# Verification of Voting Protocols

Sieuwert van Otterloo
Department of Computer Science
University of Liverpool
United Kingdom

Olivier Roy
ILLC
University of Amsterdam
The Netherlands

{sotterlo,oroy}@science.uva.nl

**Abstract**

Verification of strategic properties is complicated, even for very small protocols. In this work we look at voting protocols that allow three agents to vote over three alternatives, and investigate what properties of those protocols can be expressed in different logics. We first verify some basic requirements using a logic of coalition powers. Then we investigate a few advanced properties in complete and incomplete information context, using two more expressive logics. Along the way, we propose a new compact way to represent game trees.

## 1 Introduction

Voting is often used as a method for group decision making. The nice thing about voting is that it can easily be formalized, compared to alternatives such as debating. Voting protocols can for instance be modeled as extensive games, and then one can use game theory or logical methods to study their properties. For a survey of game theoretic approaches, see Myerson [11, p.196-201]. Logical approaches include work on social software [12, 13], and game-oriented logics [1, 14]. The verification problem is also related to the problem of automated design of suitable protocols [4, 5].

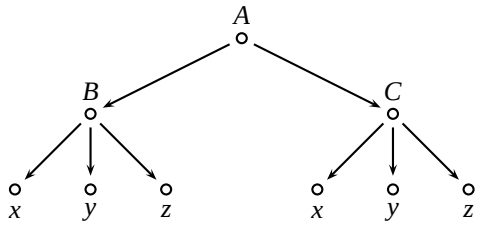
Verification of protocols can be done with different levels of detail. One can simply check certain technical properties, such that a protocol terminates or that all outcome states can be reached, and that inconsistencies cannot arise. A more thorough verification can check that all coalitions can enforce what they should be able to enforce. We call these properties "basic requirements", since they are easily understood and can be captured in many frameworks.

There can be many protocols that meet a certain set of basic requirements and in that case one would like to have a method to distinguish them. This leads us to look at two so-called 'advanced properties', namely bias and asymmetry. In order to check these properties one must specify what the agents that participate in the voting protocol know. If these agents know each other's preferences we can formalize the protocol as a complete information game. In this case backward induction is one well-known solution concept that can be used to verify many properties. We present a logic that can automate reasoning about backward induction.

One cannot always assume that all agents have complete information about agents' preferences. Therefore we also look into the incomplete information case, where agents have limited information about each others' preferences. In this case concepts from qualitative decision theory [6] become relevant. We also present a logic for modeling these decisions, and apply this logic to the example problem.

We hope that by combining different logical frameworks in one paper, it becomes more clear that a distinction between basic and advanced properties can be made in verification.

Figure 1: A voting protocol $F_1$




For verification of basic properties the main issue is efficiency. This paper shows how this depends on whether one adopts an efficient input format for extensive game forms. The second contribution that this paper makes is that it demonstrates verification of advanced properties under different knowledge assumptions.

In section 2 we define basic concepts for protocol verification. Section 3 explains advanced properties. We present a new logic *LPBI* that can be used for reasoning under complete information. This is done in section 4. In section 5 we compare the analysis with an analysis using *GLP*, a relatively new logic for reasoning under incomplete information. Section 6 is the conclusion.

# 2 Basic definitions

The example problems used throughout this paper involve three agents who have to elect one of three options. We allow protocols in which at each moment in time exactly one agent is faced with a decision. When making this decision, the agent is fully informed of all past decisions. Such protocols can be formalized as perfect information extensive game forms. We call the three agents *A*, *B*, and *C*. The options that are available are *x*, *y*, and *z*. Exactly one of those options must be chosen. If a majority supports one option, that option should be elected.

The first protocol that we analyze is a very short protocol. It allows agent *A* to choose at the beginning for either agent *B* or agent *C*. Than the agent chosen by *A* can select an option, and this chosen option is elected. A game tree for this protocol is given in figure 1.

The next two definitions define interpreted extensive game forms. These are formal description of game trees. They do not describe agents' preferences. The word 'interpreted' indicates that the game tree is annotated with atomic propositions.

**Definition 1.** *A set of finite sequences $H$ is* prefix-closed *if for any sequence $h$ and action $a$ it is the case that $ha \in H$ implies $h \in H$. For any set of sequences $H$ and $h \in H$ we define the set of next actions $A(H,h) = \{a | ha \in H\}$ and the set of terminal sequences $Z(H) = \{h \in H | A(H,h) = \emptyset\}$.*

Sequences of actions can be used to denote specific plays of a game. Such sequences are also called histories or runs. $Z(H)$ denotes the set of all sequences that cannot be extended. These are called terminal histories or sequences, and correspond to outcomes. The set $A(H,h)$ consists of all actions that can be played in $h$. The set $H$ implicitly defines a tree, since one can think of $H$ as containing all paths in the tree that start from the root and go down the tree.

**Definition 2.** *An* interpreted extensive game form *$F$ is a tuple $F = (\Sigma, H, turn, P, \pi)$, where $\Sigma$ is a finite sets of agents, $P$ is a finite set of atomic propositions, $H$ is a non-empty, prefix-*

*closed set of finite sequences, turn is a function turn* : $H \setminus Z(H) \to \Sigma$ *and* $\pi : Z(H) \to 2^P$ *returns the true atomic propositions of any terminal history.*

**Definition 3.** *Let* $F = (\Sigma, H, turn, P, \pi)$ *be a game form and* $\Gamma \subset \Sigma$ *a coalition of agents. A strategy* $\sigma_\Gamma$ *for* $\Gamma$ *is a function with domain* $\{h \in H | turn(h) \in \Gamma\}$ *such that* $\sigma_\Gamma(h)$ *is a non-empty subset of* $A(H, h)$.

The definition of a game form does not allow one to compactly specify the interpreted game form $F_1$. A more compact format is needed, and for this purpose we describe here a new way of representing those game forms, called *linear representation*. The idea behind this description method is that a game tree can be summarized by describing a typical path.

A linear representation $R$ is a string of symbols that denotes a game form. In the definition of linear representation we allow the use of variables, substitution of variables, and any commonly used mathematical operation that can be done in polynomial time. For instance we can use the string '$2+2$' in the definition of a linear representation to denote the natural number 4. If the string $s$ is equal to '$1 - x$' then the string $s[x \setminus 1]$ denotes the value $0$. The size of a representation is the number of symbols it consists of. We write $\dot{R}$ for the denotation of $R$.

**Definition 4.** *Assume the finite sets* $P, \Sigma$ *are given. The set* $\mathcal{R}$ *of all representations of game forms is the smallest set* $\mathcal{R}$ *such that*

- *If* $\dot{R} \subset P$ *then* $R \in \mathcal{R}$
- *if* $S$ *denotes a set* $\dot{S}$ *such that* $|\dot{S}| \leq |S|$, $s$ *is a variable,* $\dot{X} \in \Sigma$ *and* $R$ *is a string such that for all* $s_i \in \dot{S}$, $R[s \setminus s_i] \in \mathcal{R}$, *then* $X \xrightarrow{s \in S} R \in \mathcal{R}$
- *If* $R_0 = X \xrightarrow{s \in S_0} R_0' \in \mathcal{R}$ *and* $R_1 = X \xrightarrow{s \in S_1} R_1' \in \mathcal{R}$ *and* $\dot{S_0} \cap \dot{S_1} = \emptyset$ *then* $R_0 || R_1 \in \mathcal{R}$

The function $m$ can be applied to convert representations of game forms into game forms. The function $m$ is defined in the following way.

**Definition 5.** *Assume the finite sets* $P, \Sigma$ *are given, Define* $m(R) = m_1(\emptyset, R)$, *where* $m_1$ *is the function defined below. Let* $h$ *be a sequence of actions, and* $f_0$ *be the function with empty domain.*

- *If* $\dot{R} \subset P$ *then* $m(h, R) = (\Sigma, \{h\}, \emptyset, P, \pi)$ *where* $\pi(h) = \dot{R}$.
- *Assume* $R = X \xrightarrow{s \in S} R'$. *For any* $s_i \in \dot{S}$ *define* $m(hs_i, R'[s \setminus s_i]) = (\Sigma, H_i, turn_i, P, \pi_i)$ *where* $turn_i$ *is a function that assigns* $X$ *to* $h$. *The result* $m(h, R)$ *is defined as* $m(h, R) = (\Sigma, \bigcup_i H_i \cup \{h\}, turn, P, \bigcup_i \pi_i)$ *where* $turn = (\bigcup_i turn_i) \cup \{(h, X)\}$.
- $m(h, R_0 || R_1) = (\Sigma, H_0 \cup H_1, turn_0 \cup turn_1, P, \pi_0 \cup \pi_1)$ *where* $(\Sigma, H_i, turn_i, P, \pi_i) = m(h, R_i)$

$F_1$ has the following linear representation.

$$R_1^A = A \xrightarrow{X \in \{B,C\}} X \xrightarrow{p \in P} \{p\}$$

It says exactly what the protocol is : A chooses between $B$ and $C$, which in turn chooses his favorite alternative.

**Fact.** *For any linear representation* $R$, $m(R)$ *is an interpreted extensive form.*

*Proof.* An induction on $R$ proves this fact. □

In order to verify that $F_1$ is indeed an acceptable voting protocol, we need to verify some basic technical properties. In order to do so, we define the logic $GLP^-$. The language $GLP^-$ is a fragment of *GLP*, which was introduced in [16]. We will use full *GLP* later on, to talk about powers of agents in incomplete information contexts. The restricted version $GLP^-$ has a simpler interpretation, that does not rely on assuming incomplete information. Let presuppose that a finite set $\Sigma$ of agents and a finite set $P$ of atomic propositions are defined. The next definition defines propositional logic $\mathcal{P}$ and restricted game logic with preferences $GLP^-$.

**Definition 6.** *Let $con(L) = \{\bot, \varphi \to \psi | \varphi, \psi \in L\}$ and let propositional logic $\mathcal{P}$ be defined as the smallest language L such that $L = P \cup con(L)$. Let $glp^-(L) = \{[\Gamma : \varphi]\Box\varphi, \Box\varphi | \varphi \in \mathcal{P}\}$. The language $GLP^-$ is the smallest language L such that $L = glp^-(L) \cup con(L)$.*

The logic $GLP^-$ can be interpreted over interpreted game forms using an update function *Up*.

**Definition 7.** *Let $F = (\Sigma, H, turn, P, \pi)$ be a game form and $\sigma_\Gamma$ a strategy for $\Gamma$. Define $Up(F) = (\Sigma, H', turn', P, \pi')$ where H is the greatest subset of H such that $turn(h) \in \Gamma \wedge ha \in H'$ implies $h \in H' \wedge a \in \sigma_\Gamma(h)$. The function $turn', \pi'$ are identical to $turn, \pi$ but restricted to $H'$.*

$$
\begin{array}{lll}
F \models \bot & & \text{never} \\
F \models \varphi \to \psi & \text{iff} & \text{not } T \models \varphi \text{ or } T \models \psi \\
F \models \Box\varphi & \text{iff} & \forall h \in Z(H) : \pi(h) \models \varphi \\
F \models [\Gamma : \varphi]\Box\varphi & \text{iff} & \exists \sigma_\Gamma \forall h \in Z(H) : \pi(h) \models \varphi \\
 & & \text{where } (\Sigma, H, turn, P, \pi) = Up(F, \sigma_\Gamma)
\end{array}
$$

Since we claim that this is a simple logic, one would expect that the model checking problem for this logic is tractable. However the linear representation of a game form is a lot more compact than a naive representation of a game form. Therefore if one specifies the input using linear representation, the problem has a high computational complexity.

**Fact.** *Deciding whether a $GLP^-$ formula $\varphi$ holds on a linearly represented game form F is PSPACE-complete*

*Proof.* The definition of $F \models \varphi$ can be converted into a naive algorithm that only needs one branch of the game tree at the time. This algorithm can therefore work in polynomial space, and thus the decision problem is in PSPACE. It remains to be proven that the problem is PSPACE-hard. This can be done by reducing the PSPACE complete problem Quantified Boolean Formulas(QBF) to the $GLP^-$ decision problem. The objective of a QBF problem is to decide for a given formula of the form $\forall x_1 \exists y_1 \forall x_2 \ldots \exists y_n \forall x_{n+1} \varphi$ whether this formula holds. The formula $\varphi$ is a propositional logic formula with only propositions from the set $\{x_i, y_i, x_{n+1} | 0 < i \leq n\}$. Assume that a QBF formula $\forall x_1 \exists y_1 \forall x_2 \ldots \exists y_n \forall x_{n+1} \varphi$ is given. We have to construct an equivalent $GLP^-$ decision problem, consisting of a representation $R$ and a formula $\varphi'$. Let $\Sigma = X, Y$ and $P = \{x_i, y_i, x_{n+1} | 0 < i \leq n\}$. The representation of a game form is the following:

$$R = X \xrightarrow{v_0 \in \{0,1\}} Y \xrightarrow{w_0 \in \{0,1\}} \ldots Y \xrightarrow{w_n \in \{0,1\}} X \xrightarrow{v_{n+1} \in \{0,1\}} \{x_i | v_i = 1\} \cup \{y_i | w_i = 1\}$$

Take $\varphi' = [Y : \varphi]\Box\varphi$. It is not hard to see that there is indeed the required equivalence. If $\forall x_1 \exists y_1 \forall x_2 \ldots \exists y_n \forall x_{n+1} \varphi$, then $m(R) \models \varphi'$ and vice versa. □

$GLP^-$ can be used to determine whether one and only one alternative is to be elected in a given protocol through a validity check of the following formula :

$$\Box(\bigvee_{u \in P} u) \wedge \Box(\bigwedge_{u \in P} (u \to \neg(\bigvee_{v \in P - \{u\}} v)) \qquad \text{(One Alternative)}$$

Indeed, this formula is valid on $F_1$. A validity check on the next formula ensure us a more subtle property : that a majority of agents can force an outcome.

$$\bigwedge_{B\subseteq\Sigma:|B|>|\Sigma-B|} \bigwedge_{u\in P} [B:u]\Box u \qquad \text{(Majority Decides)}$$

We could have used alternative logical frameworks to establish the same results. Most notable, ATL [1] and coalition logic [13] are alternatives, which is not very surprising since these logics are related [7]. We have used $GLP^-$ here because we use full *GLP* later. The main point is that these properties can be verified. This allows us to call these basic properties.

# 3 Advanced Properties

The roles of *A* versus the role of *B* or *C* seems quite different. The last two agents can directly support an alternative, whereas agent *A* cannot directly support an option. One can ask whether this is fair, or whether *A* has an advantage or a disadvantage. Similarly one can imagine protocols in which *x* is treated different than *y* or *z*. If such a difference can be found, the protocol cannot be called completely fair. In the remainder of the paper, we try to formalize these two requirements :

- *Bias:* A protocols is biased if outcomes are treated differently: one outcome is more likely than others.
- *Asymmetry:* If one agent has an advantage over another, the protocol is called asymmetric.

Any logic that only looks at the powers of coalitions, such as $GLP^-$, cannot be used to establish asymmetry in protocol $F_1$. This is because in this protocol the decision powers of each coalition are symmetric. Therefore we look at more expressive logics to study asymmetry.

In our study of voting protocols we model protocols as game forms, without built-in preferences. In game theory games are usually studied with given preferences, which are known to all agents. We thus have to make a decision whether agents know each others' preferences or not. If we assume that preferences are known, then we can use standard solution concepts such as backward induction. This is done in the next section. If preferences are not known, we can use ideas from qualitative decision theory.

One can also define more protocols that satisfy our basic requirements, while being very different from protocol $F_1$. Below we define two more protocols, called $F_2 = m(R_2^{ABC})$ and $F_3 = m(R_3^x)$. Part of the game tree of protocol $F_2$ is depicted in figure 2.

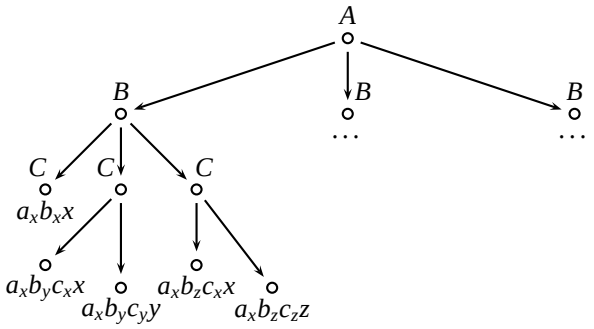
$$R_2^{ABC} = A \xrightarrow{a\in P} (B \xrightarrow{b\in P\setminus\{a\}} C \xrightarrow{c\in\{a,b\}} \{c\})||(B \xrightarrow{a} \{a\})$$

In $F_2$, *A* and *B* choose from the three possible outcomes. If they choose the same outcome, then that is the final outcome. Otherwise *C* can choose from the two outcomes they selected.

$$R_3^x = A \xrightarrow{a\in P} B \xrightarrow{b\in P} C \xrightarrow{c\in P} \{x||\{a,b,c\}|=3\} \cup \{u||\{a,b,c\}\setminus\{u\}|<2\}$$

In our third example $F_3$, *A*, *B*, can *C* vote sequentially for one of the three outcomes; the one that gets the most votes is elected. If *A*, *B* and *C* disagree then a pre-determined outcome *x* is elected. The interesting question is whether this protocol is biased towards option *x*. This seems to be the case, but cannot be proven using $GLP^-$.

Figure 2: A second voting protocol $F_2$




# 4 Complete Information

In this section we assume that agents are commonly aware of each others' preferences, and that they commonly know that each agent selects the action that optimizes the utility. Under these assumptions the protocol will have an outcome that is compatible with backward induction. In order to determine this outcome we use a logic that allows reasoning about preferences, actions and backward induction. We call this logic *LPBI* for logic of preference and backward induction. In this section we present this logic and its semantics, and we sketch a completeness proof. The logic is similar to existing logics for reasoning about rational strategies, see for instance Bonanno [3], but puts more emphasis on preferences.

LPBI has essentially three parts : a logic of preference, a logic of actions in game trees and a logic of backward induction.

**Definition 8.** *Let* $con(L) = \{\bot, \varphi \rightarrow \psi | \varphi, \psi \in L\}$, *and* $pr(L) = \{\langle i \rangle \varphi \ , \langle \Sigma \rangle \varphi \ , \langle \Sigma^+ \rangle \varphi \ , \varphi \langle Pref \rangle_i \psi \ , \langle \mathcal{BI} \rangle \varphi \ , \langle \mathcal{BI}^* \rangle \varphi | \varphi \in L, i \in \Sigma\}$. *The language LPBI is the smallest language* $L$ *such that* $L = P \cup con(L) \cup bi(L)$.

All operators that are denoted with pointy brackets, like $\langle \Sigma \rangle$, should be read as existential operators. For each of these operators we define a dual operator $[\Sigma]\varphi = \neg\langle \Sigma \rangle \neg \varphi$. For the preference operators we do something similar. Note that the arguments change place: $\varphi[Pref]\psi = \neg(\psi \langle Pref \rangle \varphi)$.

We use *interpreted games* as models for this logic. These are interpreted extensive game forms enriched with preference relations and a backward induction function. The details of this function are skipped in the following definition. We assume that, for each history, it returns the action that is chosen by the backward induction algorithm at that history. The reader will find further details in [15]. Note that atomic positions are now interpreted on all nodes.

**Definition 9.** *An* interpreted extensive game *G is a tuple* $G = (F, \{\succeq_i\}_{i \in \Sigma}, bi)$, *where F in an interpreted game form except that* $\pi : H \rightarrow 2^P$ *returns the true atomic propositions of any history.* $\succeq_i \subseteq Z(H) \times Z(H)$ *is a preference relation, one for each agent. And* bi *is a function on H such that :*

$$bi(h) = \begin{cases} ha \in A(H,h) & \text{If } h \in H - Z(H) \\ h & \text{Otherwise} \end{cases}$$

To spell out as intuitively as possible the semantic of LPBI, we need a some more machinery.

**Definition 10.** *Let $H$ be a finite prefix-closed set of sequences, and $h \in H$. Let $E(H,h)$ be the smallest subset of $H$ such that $h \in E(H,h)$ and $h' \in E(H,h)$ implies that for all $a \in A(H,h')$ we have $h'a \in E(H,h)$. Similarly, define $bi^*(h)$ to be the smallest set $B$ such that $h \in B$ and for all $h' \in B$ we have $bi(h) \in B$.*

The truth conditions for LPBI go as follows.

$$
\begin{array}{lll}
G,h \models \psi\langle Pref\rangle_i\chi & \text{iff} & \exists(h',h'') \in (\succeq_i) \text{ such that } G,h' \models \varphi \text{ and } G,h'' \models \psi \\
G,h \models \langle i\rangle\varphi & \text{iff} & turn(h) = i \text{ and } \exists a \in A(H,h) \text{ such that } G,ha \models p \\
G,h \models \langle\Sigma\rangle\varphi & \text{iff} & \exists a \in A(H,h) \text{ such that } G,ha \models p \\
G,h \models \langle\Sigma^+\rangle\varphi & \text{iff} & \exists h' \in E(H,h) \text{ such that } G,h' \models p \\
G,h \models \langle\mathcal{BI}\rangle\varphi & \text{iff} & G,bi(h) \models p \\
G,h \models \langle\mathcal{BI}^*\rangle p & \text{iff} & \exists h' \in bi^*(h) \cap Z(H) \text{ such that } G,h' \models p
\end{array}
$$

There exists some previous work on reasoning about preferences. Our method of interpreting preferences over states is similar to work by Halpern [8] and Huang [10]. For a survey on preference logic, see Hansson [9].

The important fact about this logic is that it can be axiomatized.

**Theorem 1.** *There exists a complete axiomatisation for this logic*

*Proof.* The full list of axioms can be found in [15]. Here we only sketch the proof. The proof for the preference logic can be done using standard methods in modal logic. In order to express that the logic should be interpreted over *finite* game trees, we use the Löb axiom.

$$\vdash \langle\Sigma^+\rangle\varphi \rightarrow \langle\Sigma^+\rangle(\varphi \wedge \neg\langle\Sigma^+\rangle\varphi)$$

The completeness proof for this logic of action is essentially an adaptation of Blackburn and Meyer-Viol [2]. Finally one can characterize backward induction using several axioms, of which the most interesting is the following.

$$(\langle\mathcal{BI}\rangle[\mathcal{BI}^*]\varphi \wedge \langle i\rangle[\mathcal{BI}^*]\psi) \rightarrow (\varphi\langle Pref\rangle_i\psi)$$

□

Since this logic has a complete axiomatization, we can use theorem proving as a method for protocol verification. This gives us a verification method that is effective for all properties that can be expressed in this logic, for all situations that meet our assumptions. For $F_1$, we can use it to derive the optimal strategies for each agents from a set of preference and structural assumptions. Here is a sketch of this derivation:

Assumptions

$(A1)$ $\quad \vdash z[Pref]_A y, y[Pref]_A x$

$(A2)$ $\quad \vdash x[Pref]_B \neg x$

$(A3)$ $\quad \vdash y[Pref]_C \neg y$

$(A4)$ $\quad \vdash \bigwedge_{p\in P}(\langle A\rangle\langle B\rangle([\Sigma]\bot \wedge p) \wedge \langle A\rangle\langle C\rangle([\Sigma]\bot \wedge p))$

Conclusion

$(4)$ $\quad \vdash [\mathcal{BI}^*]y$

In this context, it is easy to see which players gets the best out of the vote. Here $C$ gets his first choice while $B$ and $A$ only obtain their second best. A similar analysis can be carried for $F_2$ and $F_3$, our two alternative protocols. With the same preference setting as the one used in $F_1$, we reach the same result as $y$ is elected is both of them.

# 5 Incomplete Information

It is convenient to assume that agents know each others' preferences, but it is not always realistic. In this section we assume that agents have limited, partial information about their own and each others preferences. This means that agents are faced with decision making under uncertainty, which is something different from game theory.

**Definition 11.** *Let $F = (\Sigma, H, turn, P, \pi)$ be an interpreted game form and $h \in H$. The reduced model $r(H,h)$ is defined as $r(H,h) = (\Sigma, H', turn', P, \pi')$ where $H' = \{h' | hh' \in H\}$ and $turn', \pi'$ are restrictions of the corresponding elements of $F$ to $H'$.*

**Definition 12.** *Let $F = (\Sigma, H, turn, P, \pi)$ be an interpreted game form, $\Gamma \subset \Sigma$ and $\varphi \in \mathcal{P}$. A history $j$ is a winning position iff $r(H,j) \models [\Gamma : \varphi]\Box\varphi$. The set of winning actions $w(j)$ is defined as $w(j) = \{a | ja \text{ is a winning position}\}$. Define $\sigma^r_\Gamma(\varphi)$ such that $\sigma^r_\Gamma(\varphi)(h) = w(h)$ if $w(h) \neq \emptyset$, and $\sigma^r_\Gamma(\varphi)(h) = A(H,h)$ otherwise.*

The definition above spells out what we consider a rational strategy $\sigma^r_\Gamma(\varphi)$ for a coalition $\Gamma$ that wants to achieve $\varphi$. The strategy is defined such that it selects actions $a$ that lead to winning positions. If that is not possible, it selects all actions. The idea is that coalition $\Gamma$ tries to guarantee $\varphi$ in all positions where it can guarantee $\varphi$. In the definition below we use this strategy for interpreting this logic.

| | | | |
|---|---|---|---|
| $F \models \bot$ | | never | |
| $F \models \varphi \to \psi$ | iff | not $T \models \varphi$ or $T \models \psi$ | |
| $F \models \Box\varphi$ | iff | $\forall h \in Z(H) : \pi(h) \models \varphi$ | where $(\Sigma, H, turn, \pi) = F$ |
| $F \models [\Gamma : \varphi]\Box\varphi$ | iff | $\exists \sigma_\Gamma \forall h \in Z(H) : \pi(h) \models \varphi$ | where $(\Sigma, H, turn, P, \pi) = Up(F, \sigma_\Gamma)$ |
| $F \models [\Gamma : \varphi]\psi$ | iff | $Up(F, \sigma^r_\Gamma(\varphi)) \models \psi$ | |

If one tries to interpret $[\Gamma : \varphi]\Box\varphi$ using the fifth and then the third clause, one gets the same results as when using line four. The fourth line is included in order to stress that in *GLP* all $GLP^-$ formulas are interpreted in the same way as in $GLP^-$. In a previous paper it has been proven that this semantics can be evaluated in polynomial time [16] in the size of the game tree (without efficient representation). Thus we know the following about the complexity of $GLP^-$ and *GLP*.

| | naive representation | linear representation |
|---|---|---|
| $GLP^-$ | P | PSPACE-complete |
| *GLP* | P | PSPACE-complete |

Complexity-wise, the step from $GLP^-$ to *GLP* does not bear any extra costs.

The main result of this section is that in *GLP* we can distinguish all three protocols and point out several small points of bias and/or asymmetry in each protocol.

In $F_1$, agent $A$ has a disadvantage, because if $A$ does not know what $B$ and $C$ prefer it cannot make a good decision. Indeed one can show that learning that $A$ wants $x$ does not tell one anything. However if everybody first learned that $B$ also wants $x$, then $x$ will be selected.

$$F_1 \not\models [A : x][B : x]\Box x$$
$$F_1 \models [B : x][A : x]\Box x$$

A closer look at the two other protocols reveals the same problem: agent $A$, because he has to choose first, has an information disadvantage. Just knowing what $A$ wants is not sufficient to draw any conclusions, because $A$ cannot guarantee anything on itself. One

must know what $A$ and some other agent commonly prefer, or what $A$ knows about $B$ or $C$'s preferences.

$$F_2 \not\models [A:x][B:x]\Box x \qquad\qquad F_2 \models [B:x][A:x]\Box x$$

One can conclude that if one assumes incomplete information, almost every protocol is asymmetric since one agent always has to move first. Apparently one has to allow agents to communicate before running a protocol like this.

Having looked at asymmetry in protocols, one wonders whether *GLP* can be used for showing bias. In particular we would like to know whether protocol $F_3^x = m(R_3^x)$ is biased towards $x$. The question is thus whether there is some formula $\varphi_v$ such that $\varphi_v[v \setminus x]$ holds while $\varphi_v[v \setminus y]$ does not. Below we give an example where $\varphi_v = [AB : \neg v][A : z]\Box z$.

$$F_3 \not\models [AB:\neg y][A:z]\Box z \qquad\qquad F_3 \models [AB:\neg x][A:z]\Box z$$

In order to see that the above statements about $F_3$ are true, one has to check that these are the updated models.

$$Up(F_3, \sigma^r_{AB}(\neg y)) = A \xrightarrow{a\in\{x,z\}} (B \xrightarrow{a} a)||(B \xrightarrow{b\in\{x,z\}\setminus\{a\}} C \xrightarrow{c\in\{a,b\}} c)$$
$$Up(F_3, \sigma^r_{AB}(\neg x)) = A \xrightarrow{a\in\{x,z\}} B \xrightarrow{a} a$$

After an update of $F_3$ with $[AB : \neg y]$ we end up with a model in which $A$ and $B$ are free to choose for either $x$ or $z$. If they disagree, $C$ can choose the final outcome, if they agree on some option $v$ then $v$ results. In the model updated with $[AB : \neg x]$, $A$ and $B$ must not disagree, otherwise $C$ can vote for option $x$. Thus in the updated model $A$ can choose for either $y$ or $z$, and then $B$ selects the same action as $A$. This is an example where there is a combination of symmetry and bias: If $A$ and $B$ do not want $X$, then $A$ has an advantage in this protocol. The result of this analysis is that one can find differences between the three example protocols. The logic *GLP* is thus a very precise logic.

# 6 Conclusion and Further Work

Verification of strategic properties is complicated, even for very small protocols. In this work we have looked at voting protocols that allow three agents to vote over three alternatives, and investigated what properties of those protocols can be expressed in different logics.

In order to present the alternative protocols, we use a representation form called *linear representation*. The format allows one to describe game forms in a more efficient way than a direct encoding. A complexity proof shows that this representation is indeed more compact.

We have first shown that a basic condition, that a coalition of sufficient size can enforce any outcome, can be expressed in many logics and is relatively easy to verify. In this paper we used a fragment of the game logic with preferences ($GLP^-$), but there are alternatives. However there are many different protocols that all satisfy these basic properties. Intuitively, these protocols should be distinguishable, so more sophisticated languages are needed to express their differences. We have focused on two approaches. If agents have complete information about each other's preferences, we have shown that the logic of preferences and backward induction (*LPBI*) can be used to analyze how the preferences determine the outcome. This logic does show differences between the various example protocols. Some more differences can be expressed if we don't assume that agents are fully aware of each others' preference. We have shown that full game logic with preferences

(*GLP*) can express that some voters have information disadvantage in incomplete information context : they need information about preferences in order to make a good decisions.

In this paper we have only looked at protocols without chance moves, and with perfect information. Further work could be focused on lifting one or both of these restrictions.

**Acknowledgement.** We would like to thank Johan van Benthem for his valuable comments.

# References

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 100–109, Florida, October 1997.

[2] P. Blackburn and W. Meyer-Viol. Linguistics, logic, and finite trees. *Logic Journal of the IGPL*, 2:3–29, 1994.

[3] G. Bonanno. The logic of rational play in games of perfect information. *Economics and Philosophy*, 7:37–65, 1991.

[4] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *Proceedings of the Uncertainty in Artificial Intelligence Conference (UAI), Edmonton, Canada.*, 2002.

[5] R. K. Dash, D. C. Parkes, and N. R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18:40–47, 2003.

[6] J. Doyle and R. Thomason. Background to qualitative decision theory. *AI Magazine*, pages 55–68, Summer 1999.

[7] V. Goranko. Coalition games and alternating temporal logics. In J. van Benthem (ed.), editor, *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII, 8-10 July, 2001)*, pages 259–272. Morgan Kaufmann, 2001.

[8] J. Halpern. Defining relative likelihood in partially-ordered preferential structures. *Studia Logica*, 7:1–24, 1997.

[9] S. O. Hansson. Preference logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic (Second Edition)*, volume 4, chapter 4, pages 319–393. Kluwer, 2001.

[10] Z. Huang. *Logics for Agents with Bounded Rationality*. PhD thesis, Universiteit van Amsterdam (ILLC), 1994.

[11] R. B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.

[12] R. Parikh. Social software. *Synthese*, 132:187–211, 2002.

[13] M. Pauly. *Logic for Social Software*. PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series 2001-10.

[14] W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(4):125–157, 2003.

[15] S. van Otterloo and O. Roy. Preference logic and backward induction. *Unpublished manuscript*, 2004.

[16] S. van Otterloo, W. van der Hoek, and M. Wooldridge. Preferences in game logics. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, New York, July 2004.