Parameterized Complexity Results for Agenda Safety in Judgment Aggregation

Ulle Endriss[1], Ronald de Haan[2,*], Stefan Szeider[2,*]

[1] Institute for Logic, Language and Computation, University of Amsterdam
[2] Institute of Information Systems, Vienna University of Technology

Abstract

Many problems arising in computational social choice are of high computational complexity, and some are located at higher levels of the Polynomial Hierarchy. We argue that a parameterized complexity analysis provides a lot of insight about the factors contributing to the complexity of these problems, and can lead to practically useful algorithms. As a case study, we consider the problem of agenda safety in judgment aggregation, consider several natural parameters for this problem, and determine the parameterized complexity for each of these. Our analysis is aimed at obtaining fixed-parameter tractable (fpt) algorithms that use a small number of calls to a SAT solver. We hope that this work may initiate a structured parameterized complexity investigation of problems arising in the field of computational social choice that are located at higher levels of the Polynomial Hierarchy. A by-product of our case study is the development of complexity-theoretic techniques to provide lower bounds on the number of SAT calls needed by fpt-algorithms to solve certain problems.

1 Introduction

The field of computational social choice studies the interface of social choice theory and computer science. In particular, it is concerned with investigating properties of computational tasks related to procedures for collective decision making. Some of these computational tasks have a computational complexity that is 'beyond NP', and are thus considered to be highly intractable (cf. [2, 10, 24, 25]). We argue that the complexity analysis of problems arising in computational social choice that are 'beyond NP' benefits from a parameterized complexity approach [16, 17, 20, 32]. Recent advances in parameterized complexity theory [23] enable an investigation of the restrictions that allow an encoding of problems 'beyond NP' into the Boolean satisfiability problem (SAT). With the success that modern SAT solving algorithms have had in many practical settings over the last two decades [29, 34], this might lead to practically useful algorithms for problems that are traditionally considered to be highly intractable.

As a case study to underpin our argument, we consider the computational complexity of the problem of *agenda safety*, which is a computational problem that arises in the domain of judgment aggregation. Judgment aggregation studies the properties of procedures that combine the individual judgments on a set of related propositions (the *agenda*) of the members of a group into a collective judgment reflecting the views of the group as a whole [28]. Such procedures might, in general, yield inconsistent combined judgments. Therefore, it is useful to determine for a given agenda and a given aggregation procedure whether there exists no combination of individual judgments such that the outcome of the procedure is inconsistent (we say that the agenda is *safe* if this is the case). This is relevant, for instance,

*Supported by the European Research Council (ERC), project 239962 (COMPLEX REASON), and the Austrian Science Fund (FWF), project P26200 (Parameterized Compilation).

Parameter	Complexity
maximum formula size (ℓ)	para-Π_2^{P}-complete (Proposition 4)
maximum variable degree (d)	para-Π_2^{P}-complete (Proposition 6)
$\ell + d$	para-Π_2^{P}-complete (Proposition 6), even when restricted to 2CNF $\cap$ HORN
number of formulas	solvable in fpt-time with $f(k)$ many SAT calls, with $f(k) = 2^{O(k)}$ (Theorem 8) and $f(k) = \Omega(\log k)$ (Theorem 19)
counterexample size	$\forall^k\exists^*$-hard (Theorem 21)

Table 1: Complexity results for different parameterizations of agenda safety.

in the setting of multi-agent systems where agents need to coordinate their beliefs, intentions and actions repeatedly [36]. The problem of agenda safety is complete for the second level of the Polynomial Hierarchy (PH) [18], and is thus 'beyond NP.'

Instances of hard computational problems that occur in practice often exhibit some kind of structure. A classical complexity analysis is insensitive to any such structure. A parameterized complexity analysis, on the other hand, can take into account different forms of structure in the problem instances, by means of problem parameters. The idea underlying parameterized complexity theory is that such parameters are expected to be small in problem instances occurring in practice. By restricting the high complexity of a problem to the parameter only, these structured instances of hard computational problems can often be solved reasonably efficiently. There has been a lot of research in the field of parameterized complexity over the last two decades (cf. [9]). Most of this research is aimed at problems that are in NP. Recently, tools have been developed to analyze the parameterized complexity of problems that are located higher in the PH [22, 23]. The paradigm of parameterized complexity has been used to examine many problems in computational social choice (cf. [3, 4, 5, 15]).

Contributions. Concretely, we investigate what kind of structure helps to decrease the computational complexity of the problem of agenda safety for the majority rule. We do this by studying several natural parameterizations of the problem. The main concept of tractability that we have in mind is based on algorithms that run efficiently for small parameter values, and that use only a small number of SAT calls (depending on the parameter value only). This notion of tractability is motivated by the enormous practical success of modern SAT solvers [8, 21, 29, 34]. For precise definitions, we refer to Section 2.

Several parameterizations that we consider correspond to syntactic restrictions on the agenda (i.e., bounds on the size of formulas, bounds on variable occurrence, and bounds on the number of formulas). Another parameterization corresponds to a bound on the size of counterexamples (to the logical characterization of agenda safety), and is similar to parameterizations that have been applied successfully in other domains [6, 7]. An overview of complexity results for these parameterizations can be found in Table 1.

This parameterized complexity analysis allows us to pinpoint exactly what aspects of the problem play what role in the high computational complexity of the problem, and it helps to determine what algorithmic approach is best suited to solve the problem in practical settings. We hope that this work can help initiate a structured parameterized complexity investigation of problems arising in the field of computational social choice that are located at higher levels of the PH.

As a by-product of our case study we develop complexity-theoretic techniques to provide lower bounds on the number of SAT calls needed by fpt-algorithms to solve certain problems.

These techniques are based on novel parameterized complexity classes, related to the Boolean Hierarchy.

2 Preliminaries

In this section, we formally define the problem of agenda safety and we provide a logical characterization of the problem for a particular aggregation procedure. Moreover, we define notions from classical and parameterized complexity theory that we will need in our analysis.

Propositional Logic and Agenda Safety. A *literal* is a propositional variable x or a negated variable $\neg x$. For literals $l \in \{x, \neg x\}$, we let $\text{Var}(l) = x$ denote the variable occurring in l. A *clause* is a finite set of literals, not containing a complementary pair x, $\neg x$, and is interpreted as the disjunction of these literals. We let $\bot$ denote the empty clause. A formula in *conjunctive normal form (CNF)* is a finite set of clauses, interpreted as the conjunction of these clauses. We define the *size* $\|\varphi\|$ of a CNF formula φ to be $\sum_{c \in \varphi} |c|$; the number of clauses of φ is denoted by $|\varphi|$. For a CNF formula φ, the set $\text{Var}(\varphi)$ denotes the set of all variables x such that some clause of φ contains x or $\neg x$. We say that a clause is a *Horn clause* if it contains at most one positive literal; a CNF formula is a *Horn formula* if it contains only Horn clauses. We let the *degree* of a CNF formula φ be the maximum number of times that any variable $x \in \text{Var}(\varphi)$ occurs in φ. We define the degree of a set Φ of CNF formulas to be the maximum number of times that any variable $x \in \text{Var}(\Phi)$ occurs in Φ. We use the standard notion of *(truth) assignments* $\alpha : \text{Var}(\varphi) \to \{0, 1\}$ for Boolean formulas and *truth* of a formula under such an assignment. We let SAT denote the problem of deciding whether a given propositional formula is satisfiable, and we let UNSAT denote its co-problem, i.e., deciding whether a given formula is unsatisfiable. For every propositional formula φ, we let $\sim\varphi$ denote the *complement* of φ, i.e., $\sim\varphi = \neg\varphi$ if φ is not of the form $\neg\psi$, and $\sim\varphi = \psi$ if φ is of the form $\neg\psi$.

An *agenda* is a finite nonempty set Φ of formulas that does not contain any doubly-negated formulas and that is closed under complementation. Moreover, if $\Phi = \{\varphi_1, \dots, \varphi_n, \neg\varphi_1, \dots, \neg\varphi_n\}$ is an agenda, then we let $B(\Phi) = \{\varphi_1, \dots, \varphi_n\}$ denote the *base* of the agenda Φ. A *judgment set* J for an agenda Φ is a subset $J \subseteq \Phi$. We call a judgment set J *complete* if $\varphi \in J$ or $\sim\varphi \in J$ for all $\varphi \in \Phi$; we call it *complement-free* if for all $\varphi \in \Phi$ it is not the case that both φ and $\sim\varphi$ are in J; and we call it *consistent* if there exists an assignment that makes all formulas in J true. Let $\mathcal{J}(\Phi)$ denote the set of all complete and consistent subsets of Φ. We call a sequence $\boldsymbol{J} \in \mathcal{J}(\Phi)^{|\mathcal{N}|}$ of complete and consistent subsets a *profile*. A (resolute) *judgment aggregation procedure* for the agenda Φ and the set of individuals $\mathcal{N}$ is a function $F : \mathcal{J}(\Phi)^{|\mathcal{N}|} \to 2^{\Phi}$. An example is the *majority rule* F^{maj}, where $\varphi \in F^{\text{maj}}(\boldsymbol{J})$ if and only if φ occurs in the majority of judgment sets in $\boldsymbol{J}$, for all $\varphi \in \Phi$. We call F *complete*, *complement-free* and *consistent*, if $F(\boldsymbol{J})$ is complete, complement-free and consistent, respectively, for every $\boldsymbol{J} \in \mathcal{J}(\Phi)^n$. An agenda Φ is *safe* with respect to a class of aggregation procedures $\mathcal{F}$, if every procedure in $\mathcal{F}$ is consistent when applied to profiles of judgment sets over Φ. We say that an agenda Φ satisfies the *median property (MP)* if every inconsistent subset of Φ has itself an inconsistent subset of size at most 2. An agenda Φ is safe for the majority rule if and only if Φ satisfies the MP [18, 31]. There exist similar properties that characterize agenda safety for other aggregation procedures [18].

As an example, we consider the *discursive dilemma*, which concerns an agenda that is not safe for the majority rule. Consider the agenda $\Phi_{\text{dd}} = \{p, \neg p, q, \neg q, (p \to q), \neg(p \to q)\}$. Moreover, consider the profile $\boldsymbol{J} = (J_1, J_2, J_3)$, where $J_1 = \{p, q, (p \to q)\}$, $J_2 = \{p, \neg q, \neg(p \to q)\}$, and $J_3 = \{\neg p, \neg q, (p \to q)\}$. Clearly, $F^{\text{maj}}(\boldsymbol{J}) = \{p, \neg q, (p \to q)\}$, which is inconsistent. In other words, Φ_{dd} is not safe for the majority rule. Also, Φ_{dd} does not satisfy the MP, as it contains a subset $F^{\text{maj}}(\boldsymbol{J}) \subseteq \Phi$ that is inconsistent, but that itself contains no inconsistent subset of size 2. Intuitively, for each agenda that does not satisfy the MP, a similar discursive dilemma can be constructed, where the majority rule is forced to include an inconsistent subset (of size larger than 2), whereas the individual profiles remain consistent.

In this paper, we consider several parameterizations of the following decision problem, which is shown to be Π_2^{P}-complete [18]. For our results, we will use the fact that deciding safety of an agenda Φ for the majority rule is equivalent to checking whether Φ satisfies the median property. In fact, the technical details behind our results involve only this alternative characterization.

Agenda-Safety$^{\text{maj}}$
Instance: An agenda Φ.
Question: Is Φ safe for the majority rule?

The Boolean and Polynomial Hierarchies. There are many natural decision problems that are not contained in the classical complexity classes P or NP. The *Boolean Hierarchy* (BH) [11, 12, 26] consists of a hierarchy of complexity classes BH_i for all $i \geq 1$. Each class BH_i can be characterized as the class of problems that can be reduced to the problem BH_i-Sat, which is defined inductively as follows. The problem BH_1-Sat consists of all sequences (φ), where φ is a satisfiable propositional formula. For even $i \geq 2$, the problem BH_i-Sat consists of all sequences $(\varphi_1, \dots, \varphi_i)$ of propositional formulas such that both $(\varphi_1, \dots, \varphi_{i-1}) \in \text{BH}_{(i-1)}$-Sat and φ_i is unsatisfiable. For odd $i \geq 2$, the problem BH_i-Sat consists of all sequences $(\varphi_1, \dots, \varphi_i)$ of propositional formulas such that $(\varphi_1, \dots, \varphi_{i-1}) \in \text{BH}_{(i-1)}$-Sat or φ_i is satisfiable. The class BH_2 is also denoted by DP, and the problem BH_2-Sat is also denoted by SAT-UNSAT.

The *Polynomial Hierarchy* (PH) [30, 37, 39, 33] consists of a hierarchy of complexity classes, including the classes Σ_i^{P}, for all $i \geq 0$. The class Σ_2^{P} already contains the entire BH. We give a characterization of these classes based on the satisfiability problem of various classes of quantified Boolean formulas. A *(prenex) quantified Boolean formula* is a formula of the form $Q_1 X_1 Q_2 X_2 \dots Q_m X_m \psi$, where each Q_i is either $\forall$ or $\exists$, the X_i are disjoint sets of propositional variables, and ψ is a Boolean formula over the variables in $\bigcup_{i=1}^{m} X_i$. The quantifier-free part of such formulas is called the *matrix* of the formula. Truth of such formulas is defined in the usual way. We let $\psi[\alpha]$ denote the formula obtained from ψ by instantiation variables by their truth values given by a (partial) truth assignment α. For each $i \geq 1$ we define the following decision problem.

QSat$_i$
Instance: A quantified Boolean formula $\varphi = \exists X_1 \forall X_2 \exists X_3 \dots Q_i X_i \psi$, where Q_i is a universal quantifier if i is even and an existential quantifier if i is odd.
Question: Is φ true?

For each nonnegative integer $i \geq 0$, the complexity class Σ_i^{P} is the class of problems that can be reduced to QSat$_i$ in polynomial time [37, 39]. The Σ_i^{P}-hardness of QSat$_i$ holds already when the matrix of the input formula is restricted to 3CNF for odd i, and restricted to 3DNF for even i. Note that the class Σ_0^{P} coincides with P, and the class Σ_1^{P} coincides with NP. For each $i \geq 1$, the class Π_i^{P} is defined as co-Σ_i^{P}.

Parameterized Complexity. We introduce some core notions from parameterized complexity theory that we will use in this paper. For an in-depth treatment we refer to other sources [16, 17, 20, 23, 32]. A *parameterized problem* L is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ. For an instance $(I, k) \in \Sigma^* \times \mathbb{N}$, we call I the *main part* and k the *parameter*. The following generalization of polynomial time computability is commonly regarded as the tractability notion of parameterized complexity theory. A parameterized problem L is *fixed-parameter tractable* if there exists a computable function f and a constant c such that there exists an algorithm that decides whether $(I, k) \in L$ in time $O(f(k)\|I\|^c)$, where $\|I\|$ denotes the size of I. Such an algorithm is called an *fpt-algorithm*, and this amount of time is called *fpt-time*. FPT is the class of all fixed-parameter tractable parameterized decision problems. If the parameter is constant, then fpt-algorithms run in polynomial time where the order of the polynomial is independent of the parameter. This provides a good scalability in the parameter in contrast to running times of the form $\|I\|^k$, which are also polynomial for fixed k, but are already impractical for, say, $k > 3$. By XP we denote the class of all problems L for which it can be decided whether $(I, k) \in L$ in time $O(\|I\|^{f(k)})$, for some fixed computable function f.

Parameterized complexity also generalizes the notion of polynomial-time reductions. Let $L \subseteq \Sigma^* \times \mathbb{N}$ and $L' \subseteq (\Sigma')^* \times \mathbb{N}$ be two parameterized problems. An *fpt-reduction* from L to L' is a mapping $R : \Sigma^* \times \mathbb{N} \to (\Sigma')^* \times \mathbb{N}$ from instances of L to instances of L' such that there exist some computable function $g : \mathbb{N} \to \mathbb{N}$ such that for all $(I, k) \in \Sigma^* \times \mathbb{N}$: (i) (I, k) is a yes-instance of L if and only if $(I', k') = R(I, k)$ is a yes-instance of L', (ii) $k' \leq g(k)$, and (iii) R is computable in fpt-time. Similarly, we call reductions that satisfy properties (i) and (ii) but that are computable in time $O(\|I\|^{f(k)})$, for some fixed computable function f, *xp-reductions*.

Parameterized complexity theory also offers complexity classes corresponding to classes in the Polynomial Hierarchy. Let C be a classical complexity class, e.g., NP. The parameterized complexity class para-C is then defined as the class of all parameterized problems $L \subseteq \Sigma^* \times \mathbb{N}$, for some finite alphabet Σ, for which there exists an alphabet Π, a computable function $f : \mathbb{N} \to \Pi^*$, and a problem $P \subseteq \Sigma^* \times \Pi^*$ such that $P \in C$ and for all instances $(x, k) \in \Sigma^* \times \mathbb{N}$ of L we have that $(x, k) \in L$ if and only if $(x, f(k)) \in P$. Intuitively, the class para-C consists of all problems that are in C after a precomputation that only involves the parameter [19].

In particular, the class para-NP contains those parameterized problems that can be fpt-reduced to a single instance of SAT. Another class containing problems that can be considered fpt-reducible to SAT is the class para-DP, based on the classical complexity class $\text{DP} = \{ L_1 \cap L_2 : L_1 \in \text{NP}, L_2 \in \text{co-NP} \}$. An instance of a parameterized problem in para-DP can be solved in fpt-time by firstly reducing it to an instance of the problem SAT-UNSAT $= \{ (\varphi_1, \varphi_2) : \varphi_1 \in \text{SAT}, \varphi_2 \in \text{UNSAT} \}$, and then solving this resulting instance by invoking a SAT oracle twice.

In addition to many-one fpt-reductions to SAT, we are also interested in Turing fpt-reductions. A *Turing fpt-reduction from a problem P to SAT* is an fpt-algorithm that has access to a SAT oracle and that decides P. We are mainly interested in fpt-algorithms that only use a small number of queries to the SAT oracle (*SAT calls*). We let $\text{FPT}^{\text{NP}[f(k)]}$ denote the class of all parameterized problems P for which there exists an fpt-algorithm that decides if $(x, k) \in P$ by using at most $f(k)$ many SAT calls, for some computable function f.

On the other hand, para-Σ_2^{P}-hardness can be employed to provide evidence against the existence of fpt-reductions to SAT. However, for many interesting parameterized problems for which we want to investigate the (non-)existence of fpt-reductions to SAT, hardness for para-Σ_2^{P} cannot be used. The class para-Σ_2^{P} contains problems that cannot be reduced to SAT in polynomial time if the parameter value is a constant (unless the Polynomial Hierarchy collapses at the first level), i.e., problems in para-Σ_2^{P} do not allow an xp-reduction to SAT. Since many problems we are interested in do allow such xp-reductions to SAT, it is unlikely that these problems can be shown to be hard for the complexity class para-Σ_2^{P}.

Recent work in parameterized complexity theory has resulted in complexity classes that can be used to provide evidence for the non-existence of fpt-reductions to SAT also

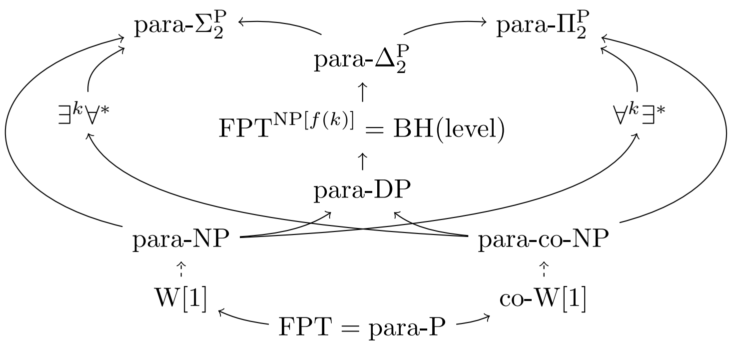

Figure 1: Parameterized complexity classes relevant to the results in this paper. Arrows indicate inclusion relations. For a definition of the classes W[1] and co-W[1], we refer to other sources [16, 17, 20].

for problems that do allow an xp-reduction to SAT [23]. The parameterized complexity class $\forall^k\exists^*$ consists of all parameterized problems that can be fpt-reduced to the following variant of quantified Boolean satisfiability that is based on truth assignments of restricted (Hamming) weight (the Hamming weight of an assignment is the number of variables that it assigns to 1).

$\forall^k\exists^*$-WSAT
Instance: A quantified Boolean formula $\varphi = \forall X.\exists Y.\psi$, and an integer k.
Parameter: k.
Question: Is it the case that for all truth assignments α to X with weight k there exists an assignment β to Y such that the assignment $\alpha \cup \beta$ satisfies ψ?

For any problem in $\forall^k\exists^*$ there exists an xp-reduction to SAT. However, there is evidence that problems that are hard for $\forall^k\exists^*$ do not allow an fpt-reduction to SAT [22, 23]. Many natural parameterized problems from various domains are complete for the class $\forall^k\exists^*$, and for none of them an fpt-reduction to SAT has been found. If there exists an fpt-reduction to SAT for any $\forall^k\exists^*$-complete problem then this is the case for all $\forall^k\exists^*$-complete problems. For an overview of parameterized complexity classes that are relevant to the results in this paper, we refer to Figure 1. For a more detailed discussion on this topic, we refer to previous work in parameterized complexity [23].

3 Parameterized Complexity Results

We start with showing that we can restrict our attention to agendas containing only formulas in CNF. We show how to transform any agenda Φ to an agenda Φ', containing only formulas in CNF (and their negations), that is safe if and only if Φ is safe, where the size of Φ' is polynomial in the size of Φ.

Lemma 1. *Let φ be a propositional formula. We can construct a CNF formula φ' such that* $\text{Var}(\varphi') \supseteq \text{Var}(\varphi)$ *and for each truth assignment* $\alpha : \text{Var}(\varphi) \to \{0,1\}$ *we have that α satisfies φ if and only if there exists an assignment* $\beta : (\text{Var}(\varphi') \backslash \text{Var}(\varphi)) \to \{0,1\}$ *such that the assignment $\alpha \cup \beta$ satisfies φ'.*

Proof (idea). The idea of the proof is to transform φ into a CNF formula ψ by using the well-known Tseitin transformation [38]. For each subformula χ of φ we add a fresh variable x_χ, and we construct the clauses of ψ in such a way that the truth value of x_χ in any satisfying assignment of ψ corresponds to the truth value of χ, for each subformula χ. A full proof can be found in the appendix. □

Proposition 2. *Let Φ be an agenda with base $B(\Phi) = \{\varphi_1, \dots, \varphi_n\}$. We can construct in polynomial time an agenda Φ' with base $B(\Phi') = \{\varphi'_1, \dots, \varphi'_n\}$ such that each φ'_i is in CNF and any subset $\Psi = \{\varphi_{i_1}, \dots, \varphi_{i_{m_1}}, \neg\varphi_{j_1}, \dots, \neg\varphi_{j_{m_2}}\}$ of Φ is consistent if and only if $\Psi' = \{\varphi'_{i_1}, \dots, \varphi'_{i_m}, \neg\varphi'_{j_1}, \dots, \neg\varphi'_{j_{m_2}}\}$ is consistent.*

Proof. Let Φ be an agenda with base $B(\Phi) = \{\varphi_1, \dots, \varphi_n\}$. By Lemma 1, we can transform in polynomial time each φ_i to a suitable CNF formula φ'_i. Because we can introduce fresh variables for constructing each φ'_i, we can assume without loss of generality that for each $1 \leq i < i' \leq n$ it is the case that $(\mathrm{Var}(\varphi'_i)\backslash\mathrm{Var}(\varphi_i)) \cap (\mathrm{Var}(\varphi'_{i'})\backslash\mathrm{Var}(\varphi_{i'})) = \emptyset$. Let $\Psi = \{\varphi_{i_1}, \dots, \varphi_{i_{m_1}}, \neg\varphi_{j_1}, \dots, \neg\varphi_{j_{m_2}}\}$ be an arbitrary subset of Φ. We claim that Ψ is consistent if and only if $\Psi' = \{\varphi'_{i_1}, \dots, \varphi'_{i_{m_1}}, \neg\varphi'_{j_1}, \dots, \neg\varphi'_{j_{m_2}}\}$ is consistent. A full proof of this claim can be found in the appendix. □

Thus, the problem AGENDA-SAFETY$^{\text{maj}}$ is Π_2^{P}-hard even for the following restricted case.

Corollary 3. *The problem* AGENDA-SAFETY$^{\text{maj}}$ *is Π_2^{P}-hard even when restricted to agendas Φ whose base $B(\Phi)$ contains only* CNF *formulas.*

Intuitively, the above results show that, using additional auxiliary variables, each agenda can be rewritten into another agenda that contains only formulas in CNF (or their negation) that are equivalent (with respect to satisfiability) to the formulas in the original agenda.

3.1 Syntactic restrictions on the agenda

We consider the following parameterizations of the agenda safety problem that correspond to syntactic restrictions on the agenda Φ. We parameterize on the size of formulas $\varphi \in \Phi$, on the maximum number of times any variable occurs in Φ (i.e., the degree of Φ), and on the number of formulas occurring in Φ.

AGENDA-SAFETY$^{\text{maj}}$(formula size)
Instance: An agenda Φ.
Parameter: $\ell = \max\{\, |\varphi| : \varphi \in \Phi \,\}$.
Question: Is Φ safe for the majority rule?

AGENDA-SAFETY$^{\text{maj}}$(degree)
Instance: An agenda Φ containing only CNF formulas.
Parameter: The degree d of Φ.
Question: Is Φ safe for the majority rule?

AGENDA-SAFETY$^{\text{maj}}$(degree + formula size)
Instance: An agenda Φ containing only CNF formulas, where $\ell = \max\{\, |\varphi| : \varphi \in B(\Phi) \,\}$, and where d is the degree of Φ.
Parameter: $\ell + d$.
Question: Is Φ safe for the majority rule?

AGENDA-SAFETY$^{\text{maj}}$(agenda size)
Instance: An agenda Φ.
Parameter: $|\Phi|$.
Question: Is Φ safe for the majority rule?

The assumption that the size of formulas in an agenda is small corresponds to the expectation that the separate statements that the individuals are judging are in a sense atomic, and therefore of bounded size. The supposition that the degree of an agenda is small corresponds to the expectation that each proposition that occurs in the statements to be judged occurs only a small number of times. The assumption that the number of formulas in the agenda is small is based on the fact that the individuals need to form an opinion on all formulas in the agenda.

Agendas with small formulas and small degree. We start by showing that parameterizing on (the sum of) the maximum formula size and the degree of the agenda Φ does not decrease the complexity of deciding whether the agenda is safe, even when (the base of) Φ contains only formulas in 2CNF $\cap$ Horn. Intuitively, these restrictions on the form and size of the formulas in the agenda do not rule out the complex interactions between the formulas in the agenda that involve many formulas simultaneously, and that give rise to the Π_2^{P}-hardness of the problem.

Proposition 4. Agenda-Safety$^{\mathrm{maj}}$(formula size) *is* para-Π_2^{P}*-complete.*

Proof. Membership in para-Π_2^{P} follows from the Π_2^{P}-membership of Agenda-Safety$^{\mathrm{maj}}$. We show para-Π_2^{P}-hardness by giving a polynomial-time reduction from $\forall\exists$-Sat(3CNF) to the problem $\{ x : (x, c) \in$ Agenda-Safety$^{\mathrm{maj}}$(formula size) $\}$, where c is bounded by the size of formulas of the form $\neg((\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge \neg z)$. This reduction is a modified variant of a reduction given by Endriss et al. [18, Lemma 11]. Let $\varphi = \forall X.\exists Y.\psi$ be an instance of $\forall\exists$-Sat, where $\psi = c_1 \wedge \dots \wedge c_m$ is in 3CNF, and where $X = \{x_1, \dots, x_m\}$. We may assume without loss of generality that none of the c_i is a unit clause. We construct the agenda $\Phi = \{x_1, \neg x_1, \dots, x_n, \neg x_n, (c_1 \wedge \neg z_1), \neg(c_1 \wedge \neg z_1), \dots, (c_m \wedge \neg z_m), \neg(c_m \wedge \neg z_m)\}$, where $Z = \{z_1, \dots, z_m\}$ is a set of fresh variables. We claim that Φ satisfies the median property if and only if φ is true. A proof of this claim can be found in the appendix. $\square$

Next, using the following technical lemma, a proof of which can be found in the appendix, and the reduction given in the proof of Proposition 4, we get para-Π_2^{P}-completeness of Agenda-Safety$^{\mathrm{maj}}$(degree + formula size).

Lemma 5. *The problem* $\forall\exists$-Sat(3CNF) *is* Π_2^{P}*-hard even when restricted to instances* $\varphi = \forall X.\exists Y.\psi$ *where each* $x \in X$ *occurs at most* 2 *times in* ψ *and each* $y \in Y$ *occurs at most* 3 *times in* ψ*.*

Proposition 6. *The parameterized problems* Agenda-Safety$^{\mathrm{maj}}$(degree + formula size) *and* Agenda-Safety$^{\mathrm{maj}}$(degree) *are* para-Π_2^{P}*-complete.*

We now show hardness even for the case where all formulas are in Horn $\cap$ 2CNF.

Proposition 7. Agenda-Safety$^{\mathrm{maj}}$(degree + formula size) *is* para-Π_2^{P}*-hard even when restricted to agendas* Φ *such that all formulas* $\varphi \in B(\Phi)$ *are in* Horn $\cap$ 2CNF.

Proof. We consider the reduction used to show Proposition 6, which is described in detail in the proof of Proposition 4. The agenda Φ that we constructed contains only formulas of the form x_i or their negation, and formulas of the form $(c_i \wedge \neg z_i)$, where c_i is a clause, or their negation. Clearly, the formulas x_i and $\neg x_i$ are (equivalent to formulas) in Horn $\cap$ 2CNF. It suffices to show that each formula $\varphi \in \Phi$ with $\varphi = (c_i \wedge \neg z_i)$ is equivalent to a formula $\varphi' \in$ Horn $\cap$ 2CNF. Let $c_i = (l_1^i \vee l_2^i \vee l_3^i)$. Observe that $(c_i \wedge \neg z_i) = ((l_1^i \vee l_2^i \vee l_3^i) \wedge \neg z_i) \equiv (l_1^i \vee \neg z_i) \wedge (l_2^i \vee \neg z_i) \wedge (l_3^i \vee \neg z_i)$. Thus, we can construct Φ in such a way that $B(\Phi)$ contains only formulas in Horn $\cap$ 2CNF. $\square$

Agendas with few formulas. Next, we parameterize the agenda safety problem on the number of formulas occurring in the agenda. We will show that instances (x, k) of the problem $\text{AGENDA-SAFETY}^{\text{maj}}$(agenda size) can be solved by an fpt-algorithm that uses $f(k)$ many SAT calls. Intuitively, the fpt-algorithm that we construct will exploit the fact that the agenda only contains few formulas, by considering all possible inconsistent subsets of the agenda, and using a SAT solver to verify that these all have an inconsistent subset of size at most 2. In particular, we will prove the following result.

Theorem 8. *There exists an algorithm that decides* $\text{AGENDA-SAFETY}^{\text{maj}}$(agenda size) *in fpt-time using at most* $2^{O(k)}$ *SAT calls, where* k *is the parameter value.*

Moreover, we give evidence that this is the best one can do, i.e., there exists no fpt-algorithm that uses a significantly smaller number of SAT calls, assuming some widely believed complexity-theoretic assumptions (Theorem 19). We will need some formal machinery to prove the latter result.

In order to perform our lower-bound analysis, we will consider two parameterized complexity classes: $\text{FPT}^{\text{NP}[f(k)]}$ and BH(level). We defined the class $\text{FPT}^{\text{NP}[f(k)]}$ above. We note that it is straightforward to verify that $\text{FPT}^{\text{NP}[f(k)]}$ is closed under fpt-reductions. Next, to define the class BH(level), we consider the following parameterized decision problem, that is based on the canonical problems BH_i-SAT of the classes BH_i in the Boolean Hierarchy.

BH(level)-SAT
Instance: a positive integer k and a sequence $(\varphi_1, \dots, \varphi_k)$ of propositional formulas.
Parameter: k.
Question: is it the case that $(\varphi_1, \dots, \varphi_k) \in \text{BH}_k$-SAT?

We then define the parameterized complexity class BH(level) to be the class of all parameterized problems that can be fpt-reduced to the problem BH(level)-SAT. In other words, the class BH(level) consists of all parameterized problems P for which there exists an fpt-reduction that reduces each instance (x, k) of P to an instance of some problem in the $f(k)$-th level of the Boolean Hierarchy, for some computable function f. As we will see below, the classes $\text{FPT}^{\text{NP}[f(k)]}$ and BH(level) coincide. Moreover, we will show that $\text{AGENDA-SAFETY}^{\text{maj}}$(agenda size) is complete for this class. We begin with showing the upper bound on the number of SAT calls needed to solve $\text{AGENDA-SAFETY}^{\text{maj}}$(agenda size).

Proposition 9. $\text{AGENDA-SAFETY}^{\text{maj}}$(agenda size) *is in* co-BH(level).

Proof. We provide an fpt-algorithm that takes an instance Φ of $\text{AGENDA-SAFETY}^{\text{maj}}$(agenda size) with $|\Phi| = k$ and produces $f(k)$ many instances $x_1, \dots, x_{f(k)}$ of co-SAT-UNSAT such that $\Phi \in \text{AGENDA-SAFETY}^{\text{maj}}$(agenda size) if and only if $\{x_1, \dots, x_{f(k)}\} \subseteq$ co-SAT-UNSAT.

Let Φ be an agenda with $B(\Phi) = \{\varphi_1, \dots, \varphi_k\}$. Let C denote the set of all complement-free subagendas $\Phi' \subseteq \Phi$ that are of size at least 3. Clearly, $|C| = 2^{O(k)}$. We know that Φ satisfies the MP if and only if for all $\Phi' \in C$ holds that either (1) Φ' is satisfiable, or (2) there exists some $\Phi'' \subseteq \Phi'$ of size 2 that is unsatisfiable.

Firstly, for each $\Phi' = \{\psi_1, \dots, \psi_\ell\} \in C$, we construct an instance $I(\Phi') = (\psi_1, \psi_2)$ of co-SAT-UNSAT such that $(\psi_1, \psi_2) \in$ co-SAT-UNSAT if and only if either (1) Φ' is satisfiable or (2) there exists some $\Phi'' \subseteq \Phi'$ of size 2 that is unsatisfiable. For any $1 \leq i < j \leq \ell$ and any propositional formula φ, we let $\varphi^{(i,j)}$ denote a copy of φ where each variable $x \in \text{Var}(\varphi)$ is replaced with a copy $x^{(i,j)}$ indexed by the pair (i, j). We define $\psi_1 = \bigwedge_{\varphi \in \Phi'} \varphi$, and $\psi_2 = \bigwedge_{1 \leq i < j \leq \ell} (\psi_i^{(i,j)} \wedge \psi_j^{(i,j)})$. It is straightforward to verify that $I(\Phi')$ satisfies the required properties.

We now straightforwardly get that $\Phi \in \text{AGENDA-SAFETY}^{\text{maj}}$(agenda size) if and only if $\{\, I(\Phi') : \Phi' \in C \,\} \subseteq$ co-SAT-UNSAT. Also, we know that $|C| = f(k) = 2^{O(k)}$ for a suitable computable function f. We know that the conjunction of $f(k)$ many instances of co-SAT-UNSAT can be reduced in polynomial time to an instance of co-$\text{BH}_{2f(k)}$-SAT [11]. □

The following lemma allows us to use this membership result to obtain the upper bound we are after.

Lemma 10. *Let P be a parameterized problem that is contained in* BH(level)*. Then there exists an algorithm A that decides P in fpt-time using at most $f(k)$ many SAT calls, where k is the parameter value and f is some computable function.*

Proof. We construct an algorithm that decides whether $(x, k) \in P$. Since $P \in$ BH(level), we know that there exists an fpt-reduction R that reduces any instance (x, k) of P to an instance $R(x, k) = (x', k')$ of BH(level)-SAT. We know that $x' = (\varphi_1, \dots, \varphi_{k'})$, and that $k' \leq g(k)$ for some computable function g. The algorithm, given an instance (x, k), firstly computes (x', k'). Then, for each $1 \leq i \leq k'$, it decides whether φ_i is satisfiable by a single SAT call. Since (x', k') corresponds to a Boolean combination of statements concerning the satisfiability of the formulas φ_i, the algorithm can then decide in fpt-time whether $(x', k') \in$ BH(level)-SAT. □

Proof of Theorem 8. The result directly follows from the proofs of Proposition 9 and Lemma 10. Moreover, the obtained algorithm decides AGENDA-SAFETY$^{\text{maj}}$(agenda size) in time $O(n \cdot 2^k)$ by making $O(2^k)$ many queries to a SAT solver consisting of formulas of size $O(n \cdot k^2)$, where n is the input size and k is the parameter value. □

Next, we will pursue the lower bound. We start with identifying an easier hardness result, which we will then extend to a hardness result for the class co-BH(level).

Lemma 11. AGENDA-SAFETY$^{\text{maj}}$(agenda size) *is* para-co-DP*-hard.*

Proof. We prove hardness for para-co-DP by giving a polynomial-time reduction from SAT-UNSAT to co-AGENDA-SAFETY$^{\text{maj}}$, such that the resulting instance is an agenda of constant size. Let (φ_1, φ_2) be an instance of SAT-UNSAT. We construct the agenda Φ with $B(\Phi) = \{\psi_1, \psi_2, \psi_3\}$ by letting $\psi_1 = r_1 \wedge p_1 \wedge \varphi_1$, $\psi_2 = r_2 \wedge p_2$, and $\psi_3 = r_3 \wedge ((p_1 \wedge p_2) \rightarrow \varphi_2)$, where $\{r_1, r_2, r_3, p_1, p_2\}$ are distinct fresh variables not occurring in φ_1 nor in φ_2. We claim that Φ does not satisfy the MP if and only if $(\varphi_1, \varphi_2) \in$ SAT-UNSAT. A proof of this claim can be found in the appendix. □

Proposition 12. AGENDA-SAFETY$^{\text{maj}}$(agenda size) *is* co-BH(level)*-hard.*

Proof. We give an fpt-reduction from BH(level)-SAT to co-AGENDA-SAFETY$^{\text{maj}}$(agenda size). For the sake of simplicity, we assume that $k \geq 2$ is even. Let the sequence $(\varphi_1, \dots, \varphi_k)$ specify an instance of BH(level)-SAT. We know that we can construct in polynomial time an sequence of formulas $(\varphi_1, \psi_1, \dots, \varphi_\ell, \psi_\ell)$, where $\ell = k/2$, such that $(\varphi_1, \dots, \varphi_k) \in \text{BH}_k$-SAT if and only if for some $1 \leq i \leq \ell$ it holds that $(\chi_i, \psi_i) \in \text{BH}_2$-SAT = SAT-UNSAT [11].

Now, for each $1 \leq i \leq \ell$, we can use the reduction in the proof of Lemma 11 to construct in polynomial time an agenda Φ_i of constant size such that Φ_i does not satisfy the median property if and only if $(\chi_i, \psi_i) \in$ SAT-UNSAT. Moreover, we can ensure that the agendas Φ_i are variable-disjoint. We now construct the agenda $\Phi = \bigcup_{1 \leq i \leq \ell} \Phi_i$. We claim that Φ does not satisfy the median property if and only if $(\chi_i, \psi_i) \in$ SAT-UNSAT for some $1 \leq i \leq \ell$. We know this latter condition holds if and only if our original instance $(\varphi_1, \dots, \varphi_k) \in \text{BH}_k$-SAT. Moreover, since $|\Phi| = O(k)$, we obtain a correct fpt-reduction. A proof of this claim can be found in the appendix. □

Corollary 13. AGENDA-SAFETY$^{\text{maj}}$(agenda size) *is* co-BH(level)*-complete.*

Now that we have established that AGENDA-SAFETY$^{\text{maj}}$(agenda size) is hard for the class co-BH(level), we will investigate what this result tells us about the number of SAT calls needed by any fpt-algorithm that decides the problem AGENDA-SAFETY$^{\text{maj}}$(agenda size). For this, it will be convenient to show that BH(level) = co-BH(level). Consider the following lemma, which allows us to relate $\text{FPT}^{\text{NP}[f(k)]}$ to BH(level). A proof of the lemma can be found in the appendix.

Lemma 14. *Let P be a parameterized problem and let A be an algorithm that decides P in fpt-time using at most $g(k)$ many SAT calls, where k is the parameter value and g is some computable function. Then there exists an fpt-reduction that reduces an instance (x, k) of P to an instance (x', k') of* BH(level)-SAT*, where $k' \leq 2^{g(k)+1}$.*

Theorem 15. $\text{FPT}^{\text{NP}[f(k)]} = \text{BH(level)}$

Proof. Since $\text{FPT}^{\text{NP}[f(k)]}$ is closed under complement, the result follows directly from Lemmas 10 and 14. □

Moreover, this also allows us to relate BH(level) and co-BH(level).

Corollary 16. BH(level) = co-BH(level).

This now immediately gives us the following characterization of the complexity of AGENDA-SAFETY$^{\text{maj}}$(agenda size).

Corollary 17. AGENDA-SAFETY$^{\text{maj}}$(agenda size) *is* $\text{FPT}^{\text{NP}[f(k)]}$*-complete and* BH(level)*-complete.*

We will now use the BH(level)-hardness of AGENDA-SAFETY$^{\text{maj}}$(agenda size), to obtain lower bounds on the number of SAT calls needed by any fpt-algorithm to solve AGENDA-SAFETY$^{\text{maj}}$(agenda size).

Proposition 18. *Let P be any* BH(level)*-hard problem. Then P is not solvable by an fpt-algorithm that uses only $O(1)$ many SAT calls, unless the Polynomial Hierarchy collapses.*

Proof. Assume that P is solvable by an fpt-algorithm that uses only c many SAT calls, where c is a constant. We will show that the PH collapses. Since P is BH(level)-hard, we know that there exists an fpt-reduction R_1 from BH(level)-SAT to P. Then, by Lemma 14, there exists an fpt-reduction R_2 from P to BH(level)-SAT, that reduces any instance (x', k') of P to an instance (x'', k'') of BH(level)-SAT, where $k'' \leq 2^{c+1}$. Then, the composition R of R_1 and R_2 is an fpt-reduction from BH(level)-SAT to itself such that any instance (x, k) of BH(level)-SAT is reduced to an equivalent instance (x'', k'') of BH(level)-SAT, where $k'' \leq m = 2^{c+1}$. We can straightforwardly modify this reduction to always produce an instance (x'', m) of BH(level)-SAT, by adding trivial instances of SAT to the sequence x''.

We now show that the Boolean Hierarchy collapses to the m-th level, where $m = 2^{c+1}$. Let y be an instance of BH_{m+1}-SAT. We can then see the reduction R as a polynomial-time reduction from BH_{m+1}-SAT to BH_m-SAT: the fpt-reduction R runs in time $f(k) \cdot n^{O(1)}$, and since $k = m + 1$ is a constant, the factor $f(k)$ is constant. From this we can conclude that $\text{BH}_m = \text{BH}_{m+1}$. Thus, the BH collapses, and consequently the PH collapses [12, 26]. □

The above lower bound holds for any BH(level)-hard problem. We can improve this bound for the particular case of AGENDA-SAFETY$^{\text{maj}}$(agenda size).

Theorem 19. *Deciding whether $(x, k) \in$ AGENDA-SAFETY$^{\text{maj}}$(agenda size) is not solvable by an fpt-algorithm that uses $o(\log k)$ many SAT calls, unless the Polynomial Hierarchy collapses.*

Proof (idea). The proof is analogous to the proof of Proposition 18. Since we know in addition that there exists an fpt-reduction from AGENDA-SAFETY$^{\text{maj}}$(agenda size) to BH(level)-SAT that increases the parameter value (only) exponentially, the argument from the proof of Proposition 18 gives us a lower bound of $O(\log k)$ many SAT calls. A full proof can be found in the appendix. □

3.2 Restricting attention to small counterexamples

Another commonly identified "hidden" structure in problem instances is a restriction on the size of counterexamples. Many computational problems ask for the non-existence of a particular counterexample, and many of such problems show a decrease in complexity if attention can be restricted to counterexamples of a particular bounded size only.

One prominent example of a decrease in complexity induced by a restriction on the size of counterexamples is the method of Bounded Model Checking [6, 7]. In a nutshell, model checking is the problem of verifying whether a model of a system meets a given specification. This problem finds applications in a myriad of domains. A commonly used formalization is the problem of deciding whether a given transition systems satisfies a specification given in the form of a linear-time temporal logic (LTL) formula. This variant of the problem is PSPACE-complete (cf. [1, 13]). The problem is equivalent to deciding whether there exists no path (potentially of exponential length) in the transition system that serves as a counterexample to the specification. If the size of such counterexamples to consider is bounded (by an upper bound given in the input), the complexity of the problem decreases to NP [6, 7]. This result has been successfully applied in practice, by implementing algorithms that iteratively search for counterexamples of increasing size (cf. [6]). In the worst-case, an exponential number of iterations is needed, but in many instances occurring in practice, small counterexamples can be found efficiently this way.

A natural question to investigate is whether we could apply a similar approach to deciding whether an agenda is safe for the majority rule. In order to do so, we would like to get an improvement in the computational complexity for the case where the size of counterexamples is bounded. Therefore, we consider the following parameterized variant of the median property problem, where the parameter measures the size of subset of the agenda that we need to consider.

AGENDA-SAFETY$^{\text{maj}}$(counterexample size)
Instance: An agenda Φ, and an integer k.
Parameter: k.
Question: Does every inconsistent subset Φ' of Φ of size k have itself an inconsistent subset of size at most 2?

Assuming that counterexamples to the MP are small in practice corresponds to the supposition that whenever several statements together imply another statement, this latter statement is already implied by a small number of the former statements. In other words, the interaction between statements is, in a sense, local.

This problem is also related to agenda safety for supermajority rules. A supermajority rule accepts any proposition in the agenda if and only if a certain supermajority of the individuals, specified by a threshold $q \in (\frac{1}{2}, 1]$, accepts the proposition. Supermajority rules always produce consistent outcomes if the threshold is greater than $\frac{k-1}{k}$, where k is the size of the largest minimally inconsistent subset of the agenda (cf. [14, 27]).

Unfortunately, it turns out that this parameterization does not lead to a significant (practically exploitable) improvement in the computational complexity. In order to prove this, we will need the following lemma, a proof of which can be found in the appendix.

Lemma 20. *Let (φ, k) be an instance of* $\forall^k\exists^*$-WSAT*. In polynomial time, we can construct an instance (φ', k) of* $\forall^k\exists^*$-WSAT *with $\varphi' = \forall X.\exists Y.\psi$, such that: (1) $(\varphi, k) \in \forall^k\exists^*$-WSAT if and only if $(\varphi', k) \in \forall^k\exists^*$-WSAT; (2) for every assignment $\alpha : X \to \{0, 1\}$ of weight $m > k$, the formula $\exists Y.\psi[\alpha]$ is false; and (3) for every assignment $\alpha : X \to \{0, 1\}$ of weigth $m < k$, the formula $\exists Y.\psi[\alpha]$ is true.*

Theorem 21. AGENDA-SAFETY$^{\text{maj}}$(counterexample size) *is $\forall^k\exists^*$-hard.*

Proof. In order to show $\forall^k\exists^*$-hardness we provide an fpt-reduction from $\forall^k\exists^*$-WSAT to AGENDA-SAFETY$^{\text{maj}}$(counterexample size). Let (φ, k) be an instance of $\forall^k\exists^*$-WSAT, where $\varphi = \forall X.\exists Y.\psi$ is a quantified Boolean formula, $X = \{x_1, \dots, x_n\}$, and k is a positive integer. We may assume without loss of generality that φ satisfies properties (2) and (3) described in Lemma 20. We define the agenda $\Phi = \{x_1, \neg x_1, \dots, x_n, \neg x_n, (\psi \land z), \neg(\psi \land z)\}$, where z is a fresh variable. We claim that for all assignments $\alpha : X \to \{0, 1\}$ of weight k it is the case that $\exists Y.\psi[\alpha]$ is true if and only if every inconsistent subset Φ' of Φ of size $k + 1$ has itself an inconsistent subset of size 2. A proof of this claim can be found in the appendix. ☐

Intuitively, restricting attention to only possible counterexamples of size k, still leaves a search space of $O(n^k)$ many possible counterexamples (where n is the input size). Moreover, since there is no restriction on the agenda, searching this space for a counterexample (or verifying that no such counterexample exists) is computationally hard.

4 Conclusion

We provided a parameterized complexity analysis of the problem of agenda safety for the majority rule in judgment aggregation, with the aim of obtaining fpt-reductions to SAT. We identified several negative cases, and one positive case, where the safety of the agenda can be decided in fpt-time using a small number of SAT calls. Moreover, for this positive case, we identified lower bounds on the number of SAT calls needed to solve the problem in fpt-time.

We hope that the initial results obtained in this paper prove to be the kick-off of a structured parameterized complexity investigation of problems in the field of computational social choice that are located at higher levels of the PH. Concretely, for the problem studied in this paper, additional parameters related to treewidth and backdoors could be considered (these notions have been applied successfully in many parameterized complexity analyses). In addition, it would be interesting to study the problem of agenda safety for other judgment aggregation procedures [18].

References

[1] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[2] Dorothea Baumeister, Felix Brandt, Felix A. Fischer, Jan Hoffmann, and Jörg Rothe. The complexity of computing minimal unidirectional covering sets. *Theory Comput. Syst.*, 53(3):467–502, 2013.

[3] Dorothea Baumeister, Gábor Erdélyi, and Jörg Rothe. How hard is it to bribe the judges? A study of the complexity of bribery in judgment aggregation. In *Proceedings of the Second International Conference on Algorithmic Decision Theory (ADT 2011), Piscataway, NJ, USA, October 26-28, 2011.*, volume 6992 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2011.

[4] Nadja Betzler, Robert Bredereck, Jiehua Chen, and Rolf Niedermeier. Studies in computational aspects of voting – a parameterized complexity perspective. In Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors, *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 318–363. Springer Verlag, 2012.

[5] Nadja Betzler, Jiong Guo, and Rolf Niedermeier. Parameterized computational complexity of Dodgson and Young elections. *Information and Computation*, 208(2):165–177, 2010.

[6] Armin Biere. Bounded model checking. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 457–481. IOS Press, 2009.

[7] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, and Yunshan Zhu. Symbolic model checking without BDDs. In Rance Cleaveland, editor, *Tools and Algorithms for Construction and Analysis of Systems, 5th International Conference, TACAS '99, Held as Part of the European Joint Conferences on the Theory and Practice of Software, ETAPS'99, Amsterdam, The Netherlands, March 22-28, 1999, Proceedings*, volume 1579 of *Lecture Notes in Computer Science*, pages 193–207. Springer Verlag, 1999.

[8] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[9] Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors. *The Multivariate Algorithmic Revolution and Beyond – Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*. Springer, 2012.

[10] Sylvain Bouveret and Jérôme Lang. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *J. Artif. Intell. Res.*, 32:525–564, 2008.

[11] Jin-yi Cai, Thomas Gundermann, Juris Hartmanis, Lane A. Hemachandra, Vivian Sewelson, Klaus W. Wagner, and Gerd Wechsung. The Boolean hierarchy I: Structural properties. *SIAM J. Comput.*, 17(6):1232–1252, 1988.

[12] Richard Chang and Jim Kadin. The Boolean hierarchy and the polynomial hierarchy: a closer connection. *SIAM J. Comput.*, 25:169–178, 1993.

[13] Edmund M. Clarke, Daniel Kroening, Joël Ouaknine, and Ofer Strichman. Completeness and complexity of bounded model checking. In Bernhard Steffen and Giorgio Levi, editors, *Verification, Model Checking, and Abstract Interpretation, 5th International Conference, VMCAI 2004*, volume 2937 of *Lecture Notes in Computer Science*, pages 85–96. Springer, 2004.

[14] Franz Dietrich and Christian List. Judgment aggregation by quota rules: Majority voting generalized. *J. of Theoretical Politics*, 19(4):391–424, 2007.

[15] Britta Dorn and Ildikó Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 64(1):126–151, 2012.

[16] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer Verlag, New York, 1999.

[17] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer Verlag, 2013.

[18] Ulle Endriss, Umberto Grandi, and Daniele Porello. Complexity of judgment aggregation. *J. Artif. Intell. Res.*, 45:481–514, 2012.

[19] Jörg Flum and Martin Grohe. Describing parameterized complexity classes. *Information and Computation*, 187(2):291–319, 2003.

[20] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series.* Springer Verlag, Berlin, 2006.

[21] Carla P. Gomes, Henry Kautz, Ashish Sabharwal, and Bart Selman. Satisfiability solvers. In *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 89–134. Elsevier, 2008.

[22] Ronald de Haan and Stefan Szeider. Fixed-parameter tractable reductions to SAT. In Uwe Egly and Carsten Sinz, editors, *Proceedings of the 17th International Symposium on the Theory and Applications of Satisfiability Testing (SAT 2014) Vienna, Austria, July 14–17, 2014*, Lecture Notes in Computer Science. Springer, 2014. To appear.

[23] Ronald de Haan and Stefan Szeider. The parameterized complexity of reasoning problems beyond NP. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014. To appear.

[24] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *J. of the ACM*, 44(6):806–825, 1997.

[25] Edith Hemaspaandra, Holger Spakowski, and Jörg Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

[26] Jim Kadin. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM J. Comput.*, 17(6):1263–1282, December 1988.

[27] Christian List. The theory of judgment aggregation: an introductory review. *Synthese*, 187(1):179–207, 2012.

[28] Christian List and Clemens Puppe. Judgment aggregation: A survey. In *Handbook of Rational and Social Choice*. Oxford University Press, 2009.

[29] Sharad Malik and Lintao Zhang. Boolean satisfiability from theoretical hardness to practical success. *Communications of the ACM*, 52(8):76–82, 2009.

[30] Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *SWAT*, pages 125–129. IEEE Computer Soc., 1972.

[31] Klaus Nehring and Clemens Puppe. The structure of strategy-proof social choice - part I: General characterization and possibility results on median spaces. *J. of Economic Theory*, 135(1):269–305, 2007.

[32] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, 2006.

[33] Christos H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[34] Karem A. Sakallah and João Marques-Silva. Anatomy and empirical evaluation of modern SAT solvers. *Bulletin of the European Association for Theoretical Computer Science*, 103:96–121, 2011.

[35] Jörg Siekmann and Graham Wrightson, editors. *Automation of reasoning. Classical Papers on Computer Science 1967–1970*, volume 2. Springer Verlag, 1983.

[36] Marija Slavkovik. *Judgment Aggregation for Multiagent Systems.* PhD thesis, University of Luxembourg, 2012.

[37] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[38] G. S. Tseitin. Complexity of a derivation in the propositional calculus. *Zap. Nauchn. Sem. Leningrad Otd. Mat. Inst. Akad. Nauk SSSR*, 8:23–41, 1968. English translation reprinted in [35].

[39] Celia Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976.

Ulle Endriss
Institute for Logic, Language and Computation
University of Amsterdam
Amsterdam, The Netherlands
Email: ulle.endriss@uva.nl

Ronald de Haan
Institute of Information Systems
Vienna University of Technology
Vienna, Austria
Email: dehaan@kr.tuwien.ac.at

Stefan Szeider
Institute of Information Systems
Vienna University of Technology
Vienna, Austria
Email: stefan@szeider.net

Appendix: Proofs

Proof of Proposition 1. Assume without loss of generality that φ contains only the connectives $\wedge$ and $\neg$. Let $\mathrm{Sub}(\varphi)$ denote the set of all subformulas of φ. We let $\mathrm{Var}(\varphi') = \mathrm{Var}(\varphi) \cup \{\, z_\chi : \chi \in \mathrm{Sub}(\varphi) \,\}$, where each z_χ is a fresh variable. We then define φ' to be the formula $\chi_\varphi \wedge \bigwedge_{\chi \in \mathrm{Sub}(\varphi)} \sigma(\chi)$, where we define the formulas $\sigma(\chi)$, for each $\chi \in \mathrm{Sub}(\varphi)$ as follows:

$$\sigma(\chi) = \begin{cases} (z_l \rightarrow l) \wedge (l \rightarrow z_l) & \text{if } \chi = l \text{ is a literal,} \\ (z_\chi \rightarrow \neg z_{\chi'}) \wedge (z_{\chi'} \rightarrow \neg z_\chi) & \text{if } \chi = \neg \chi' \text{, and} \\ (z_\chi \rightarrow z_{\chi_1}) \wedge (z_\chi \rightarrow z_{\chi_2}) \wedge (\neg z_{\chi_1} \vee \neg z_{\chi_2} \rightarrow \neg z_\chi) & \text{if } \chi = \chi_1 \wedge \chi_2. \end{cases}$$

Let $\alpha : \mathrm{Var}(\varphi) \rightarrow \{0,1\}$ be an arbitrary truth assignment. We claim that α satisfies φ if and only if there exists an assignment $\beta : (\mathrm{Var}(\varphi') \backslash \mathrm{Var}(\varphi)) \rightarrow \{0,1\}$ such that $\alpha \cup \beta$ satisfies φ'. Define the assignment β' as follows. For each $\chi \in \mathrm{Sub}(\varphi)$, we let $\beta(z_\chi) = 1$ if and only if α satisfies χ. Clearly, if α satisfies φ, then $\alpha \cup \beta'$ satisfies φ'. Conversely, for any assignment $\beta : (\mathrm{Var}(\varphi') \backslash \mathrm{Var}(\varphi)) \rightarrow \{0,1\}$ that does not coincide with β', clearly, the assignment $\alpha \cup \beta$ does not satisfy some clause of φ'. Moreover, if $\alpha \cup \beta'$ satisfies φ', then α satisfies φ. ☐

Proof of Proposition 2 (continued). We prove that Ψ is consistent if and only if $\Psi' = \{\varphi'_{i_1}, \dots, \varphi'_{i_{m_1}}, \neg\varphi'_{j_1}, \dots, \neg\varphi'_{j_{m_2}}\}$ is consistent.

($\Rightarrow$) Let $\alpha : \mathrm{Var}(\Psi) \rightarrow \{0,1\}$ be an assignment that satisfies all formulas in Ψ. By construction of the formulas φ'_i, by Lemma 1, and by the fact that for each $1 \leq i < i' \leq n$ it is the case that $(\mathrm{Var}(\varphi'_i) \backslash \mathrm{Var}(\varphi_i)) \cap (\mathrm{Var}(\varphi'_{i'}) \backslash \mathrm{Var}(\varphi_{i'})) = \emptyset$, we know that there exists an assignment $\beta : (\mathrm{Var}(\Psi') \backslash \mathrm{Var}(\Psi)) \rightarrow \{0,1\}$ such that $\alpha \cup \beta$ satisfies all formulas in Ψ.

($\Leftarrow$) Conversely, assume that there exists an assignment $\alpha : \mathrm{Var}(\Psi') \rightarrow \{0,1\}$ that satisfies all formulas in Ψ'. Then, by construction of the formulas φ'_i, we know that $\mathrm{Var}(\Psi') \subseteq \mathrm{Var}(\Psi)$. Now, by Lemma 1, we know that α satisfies all formulas in Ψ as well. ☐

Proof of Proposition 4 (continued). We prove that Φ satisfies the median property if and only if φ is true.

($\Rightarrow$) Suppose that φ is false, i.e., there exists some $\alpha : X \rightarrow \{0,1\}$ such that $\forall Y.\neg\psi[\alpha]$ is true. Let $L = \{\, x_i : 1 \leq i \leq n, \alpha(x_i) = 1 \,\} \cup \{\, \neg x_i : 1 \leq i \leq n, \alpha(x_i) = 0 \,\}$. We know that α is the unique assignment to the variables in X that satisfies L. Now consider $\Phi' = L \cup \{(c_1 \wedge z_1), \dots, (c_m \wedge z_m)\}$.

We firstly show that Φ' is inconsistent. We proceed indirectly and assume that Φ' is consistent, i.e., there exists an assignment $\beta : Y \cup Z \rightarrow \{0,1\}$ such that $\alpha \cup \beta$ satisfies Φ'. Then $\alpha \cup \beta$ must satisfy each c_i. Therefore, β satisfies $\psi[\alpha]$, which contradicts our assumption that $\forall Y.\neg\psi[\alpha]$ is true. Therefore, we can conclude that Φ' is inconsistent.

Next, we show that each subset $\Phi'' \subseteq \Phi'$ of size 2 is consistent. Let $\Phi'' \subseteq \Phi'$ be an arbitrary subset of size 2. We distinguish three cases: either (i) $\Phi'' = \{l_i, l_j\}$ for some $1 \leq i < j \leq n$; (ii) $\Phi'' = \{l_i, (c_j \wedge \neg z_j)\}$ for some $1 \leq i \leq n$ and some $1 \leq j \leq m$; or (iii) $\Phi'' = \{(c_i \wedge \neg z_i), (c_j \wedge \neg z_j)\}$ for some $1 \leq i < j \leq m$. In case (i), clearly Φ'' is consistent. In case (ii) and (iii), Φ'' is consistent because c_i and c_j are not unit clauses.

($\Leftarrow$) Conversely, suppose that Φ does not satisfy the median property, i.e., there exists an inconsistent subset $\Phi' \subseteq \Phi$ that itself does not contain an inconsistent subset of size 2. We show that φ is false. Firstly, we show that $\Psi' = \Phi' \backslash \{\neg(c_1 \wedge \neg z_1), \dots, \neg(c_m \wedge \neg z_m)\}$ is inconsistent. We proceed indirectly, and assume that Ψ' is consistent, i.e., there exists an assignment $\gamma : \mathrm{Var}(\Psi') \rightarrow \{0,1\}$ such that γ satisfies Ψ'. Now let $Z' = \{\, z_i : 1 \leq i \leq m, \neg(c_i \wedge \neg z_i) \in \Psi' \,\}$ and let $\gamma' : Z' \rightarrow \{0,1\}$ be defined by letting $\gamma'(z) = 0$ for all $z \in Z'$. Since Ψ' contains no negated pairs of formulas, we know that $Z' \cap \mathrm{Var}(\Psi') = \emptyset$. Then the assignment $\gamma \cup \gamma'$ satisfies Φ', since γ satisfies all $\psi \in \Psi'$ and γ' satisfies all $\varphi \in \Phi' \cap \Psi'$. This is a contradiction with our assumption that Φ' is inconsistent, so we can conclude that Ψ' is inconsistent.

Now let the assignment $\alpha : X \to \{0,1\}$ be defined as follows. For each $x \in X$, we let $\alpha(x) = 1$ if $x \in \Psi'$, we let $\alpha(x) = 0$ if $\neg x \in \Psi'$, and we (arbitrarily) define $\alpha(x) = 1$ otherwise. We now show that $\neg\exists Y.\psi[\alpha]$ is true. We proceed indirectly, and assume that there exists an assignment $\beta : Y \to \{0,1\}$ such that $\psi[\alpha \cup \beta]$ is true. Now consider the assignment $\gamma : Z \to \{0,1\}$ such that $\gamma(z) = 0$ for all $z \in Z$. We claim that the assignment $\alpha \cup \beta \cup \gamma$ satisfies Ψ'. Let $\chi \in \Psi'$ be an arbitrary formula. We distinguish two cases: either (i) $\chi \in \{x_i, \neg x_i\}$ for some $1 \leq i \leq n$; or (ii) $\chi = (c_i \wedge \neg z_i)$ for some $1 \leq i \leq m$. In case (i), we know that α satisfies χ. In case (ii), we know that $\alpha \cup \beta$ satisfies c_i, since $\alpha \cup \beta$ satisfies ψ, and since γ satisfies z_i. This is a contradiction with our previous conclusion that Ψ' is inconsistent, so we can conclude that $\neg\exists Y.\psi[\alpha]$ is true. From this, we know that $\forall X.\exists Y.\psi$ is false. □

Proof of Lemma 5. Let $\varphi = \forall X.\exists Y.\psi$ be an instance of $\forall\exists$-SAT(3CNF). We construct in polynomial time an equivalent instance $\varphi' = \forall X'.\exists Y'.\psi'$ of $\forall\exists$-SAT(3CNF) such that each $x \in X'$ occurs at most 2 times in ψ' and each $y \in Y'$ occurs at most 3 times in ψ'.

Firstly, we construct an equivalent formula $\varphi_1 = \forall X.\exists Y_1.\psi_1$ such that each $x \in X_1$ occurs at most 2 times in ψ_1. We do this by repeatedly applying the following transformation. Let $z \in X$ be any variable that occurs $m > 3$ times in ψ. We create m many copies $z_1, \dots, z_m$ of z, that we add to the set Y of existentially quantified variables. We replace each occurrence of z in ψ by a distinct copy z_i. Finally, we ensure equivalence of ψ_1 and ψ by letting $\psi_1 = \psi \wedge \psi^z_{\text{equiv}}$, where we define ψ^z_{equiv} to be the conjunction of binary clauses $(z_i \to z_{i+1})$ for each $1 \leq i < m$, the binary clause $(z_m \to z_1)$, and the binary clauses $(z \to z_1)$ and $(z_1 \to z)$. Repeated application of this transformation results in a formula φ_1 that satisfies the required properties.

Then, we transform φ_1 into an equivalent formula $\varphi_2 = \forall X.\exists Y_2.\psi_2$ such that each $y \in Y_2$ occurs at most 3 times in ψ_2. Moreover, each $x \in X$ occurs as many times in ψ_2 as it did in ψ_1 (i.e., twice). We use a similar strategy as we did in the first phase: we repeatedly apply the following transformation. Let $y \in Y_1$ be any variable that occurs $m > 3$ times in ψ_1. We create m many copies $y_1, \dots, y_m$ of y, that we add to the set Y_1 of existentially quantified variables. Then we replace each occurrence of y in ψ by a distinct copy y_i. Finally, we ensure equivalence of ψ_2 and ψ_1 by letting $\psi_2 = \psi^y_{\text{equiv}} \wedge \psi_1$, where we define ψ^y_{equiv} to the conjunction of the binary clauses $(y_i \to y_{i+1})$ for all $1 \leq i < m$ and the binary clause $(y_m \to y_1)$. Again, repeated application of this transformation results in a formula φ_2 that satisfies the required properties. □

Proof of Lemma 11 (continued). We claim that Φ does not satisfy the MP if and only if $(\varphi_1, \varphi_2) \in$ SAT-UNSAT.

($\Rightarrow$) Assume that Φ does not satisfy the MP. Then there exists a satisfiable complement-free subagenda $\Phi' \subseteq \Phi$ such that each subset $\Phi'' \subseteq \Phi'$ of size 2 is satisfiable. We dinstinguish several cases: either (i) $\Phi' = B(\Phi) = \{\psi_1, \psi_2, \psi_3\}$, or (ii) the above case does not hold and Φ' contains ψ_1, or (iii) the above two cases do not hold.

We show that in case (i) we can conclude that $(\varphi_1, \varphi_2) \in$ SAT-UNSAT. By assumption, every subset $\Phi'' \subseteq \Phi$ of size 2 is satisfiable. Therefore, we can conclude that the formula ψ_1 is satisfiable. Hence, φ_1 is satisfiable. Next, we show that φ_2 is unsatisfiable. We proceed indirectly, and we assume that there exists some assignment $\alpha : \text{Var}(\varphi_2) \to \{0,1\}$ that satisfies φ_2. We construct a satisfying assignment $\alpha' : \text{Var}(\Phi) \to \{0,1\}$ for Φ, which leads to a contradiction. We let α' coincide with α on the variables in $\text{Var}(\varphi_2)$. Moreover, we know that there exists some satisfying assignment $\beta : \text{Var}(\varphi_1) \to \{0,1\}$ for φ_1. We let α' coincide with β on the variables in $\text{Var}(\varphi_1)$. Finally, we let $\alpha'(x) = 1$ for each $x \in \{r_1, r_2, r_3, p_1, p_2\}$.

Clearly, α' satisfies all formulas in Φ then. This leads to a contradiction with the fact that Φ is unsatisfiable, and therefore we can conclude that φ_2 is unsatisfiable.

Next, we show that case (ii) cannot occur. We know that $\psi_1 \in \Phi'$, and that each subset $\Phi'' \subseteq \Phi$ of size 2 is satisfiable. Therefore, we know that φ_1 is satisfiable. Let $\beta : \text{Var}(\varphi_1) \rightarrow \{0,1\}$ be a satisfying assignment for φ_1. We extend the assignment β to an assignment $\beta' : \text{Var}(\Phi) \rightarrow \{0,1\}$ that satisfies Φ'. We let $\beta'(r_1) = \beta'(p_1) = 1$. If $\psi_2 \in \Phi$, we let $\beta'(r_2) = \beta'(p_2) = 1$; otherwise, if $\neg\psi_2 \in \Phi$, we let $\beta'(r_2) = 0$. If $\psi_3 \in \Phi$, we let $\beta'(r_3) = 1$ and $\beta'(p_2) = 0$; otherwise, if $\neg\psi_3 \in \Phi$, we let $\beta'(r_3) = 0$. On the other variables, we let β' be defined arbitrarily. Since not both $\psi_2 \in \Phi$ and $\psi_3 \in \Phi$, we know that β' is well-defined. It is easy to verify that β' satisfies Φ', which is a contradiction with our assumption that Φ' is unsatisfiable. From this we can conclude that case (ii) cannot occur.

Finally, we show that case (iii) cannot occur either. We construct an assignment $\beta : \text{Var}(\Phi) \rightarrow \{0,1\}$ that satisfies Φ'. We know that $\neg\psi_1 \in \Phi'$. Let $\beta(r_1) = \beta(p_1) = 0$. If $\psi_2 \in \Phi'$, we let $\beta(r_2) = \beta(p_2) = 1$; otherwise, if $\neg\psi_2 \in \Phi'$, we let $\beta(r_2) = 0$; If $\psi_3 \in \Phi'$, we let $\beta(r_3) = 1$; otherwise, if $\neg\psi_3 \in \Phi'$, we let $\beta(r_3) = 0$. It is easy to verify that β satisfies Ψ, which is a contradiction with our assumption that Φ' is unsatisfiable. From this we can conclude that case (iii) cannot occur.

($\Leftarrow$) Conversely, assume that φ_1 is satisfiable and that φ_2 is unsatisfiable. Then consider the complement-free subagenda $\Phi' \subseteq \Phi$ given by $\Phi' = B(\Phi) = \{\psi_1, \psi_2, \psi_3\}$. Since $\psi_1, \psi_2 \models p_1 \wedge p_2$ and φ_2 is unsatisfiable, we get that Φ' is unsatisfiable. However, since φ_1 is satisfiable, we get that each subset of Φ' of size 2 is satisfiable. Therefore, Φ does not satisfy the MP. □

Proof of Proposition 12 (continued). We prove that Φ does not satisfy the median property if and only if $(\chi_i, \psi_i) \in$ SAT-UNSAT for some $1 \leq i \leq \ell$.

Assume that Φ does not satisfy the median property. Then there exists a subset $\Phi' \subseteq \Phi$ that is unsatisfiable such that each $\Phi'' \subseteq \Phi'$ of size 2 is satisfiable. Moreover, we can assume Φ' to be minimal with this property. Since Φ is partitioned into the variable disjoint subsets Φ_i, and since Φ' is minimal, we know that $\Phi' \subseteq \Phi_i$, for some $1 \leq i \leq \ell$. Then Φ_i does not satisfy the median property, from which we can conclude that $(\chi_i, \psi_i) \in$ SAT-UNSAT. Conversely, assume that $(\chi_i, \psi_i) \in$ SAT-UNSAT for some $1 \leq i \leq \ell$. Then by construction of Φ_i, we know that Φ_i does not satisfy the median property. Therefore, since $\Phi_i \subseteq \Phi$, we know that Φ does not satisfy the median property. □

Proof of Lemma 14. We use the algorithm A to construct an fpt-reduction from P to BH(level)-SAT. We will use the known fact that a disjunction of m many SAT-UNSAT instances can be reduced to a single instance of BH_{2m}-SAT [11]. Let (x, k) be an instance of P. We may assume without loss of generality that A makes exactly $g(k)$ many SAT calls on any input (x, k). Consider the set $B = \{0,1\}^{g(k)}$. We interpret each sequence $\overline{b} = (b_1, \dots, b_{g(k)}) \in B$ as a sequence of answers to the SAT calls made by A; a 0 corresponds to the answer of the SAT call being "unsatisfiable" and a 1 corresponds to the answer being "satisfiable." For each $\overline{b} \in B$, we simulate the algorithm A on input (x, k) by using the answer specified by b_i to the i-th SAT call. Let us write $A_{\overline{b}}(x, k)$ to denote the simulation of A on input (x, k) where the answers to the SAT calls are specified by $\overline{b}$. By performing this simulation for each $\overline{b} \in B$, we can determine in fpt-time the set $B' \subseteq B$ of sequences $\overline{b}$ such that $A_{\overline{b}}(x, k)$ accepts.

We know that A accepts (x, k) if and only if the "correct" sequence of answers is contained in B', in other words, A accepts (x, k) if and only if there exists some $\overline{b} \in B'$ such that for each b_i it holds that if $b_i = 0$ then ψ_i is unsatisfiable, and if $b_i = 1$ then ψ_i is satisfiable, where ψ_i denotes the formula used for the i-th SAT call made by $A_{\overline{b}}(x, k)$. For each $\overline{b} \in B'$, we construct an instance $I(\overline{b}) = (\varphi_1, \varphi_0)$ of SAT-UNSAT that is a yes-instance if and only if the above condition holds for sequence $\overline{b}$, as follows. Let $(\psi_1, \dots, \psi_{g(k)})$ be the propositional formulas that $A_{\overline{b}}(x, k)$ uses for the SAT calls, i.e., ψ_i corresponds to the formula used for the

i-th SAT call of $A_{\overline{b}}(x,k)$. We may assume without loss of generality that the formulas ψ_i are variable disjoint, i.e., for each $1 \leq i < i' \leq g(k)$, it holds that $\text{Var}(\psi_i) \cap \text{Var}(\psi_{i'}) = \emptyset$. We construct the instance (φ_1, φ_0) as follows:

$$\begin{aligned} C_1 = &\ \{ 1 \leq i \leq g(k) : b_i = 1 \}; \\ \varphi_1 = &\ \bigwedge_{j \in C_1} \psi_j; \\ C_0 = &\ \{ 1 \leq i \leq g(k) : b_i = 0 \}; \text{ and} \\ \varphi_0 = &\ \bigvee_{j \in C_0} \psi_j; \end{aligned}$$

It is straightforward to verify that $I(\overline{b}) \in$ SAT-UNSAT if and only if $\overline{b}$ corresponds to the "correct" sequence of answers for the SAT calls made by A, i.e., for each b_i with $b_i = 0$ it holds that ψ_i is unsatisfiable, and for each b_i with $b_i = 1$ it holds that ψ_i is satisfiable.

We constructed ℓ many instances $I(\overline{b_1}), \dots, I(\overline{b_\ell})$ of SAT-UNSAT, for some $\ell \leq 2^{g(k)}$, such that the algorithm A accepts the instance (x,k), and thus $(x,k) \in P$, if and only if there exists some $1 \leq i \leq \ell$ such that $I(\overline{b_\ell}) \in$ SAT-UNSAT. In other words, we reduced our original instance (x,k) of P to a disjunction of $\ell \leq 2^{g(k)}$ many instances of SAT-UNSAT. We know that such a disjunction can be reduced to an instance of $\text{BH}_{2\ell}$-SAT [11]. This completes our fpt-reduction from P to BH(level). ☐

Proof of Theorem 19. Assume that AGENDA-SAFETY$^{\text{maj}}$(agenda size) is solvable by an fpt-algorithm that uses $h(k) = o(\log k)$ many SAT calls. We show that the BH collapses, and thus that consequently, the PH collapses. By Proposition 12, we know that BH(level)-SAT can be fpt-reduced to the problem AGENDA-SAFETY$^{\text{maj}}$(agenda size) in such a way that the parameter value k increases at most linearly to $h'(k) = O(k)$. By Lemma 14, we know that AGENDA-SAFETY$^{\text{maj}}$(agenda size) can be fpt-reduced to BH(level)-SAT in such a way that the resulting parameter value k' is bounded by a function $h''(k) = 2^{O(k)}$, where k is the original parameter value. We can now combine these fpt-reductions to obtain a polynomial-time reduction that witnesses the collapse of the BH. We know that there exists some integer ℓ such that $h''(h'(h(\ell))) = \ell' < \ell$. Applying the composing the fpt-reductions gives us a polynomial-time reduction from the problem BH_ℓ-SAT to the problem $\text{BH}_{\ell'}$-SAT. Since $\ell' < \ell$, this shows that the BH collapses to the ℓ'-th level. Since a collapse of the BH implies a collapse of the PH [26, 12], the result follows. ☐

Proof of Lemma 20. Let (φ, k) be an instance of $\forall^k\exists^*$-WSAT, with $\varphi = \forall X.\exists Y.\psi$. We construct the instance $\varphi' = \forall X.\exists Y \cup Z.\psi'$ as follows. We define the set Z of variables by letting $Z = \{ z_{x,i} : x \in X, 1 \leq i \leq k \}$. Intuitively, these variables keep track of how many variables in X are set to true. We define the formula $\psi' = \psi^Z_{\text{proper}} \wedge (\psi^Z_{\text{few}} \vee \psi)$, where $\psi^Z_{\text{proper}} = \bigwedge_{x \in X} \bigvee_{1 \leq i \leq k} z_{x,i} \wedge \bigwedge_{1 \leq i \leq k} \bigwedge_{x,x' \in X, x \neq x'} (\neg z_{x,i} \vee \neg z_{x',i}) \wedge \bigwedge_{x \in X} \bigwedge_{1 \leq i < i' \leq k} (\neg z_{x,i} \vee \neg z_{x,i'})$, and $\psi^Z_{\text{few}} = \bigvee_{1 \leq i \leq k} \bigwedge_{x \in X} \neg z_{x,i}$. The formula ψ^Z_{proper} enforces that for any $x \in X$ that is set to true, there must be some $1 \leq i \leq k$ such that $z_{x,i}$ is set to true as well. Moreover, it enforces that for each $x \in X$ there is at most one $1 \leq i \leq k$ such that $z_{x,i}$ is true, and for each $1 \leq i \leq k$, there is at most one $x \in X$ such that $z_{x,i}$ is true. The formula ψ^Z_{few} is true if and only if there exists some $1 \leq i \leq k$ such that $z_{x,i}$ is false for all $x \in X$.

It is now straightforward to verify that for each assignment $\alpha : X \to \{0,1\}$ it holds that (i) if α has weight k, then $\exists Y \cup Z.\psi'[\alpha]$ is true if and only if $\exists Y.\psi[\alpha]$ is true, (ii) if α has weight less than k, then $\exists Y \cup Z.\psi'[\alpha]$ is always true, and (iii) if α has weight more than k, then $\exists Y \cup Z.\psi'[\alpha]$ is never true. ☐

Proof of Theorem 21 (continued). We show that for all assignments $\alpha : X \to \{0,1\}$ of weight k it is the case that $\exists Y.\psi[\alpha]$ is true if and only if every inconsistent subset Φ' of Φ of size $k+1$ has itself an inconsistent subset of size 2.

(⇒) Assume that there exists an inconsistent subset Φ' of Φ of size $k+1$ that has itself no inconsistent subset of size 2. It is straightforward to see that for no $\varphi \in \Phi$, Φ' contains both φ and $\sim\varphi$. If Φ' does not contain $(\psi \wedge z)$, we can easily satisfy Φ' by setting z to false and satisfying all literals in Φ'. Therefore, $(\psi \wedge z) \in \Phi'$. We show that Φ' contains exactly k positive literals x_j for some $1 \leq j \leq m$. We proceed indirectly, and assume the contrary, i.e., that Φ' contains at most $k-1$ many positive literals x_j for some $1 \leq j \leq m$. Let $L = \Phi' \cap X$. Consider the assignment $\alpha : X \rightarrow \{0,1\}$ such that $\alpha(x) = 1$ if and only if $x \in \Phi$. Clearly, α has weight strictly less than k. Therefore, we know that there exists an assignment $\beta : Y \rightarrow \{0,1\}$ such that $\alpha \cup \beta$ satisfies ψ. Additionally, consider the assignment $\gamma : \{z\} \rightarrow \{0,1\}$ such that $\gamma(z) = 1$. Then $\alpha \cup \beta \cup \gamma$ satisfies Φ', which contradicts our assumption that Φ' is inconsistent. From this we can conclude that $|\Phi' \cap X| = k$.

Now, again consider the assignment $\alpha : X \rightarrow \{0,1\}$ such that $\alpha(x) = 1$ if and only if $x \in \Phi$. Clearly, α has weight k. We show that the formula $\exists Y.\psi[\alpha]$ is false. We proceed indirectly, and assume that there exists an assignment $\beta : Y \rightarrow \{0,1\}$ such that $\alpha \cup \beta$ satisfies ψ. Consider the assignment $\gamma : \{z\} \rightarrow \{0,1\}$ such that $\gamma(z) = 1$. It is straightforward to verify that $\alpha \cup \beta \cup \gamma$ satisfies Φ', which contradicts our assumption that Φ' is inconsistent. Therefore, we conclude that $\exists Y.\psi[\alpha]$ is false, and thus that it is not the case that for all assignments $\alpha : X \rightarrow \{0,1\}$ of weight k it is the case that $\exists Y.\psi[\alpha]$ is true.

(⇐) Assume that there exists an assignment $\alpha : X \rightarrow \{0,1\}$ of weight k such that $\neg\exists Y.\psi[\alpha]$ is true. Let $L = \{\, x_i : 1 \leq i \leq n, \alpha(x_i) = 1 \,\}$. Consider the subagenda $\Phi' = L \cup \{(\psi \wedge z)\}$. We show that Φ' is inconsistent. We proceed indirectly, and assume that there exists an assignment $\beta : X \cup Y \cup \{z\} \rightarrow \{0,1\}$ that satisfies Φ'. Clearly, $\beta(x_i) = 1$ for all $x_i \in L$. We show that $\beta(x) = 0$ for all $x \in X \backslash L$. We proceed indirectly, and assume the contrary, i.e., $\beta(x) = 1$ for some $x \in X \backslash L$. Then the restriction of β to the variables in X has weight $m > k$. Therefore, since for all assignments $\beta' : X \rightarrow \{0,1\,\}$ of weight strictly larger than k the formula $\exists Y.\psi[\beta']$ is false, we know that β does not satisfy ψ. From this we can conclude that $\beta(x) = 0$ for all $x \in X \backslash L$. We then know that the restriction $\beta|_X$ of β to the variables in X has weigth k. Also, since $(\psi \wedge z) \in \Phi$, we know that β satisfies ψ. This is a contradiction with our assumption that $\neg\exists Y.\psi[\beta|_X]$ is true. Therefore, we know that β cannot exist, and thus that Φ' is inconsistent.

We now show that each subset Φ'' of Φ' of size 2 is consistent. Let $\Phi'' \subseteq \Phi'$ be an arbitrary subset of size 2. We distinguish two cases: either (i) $\Phi'' = \{x_i, x_j\}$ for some $1 \leq i < j \leq n$, or (ii) $\Phi'' = \{x_i, (\psi \wedge z)\}$ for some $1 \leq i \leq n$. In case (i), clearly Φ'' is consistent. In case (ii), we get that Φ'' is consistent by the fact that for every assignment $\alpha : X \rightarrow \{0,1\}$ of weight $m < k$ the formula $\exists Y.\psi[\alpha]$ is true. This completes our proof that Φ' does not satisfy the median property. □