Dov M. Gabbay
Andrzej Szałas

Voting by Eliminating Quantifiers

Abstract. Mathematical theory of voting and social choice has attracted much attention. In the general setting one can view social choice as a method of aggregating individual, often conflicting preferences and making a choice that is the best compromise. How preferences are expressed and what is the "best compromise" varies and heavily depends on a particular situation.

The method we propose in this paper depends on expressing individual preferences of voters and specifying properties of the resulting ranking by means of first-order formulas. Then, as a technical tool, we use methods of second-order quantifier elimination to analyze and compute results of voting. We show how to specify voting, how to compute resulting rankings and how to verify voting protocols.

Keywords: voting, social choice, quantifier elimination.

1. Introduction

Mathematical theory of voting and social choice has attracted much attention (see, e.g., [3, 4, 6, 7, 11, 15]). In the general setting one can view social choice as a method of aggregating individual, often conflicting preferences and making a choice that is the best compromise. How preferences are expressed and what is the "best compromise" varies and heavily depends on a particular situation. Majority voting is an example of a popular method for making social choices.

To set up the scene, assume there are n voters $V_1, V_2, \ldots, V_n$ expressing their preferences by means of formulas $T_1, T_2, \ldots, T_n$ over signatures containing (at least) relation symbols $>_i$. The intended meaning of $o >_i o'$ is that the i-th voter considers option o to be better than option o'. The goal is to find a relation B which compromises votes expressed by voters. We postulate that the aggregation method is expressed in terms of a first-order formula $T(B)$, expressing properties of relation B which represents the compromise.

This simple setting leads to paradoxical results. Already in the 18th century Condorcet has observed that the combined votes can be in a substantial conflict. For example, suppose that three voters V_1, V_2 and V_3 decide

Special Issue: New Ideas in Applied Logic
Edited by **Dov M. Gabbay and Jacek Malinowski**

Studia Logica (2009) 92: 365–379
DOI: 10.1007/s11225-009-9200-7

whether to buy *wine*, *beer* or *juice*. Suppose that they collectively have resources to buy only one of those drinks, so decided to vote and expressed the following preferences:

$$\begin{array}{lcl} \text{voter } V_1 & : & \mathit{wine} >_1 \mathit{beer} >_1 \mathit{juice} \\ \text{voter } V_2 & : & \mathit{beer} >_2 \mathit{juice} >_2 \mathit{wine} \\ \text{voter } V_3 & : & \mathit{juice} >_3 \mathit{wine} >_3 \mathit{beer}. \end{array} \tag{1}$$

Voters V_1 and V_2 prefer *beer* to *juice*, so *juice* should not win since only voter V_3 prefers *juice* to *beer*. Similarly neither *wine* nor *beer* should win. To resolve this conflict no simple aggregation of individual votes is fair for all voters.

As shown, e.g., in [6], the situation may even be worse. For example, let thirteen voters vote for or against three propositions in a referendum and that the following number of votes has been obtained for each of eight possible outcomes, where Y stands for "yes" and N stands for "no":

YYY: 1, YYN: 1, YNY:1, NYY:1, YNN:3, NYN:3, NNY: 3, NNN:0.

Then for each proposition the number of "no" votes is 7 and the number of "yes" votes is 6. Thus, when votes are aggregated separately for each proposition, the winning combination is NNN, while there has been not a single vote for such a combination and, in fact, as a whole, this combination is the worst one. Exemplifying this situation, one might have the following three propositions in a referendum: "build a basketball field", "build a tennis court", "build a minigolf course". With the voting described above, the referendum results in doing nothing, contrary to the expressed desires of voters.

It is then desirable to have tools for specification and analysis of voting protocols. The ideal situation is when the desired compromise exists independently of voters' decisions and satisfies particular requirements. Some voting protocols, in general, violate this property, but work in many situations. Whenever the compromise exists, one would like to have tools for computing the final ranking/compromise on the basis of specification of voting protocols and voters' decisions.

The method we propose in this paper depends on expressing individual preferences of voters and specifying properties of the resulting ranking by means of first-order formulas. Then, as a technical tool, we use methods of second-order quantifier elimination to analyze and compute results of voting. We show how to specify voting, how to compute resulting rankings and

how to verify voting protocols. We show tractability of computing results for voting protocols specified by means of semi-Horn formulas which are substantially more expressive than Horn formulas.

An elimination of the suitable second-order quantifier, if possible, results in a first-order or fixpoint formula. Both cases are tractable when voting results are presented in the form of a database (see [1, 10, 13]). It is also worth emphasizing that the fixpoint theory of equality is decidable [18]. In cases when second-order quantifiers over all relational variables can be eliminated, one uses the algorithm of [18] to verify correctness of voting protocols. Moreover, using extensions of the Ackermann lemma [2] as well as of the fixpoint theorem of [17], one obtains definitions of the eliminated relations which, in turn, can be used to determine the solution. For a comprehensive overview of the area of second-order quantifier elimination see [12].

The paper is structured as follows. In Section 2 we recall some well-known notions used in the paper. Section 3 is devoted to an introductory example explaining major points we address. Section 4 introduces the formalization of social choice. Then, in Section 5, we recall two theorems concerning second-order quantifier elimination. Section 6 is devoted to applications of the introduced techniques. Finally, Section 7 concludes the paper.

2. Preliminaries

Through the paper we use the language of classical first- and second-order logic without function symbols.[1] We assume the standard first- and second-order semantics.

By a *literal* we understand a first-order formula of the form $R(\ldots)$ or $\neg R(\ldots)$, where R is a relation symbol. We say that a literal is *ground* if it contains no variables. A formula A is in the *negation normal form* if it uses no propositional connectives other than $\neg, \vee, \wedge$ and the negation sign $\neg$ does not occur in A outside of literals.

The following fact is well-known.

Fact 2.1. Every classical first- and second- order formula can equivalently be transformed into a formula in negation normal form. $\triangleleft$

Let $A(R)$ be a formula and $B(R)$ be a formula in the negation normal form equivalent to $A(R)$. Then $A(R)$ is *positive w.r.t.* R if R does not

[1] We avoid function symbols to use deductive databases [1, 10, 13] as a computational machinery.

appear under the negation sign within $B(R)$. It is *negative w.r.t.* R if R appears in $B(R)$ under negation sign only.[2]

Let $A(R)$ be a first-order formula positive w.r.t. R. Then by $\textsc{Lfp}\, R(\bar{x}).\big[A(R)\big]$ we understand the *least fixpoint* of A, i.e., the smallest S such that $S = A(S)$. Since A is positive w.r.t. R, such S exists. See [1, 10, 13] for a detailed exposition of the theory of fixpoints and their applications as database queries.

We shall use the following well-known properties of binary relations:[3]

$$R \text{ is reflexive} \quad \text{iff} \quad \forall x \forall y \big[x = y \rightarrow R(x,y)\big] \tag{2}$$

$$R \text{ is irreflexive} \quad \text{iff} \quad \forall x \forall y \big[R(x,y) \rightarrow x \neq y\big] \tag{3}$$

$$R \text{ is symmetric} \quad \text{iff} \quad \forall x \forall y \big[R(x,y) \rightarrow R(y,x)\big] \tag{4}$$

$$R \text{ is antisymmetric} \quad \text{iff} \quad \forall x \forall y \big[R(x,y) \rightarrow \neg R(y,x)\big] \tag{5}$$

$$R \text{ is weakly antisymmetric} \quad \text{iff} \quad \forall x \forall y \big[\big(R(x,y) \wedge R(y,x)\big) \rightarrow x = y\big] \tag{6}$$

$$R \text{ is transitive} \quad \text{iff} \quad \forall x \forall y \big[\exists z [R(x,z) \wedge R(z,y)] \rightarrow R(x,y)\big] \tag{7}$$

$$R \text{ is total} \quad \text{iff} \quad \forall x \forall y \big[R(x,y) \vee R(y,x)\big] \tag{8}$$

Observe that (2)–(7) are Horn formulas, while (8) is not Horn.

A relation R is a *partial order* if it satisfies (2), (6) and (7). It is a *strict partial order* if it satisfies (3), (5) and (7). It is a *total order* if it satisfies (6), (7) and (8).

Whenever we use relations like $>$, we also accept associated symbols $\geq, <, \leq$ with the standard meaning.

We also assume that writing $A(X, \bar{y})$, we indicate that A contains variables X and $\bar{y}$, but we do not exclude other arguments.

3. An Introductory Example

Consider the votes expressed by (1). The output relation, say B, should provide a ranking on drinks compromising individual votes. The relation B is then specified by a theory T constraining $B, >_1, >_2, >_3$ and frequently expressing other requirements. Assume we have a relation $M(x,y)$ which is true when majority of voters prefer x to y. This relation can easily be computed using the "database" of preferences given by inequalities (1), namely

$$M(wine, beer),\ M(beer, juice),\ M(juice, wine). \tag{9}$$

[2] Observe that formula in which R does not occur is both positive and negative w.r.t. R.

[3] A formulation of some of these properties is a bit unusual, but is easier to use in our calculations. Obviously, it provides definitions equivalent to the standard ones.

We also assume that M is transitive.

Let the aggregation, represented by B, be based on the following principle considered by Condorcet:

$$\forall x \forall y [M(x,y) \rightarrow B(x,y)], \tag{10}$$

where we require that B is antisymmetric.[4]

To check whether B exists we consider the second-order formula $\exists B[(10) \wedge (5)]$, which is equivalent to

$$\exists B \big\{ \forall x \forall y [M(x,y) \rightarrow B(x,y)] \wedge \forall x \forall y [\neg B(x,y) \vee \neg B(y,x)] \big\}. \tag{11}$$

Using the Ackermann's lemma (Lemma 5.1 in Section 5), we obtain the following formula equivalent to (11):

$$\forall x \forall y [\neg M(x,y) \vee \neg M(y,x)]. \tag{12}$$

with the definition

$$\forall x \forall y [B(x,y) \equiv M(x,y)] \tag{13}$$

of the least B satisfying (11).

When we use formula (12) as a query to (the transitive closure of) the database given by (9), we obtain FALSE as the result. This means that there is no relation B satisfying our requirements.

Instead of (9), consider a database resulting from another distribution of votes,[5] given by:

$$M(beer, juice),\ M(beer, wine),\ M(wine, juice). \tag{14}$$

Querying this database using formula (12) returns TRUE, meaning that the required B exists. The least B satisfying (11) is given by (13), so consists of the same tuples as M in (14).

The above discussion shows that certain voting protocols lead to paradoxes. An interesting question is then whether one can relax the requirements as to the final ranking to guarantee that the desired compromise exists. Applying the method we propose, in Example 6.3 we show that relaxing the antisymmetry requirement and placing certain requirements on votes, one can guarantee the existence of aggregated rankings.

---

[4] In fact, such B is often required to be an ordering, in particular to be transitive. We address this point in Example 6.1.

[5] E.g., $beer >_1 wine >_1 juice$, $beer >_2 juice >_2 wine$, $wine >_3 beer >_3 juice$.

4. Modeling Social Choice

We address the following problems:

- *specification*: how to specify voting and referenda?
- *computing results*: given a result of voting as a database and a method of aggregating votes, how to check consistency and find aggregated result?
- *correctness of aggregation*: given a method of aggregating votes, how to check its consistency for all possible results of voting?

From the practical point of view there is always a finite number of voters and a finite number of possible choices. Therefore it is reasonable to make the following assumptions, simplifying the considerations and substantially reducing the complexity of the approach:

$$\text{we restrict considerations to finite theories only} \tag{15}$$

$$\text{we consider finite models only,} \tag{16}$$

where by a finite theory we mean any theory with a finite number of axioms.

Observe that any finite theory can be considered as a single formula being the conjunction of all axioms of the theory.

4.1. Voting Specification

We address a rather general *voting problem*, where $k > 0$ voters $V_1, V_2, \ldots, V_k$ express their votes by means of (finite) theories $T_1, T_2, \ldots, T_k$ on $n > 0$ *options* $o_1, o_2, \ldots, o_n$. These theories are aggregated into a single relation on tuples of options, $B(o_1, o_2, \ldots, o_n, o'_1, o'_2, \ldots, o'_n)$, intuitively meaning that the tuple $\langle o_1, o_2, \ldots, o_n\rangle$ is better than the tuple $\langle o'_1, o'_2, \ldots, o'_n\rangle$. The aggregation, also called *ranking*, is expressed by a theory $T(B)$ not necessarily providing an explicit definition of B (see Figure 1).

The ideal situation is when the least (w.r.t. inclusion) ranking B exists and then it provides the intended aggregation. However, sometimes one can have many minimal rankings, e.g., when voters are only required to provide partial orders on their preferences being allowed to stay indifferent to particular choices. We do not make assumption as to the existence of the least ranking, however, will provide conditions under which it is guaranteed.

Observe that B does no longer have to be a binary relation. In what follows, in order to use properties (2)–(8), we treat B as a binary relation on tuples, $B(\bar{o}, \bar{o'})$.

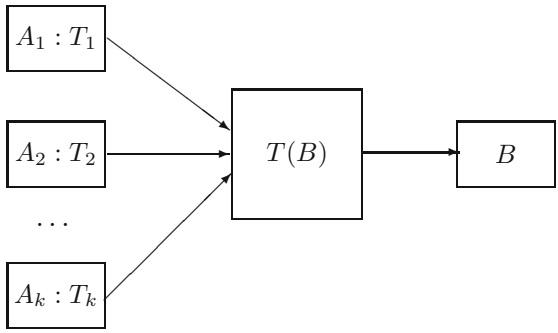

Figure 1. Voting aggregation.

For purely pragmatic reasons we separate ground literals from other formulas in $T(B)$. Namely, often $T(B)$ is equivalent to a conjunction $T_E(B) \wedge T_I(B)$, where E is a conjunction of ground literals and I is a conjunction of formulas of other shape. In such a case, using the terminology of databases [1], we call $T_E(B)$ an *extensional database* and $T_I(B)$ an *intensional database*.[6]

4.2. Computing Results

Let $T(B)$ be a theory specifying voting, as done in Section 4.1. In order to compute results of voting we consider formula

$$\exists B[T_I(B)], \tag{17}$$

where $T_I(B)$ stands for the intensional database.

Checking whether formula (17) holds in a given database $T_E(B)$, we obtain information whether the output relation exists, i.e., whether the specification is consistent. According to Fagin's theorem, existential second-order queries are not tractable (see [1, 10, 13]). Therefore we first attempt to eliminate the second-order quantifier, using results provided in Section 5. If the elimination is successful, then as a side effect we obtain the definition of the smallest relation compromising votes.

More precise description of this method requires particular techniques of second-order quantifier elimination, which is given in Section 5.2 and illustrated in Examples 6.1 and 6.2.

[6] Observe that I, as defined here, is more general than typically accepted in databases.

4.3. Correctness of Aggregation

Let a voting problem be given as in Section 4.1. One is often interested in defining methods of reaching a compromise which guarantee the existence of ranking expressed by B. In such cases we are interested to prove that formulas expressing this existence (like formula (11)) are tautologies. This can be expressed by the second order formula

$$\forall \bar{P} \exists B[T(\bar{P}, B)], \tag{18}$$

where $\bar{P}$ consists of all relation symbols, except for B, which appear in the theory T implicitly defining the compromise B. If all second-order quantifiers can be eliminated from formula (18) then the elimination results in an equivalent formula expressed in the language of the (decidable) fixpoint theory of equality (see [18]). The class of formulas where such elimination is possible is rather large. For a summary see [12].

This method is illustrated in Example 6.3.

5. Second-Order Quantifier Elimination

5.1. Basic Results

In this section we recall known results. A comprehensive study of the subject is provided in the book [12].

By $\alpha(a,\ldots)^{e_1(x_1),\ldots,e_k(x_k)}_{f_1(y_1),\ldots,f_k(y_k)}$ we understand the expression obtained from α in such a way that for any $1 \leq i \leq k$, all occurrences of e_i of the form $e_i(a)$ are replaced by $f_i(y_i)$, where y_i itself is replaced by a. For example,

$$\big(P(s) \vee P(t)\big)^{P(x)}_{(Q(a,u)\wedge Q(u,b))(u)} = \big((Q(a,s) \wedge Q(s,b))\big) \vee \big((Q(a,t) \wedge Q(t,b))\big).$$

Let us now recall a part of the theorem of [17], which we use in further calculations.

If X is a variable representing k-ary relations then we define the arity of X to be k.

THEOREM 5.1. *Let X be a relation variable and $A(X,\bar{x},\bar{z})$, $C(X)$ be classical first-order formulas, where the number of distinct variables in $\bar{x}$ is equal to the arity of X, $A(X,\bar{x},\bar{z})$ is positive w.r.t. X and $C(X)$ is negative w.r.t. X. Then*

$$\exists X\big\{\forall \bar{x}[A(X,\bar{x},\bar{z}) \rightarrow X(\bar{x})] \wedge C(X)\big\} \;\equiv\; C(X)^{X(\bar{x})}_{\textsc{Lfp}\, X(\bar{x}).\left[A(X,\bar{x},\bar{z})\right]}, \tag{19}$$

where LFP *stands for the least fixpoint operator.* ◁

As a corollary we have the following lemma first proved in [2].

LEMMA 5.1. *Let X be a relation variable and $A(\bar{x}, \bar{z})$, $C(X)$ be classical first-order formulas, where the number of distinct variables in $\bar{x}$ is equal to the arity of X. Let A contain no occurrences of X and $C(X)$ be negative w.r.t. X. Then*

$$\exists X\big\{\forall \bar{x}[A(\bar{x}, \bar{z}) \to X(\bar{x})] \wedge C(X)\big\} \;\equiv\; C(X)^{X(\bar{x})}_{A(\bar{x},\bar{z})}. \qquad \triangleleft \qquad (20)$$

5.2. Computational Aspects

In the case when Theorem 5.1 is applicable, by a *coherence condition* for formula

$$\exists X\big\{\forall \bar{x}[A(X, \bar{x}, \bar{z}) \to X(\bar{x})] \wedge C(X)\big\} \qquad (21)$$

we understand its equivalent $C(X)^{X(\bar{x})}_{\text{LFP}\, X(x).\left[A(X,\bar{x},\bar{z})\right]}$.

Moreover, if the coherence condition is satisfied, there exists the least (w.r.t. $\subseteq$) relation X satisfying $\forall \bar{x}[A(X, \bar{x}, \bar{z}) \to X(\bar{x})] \wedge C(X)$. The least such relation is defined by

$$X(\bar{x}) \equiv \text{LFP}\, X(x).\big[A(X, \bar{x}, \bar{z})\big]. \qquad (22)$$

Of course, the same holds for Lemma 5.1, where the least fixpoint operator simply reduces to $A(\bar{x}, \bar{z})$.

Assuming that the existential second-order quantifier $\exists B$ can be eliminated from formula (17) using Theorem 5.1 (or Lemma 5.1), we have the following procedure:

1. check the coherence condition for (17)
2. if the coherence condition is TRUE then compute B as a query to the database with votes, using its definition given due to (22).

Since both coherence condition and the definition of B are fixpoint formulas, checking the condition and computing B are in this case of deterministic polynomial complexity in the size of the database (see [1, 10, 13]). If Lemma 5.1 is applicable then the time complexity is still polynomial, however the space complexity becomes logarithmic in the size of the database.

REMARK 5.2. Let us point out that equality, which frequently appears in our formulas, is treated differently in logic and deductive databases. Namely, in databases a Unique Names Assumption (UNA) is accepted. According to

UNA, distinct constants represent different objects. For example, $a = b$ is always FALSE in databases, while it might be satisfied in a given model of first-order logic.

To overcome this difficulty, it suffices to represent equality in databases by a binary relation, say E and, in addition to reflexivity, symmetry and transitivity of E, to add the rule $\forall x \forall y[x = y \rightarrow E(x, y)]$.[7] This will allow one to make $E(a, b)$ true for different constants a, b. ◁

REMARK 5.3. Observe that whenever the formula under the second-order quantifier $\exists X$ is *semi-Horn*, i.e., is expressed by a conjunction of formulas negative w.r.t. X and/or formulas of the form

$$\forall \bar{x} \forall \bar{y}[A(X, \bar{x}, \bar{y}) \rightarrow X(\bar{x}, \bar{y})], \tag{23}$$

with $A(X, \bar{x}, \bar{y})$ positive w.r.t. X, then one can gather them together, using standard techniques, in particular manipulating first-order quantifiers and using the propositional tautology $[(p \rightarrow q) \wedge (r \rightarrow q)] \equiv [(p \vee r) \rightarrow q]$.[8] The resulting formula allows for a direct application of Theorem 5.1. In fact, this is a quite expressive class of formulas, substantially stronger than the class of Horn formulas. For example, $A(X, \bar{x}, \bar{y})$ can contain arbitrary first-order quantifiers, which is not allowed in Horn formulas. ◁

As a consequence of discussion provided in Remark 5.3, we have the following theorem.

THEOREM 5.2. *If the theory expressed by $T(B)$ in formula* (17) *is a conjunction of semi-Horn formulas then checking whether the aggregated relation B exists as well as computing B can be done in deterministic polynomial time w.r.t. size of the universe of the underlying database.* ◁

5.3. The Case of Partial Orders

Assume that the aggregated relation B is required to be a partial order. Then it satisfies the conjunction of conditions (2), (6) and (7). Gathering those conditions provides the following equivalent formula:

$$\begin{aligned} \forall \bar{x} \forall \bar{y} \Big[\Big(\bar{x} = \bar{y} \vee \exists \bar{z}[B(\bar{x}, \bar{z}) \wedge B(\bar{z}, \bar{y})] \Big) \rightarrow B(\bar{x}, \bar{y}) \Big] \wedge \\ \forall \bar{x} \forall \bar{y} \big[\neg B(\bar{x}, \bar{y}) \vee \neg B(\bar{y}, \bar{x}) \vee \bar{x} = \bar{y} \big]. \end{aligned} \tag{24}$$

[7] Observe that the considered rule is a Horn formula. Also, reflexivity, symmetry and transitivity can be expressed by Horn formulas - see formulas (2), (4), (7), so such a simple encoding of equality can be served by typical deductive databases.

[8] Such transformations are done automatically in the DLS algorithm presented in [8] (see also [12]).

In the case of strict partial order the conjunction of (3), (5) and (7) is equivalent to:

$$\begin{aligned}\forall\bar{x}\forall\bar{y}\Big[\exists\bar{z}[B(\bar{x},\bar{z})\wedge B(\bar{z},\bar{y})]\rightarrow B(\bar{x},\bar{y})\Big]\wedge\\ \forall\bar{x}\forall\bar{y}\big[\neg B(\bar{x},\bar{y})\vee\bar{x}\neq\bar{y})\big]\wedge\forall\bar{x}\forall\bar{y}\big[\neg B(\bar{x},\bar{y})\vee\neg B(\bar{y},\bar{x})\big].\end{aligned}\tag{25}$$

Observe that formulas (24) and (25) are in the form (19) required in Theorem 5.1. Therefore, applying the method discussed in Remark 5.3 we immediately obtain a formula in the form (19) required in Theorem 5.1 and can be used for necessary computations. We then have the following theorem.

THEOREM 5.3. *If $T(B)$ satisfies assumptions of Theorem 5.2 then its conclusion holds also for the theory $T(B)\wedge\Delta(B)$, where $\Delta(B)$ is the conjunction of conditions* (2)*,* (6) *and* (7) *applied to B.* ◁

Obviously, the above theorem can be generalized for Δ being any conjunction of conditions (2)–(7).

5.4. The General Case

Unfortunately, when $T(B)$ is not a semi-Horn formula, the method based on Theorem 5.1 is not directly applicable and the success of algorithm attempting to transform formulas into the required form is not guaranteed. As a remedy, one can try other methods of second-order quantifier elimination [12] or, given a database, transform $T(B)$ into quantifier-free formula by replacing universal quantifiers by finite conjunctions and existential quantifiers by finite disjunctions according to equivalences:

$$\begin{aligned}DB\models\forall x[A(x)]\text{ iff }DB\models A(a_1)\wedge A(a_2)\wedge\ldots\wedge A(a_m)\\ DB\models\exists x[A(x)]\text{ iff }DB\models A(a_1)\vee A(a_2)\vee\ldots\vee A(a_m),\end{aligned}$$

where the domain of the database DB is $\{a_1,a_2,\ldots,a_m\}$. As a result one obtains a Boolean combination of ground literals for which SAT solvers can be applied. Observe that modern SAT solvers are very successful even for large propositional formulas (see, e.g. [14]). However, deterministic polynomial time complexity is no longer guaranteed. Also, SAT solvers cannot be applied in verification of correctness of aggregation using the method discussed in Section 4.3. One would have to apply here solvers for quantified Boolean formulas. For a comparison of such solvers see [16].

Even if the class of semi-Horn formulas is very expressive, there are very natural properties outside of this class. For example, the totality property, as expressed by (8), is not semi-Horn.

6. Examples of Applications

Let us now show examples of applications of the proposed methods. All examples presented below concern some variations of the situation described in Section 3.

Example 6.1. Assume that we drop the assumption as to the transitivity of M, accepted in Section 3, and instead require the transitivity of B.

To check whether such B exists we consider the formula $\exists B[(10)\wedge(5)\wedge(7)]$ which, using the method discussed in Remark 5.3, can be transformed to its equivalent

$$\exists B\big\{\forall x\forall y\big[(M(x,y)\vee\exists z[B(x,z)\wedge B(z,y)])\rightarrow B(x,y)\big]\wedge \forall x\forall y[\neg B(x,y)\vee\neg B(y,x)]\big\}. \tag{26}$$

Using Theorem 5.1, we obtain the following formula equivalent to (26):

$$\forall x\forall y[\neg B(x,y)\vee\neg B(y,x)], \tag{27}$$

where B is defined by

$$B(x,y)\equiv \textsc{Lfp}\, B(x,y).\big[M(x,y)\vee\exists z[B(x,z)\wedge B(z,y)]\big]. \tag{28}$$

Of course, computing B can be done in deterministic time polynomial w.r.t. the size of the extensional database. Once B is computed, we have to check whether coherence condition (27) is satisfied. Of course, this verification is tractable too. ◁

Example 6.2. Let us consider requirements on B stronger than transitivity, discussed in Example 6.1.

- Requiring B to be a strict partial order, we use formula (25) and obtain the definition of B, given by formula (28), but the coherence condition is now

$$\forall x\forall y[\neg B(x,y)\vee x\neq y)]\wedge\forall x\forall y[\neg B(x,y)\vee\neg B(y,x)].$$

- Requiring B to be a partial order, in addition to (10) we use formula (24) and obtain the coherence condition

$$\forall x\forall y[\neg B(x,y)\vee\neg B(y,x)\vee x=y]$$

and the definition of B:

$$B(x,y)\equiv \textsc{Lfp}\, B(x,y).\big[M(x,y)\vee x=y\vee\exists z[B(x,z)\wedge B(z,y)]\big]. \quad \triangleleft$$

Example 6.3. Assume one changes the requirements of Section 3 as to aggregation by assuming now that the resulting relation is weakly antisymmetric. We are now interested in verifying whether the aggregation is correct in the sense formalized in Section 4.3.

According to (18), the correctness of aggregation is expressed by (18), where $T(\bar{P}, B) \equiv (10) \wedge (6)$, i.e.,

$$\forall M \exists B[\forall x \forall y[M(x,y) \rightarrow B(x,y)] \wedge \forall x \forall y\big[\big(B(x,y) \wedge B(y,x)\big) \rightarrow x = y\big]]. \tag{29}$$

After an application of Lemma 5.1 to $\exists B[\ldots]$ we obtain

$$\forall M \forall x \forall y\big[\big(M(x,y) \wedge M(y,x)\big) \rightarrow x = y\big]. \tag{30}$$

Obviously, formula (30) is not a tautology. However, it provides a simple criterion for the correctness of the method. Namely, if we add the requirement that M is weakly antisymmetric then (30) becomes True and, in consequence, (29) becomes True, too. ◁

7. Conclusions

In this paper we have presented a method suitable for analyzing quite general methods of verification of voting protocols, checking whether a compromise exists when particular voting results are given and, if exists, computing the ranking compromising voters. The method is quite general, as it is based on the second-order logic and second-order quantifier elimination, leading to tractable methods for large classes of theories expressing votes and aggregation of results. In fact, the fragment of first-order logic for which tractability is guaranteed is substantially more expressive than the Horn fragment and logic programming based on Horn rules.

Let us note that the method can also be used when advanced methods of expressing preferences are allowed (as, e.g., in [9]). The use of second-order formalism allows also for integration of the proposed method with commonsense reasoning, e.g., based on circumscription.

Possible applications of the proposed methodology also include multi-agent systems, where voting might be a substantial tool in reaching the agreement. Specifying votes by means of theories makes the method flexible and tractability results allow for efficient implementations.

Comparing the encoding we proposed, e.g., with the well-known Arrow's impossibility theorem [5], one can see the limitations of our method. Namely, Arrow's conditions are basically third-order. We think it is possible to work out a specialized quantifier elimination technique, applicable to Arrow's-like

conditions, but such a result would be rather far outside of scope of the current paper. We leave this topic for a future research.

Supported in part by the MNiSW grant N N206 399134.

References

[1] Abiteboul, S., R. Hull, and V. Vianu, *Foundations of Databases*, Addison-Wesley Pub. Co., 1996.

[2] Ackermann, W., 'Untersuchungen über das eliminationsproblem der mathematischen logik', *Mathematische Annalen*, 110 (1935), 390–413.

[3] Arrow, K., *Social Choice and Individual Values*, John Wiley & Sons, 1951.

[4] Arrow, K., A.K. Sen, and K. Suzumura, *Handbook of Social Choice and Welfare*, Elsevier, 2002.

[5] Arrow, K.J., 'A difficulty in the concept of social welfare', *Journal of Political Economy*, 58 (1950), 4, 328–346.

[6] Brams, S.J., D.M. Kilgour, and W.S. Zwicker, 'The paradox of multiple elections', *Social Choice and Welfare*, 15 (1998), 211–236.

[7] Conitzer, V., and T. Sandholm, 'Universal voting protocols tweaks to make manipulation hard', in *Proceedings of IJCAI-03*, 2003, pp. 781–788.

[8] Doherty, P., W. Łukaszewicz, and A. Szałas, 'Computing circumscription revisited', *Journal of Automated Reasoning*, 18 (1997), 3, 297–336.

[9] Doherty, P., and A. Szałas, 'Reasoning with qualitative preferences and cardinalities using generalized circumscription', in G. Brewka, and J. Lang, (eds.), *Proc. KR 2008, 11th International Conference on Principles of Knowledge Representation and Reasoning*, AAAI Press, 2008, pp. 560–570.

[10] Ebbinghaus, H-D., and J. Flum, *Finite Model Theory*, Springer-Verlag, Heidelberg, 1995.

[11] Gabbay, D.M., G. Pigozzi, and O. Rodrigues, 'Belief revision, belief merging and voting', in *Proceedings of the Seventh Conference on Logic and the Foundations of Games and Decision Theory (LOFT06)*, University of Liverpool, 2006, pp. 71–78.

[12] Gabbay, D.M., R. Schmidt, and A. Szałas, *Second-Order Quantifier Elimination. Foundations, Computational Aspects and Applications*, vol. 12 of *Studies in Logic*, College Publications, 2008.

[13] Immerman, N., *Descriptive Complexity*, Springer-Verlag, New York, Berlin, 1998.

[14] Kautz, H., and B. Selman, 'The state of SAT', *Discrete Applied Mathematics*, 155 (2007), 12, 1514–1524.

[15] Laslier, J.-F., *Tournament solutions and majority voting*, Springer, 1997.

[16] Narizzano, M., L. Pulina, and A. Tacchella, 'The third QBF solvers comparative evaluation', *Journal on Satisfiability, Boolean Modeling and Computation*, 2 (2006), 145–164.

[17] Nonnengart, A., and A. Szałas, 'A fixpoint approach to second-order quantifier elimination with applications to correspondence theory', in E. Orłowska, (ed.), *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, vol. 24 of *Studies in Fuzziness and Soft Computing*, Springer Physica-Verlag, 1998, pp. 307–328.

[18] Szałas, A., and J. Tyszkiewicz, 'On the fixpoint theory of equality and its applications', in R. Schmidt, (ed.), *Relations and Kleene Algebra in Computer Science*, vol. 4136 of *LNCS*, 2006, pp. 388–401.

Dov M. Gabbay
Department of Computer Science
King's College, London, UK
and
Bar-Ilan University, Israel
dov.gabbay@dcs.kcl.ac.uk

Andrzej Szałas
Institute of Informatics
Warsaw University, Poland
and
Dept. of Comp. and Information Sci.
University of Linköping, Sweden
andsz@mimuw.edu.pl