Succinctness of Languages for Judgment Aggregation

Endriss, U.; Grandi, U.; de Haan, R.; Lang, J.

Publication date
2016

Document Version
Author accepted manuscript

Published in
Proceedings, Fifteenth International Conference on Principles of Knowledge Representation and Reasoning

Succinctness of Languages for Judgment Aggregation

Ulle Endriss
ILLC
University of Amsterdam
The Netherlands
ulle.endriss@uva.nl

Umberto Grandi
IRIT
University of Toulouse
France
umberto.grandi@irit.fr

Ronald de Haan
Algorithms and Complexity Group
Technische Universität Wien
Austria
dehaan@ac.tuwien.ac.at

Jérôme Lang
LAMSADE
Université Paris-Dauphine
France
lang@lamsade.dauphine.fr

Abstract

We review several different languages for collective decision making problems, in which agents express their judgments, opinions, or beliefs over elements of a logically structured domain. Several such languages have been proposed in the literature to compactly represent the questions on which the agents are asked to give their views. In particular, the framework of judgment aggregation allows agents to vote directly on complex, logically related formulas, whereas the setting of binary aggregation asks agents to vote on propositional variables, over which dependencies are expressed by means of an integrity constraint. We compare these two languages and some of their variants according to their relative succinctness and according to the computational complexity of aggregating several individual views expressed in such languages into a collective judgment. Our main finding is that the formula-based language of judgment aggregation is more succinct than the constraint-based language of binary aggregation. In many (but not all) practically relevant situations, this increase in succinctness does not entail an increase in complexity of the corresponding problem of computing the outcome of an aggregation rule.

1 Introduction

We are interested in collective (but also, as a special case, individual) decision making problems where the domain, i.e., the set of alternatives to choose from, has a complex, combinatorial structure. A prime example is *judgment aggregation* (List and Pettit 2002), where a set of agents express their opinions on different but related questions and a consistent collective decision is sought. Other notable examples include *multiple referenda*, where agents express their preferences about each of a set of interdependent binary issues, and a collective choice on each issue has to be made; *committee* and, more generally, *multiwinner elections*, where a set of candidates, subject to some cardinality constraint, has to be elected given the voters' preferences; and *group configuration*, where a complex object, such as a path in a graph, has to be constructed, given the agents' preferences. Such domains are usually referred to as *combinatorial domains* (see Lang and Xia (2016) for a survey).

In this paper we are interested in combinatorial domains where the issues at stakes are binary, i.e., agents express a yes/no opinion on each question, and where issues may be logically correlated, thus restricting the set of admissible evaluations. Consider, for instance, the following example, which is inspired by the so-called *group-travel problem* introduced by Klamler and Pferschy (2007).

Example 1. *A tour operator using an automated planning tool needs to prepare the schedule for a day trip of a group of tourists. The group has the options of taking a guided tour of the city centre (T), visiting the nearby museum (M), or going to the beach (B). A short rest at the hotel (H) can also be included in the plan. The preferences of each tourist are elicited by means of binary questions such as "Q_1 : Do you want to include a stop-by at the hotel?" or "Q_2 : Do you want take the tour of the city centre or visit the museum (or maybe both)?", corresponding to a simple yes/no vote on propositional formulas H and $T \vee M$. More complex opinions can also be queried, such as "Q_3 : Do you require a stop-by at the hotel if a visit to the beach is scheduled?", corresponding to a vote on the propositional formula $B \to H$, or "Q_4 : Do you want to include all three activities in the schedule?", corresponding to a vote on $T \wedge M \wedge B$.*

The designer of such a tool faces a choice in the representation of the issues at stake. On the one hand, she can directly ask the individuals to vote on each of the four questions (each modelled as a formula of propositional logic), and then check that the overall individual opinion is a consistent set of formulas. However, this first option requires consistency checking, which has high computational complexity. On the other hand, the designer can pre-process the logical relations amongst the issues at stake and represent them as an integrity constraint. In our example, this would boil down to assigning a propositional symbol to each of the four questions—such as Q_1, Q_2 , Q_3, and Q_4—and checking that each individual opinion satisfies an integrity constraint expressing that a positive answer to Q_1 implies a positive answer to Q_3, that a positive answer to both Q_4 and Q_3 implies a positive answer to Q_1, and, finally, that a positive answer to Q_4 implies a positive answer to Q_2:

$$(Q_1 \to Q_3) \wedge [(Q_4 \wedge Q_3) \to Q_1] \wedge (Q_4 \to Q_2)$$

Whether the required logical relations between the issues are respected can then be verified by means of simple model checking. However, this move may come at a price, notably in the size of the integrity constraint relative to the size of the initial formulation modelling the issues themselves as propositional formulas. The aim of this paper is to quantify this price in the succinctness of representation for the two languages described above, and for two additional variants. A second aim is to clarify the extent to which changes in succinctness have an effect on the tractability of making collective decisions based on individual inputs represented in those languages.

We draw on definitions from the literature on judgment aggregation (see Grossi and Pigozzi (2014) and Endriss (2016) for two recent surveys) and more specifically on results on the computational complexity of its winner determination problem (Endriss, Grandi, and Porello 2012; Lang and Slavkovik 2014; Endriss and de Haan 2015), in combination with the notion of relative succinctness employed in different areas of artificial intelligence, such as knowledge representation languages (Cadoli et al. 2000), planning (Nebel 2000), preference representation (Coste-Marquis et al. 2004; Uckelman et al. 2009), and modal logics for multiagent systems (French et al. 2013).

Another highly relevant stream of work is belief merging (see Konieczny and Pino Pérez (2011) for a survey). The relation between belief merging and judgment aggregation is discussed by Everaere, Konieczny, and Marquis (2014; 2015). In belief merging, however, there is no given set of issues that defines a decision problem. Individuals are instead allowed to submit as many propositional formulas as they wish, making problems of succinctness less relevant for this setting. The approach followed in this paper is also reminiscent of the problem of knowledge compilation (Cadoli and Donini 1997; Darwiche and Marquis 2002; Marquis 2015), which deals amongst other things with quantifying the cost of computing a given function over a compactly represented input.

The remainder of this paper is organised as follows. In Section 2 we give the necessary background on judgment aggregation and introduce several languages for specifying aggregation problems. In Section 3 we recall the fundamental definition of relative succinctness and prove our main results, establishing the relative succinctness of four different specification languages. In Section 4 we address the issue of translating from one language to another, and we study the computational complexity of this problem. Following this, in Section 5 we analyse the influence of the choice of language on the complexity of the problem of computing the collective judgment under several aggregation rules. Finally, Section 6 concludes.

2 Languages for Judgment Aggregation

We start from a finite set of issues over which a group of agents need to take a decision. Agents express their opinions in the form of yes/no judgments over each of the issues, and these opinions are then merged into a collective one by means of an aggregation rule. Issues are related by logical constraints, making certain opinions unacceptable. A classic example for an aggregation rule is the *majority rule*, which accepts a given issue if a (strict) majority of the individuals wish to accept it. However, the majority rule has the severe drawback that it may produce an inconsistent outcome. This is the well-known *doctrinal paradox* (Kornhauser and Sager 1993; List and Pettit 2002). Therefore, many alternative rules have been considered in the literature. We will see some of them in Section 5.

The *specification* of a collective decision problem over a logically structured domain is given by a set of *issues* and a set of *feasible positions* to take on them. In much of this paper we focus on the languages that can be used to represent such restrictions, rather than on the rules for aggregating the judgments expressed in those languages. In this section, we define the four concrete languages we consider and show that they all have the same expressive power.

2.1 Basic Definitions

Let us first define the basic semantics for specifications:

Definition 1. *The* ***basic language*** *(BASIC) for the specification of collective decision problems over logically structured domains is* $\mathcal{L}_0 = \{X \mid X \subseteq \{0,1\}^m, X \neq \emptyset, m \in \mathbb{N}\}$.

For example, $X = \{(0,0,0),(0,1,0),(1,0,0),(1,1,1)\}$ is an element of $\mathcal{L}_0$ that specifies a decision problem with $m = 3$ issues for which there are four feasible positions to take, namely those for which the third issue gets accepted if and only if both the first two issues get accepted as well.

In view of the combinatorial explosion—in terms of the number of issues m—of the extensional representation of a set of all feasible evaluations, a number of languages have been proposed in the literature to specify restrictions in a compact way. We begin by giving the following definition:

Definition 2. *A* ***specification language*** $(\mathcal{L}, \tau)$ *is given by a set* $\mathcal{L}$ *and an interpretation function* $\tau : \mathcal{L} \to \mathcal{L}_0$.

Intuitively, a specification language can be used to compactly represent the restriction on feasible evaluations without having to provide the full list of them, with the function τ interpreting the representation in the basic language.

Let $\mathcal{PL}$ denote the propositional language over a countable number of atoms, and let $\mathcal{PL}_{\{p_1,\dots,p_m\}}$ denote the propositional language over the set of atoms $\{p_1,\dots,p_m\}$. Furthermore, the *length* $|\varphi|$ of a formula φ is the number of occurrences of propositional atoms in that formula.

2.2 Four Concrete Specification Languages

The first specification language we consider requires voters to directly take a stance on complex propositional formulas, restricting the set of feasible positions by requiring the set of accepted formulas to be logically consistent.[1] This setting is known as *judgment aggregation*, and is the subject of a growing number of publications in philosophy and economics (List and Pettit 2002; Pauly and van Hees 2006; Gärdenfors 2006; Dietrich and List 2007a; Miller and Osherson 2009) as well as in artificial intelligence (see, e.g., the surveys by Grossi and Pigozzi (2014) and Endriss (2016) mentioned earlier).

In this framework, issues are pairs of propositional formulas $\{\varphi, \neg\varphi\}$, where φ is not of the form $\neg\psi$, which together

[1] This setting has been generalised by Dietrich (2007) to include a larger class of logical languages. In this paper, however, we focus on the standard definitions using propositional logic.

form an *agenda* $\Phi = \{\varphi_1, \neg\varphi_1, \ldots, \varphi_m, \neg\varphi_m\}$. Given an agenda Φ, the *pre-agenda* associated with it is the set of its non-negated formulas $\Phi^+ = \{\varphi_1, \ldots, \varphi_m\}$. Individuals express their positions by means of *judgment sets* $J \subseteq \Phi$. A judgment set J is called *complete* if for all $\varphi \in \Phi^+$ we have $\varphi \in J$ or $\neg\varphi \in J$, and it is called *consistent* if there exists an assignment that makes all formulas in J true. The set of all complete and consistent judgment sets is denoted by $\mathcal{J}(\Phi)$.

Definition 3. *The language of* ***judgment aggregation*** *(JA) is defined as follows:*

$$\begin{aligned}\mathcal{L}_{JA} &= \{\{\varphi_1, \neg\varphi_1, \ldots, \varphi_m, \neg\varphi_m\} \mid \varphi_j \in \mathcal{PL},\ m \in \mathbb{N}\}\\ \tau_{JA}(\Phi) &= \mathcal{J}(\Phi),\end{aligned}$$

where we assume that a complete and consistent judgment set $J \subseteq \Phi$ is represented by a binary vector $(v_1, \ldots, v_m)$ with $v_j = 1$ if $\varphi_j \in J$ and $v_j = 0$ if $\neg\varphi_j \in J$.

For example, the pre-agenda $\Phi^+ = \{a, b, a \wedge b\}$ specifies a decision problem where each individual has to choose for each of these three formulas whether to accept it or its negation. To be consistent, an individual has to accept the third formula if and only if she accepts the first two, i.e., if we translate into the basic language we obtain $\tau_{JA}(\Phi) = \{(0,0,0),(0,1,0),(1,0,0),(1,1,1)\}$, the exact same set of feasible positions discussed directly after Definition 1.

A second possibility for staging a vote on a binary combinatorial domain is that of directly querying individuals on simple binary issues, and representing the logical relations amongst them by means of a propositional constraint. This is the approached followed by Grandi and Endriss (2013), building on earlier work by Dokow and Holzman (2009; 2010). In this setting, a set of binary *issues* $\mathcal{I} = \{1, \ldots, m\}$ is given, and the individuals have to express their opinions in the form of binary *ballots* $B \in \{0,1\}^{\mathcal{I}}$. Feasible positions are specified by means of an *integrity constraint* Γ, built using only the propositional symbols $PS = \{p_1, \ldots, p_m\}$, one for each issue.

Definition 4. *The language of* ***binary aggregation with integrity constraints*** *(IC) is defined as follows:*

$$\begin{aligned}\mathcal{L}_{IC} &= \{\Gamma \mid \Gamma \in \mathcal{PL}_{\{p_1,\ldots,p_m\}} \textit{ satisfiable}, m \in \mathbb{N}\}\\ \tau_{IC}(\Gamma) &= \mathrm{Mod}(\Gamma),\end{aligned}$$

where $\mathrm{Mod}(\Gamma)$ *is the set of models of the formula* Γ*, each of which is represented as a binary vector.*

In this framework, our running example is represented by the integrity constraint $\Gamma = (p_1 \wedge p_2 \leftrightarrow p_3)$. Indeed, we again have $\tau_{IC}(\Gamma) = \{(0,0,0),(0,1,0),(1,0,0),(1,1,1)\}$.

Finally, we also consider two generalisations of this last framework. The first of these appears not to have been studied in previous work. It generalises *IC* by allowing the use of additional variables in the integrity constraint—beyond those directly corresponding to the m issues:

Definition 5. *The language of* ***binary aggregation with integrity constraints with additional variables*** *(IC+AV) is defined as follows:*

$$\begin{aligned}\mathcal{L}_{IC+AV} &= \{(\{p_1, \ldots, p_m\}; \Gamma) \mid \Gamma \in \mathcal{PL} \textit{ sat.}, m \in \mathbb{N}\}\\ \tau_{IC+AV}(\Gamma) &= \mathrm{Mod}(\Gamma)_{\{p_1,\ldots,p_m\}},\end{aligned}$$

where $\mathrm{Mod}(\Gamma)_{\{p_1,\ldots,p_m\}}$ *is the set of models of* Γ *restricted to the propositional variables in* $\{p_1, \ldots, p_m\}$.

The second generalisation combines the formula-based setting of the language *JA* with the use of integrity constraints (with additional variables). This framework has been used, amongst others, by Dietrich and List (2008) and Lang and Slavkovik (2014).

Definition 6. *The language of* ***judgment aggregation with constraints*** *(JAC) is defined as follows:*

$$\begin{aligned}\mathcal{L}_{JAC} &= \{(\{\varphi_1, \neg\varphi_1, \ldots, \varphi_m, \neg\varphi_m\}; \Gamma) \mid\\ &\qquad \varphi_j \in \mathcal{PL}, \Gamma \in \mathcal{PL} \textit{ satisfiable}, m \in \mathbb{N}\}\\ \tau_{JAC}(\Phi; \Gamma) &= \mathcal{J}(\Phi; \Gamma),\end{aligned}$$

where $\mathcal{J}(\Phi;\Gamma)$ is composed of all complete and consistent judgment sets for Φ that are consistent with Γ.

Observe that *IC+AV* is recovered as a special case of *JAC* if we restrict pre-agenda formulas to atomic propositions.

We mention in passing two additional languages that have been considered in the literature on social choice theory. The language of *binary aggregation* (Dokow and Holzman 2009; 2010) represents explicitly the set of feasible positions, and therefore corresponds directly to our basic language *BASIC*. The language of *abstract Arrovian aggregation* (Nehring and Puppe 2010) represents a decision problem in terms of a set of Boolean properties, specifying for each feasible alternative a full list of the properties that are satisfied, resulting in a representation of size comparable to that of the basic language. We omit formal definitions in the interest of space.

2.3 Expressivity

It is straightforward to show that all of the languages introduced earlier are equally expressive, and that they can represent all problem specifications from the basic language.

Proposition 1. *τ_{JA}, τ_{IC}, τ_{IC+AV}, and τ_{JAC} are surjective.*

Proof. A proof of the surjectivity of τ_{JA} can be found in the work of Dokow and Holzman (2009, Proposition 2.1). τ_{IC} is surjective by the full expressivity of propositional logic with respect to sets of Boolean assignments. The remaining part of the proof follows from the facts that *IC* is a special case of *IC+AV* and that *JA* is a special case of *JAC*. □

Thus, expressivity is not a relevant criterion when choosing the best specification language for a given problem.

3 Succinctness of Specification Languages

In this section, we present our results on the relative succinctness of different languages for judgment aggregation. Our main result shows that voting directly on propositional formulas (*JA*) is strictly more succinct than voting on issues related by an integrity constraint (*IC*). We also show that languages that combine formulas and constraints have the same succinctness as the formula-based setting.

3.1 Definition of Relative Succinctness

We now provide a definition of relative succinctness between two specification languages, inspired by the work of Gogic et al. (1995) and Cadoli et al. (2000).

Let the *size* of a specification in the *BASIC* language be defined as $size(X) = |X| \cdot m$. The *size* of a specification in the language *JA* is the size of the corresponding agenda Φ, i.e., the sum of the lengths of the formulas in Φ, adding the length of the integrity constraint in the case of the language *JAC*. The size of a specification in the language *IC* is the length of the integrity constraint Γ, with the addition of m for the case of the language *IC+AV*. Call two specifications $X_1 \in \mathcal{L}_1$ and $X_2 \in \mathcal{L}_2$ in languages $\mathcal{L}_1$ and $\mathcal{L}_2$ *equivalent*, and write $X_1 \equiv X_2$, if and only if $\tau_1(X_1) = \tau_2(X_2)$.

Definition 7. *Given two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ for specification, we say that $\mathcal{L}_1$ is* ***at least as succinct as*** *$\mathcal{L}_2$, and write $\mathcal{L}_1 \preccurlyeq \mathcal{L}_2$, if there exist a function $f : \mathcal{L}_2 \to \mathcal{L}_1$ and a polynomial p such that, for all $X \in \mathcal{L}_2$, we have:*

- *$f(X) \equiv X$, and*
- *$size(f(X)) \leqslant p(size(X))$.*

Thus, language $\mathcal{L}_1$ is at least as succinct as language $\mathcal{L}_2$, if any specification given to us in $\mathcal{L}_2$ can be translated into an equivalent specification in $\mathcal{L}_1$ without a super-polynomial blow-up in the size of the representation. $\mathcal{L}_1$ is *strictly more succinct* than $\mathcal{L}_2$, denoted $\mathcal{L}_1 \prec \mathcal{L}_2$, if $\mathcal{L}_1 \preccurlyeq \mathcal{L}_2$ but $\mathcal{L}_2 \not\preccurlyeq \mathcal{L}_1$, i.e., if such a correspondence can be found in only one direction. $\mathcal{L}_1$ and $\mathcal{L}_2$ are *equally succinct*, denoted $\mathcal{L}_1 \sim \mathcal{L}_2$, if both $\mathcal{L}_1 \preccurlyeq \mathcal{L}_2$ and $\mathcal{L}_2 \preccurlyeq \mathcal{L}_1$.

3.2 Basic Result

We begin by proving the following fact, establishing the increased succinctness of using integrity constraints with respect to the explicit representation:

Proposition 2. *IC is strictly more succinct than BASIC.*

Proof. (*IC* $\preccurlyeq$ *BASIC*) We need to show that every $X \subseteq \{0,1\}^m$ can be represented by a propositional formula that is at most polynomial in the size of X. Each element $x \in \{0,1\}^m$ can be specified by a conjunction of literals $\varphi_x = \ell_1 \wedge \cdots \wedge \ell_m$ such that $\ell_j = p_j$ if $x_j = 1$ and $\ell_j = \neg p_j$ iff $x_j = 0$. Let $f(X) = \bigvee_{x \in X} \varphi_x$. It is easy to see that $f(X) \equiv X$ and that $size(f(X)) = O(size(X))$.

(*BASIC* $\not\preccurlyeq$ *IC*) To show that there is no polynomial translation from *IC* to *BASIC*, we give a set of integrity constraints whose unique equivalent in $\mathcal{L}_0$ is of exponential size. Consider $\Gamma = p_1$ as integrity constraint, and let the number of issues grow. For every m, the set of models of Γ has size 2^{m-1}, and thus is exponential in m.[2] $\square$

Recall that *BASIC* is identical to the binary aggregation framework of Dokow and Holzman (2009; 2010) and that, in terms of succinctness, it is closely related to the abstract Arrovian aggregation framework of Nehring and Puppe (2010). Thus, these two frameworks are also strictly less succinct than binary aggregation with integrity constraints.

[2]Since the set of models of $\neg p_1$ has also size exponential in m, this proof shows that representing the set of *infeasible* positions explicitly is also less succinct than using a constraint.

3.3 Formulas and Constraints

In this section, we prove our main result (Theorem 6) that shows that the formula-based setting of judgment aggregation is more succinct than the constraint-based formalism, and that this relation is subject to a well-known conjecture in computational complexity theory. We begin by showing a non-trivial reduction from *JA* to *IC* and then illustrate that reduction with an example.[3]

Proposition 3. *JA is at least as succinct as IC.*

Proof. Let Γ be a satisfiable propositional formula over the propositional variables $\{p_1, \dots, p_m\}$. To prove the statement we show that there exists an agenda Φ with $\Phi^+ = \{\varphi_1, \dots, \varphi_m\}$ such that $\tau_{JA}(\Phi) = \tau_{IC}(\Gamma)$, and such that the size of Φ is polynomial in the size of Γ.

Since Γ is satisfiable (see Definition 4), we can pick an assignment $\alpha : \{p_1, \dots, p_m\} \to \{0,1\}$ that satisfies Γ. Let $I^+ = \{1 \leqslant j \leqslant m \mid \alpha(p_j) = 1\}$ be the set of indices of propositions p_j that α sets to true, and let $I^- = \{1 \leqslant j \leqslant m \mid \alpha(p_j) = 0\}$ be the set of indices of propositions p_j that α sets to false. We can now define a pre-agenda $\Phi^+ = \{\varphi_1, \dots, \varphi_m\}$ composed of the following formulas:

$$\varphi_j = \begin{cases} p_j \vee \neg\Gamma & \text{if } j \in I^+ \\ p_j \wedge \Gamma & \text{if } j \in I^- \end{cases}$$

Clearly, the size of Φ is polynomial in the size of Γ. Note that for $j \in I^+$ we have that $\neg\varphi_j \equiv \neg p_j \wedge \Gamma$.

We now need to show that there exists a one-to-one correspondence between truth assignments on $\{p_1, \dots, p_m\}$ that models Γ on the one hand, and consistent judgment sets over the agenda Φ on the other hand. Let β be a truth assignment, and define $J_\beta = \{\varphi_j \mid \beta(p_j) = 1\} \cup \{\neg\varphi_j \mid \beta(p_j) = 0\}$. We now show that for any assignment β it holds that J_β is consistent if and only if β is a model of Γ, which in turn implies that $\tau_{JA}(\Phi) = \tau_{IC}(\Gamma)$ and concludes the proof.

First, recall that α is the initially chosen model of Γ, and observe that $J_\alpha = \{p_j \vee \neg\Gamma \mid j \in I^+\} \cup \{\neg p_j \vee \neg\Gamma \mid j \in I^-\}$, thus α satisfies J_α and J_α is consistent. Now let β be a truth assignment such that $\alpha \neq \beta$. We begin by showing that $J_\beta \models \Gamma$. By definition, there exists some $1 \leqslant i \leqslant m$ such that $\alpha(p_i) \neq \beta(p_i)$. Let $\chi_i = \neg\varphi_i$ if $\beta(p_i) = 0$, and $\chi_i = \varphi_i$ otherwise. Observe that $\chi_i \in J_\beta$, by definition of J_β. If $\alpha(p_i) = 1$ then, since $\alpha(p_i) \neq \beta(p_i)$, we have $\beta(p_i) = 0$ and $\chi_i = \neg p_i \wedge \Gamma$. Similarly, if $\alpha(p_i) = 0$ then $\beta(p_i) = 1$ and $\chi_i = p_i \wedge \Gamma$. Therefore, we know that $\chi_i \models \Gamma$, which, because of $\chi_i \in J_\beta$, implies that $J_\beta \models \Gamma$.

We now show that that for every $1 \leqslant j \leqslant m$ it holds that $J_\beta \models p_j$ if and only if $\beta(p_j) = 1$. There are two cases:

(i) If $\beta(p_j) = 1$ then by construction of J_β, we have $\varphi_j \in J_\beta$. Thus, J_β contains either $p_j \wedge \Gamma$ or $p_j \vee \neg\Gamma$. Because $J_\beta \models \Gamma$, if we are in the case where J_β contains $p_j \vee \neg\Gamma$, then $J_\beta \models p_j$, and if J_β contains $p_j \wedge \Gamma$ then $J_\beta \models p_j$.

[3]An equivalent shorter proof of this theorem can be obtained by combining our Proposition 8 with the straightforward observation that *IC+AV* $\preccurlyeq$ *IC*. We chose to present a direct proof here as it provides better insights into the properties of the two languages.

(ii) If instead $\beta(p_j) = 0$ then by construction of J_β, we have $\neg\varphi_j \in J_\beta$. Thus, J_β contains either $\neg p_j \vee \neg\Gamma$ or $\neg p_j \wedge \Gamma$. Again, because $J_\beta \models \Gamma$, if we are in the case where J_β contains $\neg p_j \vee \neg\Gamma$, then $J_\beta \models \neg p_j$; and if we are in the case where J_β contains $\neg p_j \wedge \Gamma$ then $J_\beta \models \neg p_j$.

Summing up, we have shown that $J_\beta \models \Gamma$, and that $J_\beta \models p_j$ iff $\beta(p_j) = 1$. We can therefore conclude that J_β is satisfiable if and only if β satisfies Γ. □

The intuition behind the construction used in the proof of Proposition 3 is the following. Since α satisfies Γ, the complete judgment set J_α corresponding to α should be consistent, which is clearly the case. For all other complete judgment sets, there is at least one formula that implies Γ. Under these conditions, each formula φ_j essentially just enforces p_j and each formula $\neg\varphi_j$ essentially just enforces $\neg p_j$. Therefore, Γ and Φ have exactly the same semantics. We illustrate this construction with an example.

Example 2. *Let PS* $= \{p_1, p_2, p_3\}$ *and let* $\Gamma = \neg p_1 \leftrightarrow (p_2 \vee p_3)$. *Let* $\alpha = (1, 0, 0)$*; clearly,* $\alpha \models \Gamma$. *We can now define the two sets* $I^+ = \{1\}$ *and* $I^- = \{2, 3\}$*, and the corresponding pre-agenda consisting of the formulas* $\varphi_1 = p_1 \vee \neg\Gamma \equiv p_1 \vee \neg(p_2 \vee p_3)$, $\varphi_2 = p_2 \wedge \Gamma \equiv \neg p_1 \wedge p_2$*, and* $\varphi_3 = p_3 \wedge \Gamma \equiv \neg p_1 \wedge p_3$. *Out of the eight judgment sets over* $\{\varphi_1, \varphi_2, \varphi_3\}$*, four of them are consistent:* $\{\varphi_1, \neg\varphi_2, \neg\varphi_3\}$, $\{\neg\varphi_1, \varphi_2, \varphi_3\}$, $\{\neg\varphi_1, \varphi_2, \neg\varphi_3\}$ *and* $\{\neg\varphi_1, \neg\varphi_2, \varphi_3\}$*, and they correspond to the four interpretations satisfying* Γ*, namely* $(1, 0, 0)$, $(0, 1, 1)$, $(0, 1, 0)$ *and* $(0, 0, 1)$. *Take for instance* $\beta = (0, 0, 1)$. *The corresponding judgment set* $J_\beta = \{\neg\varphi_1, \neg\varphi_2, \varphi_3\} = \{\neg p_1 \wedge (p_2 \vee p_3), p_1 \vee \neg p_2, \neg p_1 \wedge p_3\}$. *As expected, we have that* J_β *is consistent and* $\beta \models \Gamma$. *Take now* $\beta' = (1, 1, 0)$. *We have* $J_{\beta'} = \{\varphi_1, \varphi_2, \neg\varphi_3\} = \{p_1 \vee \neg(p_2 \vee p_3), \neg p_1 \wedge p_2, p_1 \vee \neg p_3\}$. *In this case* $J_{\beta'}$ *is inconsistent and* $\beta' \models \neg\Gamma$.

We now show that the converse of Proposition 3 is not true, subject to a well-known conjecture in computational complexity theory. We do so by using a known result by Cadoli et al. (2000). For this, we consider the following problem. Let $\mathcal{L}$ be a language of specification. The *admissibility* (or model checking) problem for $\mathcal{L}$ consists in deciding, given some $X \in \mathcal{L}$ and a vector $v \in \{0, 1\}^m$, whether $v \in \tau(X)$. In order to use the result by Cadoli et al., we show a difference in complexity for this admissibility problem, for the languages *IC* and *JA*. For *IC*, the admissibility problem coincides with checking whether a given (complete) truth assignment satisfies a propositional formula, which can be done in polynomial time.

Observation 4. *The admissibility problem for IC is in P.*

For *JA*, on the other hand, the admissibility problem involves checking satisfiability of a propositional formula.

Lemma 5. *The admissibility problem for JA is NP-hard.*

Proof. We give a reduction from SAT, the propositional satisfiability problem. Let φ be an arbitrary propositional formula (in conjunctive normal form). We construct the agenda $\Phi = \{\varphi, \neg\varphi\}$. It is then easy to see that the vector (1), i.e., a vector of length one with a single element, is in $\tau_{JA}(\Phi)$ if and only if φ is satisfiable. □

We are now ready to state our main result. This result is based on the widely-believed complexity-theoretic assumption that the Polynomial Hierarchy is strict (cf. Arora and Barak, 2009; Chapter 5).

Theorem 6. *JA is strictly more succinct than IC, unless the Polynomial Hierarchy collapses.*

Proof. By Proposition 3, we know that *JA* is at least as succinct as *IC*. To see that *IC* is not as succinct as *JA*, unless the Polynomial Hierarchy collapses, suppose that the Polynomial Hierarchy is strict. Then NP $\neq$ P, and we can apply Theorem 5 of Cadoli et al. (2000) using the results of Observation 4 and Lemma 5. This result directly implies that *IC* is not as succinct as *JA*. □

The resulting collapse of the Polynomial Hierarchy (if *IC* $\preccurlyeq$ *JA*) takes place at the second level, which can be shown with a direct (but lengthier) proof, which we omit here in the interest of space.

3.4 Additional Languages

Next, we show that the two additional languages that combine constraints and formulas (see Definitions 5 and 6) are as succinct as the language *JA*.

Proposition 7. *JAC and JA are equally succinct.*

Proof. (*JAC* $\preccurlyeq$ *JA*) The inequality is straightforward from the fact that *JA* is a special case of *JAC*.

(*JA* $\preccurlyeq$ *JAC*) To prove that every agenda and additional constraint can be translated into a single agenda with the same semantics we use a similar translation to the one presented in Proposition 3. Let $(\Phi; \Gamma)$ be an element of *JAC*, where $\Phi = \{\varphi_1, \neg\varphi_1, \ldots, \varphi_m, \neg\varphi_m\}$. Let $\{p_1, \ldots, p_u\}$ be the propositional variables occurring in $\Phi \cup \{\Gamma\}$. We show that there exists an agenda Δ with $\Delta^+ = \{\delta_1, \ldots, \delta_m\}$ such that $\tau_{JAC}(\Phi, \Gamma) = \tau_{JA}(\Delta)$, and $|\delta|$ is polynomial in $|\Phi \cup \{\Gamma\}|$, for each $\delta \in \Delta$.

Since Γ is satisfiable, there exists an assignment $\alpha : \{p_1, \ldots, p_u\} \to \{0, 1\}$ that satisfies Γ. We can therefore repeat the construction presented in the proof of Proposition 3. Let $I^+ = \{1 \leqslant j \leqslant m \mid \alpha \models \varphi_j\}$ and $I^- = \{1 \leqslant j \leqslant m \mid \alpha \models \neg\varphi_j\}$. For each $j \in I^-$, we let $\delta_j = \varphi_j \wedge \Gamma$; and for each $j \in I^+$, we let $\delta_j = \varphi_j \vee \neg\Gamma$, We now define a one-to-one correspondence between judgment sets over Φ and judgment sets over the agenda Δ: for each judgment set $J \in \mathcal{J}(\Phi)$, we define the judgment set $J' = \{\delta_j \mid \varphi_j \in J\} \cup \{\neg\delta_j \mid \neg\varphi_j \in J\}$. Using the same arguments as in the proof of Proposition 3, we show that for any J it holds that $J \cup \{\Gamma\}$ is consistent if and only if J' is consistent. □

Proposition 8. *IC+AV and JA are equally succinct.*

Proof. (*IC+AV* $\preccurlyeq$ *JA*) Take any pre-agenda $\Phi^+ = \{\varphi_1, \ldots, \varphi_m\}$ and let $T(\Phi) = (\{p_1, \ldots, p_m\}; \Gamma)$ be an instance of $\mathcal{L}_{IC+AV}$, where $\Gamma = \bigwedge(p_i \leftrightarrow \varphi_i)$. Now let α be the propositional formula corresponding to an assignment to the propositional variables $\{p_1, \ldots, p_m\}$. The consistency of $\alpha \wedge \Gamma$ is equivalent to the consistency of judgment set $\{\varphi_i \mid \alpha(p_i) = 1\} \cup \{\neg\varphi_i \mid \alpha(p_i) = 0\}$. Therefore, we have a one-to-one correspondence between Γ-consistent judgment sets of $T(\Phi)$ and consistent judgment sets of Φ.

(*JA* $\preccurlyeq$ *IC+AV*) The inequality is a straightforward consequence of the fact that *IC+AV* is a special case of *JAC*, combined with the result of Proposition 7. $\square$

4 From Formulas to Constraints and Back

In Section 3.3 we showed that *JA* $\preccurlyeq$ *IC*, which means that every formula $\Gamma \in \mathcal{L}_{IC}$ can be equivalently expressed as an agenda $\Phi \in \mathcal{L}_{JA}$ of a size that is polynomial in that of Γ. We now show that this translation is hard, namely as hard as computing a model of a propositional formula, if it exists.

Proposition 9. *Given a satisfiable propositional formula* $\Gamma \in \mathcal{PL}_{\{p_1,\dots,p_m\}}$*, the problem of computing an agenda* Φ *of size polynomial in* $|\Gamma|$ *with* $|\Phi^+| = m$ *such that* $\tau_{IC}(\Gamma) = \tau_{JA}(\Phi)$ *is FNP-complete under Turing reductions.*[4]

Proof. We first show membership in FNP. Given Γ, we can guess a satisfying assignment α of Γ in polynomial time. Using α, we can use the construction in the proof of Proposition 3 to construct an agenda Φ of polynomial size such that $\tau_{IC}(\Gamma) = \tau_{JA}(\Phi)$. If Γ is unsatisfiable, we are allowed to output any agenda containing the right number of formulas.

Next, to show FNP-hardness, we reduce from the problem FSAT. In this problem, given a propositional formula Γ, one needs to output a satisfying assignment of Γ if Γ is satisfiable, and "unsatisfiable" otherwise. In our reduction, we produce in polynomial time an instance x of the problem P that we are proving hardness of, and from the output of P on input x we produce in polynomial time an output for FSAT.

The idea of this reduction is the following. Given any agenda Φ, we can efficiently construct a vector $\overline{a} \in \tau_{JA}(\Phi)$, by taking an arbitrary truth assignment α to the variables in Φ, and checking what formulas in Φ are satisfied by α. Therefore, if we can express a propositional formula $\Gamma \in \mathcal{L}_{IC}$ as an equivalent agenda $\Phi \in \mathcal{L}_{JA}$, we can use this trick to efficiently produce a satisfying assignment for Γ.

Let Γ be an instance of FSAT on variables $\{p_1,\dots,p_m\}$. We let Γ be the instance of P. Then, the output of P on input Γ is an agenda Φ. Then, let $\gamma : \mathrm{Var}(\Phi) \to \{0,1\}$ be an arbitrary truth assignment to the variable occurring in Φ (e.g., the all-zeroes assignment). We now construct an assignment $\alpha : \{y_1,\dots,y_m\} \to \{0,1\}$ as follows. For each $1 \leqslant j \leqslant m$, we let $\alpha(y_j) = 1$ if and only if $\gamma \models \varphi_j$. Clearly, the complete judgment set corresponding to α is satisfiable, since γ witnesses this. Now, if $\alpha \not\models \Gamma$, we can conclude that Γ is unsatisfiable, since otherwise it would have been the case that $\tau_{IC}(\Gamma) = \tau_{JA}(\Phi)$. Thus, if $\alpha \models \Gamma$, we can output α, and if $\alpha \not\models \Gamma$, we can output "unsatisfiable". $\square$

In the other direction, the main interest of the language *IC+AV* is that it gives us a practical way of translating (with worst-case exponential-size growth) an element of $\mathcal{L}_{JA}$ into an element of $\mathcal{L}_{IC}$. Before we start, recall that the forgetting $\exists X.\varphi$ of a set of variables X in a propositional formula φ (Lin and Reiter 1994) is defined inductively as follows:

- $\exists\emptyset.\varphi = \varphi$
- $\exists\{x\}.\varphi = \varphi_{x\leftarrow\bot} \vee \varphi_{x\leftarrow\top}$
- $\exists(X \cup \{x\}).\varphi = \exists X.(\exists\{x\}.\varphi)$

The problem of translating an element of *JA*, that is, an agenda Φ, into an equivalent element of $(\{p_1,\dots,p_m\};\Gamma)$ with $\Gamma \in \mathcal{PL}_{\{p_1,\dots,p_m\}}$, amounts to variable forgetting in a propositional formula, as shown by the following construction.[5] Let Φ with $\Phi^+ = \{\varphi_1,\dots,\varphi_m\}$ be an element of $\mathcal{L}_{JA}$, and let $Y = \mathit{Var}(\varphi_1,\dots,\varphi_m)$ be the set of propositional variables appearing in $\varphi_1,\dots,\varphi_m$. Define $\Gamma^* := \bigwedge_i(p_i \leftrightarrow \varphi_i)$, and define an instance of $\mathcal{L}_{IC+AV}$ as $(\{p_1,\dots,p_m\};\Gamma^*)$. Now, forget from Γ^* all variables with the exception of $p_1,\dots,p_m$:

$$\Gamma \quad := \quad \exists\big(Y \setminus \{p_1,\dots,p_m\}\big).\Gamma^*$$

Thus, we obtain an instance Γ of $\mathcal{L}_{IC}$ that is equivalent to the initial instance Φ of $\mathcal{L}_{JA}$.

Example 3. *Let* Φ *be an instance of* $\mathcal{L}_{JA}$ *with pre-agenda* $\Phi^+ = \{a \wedge b, a \to c, c, b \leftrightarrow d, d\}$*. Following the construction described above, we have:*

$$\begin{aligned}\Gamma^* \quad = \quad & (p_1 \leftrightarrow (a \wedge b)) \wedge (p_2 \leftrightarrow (a \to c)) \wedge \\ & (p_3 \leftrightarrow c) \wedge (p_4 \leftrightarrow (b \leftrightarrow d)) \wedge (p_5 \leftrightarrow d)\end{aligned}$$

Forgetting a,b,c,d *in* Γ^**, after calculations, provides us with the following integrity constraint:*

$$\begin{aligned}\Gamma \quad = \quad & \ (p_1 \wedge (p_2 \leftrightarrow p_3) \wedge (p_4 \leftrightarrow p_5)) \\ & \vee\ (\neg p_1 \wedge p_2 \wedge (p_4 \leftrightarrow p_5)) \\ & \vee\ (\neg p_1 \wedge (p_2 \leftrightarrow p_3) \wedge (p_4 \leftrightarrow \neg p_5)) \\ & \vee\ (\neg p_1 \wedge p_2 \wedge (p_4 \leftrightarrow \neg p_5))\end{aligned}$$

It is then a straightforward exercise to check that the assignments to $\{p_1,\dots,p_5\}$ *that verify* Γ *correspond one-to-one to complete and consistent judgment sets over* Φ*.*

The interest of this construction is that it can also be used to build fragments of *JAC* that are intermediate between *JA* and *IC*, by replacing *some* elements φ of the agenda by a propositional variable p_i, adding $p_i \leftrightarrow \varphi$ to the constraint Γ, and forgetting all variables (if any) that appeared in φ_i. The more we move from *JA* to *IC*, the more space we need (cf. Theorem 6). At the same time, as we discuss below, some basic tasks, such as checking consistency of a judgment set, become computationally easier to solve.

Finally, we note that in case an agenda Φ is such that every propositional variable occurring as a subformula within one of the elements of Φ is also an agenda formula itself, translating from *JA* to *IC* is simple and does not involve a combinatorial explosion. Indeed, if $\Phi^+ = \{p_1,\dots,p_\ell,\varphi_{\ell+1},\dots,\varphi_m\}$ with $\mathit{Var}(\Phi^+) = \{p_1,\dots,p_\ell\}$, we can build an equivalent constraint $\Gamma \in \mathcal{PL}_{\{p_1,\dots,p_m\}}$ as follows, using fresh variable names $p_{\ell+1},\dots,p_m$:

$$\Gamma \quad = \quad (p_{\ell+1} \leftrightarrow \varphi_{\ell+1}) \wedge \dots \wedge (p_m \leftrightarrow \varphi_m)$$

[4]Here we allow the output to be any agenda if Γ is unsatisfiable.

[5]Forgetting may give rise to exponentially long formulas, unless NP $\cap$ coNP $\subseteq$ P / poly (Lang, Liberatore, and Marquis 2003, Proposition 23). This fact can be used to give an alternative proof that *IC* cannot be at least as succinct as *JA* (i.e., of Theorem 6), again under standard assumptions of complexity theory.

5 Does the Choice of Language Affect the Complexity of Aggregation?

Our results on the succinctness of specification languages for judgment aggregation need to be interpreted in view of some considerations of computational complexity. When confronted with an aggregation problem, we want to be able to choose the most succinct language with respect to the specific aggregation rule used. It is therefore important to know whether the computational complexity of computing judgment aggregation rules changes significantly or not when a different specification language is being used.

It is clear that basic tasks related to the admissibility of judgments will be easier in the less succinct setting of the *IC* language, rather than in the formula-based languages *JA* and *JAC*. Concretely, deciding whether either an individual judgment or the result of an aggregation result is admissible is polynomial in the case of *IC*—as it corresponds to model-checking—and NP-complete in any formula-based language—as it corresponds to satisfiability checking.

However, in this section we show that this kind of gain does not always transfer to more complex tasks, such as determining the result of using an aggregation rule, a problem which often is called the *winner determination problem* in the literature, due to its close links with the problem of computing the winners in an election. In what follows we focus on the two languages of *JA* and *IC*, for which an extensive analysis of the computational complexity of the winner determination problem already exists in the literature (Endriss, Grandi, and Porello 2012; Grandi 2012; Lang and Slavkovik 2014; Endriss and de Haan 2015).

5.1 Aggregation Rules

Let $\mathcal{N} = \{1, \dots, n\}$ be a finite set of n individuals. For the sake of simplicity, we will always assume that n is odd. Let a *profile* be a collection of individual views, be they binary ballots $\boldsymbol{B} = (B_1, \dots, B_n)$ or judgment sets $\boldsymbol{J} = (J_1, \dots, J_n)$, one for each individual. An *aggregation rule* is a function that associates with each profile of ballots/judgment sets a single collective ballot/judgment set (or possibly a set of such collective ballots/judgment sets, in case there is a tie).

We now introduce the specific aggregation rules for which we study the winner determination problem. We use the formalism of *IC* to state our definitions. Equivalent formulations can be easily obtained in *JA*.[6] If B is a ballot, i.e., $B \in \{0,1\}^{\mathcal{I}}$, we indicate with b_j the value it takes on issue j. If $\boldsymbol{B}$ is a profile of ballots, then $b_{i,j}$ is the opinion of voter i on issue j. The *Hamming distance* between two ballots is defined as the number of issues on which they differ: $H(B, B') = |\{j \in \mathcal{I} \mid b_j \neq b'_j\}|$.

Definition 8. *The following are aggregation rules. In each case, Γ is an integrity constraint.*

(i) *For a given quota $k \leqslant n$, the* ***uniform quota rule*** Q_k *is defined as $Q_k(\boldsymbol{B})_j = 1$ iff $|\{i \in \mathcal{N} \mid b_{i,j} = 1\}| \geqslant k$.*

(ii) *The* ***majority rule*** Maj *is defined as Q_k with $k = \frac{n+1}{2}$.*

(iii) *The* ***Kemeny rule*** *is defined as follows:*

$$\text{Kemeny}^{\Gamma}(\boldsymbol{B}) = \underset{B \models \Gamma}{\text{argmin}} \sum_{i \in \mathcal{N}} H(B, B_i)$$

(iv) *The* ***Slater rule*** *is defined as follows:*

$$\text{Slater}^{\Gamma}(\boldsymbol{B}) = \underset{B \models \Gamma}{\text{argmin}}\, H(B, \text{Maj}(\boldsymbol{B}))$$

(v) *The* ***maximum subagenda rule*** *is defined as follows:*[7]

$$\text{MSA}^{\Gamma}(\boldsymbol{B}) = \underset{B \models \Gamma}{\text{argmax}^{\subseteq}} \{j \in \mathcal{I} \mid b_j = \text{Maj}(\boldsymbol{B})_j\}$$

(vi) *For every $k \leqslant n$, the* ***binomial-k rule*** *is defined as:*

$$\text{Bin}_k^{\Gamma}(\boldsymbol{B}) = \underset{B \models \Gamma}{\text{argmax}} \sum_{i \in \mathcal{N}} \binom{n - H(B, B_i)}{k}$$

Thus, a uniform quota rule simply accepts an issue when at least a given number k of the individuals do. The quota rules have been studied in depth by Dietrich and List (2007b). The Kemeny rule returns those of the admissible outcomes that minimise the sum of the Hamming distances to the individual ballots in the profile, while the Slater rule minimises the distance to the majority outcome (which itself may not be admissible). The names of these two rules are due to their close similarity to well-known preference aggregation rules with the same names. The maximum subagenda rule returns those admissible ballots that, in terms of set-inclusion, have maximal agreement with the majority outcome. The latter three rules appear in the literature under a variety of different names (Miller and Osherson 2009; Lang et al. 2011; Endriss, Grandi, and Porello 2012; Nehring, Pivato, and Puppe 2014; Dietrich 2014). The binomial-k rules maximise the number of subsets of issues of size k that the outcome agrees on with the individual ballots. So for $k = 1$ we obtain the Kemeny rule as a special case (Costantini, Groenland, and Endriss 2016).

Not all aggregation rules are *collectively rational*, i.e., they may output an inadmissible ballot on an admissible profile. Characterisation results in the literature identify for what integrity constraint quota rules, and in particular the majority rule, are collectively rational (see, e.g., the surveys by Grossi and Pigozzi (2014) and Endriss (2016)). The Kemeny, Slater, maximum subagenda, and binomial-k rules are collectively rational by definition.

5.2 Winner Determination Problems

We consider the following decision problems, that allow us to compute the result of an aggregation rule on a given profile of ballots or judgments:

WinDet$^{IC}(F)$

Instance:	Integrity constraint IC, admissible profile $\boldsymbol{B}$, subset $I \subseteq \mathcal{I}$, partial ballot $\rho : I \to \{0,1\}$
Question:	Is there a $B^\star \in F(\boldsymbol{B})$ s.t. $\forall j \in I,\ b^\star_j = \rho(j)$?

[6] Formally, we say that rule F_1 for language $\mathcal{L}_1$ and rule F_2 for language $\mathcal{L}_2$ are equivalent if $\tau_1(F_1(x_1, \dots, x_n)) = \tau_2(F_2(y_1, \dots, y_m))$ whenever $\tau_1(x_i) = \tau_2(y_i)$ for all $i \in \mathcal{N}$.

[7] The operator $\text{argmax}^{\subseteq}$ is understood to return arguments that produce a result that is maximal with respect to set-inclusion.

$\textsc{WinDet}^{JA}(F)$
Instance: Agenda Φ, admissible profile $\boldsymbol{J}$, set $L \subseteq \Phi$
Question: Is there a $J^\star \in F(\boldsymbol{J})$ with $L \subseteq J^\star$?

A slightly different type of winner determination problem has been introduced by Lang and Slavkovik (2014). In that formulation, we ask whether all judgment sets in $F(\boldsymbol{B})$ contain a given element of the agenda. Thus, this formulation is dual to the formulation used here. It is straightforward to check that all the results we present in the sequel also hold, *mutatis mutandis*, for this alternative definition of the winner determination problem (sometimes replacing NP by coNP).

5.3 Computational Complexity: No Gap

In this section we show that for most of the aggregation rules defined in Definition 8 the computational complexity of their winner determination problem is independent of the specification language used. We also prove a general result that, to a certain extent, formalises the intuition that winner determination should in principle not become any harder when we move from the language *JA* to the less succinct language *IC*. We start with the following straightforward observation:

Observation 10. *Both of the problems* $\textsc{WinDet}^{JA}(Q_k)$ *and* $\textsc{WinDet}^{IC}(Q_k)$ *are in P for any quota* $k \leqslant n$.

As an aside, and without stating precise definitions here, we note that the same is true for all of the so-called *representative-voter rules* studied by Endriss and Grandi (2014), which are rules that always select an outcome from the set of feasible positions explicitly provided by one of the individuals. All of these rules clearly have trivial winner determination problems for any of our specification languages.

The following is a known result:

Proposition 11. *Both of the problems* $\textsc{WinDet}^{JA}(\text{Kemeny})$ *and* $\textsc{WinDet}^{IC}(\text{Kemeny})$ *are* Θ_2^p*-complete.*

Proof. For the language *JA*, a proof is given by Endriss, Grandi, and Porello (2012), and for the language *IC* a proof is given by Grandi (2012). □

Many hardness results for $\textsc{WinDet}^{JA}$ in the literature carry over to the problem $\textsc{WinDet}^{IC}$. This holds for hardness proofs where only agendas are used that satisfy the property that every propositional variable occurring in the agenda also appears as a separate formula in the agenda. In this case, we can employ the translation described at the end of Section 4 to transform such an agenda into an equivalent integrity constraint in polynomial time. This gives us a method of transforming the hardness proof for $\textsc{WinDet}^{JA}$ into a hardness proof for $\textsc{WinDet}^{IC}$. We give an example of a hardness result that carries over in this manner.

Proposition 12. *Both of the problems* $\textsc{WinDet}^{JA}(\text{Slater})$ *and* $\textsc{WinDet}^{IC}(\text{Slater})$ *are* Θ_2^p*-complete.*

Proof. $\textsc{WinDet}^{JA}(\text{Slater})$ has been shown to be Θ_2^p-complete by Lang and Slavkovik (2014) and Endriss and de Haan (2015). The hardness proof given in the latter paper only involves agendas that satisfy the property described above. Therefore, $\textsc{WinDet}^{IC}(\text{Slater})$ is Θ_2^p-hard as well. Membership in Θ_2^p for $\textsc{WinDet}^{IC}(\text{Slater})$ is routine. □

The final rule for which we are going to establish language-independence for the complexity of its winner determination problem is the maximum subagenda rule.

Proposition 13. *Both of the problems* $\textsc{WinDet}^{JA}(\text{MSA})$ *and* $\textsc{WinDet}^{IC}(\text{MSA})$ *are* Σ_2^p*-complete.*

Proof. The membership proof is routine in both cases. For $\textsc{WinDet}^{JA}(\text{MSA})$, the Π_2^p-hardness of the complementary problem was proven by Lang and Slavkovik (2014), even for the case of $|L| = 1$ (see Proposition 1 and the discussion at the end of Section 3 in their paper).

For $\textsc{WinDet}^{IC}(\text{MSA})$, consider the following reduction from $\textsc{WinDet}^{JA}(\text{MSA})$ with $|L| = 1$. Let $\Phi = \{\varphi_1, \neg\varphi_1, \dots, \varphi_m, \neg\varphi_m\}$ be an agenda, $\boldsymbol{J}$ a profile over Φ, and $\alpha \in \Phi$. Without loss of generality, let $\alpha = \varphi_1$. Let $\{p_1, \dots, p_r\}$ be the propositional variables appearing in $\varphi_1, \dots, \varphi_m$. Moreover, let $u_1 = |\boldsymbol{J}|$, and let $u_2 = r + m + 1$. We construct the set $P = \{p_i, p'_i \mid 1 \leqslant i \leqslant r\} \cup \{q_1, \dots, q_m\} \cup \{z_{i,j} \mid 1 \leqslant i \leqslant u_1, 1 \leqslant j \leqslant u_2\}$ of propositions. Then, we define the integrity constraint $\Gamma = (\bigvee_{i=1}^{u_1}(\bigwedge_{j=1}^{u_2} \neg z_{i,j})) \to (\bigwedge_{j=1}^{m}(q_j \leftrightarrow \varphi_j) \wedge \bigwedge_{j=1}^{r}(p_j \leftrightarrow \neg p'_j))$. Finally, if $\boldsymbol{J} = (J_1, \dots, J_{u_1})$, we define $\boldsymbol{B} = (B_1, \dots, B_{u_1})$, where for each $1 \leqslant \ell \leqslant u_1$ the ballot B_ℓ is defined as follows. For each variable $z_{i,j}$, B_ℓ sets the corresponding issue to 1 if and only if $i = \ell$. For each variable q_j, B_ℓ sets the corresponding issue to 1 if and only if $\varphi_j \in J_\ell$. Finally, B_ℓ sets all issues corresponding to variables p_i and p'_i to 0. It is readily verified that each B_ℓ satisfies Γ, due to the assignment to the variables $z_{i,j}$.

Assume $(\Phi, \boldsymbol{J}, \{\alpha\})$ is a positive instance of $\textsc{WinDet}^{JA}(\text{MSA})$; that is, there exists a maximally consistent subset S of $\text{Maj}(\boldsymbol{B})$ that contains φ_1. Let this maximally consistent subset be $\{\epsilon_i\varphi_i \mid i \in I\}$ where $\epsilon_i\varphi$ is either φ_i or $\neg\varphi_i$, and $I \subseteq \{1, \dots, m\}$. Then the following assignment α satisfies Γ, and it agrees with $\text{Maj}(\boldsymbol{B})$ on a maximal set of issues amongst all assignments satisfying Γ. For each variable q_i, α sets q_i to true if and only if $\epsilon_i\varphi_i = \varphi_i$. The assignment α sets each variable $z_{i,j}$ to false. Finally, the variables p_i and p'_i are set according to some assignment β that satisfies S, as follows. For each $1 \leqslant i \leqslant r$, if β sets p_i to true, then α sets p_i to true and p'_i to false, and if β sets p_i to false, then α sets p'_i to true and p_i to false.

Similarly, for any assignment α that satisfies Γ, that agrees with $\text{Maj}(\boldsymbol{B})$ on a maximal set of issues amongst all assignments satisfying Γ, and that contains p_1, there is a corresponding maximally consistent subset of $\text{Maj}(\boldsymbol{B})$ that contains φ_1. □

We have thus shown that, for a good number of well-known aggregation rules, the complexity of the aggregation problem is invariant under changes of the specification language, at least for what concerns the languages *JA* and *IC*. Intuitively, the problem of $\textsc{WinDet}^{JA}$ should always be at least as hard as that of $\textsc{WinDet}^{\mathit{IC}}$, but a result that formalises this intuition in its full generality seems difficult to obtain. For the classes NP, Θ_2^p, Δ_2^p, and Σ_2^p, however, membership carries over from the case of *JA* to the case of *IC*. We give a proof of this fact for the case of NP.

In what follows, we refer to arbitrary aggregation rules F. We use F to refer to the equivalent rules, for both *JA* and *IC*.

Proposition 14. *If* $\textsc{WinDet}^{\mathit{JA}}(F)$ *is in NP, then also* $\textsc{WinDet}^{\mathit{IC}}(F)$ *is in NP.*

Proof. Suppose that $\textsc{WinDet}^{\mathit{JA}}(F)$ is in NP, i.e., that there is a non-deterministic polynomial-time algorithm that solves $\textsc{WinDet}^{\mathit{JA}}(F)$. We describe a non-deterministic polynomial-time algorithm A that solves $\textsc{WinDet}^{\mathit{IC}}(F)$ (showing membership in NP). Let $(\Gamma, \boldsymbol{B}, \rho)$ specify an arbitrary input for the problem $\textsc{WinDet}^{\mathit{IC}}(F)$. Firstly, the algorithm A guesses an assignment α for Γ. If α satisfies Γ, it continues, and otherwise, it rejects the input. Then, the algorithm A uses the model α of Γ to construct an equivalent agenda Φ, using the construction in the proof of Proposition 3. Finally, the algorithm A simulates the non-deterministic polynomial-time algorithm for $\textsc{WinDet}^{\mathit{JA}}(F)$, using as input $(\Phi, \boldsymbol{J}, L)$, where $\boldsymbol{J}$ and L correspond directly to $\boldsymbol{B}$ and ρ, respectively. Since the algorithm for $\textsc{WinDet}^{\mathit{JA}}(F)$ accepts (for some sequence of non-deterministic choices) if and only if $(\Phi, \boldsymbol{J}, L)$ is a yes-instance, we get that A accepts (for some sequence of non-deterministic choices) if and only if $(\Gamma, \boldsymbol{B}, \rho)$ is a yes-instance for $\textsc{WinDet}^{\mathit{IC}}(F)$. $\square$

For the case of Δ_2^p and Σ_2^p, an analogous statement can be proven, using similar arguments (constructing an algorithm that firstly determines a model for Γ, and subsequently simulates the algorithm for the case of *JA*). For the case of Θ_2^p, a technically more involved argument is required.

5.4 Computational Complexity: Gap

The results above notwithstanding, it is not the case that the complexity of the winner determination remains *always* unaffected when we switch between languages. One aggregation rule where winner determination is easy in binary aggregation with integrity constraints but intractable in the formula-based judgment aggregation framework is the binomial-k rule for $k = m - 1$.[8]

Proposition 15. $\textsc{WinDet}^{\mathit{IC}}(\mathrm{Bin}^{\Gamma}_{m-1})$ *is polynomial, while* $\textsc{WinDet}^{\mathit{JA}}(\mathrm{Bin}^{\Gamma}_{m-1})$ *is NP-complete.*

Proof. $\textsc{WinDet}^{\mathit{IC}}(\mathrm{Bin}^{\Gamma}_{m-1})$ has been shown to be polynomial by Costantini, Groenland, and Endriss (2016). Membership of $\textsc{WinDet}^{\mathit{JA}}(\mathrm{Bin}^{\Gamma}_{m-1})$ in NP is routine. To establish NP-hardness, we provide a reduction from SAT. Suppose we want to check whether a given formula φ is consistent. Let a, b, c be propositional variables not occurring in φ, and construct an agenda Φ with $\Phi^+ = \{a, b, \psi\}$ for $\psi = a \wedge b \rightarrow (\varphi \wedge c)$. Now consider the profile consisting of the following three judgment sets: $J_1 = \{a, b, \neg\psi\}$, $J_2 = \{\neg a, b, \psi\}$, and $J_3 = \{a, \neg b, \psi\}$, all of which are consistent. Suppose we want to know whether $\{a, b, \psi\}$ is one of the winners. Under the binomial-2 rule, both this judgment set and each of the three judgment sets in the profile would receive a score of 3, while no other judgment set can possibly obtain a higher score. Thus, $\{a, b, \psi\}$ is a winner if and only if it is consistent, which is the case if and only if φ is consistent. Thus, we can solve SAT for φ by solving this particular instance of the winner determination problem. $\square$

Let us informally describe one further family of rules for which there also is a complexity gap. Consider any aggregation rule that returns the majority outcome when it is admissible and that carries out some simple computation—that is polynomial for both of our languages—in all other cases. For any such rule, winner determination will be polynomial for *IC* but NP-hard for *JA*, because for the latter language we need to carry out a satisfiability check to determine whether or not the majority outcome is admissible. A natural rule of this kind, which however has not been considered in the literature before, would be the rule that returns the majority outcome when it is admissible, and that in all other cases returns the outcome of the *majority-voter rule* defined by Endriss and Grandi (2014). The latter rule returns the "most representative" individual judgment as the collective judgment, in the sense of being the individual judgment that minimises the Hamming distance to the majority outcome.

6 Conclusions

We have studied the relative succinctness of four compact languages for the representation of a collective decision making problem in a logically structured combinatorial domain. Our main result shows that the formula-based approach used in judgment aggregation is strictly more succinct than the constraint-based one used in binary aggregation. We have also studied the translation from one language to the other, and its computational complexity, in particular by means of additional languages that combine a formula-based description of the issues with a constraint.

One of the most obvious problems we face in judgment aggregation is that of winner determination: given a profile of judgments and an aggregation rule, we want to compute the collective judgment returned by the rule. We have related our study of succinctness with results on the computational complexity of this winner determination problem, showing that for a number of well-known rules the complexity of the associated problem does not change when we change the specification language that is used. At the same time, we have seen that this tendency is not a universal law and that there are meaningful, albeit much less widely used, aggregation rules where complexity is positively affected by switching to the less succinct specification language.

A full characterisation of rules for which the complexity of winner determination is language-dependent represents an interesting open problem, as does the identification of fragments of the formula-based language for which the corresponding integrity constraints are polynomially bounded.

[8] The following result easily generalises to any $k > 0$ for which the difference $m - k$ is a constant.

Acknowledgements. This work has been partly supported by COST Action IC1205 on Computational Social Choice, by the FWF Austrian Science Fund (Parameterized Compilation, P26200), and and by the French National Research Agency (Project ANR-14-CE24-0007-01 "CoCoRICo-CoDec").

References

Arora, S., and Barak, B. 2009. *Computational Complexity — A Modern Approach*. Cambridge University Press.

Cadoli, M., and Donini, F. M. 1997. A survey on knowledge compilation. *AI Communications* 10(3–4):137–150.

Cadoli, M.; Donini, F. M.; Liberatore, P.; and Schaerf, M. 2000. Space efficiency of propositional knowledge representation formalisms. *J. of Artificial Intelligence Research (JAIR)* 13:1–31.

Costantini, M.; Groenland, C.; and Endriss, U. 2016. Judgment aggregation under issue dependencies. In *Proc. 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*.

Coste-Marquis, S.; Lang, J.; Liberatore, P.; and Marquis, P. 2004. Expressive power and succinctness of propositional languages for preference representation. In *Proc. 9th International Conference on Principles of Knowledge Representation and Reasoning (KR-2004)*.

Darwiche, A., and Marquis, P. 2002. A knowledge compilation map. *J. of Artificial Intelligence Research (JAIR)* 17:229–264.

Dietrich, F., and List, C. 2007a. Arrow's theorem in judgment aggregation. *Social Choice and Welfare* 29(1):19–33.

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

Dietrich, F., and List, C. 2008. Judgment aggregation without full rationality. *Social Choice and Welfare* 31(1):15–39.

Dietrich, F. 2007. A generalised model of judgment aggregation. *Social Choice and Welfare* 28(4):529–565.

Dietrich, F. 2014. Scoring rules for judgment aggregation. *Social Choice and Welfare* 42(4):873–911.

Dokow, E., and Holzman, R. 2009. Aggregation of binary evaluations for truth-functional agendas. *Social Choice and Welfare* 32(2):221–241.

Dokow, E., and Holzman, R. 2010. Aggregation of binary evaluations. *Journal of Economic Theory* 145(2):495–511.

Endriss, U., and de Haan, R. 2015. Complexity of the winner determination problem in judgment aggregation: Kemeny, Slater, Tideman, Young. In *Proc. 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2015)*.

Endriss, U., and Grandi, U. 2014. Binary aggregation by selection of the most representative voter. In *Proc. 28th AAAI Conference on Artificial Intelligence (AAAI-2014)*.

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

Endriss, U. 2016. Judgment aggregation. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press.

Everaere, P.; Konieczny, S.; and Marquis, P. 2014. Propositional merging and judgment aggregation: Two compatible approaches? In *Proc. 21st European Conference on Artificial Intelligence (ECAI-2014)*.

Everaere, P.; Konieczny, S.; and Marquis, P. 2015. Belief merging versus judgment aggregation. In *Proc. 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2015)*.

French, T.; van der Hoek, W.; Iliev, P.; and Kooi, B. 2013. On the succinctness of some modal logics. *Artif. Intelligence* 197:56–85.

Gärdenfors, P. 2006. A representation theorem for voting with logical consequences. *Economics and Philosophy* 22(2):181–190.

Gogic, G.; Kautz, H. A.; Papadimitriou, C. H.; and Selman, B. 1995. The comparative linguistics of knowledge representation. In *Proc. 14th International Joint Conference on Artificial Intelligence (IJCAI-1995)*.

Grandi, U., and Endriss, U. 2013. Lifting integrity constraints in binary aggregation. *Artificial Intelligence* 199–200:45–66.

Grandi, U. 2012. *Binary Aggregation with Integrity Constraints*. Ph.D. Dissertation, ILLC, University of Amsterdam.

Grossi, D., and Pigozzi, G. 2014. *Judgment Aggregation: A Primer*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Klamler, C., and Pferschy, U. 2007. The traveling group problem. *Social Choice and Welfare* 29(3):429–452.

Konieczny, S., and Pino Pérez, R. 2011. Logic based merging. *Journal of Philosophical Logic* 40(2):239–270.

Kornhauser, L. A., and Sager, L. G. 1993. The one and the many: Adjudication in collegial courts. *California Law Review* 81(1):1–59.

Lang, J., and Slavkovik, M. 2014. How hard is it to compute majority-preserving judgment aggregation rules? In *Proc. 21st European Conference on Artificial Intelligence (ECAI-2014)*.

Lang, J., and Xia, L. 2016. Voting in combinatorial domains. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press.

Lang, J.; Pigozzi, G.; Slavkovik, M.; and van der Torre, L. 2011. Judgment aggregation rules based on minimization. In *Proc. 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2011)*.

Lang, J.; Liberatore, P.; and Marquis, P. 2003. Propositional independence: Formula-variable independence and forgetting. *Journal of Artificial Intelligence Research (JAIR)* 18:391–443.

Lin, F., and Reiter, R. 1994. Forget it! In *Proc. AAAI Fall Symposium on Relevance*.

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

Marquis, P. 2015. Compile! In *Proc. 29th AAAI Conference on Artificial Intelligence (AAAI-2015)*.

Miller, M. K., and Osherson, D. 2009. Methods for distance-based judgment aggregation. *Social Choice and Welfare* 32(4):575–601.

Nebel, B. 2000. On the compilability and expressive power of propositional planning formalisms. *Journal of Artificial Intelligence Research (JAIR)* 12:271–315.

Nehring, K. D., and Puppe, C. 2010. Abstract Arrowian aggregation. *Journal of Economic Theory* 145(2):467–494.

Nehring, K.; Pivato, M.; and Puppe, C. 2014. The Condorcet set: Majority voting over interconnected propositions. *Journal of Economic Theory* 151:268–303.

Pauly, M., and van Hees, M. 2006. Logical constraints on judgement aggregation. *Journal of Philosophical Logic* 35(6):569–585.

Uckelman, J.; Chevaleyre, Y.; Endriss, U.; and Lang, J. 2009. Representing utility functions via weighted goals. *Mathematical Logic Quarterly* 55(4):341–361.