Assume that there is a Γ-rational profile $\boldsymbol{B}$ which is value-restricted with respect to Γ. However, for the sake of a contradiction we assume that the outcome is not Γ-feasible. As the outcome is such that $F(\boldsymbol{B}) \nvDash \Gamma$ holds, there exists a prime implicate π of Γ such that $F(\boldsymbol{B}) \nvDash \pi$.

Without loss of generality, we can assume that π is a positive clause. Therefore, it can be denoted as such, i.e. $\pi = \bigvee_{\varphi \in \text{Var}(\pi)} \varphi$. This rewriting is possible due to the symmetric treatment of propositions under the majority rule in binary aggregation when n is odd. Therefore, any prime implicate can be rewritten as a positive clause where the roles of φ and $\neg\varphi$ are switched.[1] It is worth noting that disagreeing with a positive clause π is equivalent to rejecting all of the issues in π.

As $\boldsymbol{B}$ is a value-restricted profile with respect to Γ, for two distinct propositions $\varphi, \psi \in \text{Var}(\pi)$, no voter rejects both of them. Equivalently, everyone votes for at least one of φ or ψ. From this and the assumption that n is odd, it is clear that at least one of φ and ψ will gain a majority. This literal will be accepted under $F(\boldsymbol{B})$, and thus we have that $F(\boldsymbol{B}) \vDash \pi$ holds.

We have reached a contradiction, as we have that both $F(\boldsymbol{B}) \nvDash \pi$ and $F(\boldsymbol{B}) \vDash \pi$. □

It is worth observing that this is only the case if the number of voters is odd. Next, we will see an example showing that this is not the case if n is even.

Example 10. Consider two voters, $\mathcal{N} = \{a_1, a_2\}$ who are deciding on two issues, $\Phi = \{\varphi_a, \varphi_b\}$. The constraint which the agents have to vote in accordance with is $\Gamma = \varphi_a \vee \varphi_b$. Thus, both voters have to vote for one of the two issues. Consider the binary profile depicted in Table 4.1.

We see that the profile depicted in Table 4.1 is a Γ-rational profile which is also value-restricted with respect to Γ. However, the outcome is not Γ-feasible. This

[1] If π contains the literal $l = \neg\varphi$, then it can be replaced by $\varphi' = \neg\varphi$, thus $l = \varphi'$.

	φ_a	φ_b
a_1	✓	×
a_2	×	✓
Majority	×	×

Table 4.1: Value-restricted profile with respect to Γ, without a Γ-feasible outcome

is due to the majority rule favouring the rejection of the issues when there is a tie (rejection of an issue requires $\frac{n}{2}$ votes against the issue, while the issue needs $\frac{n}{2} + 1$ votes of support for it to be accepted). This is connected to the notion of completeness in judgment aggregation (see Subsection 2.2.1), where we want that either a formula or its complement will be in the outcome. In binary aggregation, this completeness is built into our model with respect to quota rules. If φ is not accepted, then $\neg\varphi$ is accepted.

Now suppose that $\Gamma = \neg\varphi_a \vee \neg\varphi_b$. Then the outcome would be Γ-feasible. The theorem would hold in this case. This mirrors the bias of the majority rule towards negated issues. Similarly, we see that when our constraint Γ has at least one negated issue, then we are guaranteed Γ-feasible outcomes. $\triangle$

Here we see a distinction between the theorem by Dietrich and List (Theorem 4.1 in this thesis) and our translation of this result in Theorem 4.2. To guarantee consistent outcomes, or Γ-feasible results, we need to assume that n is odd. Dietrich and List do not need this extra assumption to guarantee consistent results in judgment aggregation, as the outcomes do not need to be complete. In the case when n is even, and exactly half of the voters vote for φ and the remaining voters for $\neg\varphi$, the outcome would include neither φ nor $\neg\varphi$. However, in binary aggregation, we are guaranteed that the result will be complete due to the way in which the majority rule and more generally, quota rules are defined. However, as alluded to in the previous example, this does not mean that a similar notion of value restriction cannot be used when n is even. Still, it is more restrictive than Definition 4.2.

Definition 4.3 (Negatively value-restricted profile with respect to a constraint)**.** A binary profile $\boldsymbol{B}$ is *negatively value-restricted with respect to a constraint* Γ if and only if for all prime implicates π of Γ, there exist two distinct literals ℓ_i and ℓ_j of π such that no voter disagrees with both ℓ_i and ℓ_j, and at least one of them is a negated issue.

It is worth noting here that all profiles that are negatively value-restricted with respect to some constraint Γ are also value-restricted with respect to Γ.

Theorem 4.3. *Let $\boldsymbol{B}$ be a Γ-rational profile. If $\boldsymbol{B}$ is negatively value-restricted with respect to a constraint Γ, then the majority rule guarantees Γ-feasible outcomes.*

Proof. When n is odd the claim follows from Theorem 4.2, so suppose n is even.

When n is even, we assume that there is a Γ-rational profile $\boldsymbol{B}$ which is negatively value-restricted with respect to Γ. However, for the sake of a contradiction we assume that the outcome is not Γ-feasible. As the outcome is such that $F(\boldsymbol{B}) \nvDash \Gamma$, there exists a prime implicate π of Γ such that $F(\boldsymbol{B}) \nvDash \pi$.

As the profile is negatively value-restricted with respect to Γ, the prime implicate π has two literals, namely ℓ_i and ℓ_j, such that no voter disagrees with both of them. Furthermore, we know that one of them is a negated issue. Without loss of generality we will assume that ℓ_i is a negated issue, without making any assumption about whether ℓ_j is a negated issue or not.

As $F(\boldsymbol{B}) \nvDash \pi$ holds and ℓ_i is the a negated issue, there can be at most $\frac{n}{2} - 1$ votes agreeing with the literal ℓ_i. However, the remaining agents need to agree with ℓ_j. Thus, we have $\frac{n}{2} + 1$ votes agreeing with the literal ℓ_j. Therefore, ℓ_j will have received enough support such that it is consistent with the outcome, whether it is a positive or negative literal. Therefore, we have that $F(\boldsymbol{B}) \vDash \pi$.

We have reached a contradiction as we have that both $F(\boldsymbol{B}) \vDash \pi$ and $F(\boldsymbol{B}) \nvDash \pi$. □

The consequence of Theorem 4.3 is that although we have a more limiting sense of a profile being value-restricted with respect to a constraint, it can account for any number of voters. Thus, there is a trade-off in usefulness in two senses: one theorem can speak about only specific groups of voters and the other restricts the domain of profiles a lot further. However, we see that when n is odd we can look to Theorem 4.2 for guidance and to Theorem 4.3 when n is even.

Next, we will move away from the single-constraint case and extend the work in this section to the two-constraint case.

4.3 Binary Value Restriction with Rationality and Feasibility Constraints

In this section, we will extend the results from the previous one, moving from the single-constraint setting to the rationality and feasibility setting of binary aggregation. Next we take Definition 4.2 and extend to be with respect to a pair of constraints, rather than one.

Definition 4.4 (Binary value-restricted profile with respect to a pair of constraints). A binary profile $\boldsymbol{B}$ is *value-restricted with respect to a pair of constraints* (Γ, Γ') if and only if for every prime implicate π' of Γ' there is exists a prime implicate π of Γ such that $\pi \vDash \pi'$ holds, and there are two distinct literals ℓ_i and ℓ_j of π such that no voter disagrees with both.

It is worth noting here, that if the profile is value-restricted with respect to the pair of formulas, we see that for all prime implicates of the feasibility constraint there exists a prime implicate of the rationality constraint such that the latter entails the former. By Lemma 2.5, this holds if the rationality constraint entails the feasibility constraint. We see that the converse of Lemma 2.5 holds as well.

Lemma 4.4. *If for all prime implicates π' of Γ', there exists a prime implicate π of Γ such that $\pi \vDash \pi'$, then $\Gamma \vDash \Gamma'$.*

The proof of Lemma 4.4 can be found in Appendix A.1. Therefore, we can see that if a profile is value-restricted with respect to a pair of formulas, this means that the rationality constraint entails the feasibility constraint.

Next, as in Section 4.2, we will recreate the result of Dietrich and List (2010). However, now we are considering a pair of constraints.

Theorem 4.5. *Let n be odd and $\boldsymbol{B}$ be a Γ-rational profile. If $\boldsymbol{B}$ is value-restricted with respect to the pair of constraints (Γ, Γ'), then the majority rule guarantees Γ'-feasible outcomes.*

Proof. Assume that there is a Γ-rational profile $\boldsymbol{B}$ which is also value-restricted with respect to (Γ, Γ'). For the sake of a contradiction, assume that $F(\boldsymbol{B}) \nvDash \Gamma'$.

We see that as $F(\boldsymbol{B}) \nvDash \Gamma'$, there must exist a prime implicate π' of Γ', such that $F(\boldsymbol{B}) \nvDash \pi'$.

From the assumption that $\boldsymbol{B}$ is a value-restricted profile with respect to (Γ, Γ'), there exists a prime implicate π of Γ which entails π'. Furthermore, π has two literals such that no voter disagrees with them both. Since π entails π', we also have $F(\boldsymbol{B}) \nvDash \pi$.

Due to the symmetric treatment of issues under the majority rule when n is odd, we can assume that π is a positive clause. Therefore, there exists $\varphi, \psi \in \text{Var}(\pi)$ that are literals of π. Furthermore, no voter disagrees with both φ and ψ. Equivalently, all voters vote for at least one of them. It is clear that the this implies that one of

φ or ψ will be accepted by the majority rule. Therefore, the outcome will satisfy π, thus $F(\boldsymbol{B}) \vDash \pi$.

Therefore, we have reached a contradiction, as we have that both $F(\boldsymbol{B}) \nvDash \pi$ and $F(\boldsymbol{B}) \vDash \pi$ hold. □

We see that a profile being value-restricted with respect to a single constraint is a special case of a profile being value-restricted with respect to a pair of formulas, where Γ and Γ' are logically equivalent. Next, we see an example where Γ and Γ' are different, thus showing that Theorem 4.5 is more expressive than Theorem 4.2.

Example 11. We consider three voters, $\mathcal{N} = \{a_1, a_2, a_3\}$, who are voting on agenda $\Phi = \{\varphi_a, \varphi_b, \varphi_c, \varphi_d, \varphi_e, \varphi_f\}$. The voter's judgments have to abide by the following rationality constraint: $\Gamma = (\varphi_a \vee \varphi_b \vee \varphi_c) \wedge (\varphi_d \vee \varphi_e \vee \varphi_f)$. Furthermore, the outcome should abide by the following feasibility constraint: $\Gamma' = (\varphi_a \vee \varphi_b \vee \varphi_c \vee \varphi_d \vee \varphi_e \vee \varphi_f)$ (observe that $\Gamma \vDash \Gamma'$). Now consider the Γ-rational profile depicted in Table 4.2.

	φ_a	φ_b	φ_c	φ_d	φ_e	φ_f
a_1	✓	×	×	✓	×	×
a_2	×	✓	×	×	✓	×
a_3	×	×	✓	✓	×	×
Majority	×	×	×	✓	×	×

Table 4.2: Value-restricted profile with respect to (Γ, Γ')

We see that the profile is value-restricted with respect to (Γ, Γ'). The only prime implicate of Γ' is itself. Γ' is entailed by the clause $(\varphi_d \vee \varphi_e \vee \varphi_f)$, and all voters have accepted either φ_d or φ_e. This leads to φ_d being accepted by the majority rule.

We see that this is a Γ-rational profile with a Γ'-feasible outcome under the majority rule, even though the pair (Γ, Γ') is not simple. Therefore, our result shows a situation with Γ'-feasible outcome. However, it does not abide by the conditions of Theorem 2.6.

Furthermore, observe that this profile is not value-restricted with respect to the rationality constraint Γ, showing that Theorem 4.5 can be more expressive than Theorem 4.2. However, we see that this profile is value-restricted with respect to the feasibility constraint. △

As with Theorem 4.2, we see that Theorem 4.5 only holds for the case when n is odd. However, we can define another notion of value restriction in this setting which we can guarantee feasible outcomes for any n. Thus, as before, we use a more limiting definition to restrict our domain.

Definition 4.5 (Negatively value-restricted profile with respect to a pair of formulas). A binary profile $\boldsymbol{B}$ is *negatively value-restricted with respect to a pair of formulas* (Γ, Γ') if and only if for every prime implicate π' of Γ', there exists a prime implicate π of Γ such that $\pi \vDash \pi'$ holds, and there exist two distinct literals ℓ_i and ℓ_j of π such that no voter disagrees with them both, and at least one of them is a negated issue.

Observe that all profiles that are negatively value-restricted with respect to some pair of constraints (Γ, Γ'), are also value-restricted with respect to (Γ, Γ'). Furthermore, if a profile is negatively value-restricted with respect to some pair of constraints (Γ, Γ'), then it is also the case that Γ entails Γ' by Lemma 4.4. Now we have the following result.

Theorem 4.6. *Let $\boldsymbol{B}$ be a Γ-rational profile. If $\boldsymbol{B}$ is negatively value-restricted with respect to a pair of constraints (Γ, Γ'), then the majority rule guarantees Γ'-feasible outcomes.*

Proof. When n is odd the claim follows from Theorem 4.5, so suppose n is even.

When n is even we assume that there is a Γ-rational profile $\boldsymbol{B}$ which is negatively value-restricted with respect to (Γ, Γ'). However, for the sake of a contradiction we assume that the outcome is not Γ-feasible, i.e. $F(\boldsymbol{B}) \nvDash \Gamma'$.

We see that as $F(\boldsymbol{B}) \nvDash \Gamma'$, there must exist a prime implicate π' of Γ', such that $F(\boldsymbol{B}) \nvDash \pi'$.

From the assumption that $\boldsymbol{B}$ is a negatively value-restricted profile with respect to (Γ, Γ'), there exists a prime implicate π of Γ such that the conditions in Definition 4.5 hold. Thus, π has two literals, namely ℓ_i and ℓ_j, such that no voter disagrees with them both. Moreover, at least one of these literals is a negated issue. Without loss of generality we will assume that ℓ_i is a negated issue without making any assumptions on whether ℓ_j is negated or not. Additionally, as π' cannot be entailed by the outcome, we also have $F(\boldsymbol{B}) \nvDash \pi$.

As $F(\boldsymbol{B}) \nvDash \pi$ and ℓ_i is a negated issue, there can be at most $\frac{n}{2} - 1$ votes agreeing with the literal ℓ_i. However, the remaining voters need to agree with ℓ_j. Therefore, we have at least $\frac{n}{2} + 1$ votes agreeing with the literal ℓ_j. Thus, ℓ_j will have received enough support such that the outcome is consistent with it, whether it is a positive or negative literal. Therefore, we have that $F(\boldsymbol{B}) \vDash \pi$.

We have reached a contradiction as we have that both $F(\boldsymbol{B}) \vDash \pi$ and $F(\boldsymbol{B}) \nvDash \pi$. $\square$

The consequences of Theorem 4.6 are similar to the consequences of Theorem 4.3.

Therefore, we see that when n is odd, we can look to Theorem 4.5 and to Theorem 4.6 when n is even.

4.4 Summary of Chapter 4

In this chapter, we looked at translating domain restriction to binary aggregation with constraints. We focussed on value restriction, rather than the other types of domain restriction- such as unidimensional alignment. We translated value restriction from judgment aggregation to both the single-constraint setting and the two-constraint setting of binary aggregation.

For both settings, we recreated the consistency result of Dietrich and List (2010) (Proposition 4.1 in this thesis). Therefore, if our binary profile is value-restricted with respect to a constraint (or a pair of constraints), then we are guaranteed feasible outcomes with respect to the constraint (or with respect to the feasibility constraint). These results required an extra assumption that n is odd, whereas in the original Dietrich and List (2010) result, this extra assumption is not needed. This is due to the majority rule in binary aggregation favouring the rejection of an issue when n is even. We then go on to define a new notion of value restriction. Namely, *negatively value-restricted* profiles with respect to either a single constraint or a pair of constraints. We see in both of the binary aggregation settings that this guarantees feasible outcomes, with no assumption of whether n is odd or even.

In this chapter, we have shed light on another method with which we can guarantee feasible outcomes on rational profiles. This way can cover more cases than the method from Endriss (2018) (see Theorem 2.6). Moreover, all rational profiles for which the pair of formulas is simple, are also value-restricted with respect to the pair of formulas.

Chapter 5

Computational Complexity of Guaranteeing Feasible Outcomes

In this chapter, we provide computational complexity results relating to some of the properties explored in Chapters 3 and 4. We focus on three decision problems in this chapter, regarding the simplicity of pairs of constraint formulas and value-restricted profiles. Note that here we will focus problems from both judgment aggregation and binary aggregation with rationality and feasibility constraint. We do not focus on the single-constraint case as it is a special case of two-constraint case.

In Section 5.1, we will give an introduction to the relevant background of computational complexity in order to understand the remainder of the chapter.We introduce the complexity class Π_2^p, as well as some problems that will appear in the proofs in this chapter.

In Section 5.2, we will inspect a decision problem, we call PAIRSIMPLE. This problem relates to checking if a pair of formulas is *simple*, according to Definition 2.18 (Endriss, 2018). Throughout this thesis, we have seen thata formulas being simple corresponds to the median property (Definition 2.13) in judgment aggregation. As an analysis of the complexity of checking if an agenda has the median property has been carried out by Endriss et al. (2012). It is of interest to compare the complexity classes which these two similar problems belong to. After this, we will prove membership of PAIRSIMPLE in Π_2^p and then we will go on to show that PAIRSIMPLE is also a **coNP**-hard problem. This supports our claim PAIRSIMPLE is Π_2^p-complete.

In Section 5.3, we will look at the complexity of checking if a profile is value-restricted with respect to an agenda in the formula-based model of judgment aggregation. We will call this problem VALUERESTRICTED. Here we will prove that

VALUERESTRICTED is in Π_2^p. We then look at the analogue of this problem in binary aggregation (Definition 4.4), which we will call BINVALUERESTRICTED. We go on to show that this problem is too in Π_2^p and $\mathbf{coNP}$-hard. After which we prove that VALUERESTRICTED is at least as hard as BINVALUERESTRICTED, by giving a reduction.

In Section 5.4, we will explore the results of this chapter, comparing them to the aims of the chapter. Furthermore, we will draw on the results of Endriss et al. (2016), which focusses on the succinctness of the languages of judgment aggregation and binary aggregation with constraints. Following this we will summarise the chapter.

5.1 An Introduction to Computational Complexity

In this section, we will cover enough background information on computational complexity for the reader to be able to understand the remainder of the chapter. However, we assume some prior understanding of computational complexity. We direct the reader to the textbook by Arora and Barak (2009) for more details. Furthermore, we assume throughout this chapter that $\mathbf{P} \neq \mathbf{NP}$, and that the polynomial hierarchy does not collapse to any point below the third level.

5.1.1 Complexity Classes and Complete Problems

In this chapter, the main complexity class that we will be working with is Π_2^p, a class in the second level of the polynomial hierarchy. Π_2^p is also known as $\mathbf{coNP^{NP}}$, or "$\mathbf{coNP}$ with an $\mathbf{NP}$-oracle" (Arora and Barak, 2009, Section 5.2).

This chapter aims to prove that the problems in question are Π_2^p-complete. For a problem to be Π_2^p-complete it has to belong to the class Π_2^p, as well as be Π_2^p-hard. To prove that a problem L is in Π_2^p for a given input x, we need to find a certificate to affirm that $x \notin L$. This certificate must be checked in polynomial time, with access to an oracle which answers queries of an $\mathbf{NP}$-complete problem. This could be a problem such as SAT. Once membership has been shown, this gives the problem an upper bound to the class that the problem can be complete for.

To prove Π_2^p-hardness of a candidate problem, we take a problem known to be a Π_2^p-complete problem and show that we can reduce the Π_2^p-complete problem to our candidate problem with a polynomial reduction (Arora and Barak, 2009, Definition 2.7). This reduction has to be computable in polynomial time and consists of a mapping from our Π_2^p-complete problem to our candidate problem. This mapping means we can use an algorithm for our candidate problem to give a solution for the Π_2^p-complete problem. Therefore, we can informally see this as giving a lower bound on the hardest class that the problem can belong to. Note that other hardness results for other complexity classes follow the same structure. However, they use a problem complete for that class.

5.1.2 The Complexity of Checking Implicates of a Formula

In this subsection, we will introduce the problem NOTIMPLICATE, a problem which will check if a clause π is not an implicate of a formula Γ.

NOTIMPLICATE

- Input: A formula Γ and a clause π
- Question: Is π not an implicate of Γ?

Throughout this thesis, we have used Definition 2.16 to characterise the prime implicates of a formula. Checking if a clause is a prime implicate by this definition requires finding a list of all other implicates to check that the second condition holds- that any implicate that entails the clause is also logically equivalent to it. However, this is computationally expensive. Therefore, we will describe a more efficient way of checking if a clause is a prime implicate of a formula, using our decision problem NOTIMPLICATE. However, first we will define what is meant by a *sub-clause* of clause.

Definition 5.1 (sub-clause)**.** Let π be a clause. A clause π' is a *sub-clause* of π if and only if every literal of π' is also a literal of π.

Next, we will outline a different method of checking if a clause is a prime implicate using a more efficient method.

Remark 1. To check if a clause π is a prime implicate of a formula Γ, we need to check two properties. First, the clause needs to be an implicate of Γ. Thus, we need to check that $\Gamma \vDash \pi$. Second, we need to check that no sub-clause of π is an implicate of Γ. It is sufficient to check if all sub-clauses of π containing just one literal less than π are not implicates of Γ, as this entails that any shorter sub-clause will also not be an implicate of Γ.

Therefore, we can see that this method of checking a single clause does not require a list of candidate prime implicates of the formula. Furthermore, we can use the problem NOTIMPLICATE to help us check if a clause is a prime implicate. Now we will see an example of how we can use the previous method.

Example 12. Consider the formula $\Gamma = (\varphi_a \vee \varphi_b \vee \neg\varphi_c) \wedge (\varphi_a \vee \varphi_b)$ and the clause $\pi = (\varphi_a \vee \varphi_b \vee \neg\varphi_c)$. It is clear that $\Gamma \vDash \pi$, therefore π is an implicate of Γ. Next, we have to check that all sub-clauses of π containing two literals are not implicates of Γ. As $\Gamma \not\models (\varphi_a \vee \neg\varphi_c)$ and $\Gamma \not\models (\varphi_b \vee \neg\varphi_c)$, neither $(\varphi_a \vee \neg\varphi_c)$ nor $(\varphi_b \vee \neg\varphi_c)$ are implicates of Γ. However, $(\varphi_a \vee \varphi_b)$ is an implicate of Γ as $\Gamma \models (\varphi_a \vee \varphi_b)$. Therefore, we can conclude that π is not a prime implicate of Γ, as there exists a sub-clause of π which is an implicate of Γ. △

Hence, we can use the problem NOTIMPLICATE to check if a clause is a prime implicate of a formula. Furthermore, observe that NOTIMPLICATE is a variation of the **NP** decision problem SAT.[1] Therefore, we see that NOTIMPLICATE is an **NP**-complete problem. In Section 5.2, we will use the problem NOTIMPLICATE as an oracle for the proof that PAIRSIMPLE is in Π_2^p. Therefore, the oracle will be used to check if a clause will be a prime implicate of the formula.

5.2 The Complexity of Simplicity

In this section, we will look at the complexity of checking if a pair of constraints is simple by Definition 2.18 (Endriss, 2018). We will first inspect the problem PAIRSIMPLE, then we will prove some results supporting the problem being Π_2^p-complete.

PAIRSIMPLE
- Input: A pair of constraints (Γ, Γ')
- Question: Is the pair of constraints (Γ, Γ') simple?

The analogue of this problem in judgment aggregation, checking if an agenda has the median property, is a Π_2^p-complete problem (Endriss et al., 2012). This motivates us to look at the membership of PAIRSIMPLE in Π_2^p.

Lemma 5.1. *The problem* PAIRSIMPLE *is in* Π_2^p.

Proof. To show membership of PAIRSIMPLE in the complexity class Π_2^p, we need an algorithm that decides on the correctness of a certificate for the violation of the input (Γ, Γ') being simple. This certificate must be checked in polynomial time with access to an **NP**-oracle. Here we let the oracle have access to the **NP**-complete problem, NOTIMPLICATE, described in Section 5.1.2.

Our certificate π' is a clause, with $\pi' \in \mathcal{L}(\mathrm{Var}(\Gamma'))$. This certificate needs to have the following two properties in order for (Γ, Γ') not to be simple: (1) π' needs to be a prime implicate of Γ', and (2) there are no prime implicates of Γ, which entail π' and are simple. We let the number of literals in π' be m ($|\mathrm{Var}(\pi')| = m$).

[1] If π is not an implicate of a formula Γ, then $\Gamma \not\models \pi$ holds. Therefore, we need a certificate which satisfies Γ, but not π. This would be the same as checking if $\Gamma \wedge \neg\pi$ is satisfiable.

1. First we need to check that π' is an implicate of Γ', we can check this with one call to the oracle. Once, we have established that it is an implicate of Γ', we check that every sub-clause of π' containing $m-1$ literals is not an implicate of Γ'. This takes m queries to the oracle. In total checking that π' is a prime implicate of Γ' takes $m+1$ queries to the oracle.

2. Second, we let P be the set of candidate simple prime implicates of Γ which entail π', $P = \{\pi \mid \pi \text{ is a simple clause}, \pi \vDash \pi' \text{ holds, and } \mathrm{Var}(\pi) \subseteq \mathrm{Var}(\pi')\}$. As $\pi \vDash \pi'$, it is clear that each $\pi \in P$ is a sub-clause of π' with either one or two literals. As π' has m literals, the size of the set P is $|P| \leqslant \sum_{i=1}^{2} \binom{m}{i} = \frac{m(m+1)}{2}$. We need to call to the oracle for each $\pi \in P$ to check that none of them are implicates of Γ. This takes at most $\frac{m(m+1)}{2}$ calls to the oracle.

Therefore, we can verify the correctness of a certificate π' for the violation of the input (Γ, Γ') being simple in $\frac{m(m+3)}{2} + 1$ calls to the oracle. Therefore, this can be computed in polynomial time. Thus, we can conclude that PAIRSIMPLE is in Π_2^p. □

With this result, we know that the decision problem cannot be complete for any class higher in the hierarchy than Π_2^p. Our next step is to prove hardness. However, here we do not prove Π_2^p-hardness, but rather **coNP**-hardness. We will expand on the consequence of this in Section 5.4.

Lemma 5.2. PAIRSIMPLE *is* **coNP**-*hard.*

Proof. We shall show with a reduction from UNSAT (a **coNP**-complete problem) that PAIRSIMPLE is **coNP**-hard.

Take an input formula of UNSAT, φ, then map this onto the pair of constraints (Γ, Γ'). Let $\Gamma = a$, where $a \notin \mathrm{Var}(\varphi)$ and $\Gamma' = \neg\varphi$. It is clear that the function which takes φ to the pair (Γ, Γ') is computable in polynomial time. Now we will show that a positive (negative) answer to the problem PAIRSIMPLE under this mapping means that we have a positive (negative) answer to UNSAT.

First consider the case where $\varphi \in$ UNSAT. As φ is not satisfiable, it is clear that $\neg\varphi$ is a tautology. By definition 2.15, an implicate cannot be a tautology, therefore, Γ' has no prime implicates. Therefore, it is vacuously true that for all prime implicates of Γ' that there exists a prime implicate of Γ such that the latter entails the former and the latter has at most two literals.

In the second case, we consider the case when $\varphi \notin$ UNSAT. Then, φ is satisfiable, and therefore, $\neg\varphi$ is not a tautology. Therefore, there will exist a prime implicate π' of Γ' (such that $\mathrm{Var}(\pi') \subseteq \mathrm{Var}(\Gamma')$). As the only prime implicate of Γ is a, and

a is a fresh variable, it is clear that this prime implicate will not be entailed by a. Therefore, the pair of constraints is not simple. □

We know that the problem PAIRSIMPLE is at least as hard as the hardest problems in the class **coNP**, as we are able to solve a **coNP**-complete problem using the algorithm for PAIRSIMPLE. As we have shown membership of PAIRSIMPLE in Π_2^p and that it is **coNP**-hard, we can conclude that PAIRSIMPLE is complete for a class which is at least as hard as **coNP** and no harder than the class Π_2^p.

5.3 The Complexity of Value Restriction

In this section, we investigate the complexity of checking if a profile is value-restricted (Definition 4.1). Then we will inspect the problem of a binary profile being value-restricted (Definition 4.4). Therefore, this section relates to Chapter 4 of this thesis. First, we will introduce the problem, which we will call VALUERESTRICTED. Following this, we show that this problem and its binary aggregation analogue, BINVALUERESTRICTED, are both in Π_2^p. We go on to show that BINVALUERESTRICTED is **coNP**-hard. To prove that VALUERESTRICTED is **coNP**-hard as well, we will reduce VALUERESTRICTED to BINVALUERESTRICTED.

VALUERESTRICTED
- Input: A profile $\boldsymbol{J}$ and an agenda X
- Question: Is the profile $\boldsymbol{J}$ value-restricted with respect to X?

In the following lemma, we will see that the problem of checking if a profile is value restricted is in Π_2^p. An intuitive argument for this is due to the structure of the polynomial hierarchy. We see that Π_2^p fits problems which follow the quantification pattern 'for all ... there exists ...'. This pattern fits with the definition of value restriction, as we need to check that for all minimally inconsistent subsets of the agenda, there exists a two-element subset that has a certain property.

Lemma 5.3. *The problem* VALUERESTRICTED *is in* Π_2^p.

Proof. To prove that the decision problem VALUERESTRICTED is in Π_2^p we need to provide an algorithm with access to an **NP**-oracle. This algorithm decides the correctness of a certificate for the violation of $\boldsymbol{J}$ being a value-restricted profile. We will take the oracle's **NP**-complete problem to be SAT.

We consider the certificate $\Delta \subseteq X$, where $|\Delta| = m$. Our certificate Δ should be a minimally inconsistent subset of the agenda, X. Furthermore, Δ should be such that for every two element subset of Δ, there is an agent who accepts both of them.

Therefore, Δ has to satisfy the following: (1) Δ is an inconsistent subset, (2) Δ is minimally inconsistent and (3) we need to check that the profile property does not hold. We now check the time required to check these three criteria.

1. The set Δ needs to be inconsistent. This is equivalent to checking if the conjunction of the formulas in Δ is not satisfiable. This can be done with one query to the SAT oracle.
2. Next we need to check that Δ is a minimally inconsistent subset of X. Therefore, we need to check that each $\Theta \subseteq \Delta$, with $|\Theta| = m - 1$, is consistent (note that there are m of these subsets). We do this by taking the conjunction the formulas in each Θ. Then we query if the conjunction of the members of Θ is satisfiable. Therefore, to check that all of these m subsets of Δ are consistent we need m queries to the SAT oracle.
3. The last part we have to check is that there are no two-element subsets of Δ such that each voter rejects one of them. Alternatively, for each two-element subset of Δ, one agent must accept both elements. Therefore, there are $\frac{m(m-1)}{2}$ two-element subsets for which we have to check that at least one voter accepted both elements in the subset. This can be done in $\mathcal{O}(\frac{m(m-1)}{2})$ steps.

Therefore, we have a polynomial time algorithm that checks the correctness of a certificate for the violation of $\boldsymbol{J}$ being a value-restricted profile with respect to an agenda X, with access to a SAT oracle. □

We have shown the membership of VALUERESTRCITED in Π_2^p. Thus, the problem cannot be complete for any class harder than Π_2^p. Now we move on to look at the analogue problem to VALUERESTRICTED in the binary aggregation setting (described in Section 4.2), which we will call BINVALUERESTRICTED.

BINVALUERESTRICTED
- Input: A pair of constraints, (Γ, Γ'), a Γ-rational profile $\boldsymbol{B}$
- Question: Is the profile $\boldsymbol{B}$ value-restricted with respect to (Γ, Γ')?

Next we shall show that this problem, like the judgment aggregation analogue, is in Π_2^p. Again, this is intuitively the correct class for this problem, as we need to check that for all prime implicates of the feasibility constraint, that there exists a prime implicate of the rationality constraint such that it has a certain property.

Lemma 5.4. *The problem* BINVALUERESTRICTED *is in* Π_2^p.

Proof. To prove that the decision problem BINVALUERESTRICTED is in Π_2^p we need to provide an algorithm with access to an **NP**-oracle. This algorithm decides the correctness of a certificate for the violation of $\boldsymbol{B}$ being a value-restricted profile. We let the oracle have access to the $\mathbf{NP}$-complete problem, NOTIMPLICATE.

We take our certificate to be a pair of clauses, (π, π'). For this to be a certificate that verifies that the input is not value-restricted, it needs to be that case that π and π' are prime implicates of Γ and Γ', respectively. Furthermore, it should be the case that $\pi \vDash \pi'$. Once this has been done, we need to check that for every pair of literals of π, there is an agent who disagrees with both of these literals. This is broken down into the following four steps, where we assume that $|\text{Var}(\pi)| = m, |\text{Var}(\pi')| = m'$ with $m \leqslant m'$.

1. Checking if π' is a prime implicate of Γ' first requires us to check that π' is an implicate of Γ'. This takes one call to the oracle. Then to check that it is a prime implicate, we have to check that any sub-clause of π' (containing one literal less than π') cannot be a implicate of Γ'. This requires m' calls to the NOTIMPLICATE oracle for each sub-clause of π' containing $m' - 1$ literals. This step takes $m' + 1$ calls to the oracle.

2. As with the previous point, we need to check if π is a prime implicate of Γ. This requires $m+1$ calls to the NOTIMPLICATE oracle. One for each sub-clause of π containing $m - 1$ literals, and one to check that π is an implicate of Γ.

3. Next to check is that $\pi \vDash \pi'$ holds, or equivalently, that π' is an implicate of π. Therefore, we need a negative response from the NOTIMPLICATE oracle. This takes one call to the oracle.

4. Finally, we need to check that for all pairs of literals in π, there is an agent that disagrees with both of the literals. This can be done in time $\mathcal{O}(\frac{m}{2}(m-1))$.

Therefore, we can check if a certificate shows that an input has a negative answer in polynomial time with access to an $\mathbf{NP}$-oracle. Therefore we have shown membership of BINVALUERESTRICTED in Π_2^p. □

Next, we will not show that BINVALUERESTRICTED is Π_2^p-hard. Instead, we will show that it is $\mathbf{coNP}$-hard. This is a weaker result, but it does reduce the number classes for which BINVALUERESTRICTED can be complete.

Lemma 5.5. BINVALUERESTRICTED *is* $\mathbf{coNP}$*-hard.*

Proof. We show that BINVALUERESTRICTED is $\mathbf{coNP}$-hard by giving a polynomial reduction from UNSAT to BINVALUERESTRICTED. Therefore, we have an input for UNSAT, a well-formed formula φ. We map this onto the following input of BIN-VALUERESTRICTED. We let the positive section of the agenda be $\Phi^+ = \text{Var}(\varphi) \cup \{a\}$, where $a \notin \text{Var}(\varphi)$. Our constraints are $\Gamma = a$ and $\Gamma' = \neg\varphi$. We have a unanimous profile $\boldsymbol{B}$, where for all $i \in \mathcal{N}$ we have that $B_i \vDash \Phi^+$. This translation can be computed in polynomial time.

Now we have to show that there is a positive answer to the input UNSAT if and only if the translation of the input gives a positive answer of BINVALUERESTRICTED. Therefore, we first assume that φ is not satisfiable. Then, $\neg\varphi$ is a tautology. Therefore, there are no prime implicates of Γ'. As before, we see that our profile is value-restricted with respect to the pair of formulas.

Now we assume that φ is a negative input for UNSAT. Therefore, φ is satisfiable. Furthermore, we do not know if $\Gamma' = \neg\varphi$ is satisfiable or not, but we know that $\neg\varphi$ is not a tautology. Therefore, we know that it has a prime implicate π', such that $\text{Var}(\pi') \subseteq \text{Var}(\varphi)$. The only prime implicate of Γ is $\pi = a$ and it is clear that for the prime implicate π' there is no prime implicate of Γ such that it entails π', as $\pi \nvDash \pi'$. Therefore, this is not a binary value restricted profile with respect to the constraints. □

From this result, we know that the problem BINVALUERESTRICTED will be complete for a class at least as hard than **coNP** and no harder than Π_2^p. Now we want to see the relationship between BINVALUERESTRICTED and VALUERESTRICTED, which we will see in the following lemma.

Lemma 5.6. VALUERESTRICTED *is at least as hard as* BINVALUERESTRICTED.

Proof. We show this by a reduction from BINVALUERESTRICTED to VALUERESTRICTED. Therefore we start with a valid input for the problem BINVALUERESTRICTED, namely (Γ, Γ') and a Γ-rational profile $\boldsymbol{B}$. We need to translate this input to a valid input of VALUERESTRICTED. We let the new agenda X be such that $X^+ = \{\Gamma'\} \cup \text{Var}(\Gamma')$. Then we translate $\boldsymbol{B}$ into $\boldsymbol{J}$ as such: given B_i and $\varphi \in X$, if $B_i \vDash \varphi$ then $\varphi \in J_i$, otherwise $\neg\varphi \in J_i$. This translation takes polynomial time, where the only computational expensive part is checking if $B_i \vDash \Gamma'$. However, this is computable in polynomial time and is repeated n time. Thus, the overall time is polynomial.

Now we have to show that we can use this translation to decide the outcome of BINVALUERESTRICTED using the algorithm for VALUERESTRICTED. We first assume that $\boldsymbol{B}$ is value-restricted with respect to (Γ, Γ'). As this is the case, we can assume that $\Gamma \vDash \Gamma'$, as this is entailed by the profile being value restricted with respect to a pair of constraints (following from Lemma 4.4). Therefore, as $\text{Mod}(\Gamma) \subseteq \text{Mod}(\Gamma')$, we see that $\boldsymbol{B}$ is also a Γ'-rational profile and $\Gamma' \in J_i$ for all $i \in \mathcal{N}$. We see that any minimally inconsistent subset of X (other than a member of X and its complement) will either contain Γ or $\neg\Gamma'$. If the minimally inconsistent subset contains $\neg\Gamma'$, then as $\boldsymbol{J}$ is a Γ-rational profile, it will also be a Γ'-rational profile. Everyone will reject $\neg\Gamma'$. Therefore, $\boldsymbol{J}$ will be value-restricted when considering this minimally inconsistent subset. If the minimally inconsistent subset contains Γ', then it will contain Γ' and the negation of literals of one of its prime implicates. That is, it will contain: $\{\Gamma', \neg\ell_1, ..., \neg\ell_k\}$, where $\pi' = \bigvee_{i=1}^{k} \ell_i$ is a prime implicate of Γ'. As $\boldsymbol{B}$ is a value-restricted profile with respect to (Γ, Γ'), it is clear that $\boldsymbol{J}$ is value-restricted with respect to the agenda X.

Now assume that $\boldsymbol{B}$ is not a value-restricted profile with respect to (Γ, Γ'). Then there exists a prime implicate π' of Γ' such that for all prime implicates π of Γ, such that $\pi \models \pi'$ and for all pairs of literals of π, there is an agent who disagrees with both of them. We see that after mapping the input to the judgment aggregation setting, this means that there is a minimally inconsistent subset containing Γ' and the negation of the literals of π' (like in the previous case). However, there will be no two items in this minimally inconsistent subset such that every agent rejects one of the two items. Therefore, the translation of $\boldsymbol{B}$ is not value-restricted. □

Now we can conclude the following corollary.

Corollary 5.7. VALUERESTRICTED *is* **coNP**-*hard.*

This follows from Lemma 5.5 and Lemma 5.6. Now we have seen all of the results from this chapter, we will go on to elaborate further on what we can conclude from these results.

5.4 coNP-Hardness and Membership in Π_2^p

In this chapter, we have seen a lot of complexity results, which have not lead to any of our problems being complete for a class. Note that, throughout this discussion we will assume that $\mathbf{P} \neq \mathbf{NP}$, and furthermore, the polynomial hierarchy does not collapse below the third level. The aim of this chapter was to show that our three candidate problems were all Π_2^p-complete problems. Unfortunately, we haven't shown this here. However, we have narrowed down the scope of classes which the problems can be complete for. For each of the problems, we have shown that they have membership in Π_2^p, this tells us that our problems can be complete for Π_2^p, or any class which lies within it.

Furthermore, we have shown that each of the problems are **coNP**-hard, either by giving a reduction from UNSAT, or a reduction from a **coNP**-hard problem. A problem being **coNP**-hard entails that it is as hard as the hardest problems in the class **coNP**. We see in Figure 5.1 the classes which contain **coNP**-hard problems up until the second level of the hierarchy. [2]

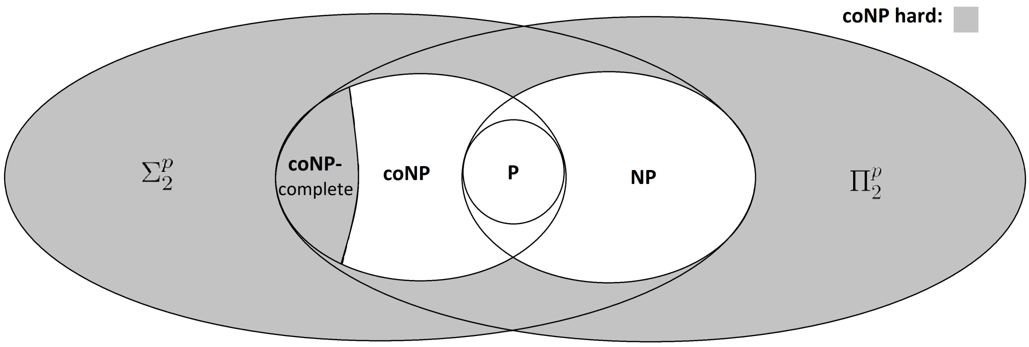

Figure 5.1: Problems which are **coNP**-hard up until the second level of the polynomial hierarchy.

As we have membership in Π_2^p the class the problems are complete for cannot be harder than Π_2^p. Furthermore, membership in Π_2^p means that the completeness of the problems in Σ_2^p can be ruled out, otherwise the polynomial hierarchy would collapse to the second level. There is a class between **coNP** and Π_2^p. Namely, Δ_2^p (also known as $\mathbf{P}^{\mathbf{NP}}$). This is a class of problems which can be solved by a deterministic Turning Machine in polynomial time with access to an **NP**-oracle. However, as stated with the introduction of each problem, the structure of these problems suggest that they are Π_2^p-complete problems. This is due to the class having problems which follow the quantification pattern of 'for all ... there exists...', which is what we see in each of these problems. This still makes us see that these problems are likely to be Π_2^p-complete problems.

Our problem PAIRSIMPLE is the binary aggregation analogue to the problem of checking if an agenda has the median property, MP. The problem MP was shown to be Π_2^p-complete (Endriss et al., 2012, Lemma 20, Lemma 24). We here showed that in the binary aggregation setting, this property is no harder to check. Future research may extend this result to a full completeness proof.

Similarly, we investigated the problems VALUERESTRICTED and BINVALUERESTRICTED, checking a property in judgment aggregation and binary aggregation, respectively. In Lemma 5.6, we saw that we can compute the problem BINVALUERESTRICTED by using an algorithm for VALUERESTRICTED, meaning that VALUERESTRICTED is at

[2]This diagram assumes that $\mathbf{P} \neq \mathbf{NP}$, as well as that the polynomial hierarchy does not collapse below the third level of the hierarchy. Furthermore, the exact relationship between **coNP**-complete and **NP**-complete problems is not known. However, this diagram represents what is commonly thought of the structure of the polynomial hierarchy up until its second level.

least as hard as BINVALUERESTRICTED. This means that using value restriction in our binary aggregation setting is no harder than in the judgment aggregation setting. So although, we believe that they will both be complete for the same class we also see that our binary value restriction can be no harder than in the judgment aggregation setting.

There is already work regarding the comparison between problems in the judgment aggregation and binary aggregation settings. This is looking at the succinctness of the languages. Endriss et al. (2016, Theorem 6) prove that the language of judgment aggregation is strictly more succinct than that of binary aggregation with constraints, unless the polynomial hierarchy collapses.

To give an idea of what this means, we see that checking if a binary profile is rational can be done in polynomial time, as it is model checking. Whereas, checking if a judgment aggregation profile is consistent is an **NP**-complete problem, as it boils down to the problem SAT.

Since judgment aggregation is more succinct than binary aggregation, this entails that we can translating a binary aggregation input to judgment aggregation without a super-polynomial blow up of the input size (Endriss et al., 2016). Whereas, the same is not true for a translation from judgment aggregation to binary aggregation. Following from this, Endriss et al. (2016) prove results such as, if a winner determination problem in judgment aggregation is in a class, such as **NP**, then the equivalent winner determination problem in binary aggregation will also be in that class (Endriss et al., 2016, Propsition 14). Although, in our setting we are not looking at winner determination problems, we see that the results shown in this chapter follow in the same direction as this work. In this chapter, we showed that the problems in binary aggregation are no harder than their analogue results in judgment aggregation.

5.5 Summary of Chapter 5

In this chapter, we have looked at three problems. PAIRSIMPLE, checking if a pair of constraints is simple. VALUERESTRICTED, checking that a profile is value-restricted with respect to an agenda. BINVALUERESTRICTED, checking if a profile is value-restricted with respect to a pair of constraints. The results in this chapter showed that they all had membership in Π_2^p and were all **coNP**-hard. Therefore, we have narrowed down the complexity classes for which they will be complete: at least as hard as **coNP**-complete problems, but no harder than Π_2^p-complete problems. Although it hasn't been shown, the structure of the problems suggest that they will be Π_2^p-complete, this is due to their 'for all ..., there exists...' quantification pattern.

Another conclusion we can draw from this chapter is that, we can say that our translation of the median property and value restriction to the binary aggregation does not increase the complexity of checking the properties. This is follows from the work of Endriss et al. (2016), stating that judgment aggregation is a more succinct than binary aggregation. As MP is Π_2^p-complete and it's analogue is the pair of formulas being simple, PAIRSIMPLE, has membership in Π_2^p then we know that PAIRSIMPLE cannot be any computational harder than MP. Similarly, for value restriction, we saw that both VALUERESTRICTED and BINVALUERESTRICTED both have membership in Π_2^p. therefore, showing that both problems cannot be complete for any harder class than Π_2^p. However, in Lemma 5.6 we saw that we can reduce the binary aggregation problem to the judgment aggregation problem. This shows that the judgment aggregation version of value restriction is at least as hard as the binary aggregation version of value restriction. Therefore, we can conclude that moving from the binary aggregation setting does not increase the complexity of the problem.

Chapter 6

Conclusion

In this chapter, we will give an overview of the chapters and the results that lie in them. We aimed to investigate binary aggregation with rationality and feasibility constraints. In particular, looking at when we can guarantee consistency, in the form of the outcomes abiding by the feasibility constraint. First, we will give an overview of the chapters and then we move on to look at some future research directions.

6.1 A Summary of the Chapters

In Chapter 2, we introduced the formal model of binary aggregation with constraints. We saw a translation of the formula-based model of judgement aggregation to binary aggregation with constraints (Grandi and Endriss, 2011). This shows the connection between the two models. Furthermore, it shows that the binary aggregation model can be seen as more expressive than the formula-based model. The constraints can give complete, consistent and complement-free judgments, as well as allowing us to add clauses which reflect the restraints of a situation, e.g. that the voters must accept at least one of the items. Then we looked at the work already carried out investigating binary aggregation with constraints. First, we looked at the work of Grandi and Endriss (2013). Their work focusses on the translation of agenda properties to binary aggregation with a single constraint. Following this we saw the extensions of this work by Endriss (2018); introducing binary aggregation with rationality and feasibility constraints. Following this, we looked at the first translation of an agenda property into this setting.

In Chapter 3, we continued to translate agenda properties from judgment aggregation to binary aggregation with rationality and feasibility constraints, focussing on the k-median property. We recreated Proposition 2.2 by Dietrich and List (2007) in the rationality and feasibility setting. The judgment aggregation result has been translated by Grandi and Endriss (2013) into their single-constraint case and we followed the same steps to arrive at the rationality and feasibility analogues. We looked at rationality constraints which were either positive, negative and mixed clauses, giving Proposition 3.3, Proposition 3.5 and Theorem 3.7, respectively. From this, we moved to inspect the translation of the k-median property to constraints with any finite number of clauses. We looked at the results already given for the single-constraint case and extended them to any constraint, resulting in Theorem 3.11. From this, we moved to the two-constraint setting. First, we recreated Lemma 3.9 in the rationality and feasibility setting, giving Lemma 3.12. Then, as before, we used this result to prove Theorem 3.14.

In Chapter 4, we looked to judgment aggregation for more ways in which consistent outcomes can be guaranteed. We then translated them into our binary aggregation setting. We first introduced domain restriction, which is linked to consistent outcomes under the majority rule in judgment aggregation. We focus on value restriction rather than other well-known domain restrictions, as it relies on the logical properties, rather than the organisation of the issues. In Section 4.2, we focussed on translating value restriction to the single-constraint setting and in Section 4.3, we translated value restriction to the two-constraint setting. In these sections, we produce similar results in both the single-constraint setting and the two-constraint setting. However, in the single-constraint setting, we found that the profile is value restricted with respect to a single constraint. Whereas, in the two-constraint setting, the profile is value restricted with respect to a pair of constraints. In this chapter, and in both settings, we recreated the result of Dietrich and List (2010) which states that if a profile is value-restricted, then it guarantees consistent outcomes under the majority rule. However, after translating this result, to both settings, we see that it only holds under the assumption that there is an odd number of agents. From this we created a new notion of value restriction. Namely, a profile being *negatively value-restricted* which guarantees feasible outcomes, for any number of agents.

In Chapter 5, we investigated the complexity of checking some of the properties that we encountered, namely agenda properties and domain restrictions. We focussed on the median property (which the complexity has already been investigated by Endriss et al. (2012) and shown to be Π_2^p-complete) and the property of a pair of constraints being simple. For domain restriction, we focussed on a profile being value-restricted in both judgment aggregation and binary aggregation with rationality and feasibility constraints. For the three problems which we studied, we showed that they all have membership in Π_2^p and that they are all **coNP**-hard. Another conclusion we can make from our results is that the problems in the binary setting are no harder than the problems in judgment aggregation. These results also align with the work of Endriss et al. (2016), stating that judgment aggregation is more succinct than binary aggregation. Furthermore, although we do not have completeness results for the problems in question, we have narrowed down the possible complexity classes wherein the problems will be complete. We also gave motivation that these problems are intuitively in Π_2^p as each of the problems follow the same quantification pattern, 'for all ... there exists ...', and it is this quantification pattern which defines the class Π_2^p.

6.2 Future Work

This thesis has expanded the study of binary aggregation with rationality and feasibility constraints. However, there are still some unanswered questions which we raised. In this section, we will look at some of the possible research directions which can follow on from our results.

In Section 3.2, we saw results such as Theorem 3.14. Here we saw that a quota rule guarantees Γ'-feasible outcomes on Γ-rational profiles if and only if this quota rule guarantees π'-feasible outcomes on some π-rational profile, with π and π' being prime implicates of Γ and Γ', respectively. The result has only been shown under the condition that the aggregation rule is a *quota rule*. The next steps in this research would be to find the characterisation of the rules for which the result holds. For example, it could be said that the rule has to be independent, as if it were not, then we would not be able to move between Γ'-feasible outcomes and π'-feasible outcomes.

In Chapter 5, we gave support for why the problems in question are Π_2^p-complete. However, there is still a completeness proof still missing. Thus, these completeness proofs would be the next step in this research, allowing us to know precisely where the problems sit in the hierarchy.

We could also extend our research by considering irresolute rules and how this would interact with our results. For instance, we assumed that the majority rule favours the rejection of issues when n is even. There are arguments against this asymmetric definition, as it does not fit how we intuitively want the majority rule to work, so we may move towards an irresolute rule. Furthermore, with the rule not being resolute, this may interact with some of our methods of attaining feasible outcomes. In particular, our results in Chapter 4 using value-restriction and negative value-restriction.

Appendix A

Appendix

A.1 Chapter 4, Proof of Lemma 4.4

Proof. We assume that for all prime implicates π' of Γ', there exists a prime implicate π of Γ such that $\pi \models \pi'$ and for the sake of a contradiction we assume that $\Gamma \not\models \Gamma'$.

As $\Gamma \not\models \Gamma'$, there exists a $B \in \text{Mod}(\Gamma) \backslash \text{Mod}(\Gamma')$. As $B \notin \text{Mod}(\Gamma')$, there exists a prime implicate π' of Γ', such that $B \notin \text{Mod}(\pi')$. By assumption we have that there exists a prime implicate π of Γ such that $\pi \models \pi'$, or equivalently $\text{Mod}(\pi) \subseteq \text{Mod}(\pi')$. However, $B \in \text{Mod}(\Gamma) \subseteq \text{Mod}(\pi) \subseteq \text{Mod}(\pi')$. Therefore, $B \in \text{Mod}(\pi')$ and we have reached a contradiction. $\square$

Bibliography

Arora, S. and B. Barak (2009). *Computational complexity: A modern approach*. Cambridge University Press.

Black, D. (1948). On the rationale of group decision-making. *Journal of Political Economy 56*(1), 23–34.

Chen, W. and U. Endriss (2018). Aggregating alternative extensions of abstract argumentation frameworks: Preservation results for quota rules. In *Computational Models of Argument*, pp. 425–436.

Dietrich, F. and C. List (2007). Judgment aggregation by quota rules: Majority voting generalized. *Journal of Theoretical Politics 19*(4), 391–424.

Dietrich, F. and C. List (2010). Majority voting on restricted domains. *Journal of Economic Theory 145*(2), 512–543.

Endriss, U. (2018). Judgment aggregation with rationality and feasibility constraints. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 946–954. International Foundation for Autonomous Agents and Multiagent Systems.

Endriss, U., U. Grandi, R. De Haan, and J. Lang (2016). Succinctness of languages for judgment aggregation. In *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Endriss, U., U. Grandi, and D. Porello (2012). Complexity of judgment aggregation. *Journal of Artificial Intelligence Research 45*, 481–514.

Grandi, U. (2012a). *Binary Aggregation with Integrity Constraints*. Ph. D. thesis, Institute for Logic, Language and Computation, University of Amsterdam.

Grandi, U. (2012b). The common structure of paradoxes in aggregation theory. In *Proceedings of the 4th International Workshop on Computational Social Choice*.

Grandi, U. and U. Endriss (2011). Binary aggregation with integrity constraints.

In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*.

Grandi, U. and U. Endriss (2013). Lifting integrity constraints in binary aggregation. *Artificial Intelligence 199*, 45–66.

Inoue, K. (1992). Linear resolution for consequence finding. *Artificial Intelligence 56*(2-3), 301–353.

List, C. (2003). A possibility theorem on aggregation over multiple interconnected propositions. *Mathematical Social Sciences 45*(1), 1–13.

List, C. and P. Pettit (2002). Aggregating sets of judgments: An impossibility result. *Economics & Philosophy 18*(1), 89–110.

List, C. and C. Puppe (2009). Judgment aggregation: A survey. In C. List and C. Puppe (Eds.), *Handbook of Rational and Social Choice*. Oxford University Press.

Marquis, P. (2000). Consequence finding algorithms. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pp. 41–145. Springer.

Nehring, K. and C. Puppe (2007). The structure of strategy-proof social choice — part i: General characterization and possibility results on median spaces. *Journal of Economic Theory 135*(1), 269–305.

Pettit, P. (2001). Deliberative democracy and the discursive dilemma. *Philosophical Issues 11*(1), 268–299.

Sen, A. K. (1966). A possibility theorem on majority decisions. *Econometrica: Journal of the Econometric Society*, 491–499.

Tourret, S. (2012). *Abduction and prime implicates, from propositional logic to equational logic*. Ph. D. thesis, Grenoble InP.