Bisimulations for Verifying Strategic Abilities with an Application to the ThreeBallot Voting Protocol

Francesco Belardinelli[a], Rodica Condurache[b], Cătălin Dima[b], Wojciech Jamroga[c], Michal Knapik[c]

[a]*Imperial College London, UK, and Université d'Evry, France*
[b]*LACL, Université Paris-Est Créteil, France*
[c]*Institute of Computer Science, Polish Academy of Sciences, Poland*

Abstract

We propose a notion of alternating bisimulation for strategic abilities under imperfect information. The bisimulation preserves formulas of ATL* for both the *objective* and *subjective* variants of the state-based semantics with imperfect information, which are commonly used in the modeling and verification of multi-agent systems. Furthermore, we apply the theoretical result to the verification of coercion-resistance in the ThreeBallot voting system, a voting protocol that does not use cryptography. In particular, we show that natural simplifications of an initial model of the protocol are in fact bisimulations of the original model, and therefore satisfy the same ATL* properties, including coercion-resistance. These simplifications allow the model-checking tool MCMAS to terminate on models with a larger number of voters and candidates, compared with the initial model.

Keywords: Alternating-time Temporal Logic, Bisimulations, Voting Protocols, Formal Verification.

1. Introduction

Formal languages for expressing strategic abilities of rational agents have witnessed a steady growth in recent years [1, 2, 3]. Among the most significant contributions we mention Alternating-time Temporal Logic [4, 5], (possibly enriched with strategy contexts [6]), Strategy Logic [7, 8], and Coalition Logic [9]. These languages allow to express that a group of agents has a strategy to enforce a certain outcome, regardless of the behavior of the other agents. That provides syntactical and semantic means to characterize winning conditions in multi-player games, notions of equilibrium (e.g. Nash), strategy-proofness, and so on [7, 10, 11, 12].

However, if logics for strategies are to be applied to the specification and verification of multi-agent systems (MAS) [13, 14, 15], they need to be coupled with efficient model checking techniques. Unfortunately, while in contexts of perfect information we benefit from tractable algorithms for model checking [5], the situation is rather different once we consider imperfect information. In contexts of imperfect information the complexity of the verification task ranges between Δ_2^P-completeness [16] to undecidability [17], depending on whether we assume perfect recall. In this setting, complementary model checking techniques are being investigated, in order to make the problem feasible in practice, including semantic [18, 19] and syntactic restrictions [20], as well as approximations [21].

Amongst these, model reduction seems one of the more promising paths. In particular, reductions by state- and action-space abstraction have proved to be a valuable tool for efficient verification [22, 23, 24], also in the context of strategic abilities [10, 25, 26, 27]. Within that approach, the "concrete" system S to be model-checked is abstracted into a "simpler" model S^A, which typically contains less states and transitions and therefore can be easier to verify. Then, the verification result is transferred from abstraction S^A to the concrete S by virtue of some preservation result. Normally, preservation is guaranteed by proving that abstraction S^A is *similar* or *bisimilar* to S. (Bi)simulations are a powerful tool to analyze the expressiveness of modal languages, starting with van Benthem's characterisation of modal logic as the bisimulation-invariant fragment of first-order logic [28]. However, (bi)simulations are a lot less understood in logics for strategies, where they have been studied mostly in the contexts of perfect information scenarios [29, 30, 31].

In this paper, we advance the state-of-the-art by introducing simulations and bisimulations for alternating-time temporal logic under imperfect information. We prove that these (bi)simulations preserve the interpretation of formulas for the whole syntax of ATL*, when interpreted with imperfect information and imperfect recall, for both the objective and subjective variants of the semantics [2, 32]. Most interestingly for MAS verification, we apply the (bi)simulations to the model reduction of a class of electronic voting protocols without encryption.

Electronic voting is often considered as an attractive alternative to paper-based elections and referendums due to a number of advantages: increased accessibility, availability, voter turnout, cost-efficiency, and usability, as well as speed and accuracy of the voting, counting and publication processes. However, electronic voting poses a number of challenges: resistance to coercion and other types of fraud, secrecy, anonymity, verifiability, democracy (the right to vote at most once), accountability, etc. Those threats exist also in paper voting, but the use of technology magnifies their scope and potential impact. There are also other issues, specific to electronic voting, such as limited access to technology and complex interaction between technology and public understanding and trust in the procedure [33, 34]. In this paper, we focus on the property of *coercion resistance* that captures the voters' ability to choose freely how they vote: whatever course of action the coercer adopts, the voter always has a strategy to vote as they intend while appearing to comply with the coercer's requirements.

An increasing amount of research has focused recently on the verification of many of these properties for various types of voting protocols [35, 36]. The frameworks used for modeling and verifying security properties of voting protocols include, to mention only a few, process calculi such as the *applied π-calculus* or *CSP* [37, 38, 39], rewriting-based approaches [40, 41, 42], approaches based on flat transition systems etc.

In this paper, we show how our bisimulation can be used to obtain significant model reductions for some voting protocols. In consequence, we develop a verification procedure for those voting protocols, based on the multi-agent approach. The main advantage of multi-agent logics is the provision of a unified specification language for a vast array of properties. Due to that feature, the logics can for instance serve to disambiguate the variety of informal intuitions behind coercion resistance that are found in the literature, cf. [43]. Moreover, multi-agent logics allow for a more flexible and expressive specification of variants of coercion-resistance, involving explicit references to the dynamics of attacker's knowledge, as well as the (non)existence of strategies suitable for the attacker and/or the voter. The same applies to specification of other important properties, such as individual verifiability, end-to-end (universal) verifiability, and accountability.

Here, we focus on a particular formalization of coercion resistance, and verify it for a simplified version of the ThreeBallot voting protocol [44, 45]. Our first results in this line already allow for verification of models with a larger number of voters and candidates compared to the approach based on process calculi in [46, 47]. Even more importantly, our experiments show that bisimulation-based reductions can turn verification from a utterly difficult task to something feasible in practice.

1.1. Structure of the Paper and Technical Contributions

The article is structured as follows. In Section 2, we introduce the syntax and semantics of ATL* interpreted under the assumption of imperfect information and imperfect recall. In Section 3, we propose the novel relations of simulation and bisimulation for the setting. Moreover, we prove the main theoretical result of this paper, namely that the bisimulation preserves the interpretation of ATL* formulas. In Section 4, we show that our notion of bisimulation does not enjoy the Hennessy-Milner property, and we discuss some necessary conditions for two models to be logically equivalent. The results highlight some interesting features of the current semantics of ATL* with imperfect information, and might point to a novel semantics for logics of strategic ability. Further, in Section 5, we present our case study based on the ThreeBallot voting protocol. We formalize ThreeBallot as a game structure, and provide two abstractions of the protocol. Then, we point out that the abstractions are indeed bisimilar with the original game structure. Finally, in Section 6, we evaluate the gains in terms of verification time, by comparing model checking of the abstractions to model checking of the original model. We conclude in Section 7, and indicate some future directions of research.

Previous version of the material. This work extends and revises the results presented in the conference paper [48]. The main technical differences are as follows. First, our results are now given for the whole language of ATL*, while [48] considered only its restricted fragment ATL. Secondly, we correct a mistake in the main preservation result. In the previous version, we erroneously used a weaker notion of bisimulation, which actually does not preserve even formulas of ATL. We discuss the details, and point out some differences between the two definitions in Section 3.3. Thirdly, we report completely new theoretical research in Section 3.4 (computational complexity of checking for bisimulation) and Section 4 (Hennessy-Milner property and necessary conditions for logical equivalence of models). Finally, we revise the implementation of our ThreeBallot models, and present entirely new experimental results. In particular, we extend our model checking experiments to the imperfect information semantics of ATL, for which the bisimulation-based reductions were in fact designed.

Regarding the presentation, we add a running example, and streamline the introduction of the bisimulation, in order to make the concept easier to understand.

1.2. Related Work

The literature on both logics for strategies and the formal verification of voting protocols is extensive and rapidly growing. Hereafter we only consider the works most closely related to the present contribution.

Bisimulations for ATL. An in-depth study of model equivalences induced by various temporal logics appears in [30]. Bisimulations for ATL* with perfect information have been introduced in [29]. Since then there have been various attempts to extend these to more expressive languages (including Strategy Logic recently [49]), as well as to contexts of imperfect information [31, 50]. In [50, 51] non-local model equivalences for ATL with imperfect information have been put forward. However, these works do not deal with the imperfect information/imperfect recall setting here considered, nor do they provide a local account of bisimulations. Finally, in [48] we provided a different notion of bisimulation for ATL only with imperfect information and imperfect recall. We remarked in the introduction that such definition was flawed and provided a counterexample in Section 3.3. The present contribution is also aimed at rectifying this result.

Verification of Voting Protocols. Our work is inspired by recent contributions on the verification of voting protocols, mostly by using the π-calculus and *CSP* [37, 38, 39]. Some existing attempts at verification of e-voting protocols follow this approach by using one of the security verifiers [52, 53], or even a general equivalence checker for a process algebra [46, 47, 54], where privacy-type and anonymity properties of the protocols are verified by using CSP. An important strand in verification of voting protocols is based on specifications in first-order logic or linear logic. The verification is done by means of theorem proving [55, 56, 57, 58] or bounded model checking (BMC) [35, 59, 60, 61]. In particular [61] presents a BMC-based analysis of risk-limiting audits and [58] applies theorem proving to automated verification of the Selene voting protocol. In [35] the authors define two semantic criteria for single transferable vote (STV) schemes, then show how BMC and SMT solvers can be used to check whether these criteria are met. In [47] the authors construct CSP models of the ThreeBallot system and use them to produce an automated formal analysis of their anonymity properties. Finally, some challenges and solutions for verification of end-to-end verifiable systems were addressed in [62, 63].

An issue that can be identified with many of the above approaches is that the description of the system, and the property to be verified, are not clearly distinguished. We emphasize that multi-agent logics allow for a clear separation of the two. Moreover, they provide a wider variety of properties, including ones that are related to the existence of the attacker's strategies. Last but not least, they give reasonable hope for interesting verification performance. In our experiments, we have been able to model-check ThreeBallot models with 5 voters and 2 candidates, or 4 candidates and 3 voters, while e.g. the results in [47] are provided for at most 3 voters and 2 candidates. In this respect, the closest work to ours is the recent paper [64], presenting some experimental results for verification of the voting protocol SELENE, based on the multi-agent logic ATL_{ir}.

2. The Formal Setting

In this section, we introduce the syntax of the Alternating-time Temporal Logic ATL* [5], and present its semantics for agents with imperfect information and imperfect recall, defined on imperfect information concurrent game structures (iCGS). The assumption of *imperfect information* means that agents can only partially observe the global state of the system. Thus, they have to make their decisions, and execute their strategies, with only partial knowledge about the current situation. The assumption of *imperfect recall* means that they do not necessarily memorize all of their past observations. This does not mean that the agents have no memory at all. However, an agent's recall of the past must be encapsulated in the current local state of the agent.

The formal definitions and notation follow [17].

2.1. Concurrent Game Structures

Concurrent game structures have originally been introduced in [5] in a perfect information setting. Here we consider their version for contexts of imperfect information [65].

Definition 1. *A* concurrent game structure with imperfect information*, or iCGS, is a tuple* $\mathcal{G} = \langle Ag, AP, S, s_0, Act, \{\sim_i\}_{i \in Ag}, d, \rightarrow, \pi \rangle$ *such that*

- *Ag is a nonempty and finite set of* agents*. Subsets $A \subseteq Ag$ of agents are called* coalitions*.*

- *AP is a set of* atomic propositions*, or atoms.*

- *S is a non-empty set of* states *and $s_0 \in S$ is the* initial state *of $\mathcal{G}$.*

- *Act is a finite non-empty set of* actions*. A tuple $\vec{a} = (a_i)_{i \in Ag} \in Act^{Ag}$ is called a* joint action*.*

- *For every agent $i \in Ag$, $\sim_i$ is an equivalence relation on S, called the* indistinguishability relation *for i.*

- *$d : Ag \times S \to (2^{Act} \setminus \{\varnothing\})$ is the* protocol function*, satisfying the property that, for all states $s, s' \in S$ and any agent i, $s \sim_i s'$ implies $d(i,s) = d(i,s')$. That is, the same (non-empty) set of actions is available to agent i in indistinguishable states.*

- *$\to \subseteq S \times Act^{Ag} \times S$ is the* transition relation *such that, for every state $s \in S$ and joint action $\vec{a} \in Act^{Ag}$, $(s, \vec{a}, s') \in \to$ for some state $s' \in S$ iff $a_i \in d(i,s)$ for every agent $i \in Ag$. We normally write $s \xrightarrow{\vec{a}} r$ for $(s, \vec{a}, r) \in \to$.*

- *$\pi : S \to 2^{AP}$ is the* state-labeling function*.*

By Def. 1 at any given state s, every agent $i \in Ag$ can perform the enabled actions in $d(i,s)$. A joint action $\vec{a}$ fires a transition from state s to some state s' only if each a_i is enabled for agent i in s. Further, every agent i is equipped with an indistinguishability relation $\sim_i$, with $s \sim_i s'$ meaning that i cannot tell state s from state s', i.e., agent i possesses the same information, makes the same observation in the two states. In particular, the same actions are enabled in indistinguishable states.

Example 1. *Consider a two-stage voting system with a simple anti-coercion mechanism. In the first stage, the voter casts her vote for one of the candidates; we assume for simplicity that there are only two candidates in the election. In the second stage, the voter has the option to send a signal – by an independent communication channel – that flips her vote to the other candidate. After that, the election official publishes the result of the election (action* pub*). However, a dishonest official can also decide not to publish the outcome (action* np*). An iCGS modeling the scenario for a single voter (represented by agent* 1*) is depicted in Figure 1. The election official is represented by agent* 2*; the dotted lines indicate that the official does not directly*

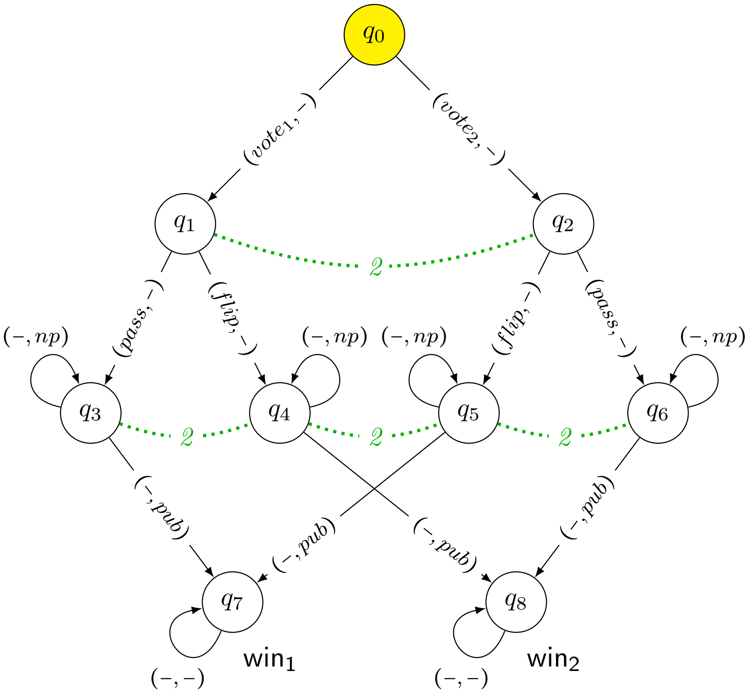


Figure 1: Simple model of 2-stage voting. Dotted lines represent indistinguishability for agent 2 (the election official).

see the choices made by the voter. Note that this toy model is in fact a turn-based game, i.e., in every state only a single agent has a choice of decision while the other agent is passive.

Runs. Given an iCGS $\mathcal{G}$ as above, a *run* is a finite or infinite sequence $\lambda = s_0 \vec{a}_0 s_1 \dots$ in $((S \cdot Act^{Ag})^* \cdot S) \cup (S \cdot Act^{Ag})^\omega$ such that for every $j \geqslant 0$, $s_j \xrightarrow{\vec{a}_j} s_{j+1}$. Given a run $\lambda = s_0 \vec{a}_0 s_1 \dots$ and $j \geqslant 0$, $\lambda[j]$ denotes the $j+1$-th state s_j in the sequence; while $\lambda_{\geqslant j}$ denotes run $s_j \vec{a}_j s_{j+1} \dots$ starting from $\lambda[j]$. We denote by $Run(\mathcal{G})$ the set of all runs in iCGS $\mathcal{G}$. For a coalition $A \subseteq Ag$ of agents, a *joint A-action* denotes a tuple $\vec{a}_A = (a_i)_{i \in A} \in Act^A$ of actions, one for each agent in A. For coalitions $A \subseteq B \subseteq Ag$ of agents, a joint A-action $\vec{a}_A$ *is extended* by a joint B-action $\vec{b}_B$, denoted $\vec{a}_A \sqsubseteq \vec{b}_B$, if for every $i \in A$, $a_i = b_i$. Also, a joint A-action $\vec{a}_A$ is *enabled* at state $s \in S$ if for every agent $i \in A$, $a_i \in d(i, s)$.

Epistemic neighbourhoods. Given a coalition $A \subseteq Ag$ of agents, the *collective knowledge relation* $\sim_A^E$ is defined as $\bigcup_{i \in A} \sim_i$, while the *common* *knowledge relation* $\sim_A^C$ is the transitive closure $(\bigcup_{i\in A} \sim_i)^+$ of $\sim_A^E$. Then, $C_A^{\mathcal{G}}(q) = \{q' \in S \mid q' \sim_A^C q\}$ is the *common knowledge neighbourhoods* (CKN), of state q for coalition A in the iCGS $\mathcal{G}$. We will omit the superscript $\mathcal{G}$ whenever it is clear from the context.

Uniform strategies. We now recall a notion of strategy adapted to iCGS with imperfect information [65].

Definition 2. *A* (uniform) strategy *for an agent $i \in Ag$ is a function $\sigma : S \to Act$ that is compatible with d and $\sim_i$, i.e.,*

- *for every state $s \in S$, $\sigma(s) \in d(i, s)$;*
- *for all states $s, s' \in S$, $s \sim_i s'$ implies $\sigma(s) = \sigma(s')$.*

By Def. 2 a strategy has to be uniform in the sense that in indistinguishable states it must return the same action. Such strategies are also known as *observational* in the literature on game theory and control theory. Note that in this paper we use memoryless strategies, whereby only the current state determines the action to perform. This choice is dictated by the application in hand, namely voting protocols, where the unbounded recall is not needed. Namely, the information available to the agents is fully encoded in global states together with help of indistinguishability relations[1]. Note that the finite memory of a given depth of recall can be obtained in practical setting by introducing an adequate number of special variables, i.e., dimensions in iCGSs whose states are valuations of variables.

Perfect recall strategies with imperfect information can be defined similarly, as memoryless strategies on tree unfoldings of iCGS. We leave this extension for future work.

A strategy for a coalition A of agents is a set $\sigma_A = \{\sigma_a \mid a \in A\}$ of strategies, one for each agent in A. Given coalitions $A \subseteq B \subseteq Ag$, a strategy σ_A for coalition A, a state $s \in S$, and a joint B-action $\vec{b}_B \in Act^B$ that is enabled at s, we say that $\vec{b}_B$ is *compatible with σ_A (in s)* whenever $\sigma_A(s) \sqsubseteq \vec{b}_B$. For states $s, s' \in S$ and strategy σ_A, we write $s \xrightarrow{\sigma_A(s)} r$ if $s \xrightarrow{\vec{a}} r$ for some joint action $\vec{a} \in Act^{Ag}$ that is compatible with σ_A.

There exist two alternative semantics of strategic operators under imperfect information, corresponding to two different notions of success for a

[1] Therefore memoryless strategies already encode the agent's memory of all her past observations.

strategy. The idea of *subjective* ability [66, 65] requires that a winning strategy must succeed from all the states that the coalition considers possible in the initial state of the play. The alternative notion of *objective* ability [2] assumes that it suffices for the strategy to succeed from the initial state alone. Accordingly, we define two notions of the *outcome* of a strategy σ_A at state s, corresponding to the objective and the subjective interpretation of alternating-time operators. Fix a state s and a strategy σ_A for coalition A.

1. The set of *objective outcomes of σ_A at s* is defined as $out^{\mathcal{G}}_{obj}(s, \sigma_A) = \big\{\lambda \in Run(\mathcal{G}) \mid \lambda[0] = s \text{ and for all } j \geqslant 0, \lambda[j] \xrightarrow{\sigma_A(\lambda[j])} \lambda[j+1]\big\}$.
2. The set of *subjective outcomes of σ_A at s* is defined as $out^{\mathcal{G}}_{subj}(s, \sigma_A) = \bigcup_{i \in A, s' \sim_i s} out^{\mathcal{G}}_{obj}(s', \sigma_A)$.

2.2. Alternating-Time Temporal Logic.

We now introduce the alternating-time temporal logic ATL* to be interpreted on iCGS.

Definition 3 (ATL*)**.** *The language of ATL* is formally defined by the following grammar, where $p \in AP$ and $A \subseteq Ag$:*

$$\begin{array}{rcl} \varphi & ::= & p \mid \neg\varphi \mid \varphi \to \varphi \mid \langle\!\langle A \rangle\!\rangle \psi \\ \psi & ::= & \varphi \mid \neg\psi \mid \psi \to \psi \mid X\psi \mid \psi U \psi \end{array}$$

Formulas φ are called state formulas of ATL**, or simply* formulas of ATL**. Formulas ψ are sometimes called* path formulas of ATL**.*

The ATL* operator $\langle\!\langle A \rangle\!\rangle$ intuitively means that 'the agents in coalition A have a (collective) strategy to achieve . . . ', where the goals are LTL formulas built by using operators 'next' X and 'until' U. We define A-formulas as the formulas in ATL* in which A is the only coalition appearing in ATL* modalities.

Traditionally, ATL* under imperfect information has been given either state-based or history-based semantics, and several variations have been considered on the interpretation of strategy operators. Here we present both the objective and subjective variants of the state-based semantics with imperfect information and imperfect recall.

Definition 4. *Given an iCGS $\mathcal{G}$, a state formula φ, and path formula ψ, the* subjective *(resp.* objective*) satisfaction of φ at state s and of ψ in path λ, denoted $(\mathcal{G},s)\models_x\varphi$ and $(\mathcal{G},\lambda)\models_x\psi$ for $x=subj$ (resp. $x=obj$), is defined recursively as follows:*

$$\begin{array}{ll}
(\mathcal{G},s)\models_x p & \textit{iff } p\in\pi(s)\\
(\mathcal{G},s)\models_x \neg\varphi & \textit{iff } (\mathcal{G},s)\not\models_x \varphi\\
(\mathcal{G},s)\models_x \varphi\rightarrow\varphi' & \textit{iff } (\mathcal{G},s)\not\models_x \varphi \textit{ or } (\mathcal{G},s)\models_x\varphi'\\
(\mathcal{G},s)\models_x \langle\!\langle A\rangle\!\rangle\psi & \textit{iff for some } \sigma_A, \textit{ for all } \lambda\in out_x^{\mathcal{G}}(s,\sigma_A), (\mathcal{G},\lambda)\models_x\psi\\
(\mathcal{G},\lambda)\models_x \varphi & \textit{iff } (\mathcal{G},\lambda[0])\models_x\varphi\\
(\mathcal{G},\lambda)\models_x \neg\psi & \textit{iff } (\mathcal{G},\lambda)\not\models_x \psi\\
(\mathcal{G},\lambda)\models_x \psi\rightarrow\psi' & \textit{iff } (\mathcal{G},\lambda)\not\models_x \psi \textit{ or } (\mathcal{G},\lambda)\models_x\psi'\\
(\mathcal{G},\lambda)\models_x X\psi & \textit{iff } (\mathcal{G},\lambda_{\geqslant 1})\models_x\psi\\
(\mathcal{G},\lambda)\models_x \psi U\psi' & \textit{iff for some } j\geqslant 0, (\mathcal{G},\lambda_{\geqslant j})\models_x\psi'\\
 & \quad\textit{and for all } k, 0\leqslant k<j \textit{ implies } (\mathcal{G},\lambda_{\geqslant k})\models_x\psi
\end{array}$$

Example 2. *Consider the simple voting model in Example 1. Clearly, the voter cannot enforce a win of any candidate, since the election official might block the publication of the outcome: $\mathcal{G},q_0\models\neg\langle\!\langle 1\rangle\!\rangle\mathrm{F}\mathsf{win}_1\wedge\neg\langle\!\langle 1\rangle\!\rangle\mathrm{F}\mathsf{win}_2$. On the other hand, she can prevent any given candidate from being elected, by casting her vote for the other candidate: $\mathcal{G},q_0\models\langle\!\langle 1\rangle\!\rangle\mathrm{G}\neg\mathsf{win}_1\wedge\langle\!\langle 1\rangle\!\rangle\mathrm{G}\neg\mathsf{win}_2$. Finally, the voter and the official together can get any arbitrary candidate elected: $\mathcal{G},q_0\models\langle\!\langle 1,2\rangle\!\rangle\mathrm{F}\mathsf{win}_1\wedge\langle\!\langle 1,2\rangle\!\rangle\mathrm{F}\mathsf{win}_2$. Note that the truth value of formulae in q_0 does not depend on whether we use the subjective or the objective semantics.*

The difference between the two semantics can be demonstrated, e.g., in state q_3. In that state, the election official has the objective ability to make candidate 1 win ($\mathcal{G},q_3\models_{obj}\langle\!\langle 2\rangle\!\rangle\mathrm{F}\mathsf{win}_1$) by playing pub *regardless of anything. On the other hand, there is no uniform strategy for 2 that guarantees the same from all the states indistinguishable from q_3 (i.e., from q_3,q_4,q_5,q_6). In consequence, $\mathcal{G},q_3\models_{subj}\neg\langle\!\langle 2\rangle\!\rangle\mathrm{F}\mathsf{win}_1$.*

The individual knowledge operator K_i can be added to the syntax of ATL* with the following semantics:

$$(\mathcal{G},s)\models_x K_i\varphi \text{ iff for all } s'\in S, s'\sim_i s \text{ implies } (\mathcal{G},s')\models_x\varphi$$

By considering the subjective interpretation of ATL*, this operator can be derived: $(\mathcal{G},s)\models_{subj} K_i\varphi$ iff $(\mathcal{G},s)\models_{subj}\langle\!\langle i\rangle\!\rangle\varphi U\varphi$. There exists no such definition for the knowledge operator in ATL* with the objective semantics. The latter is easy to see, as the objective semantics of $\langle\langle i \rangle\rangle$ in $(\mathcal{G}, s)$ does not refer in any way to the states epistemically indistinguishable from s.

3. Simulations and Bisimulations

In this section we introduce a notion of bisimulation for imperfect information concurrent game structures. Then, we show that it preserves the meaning of formulas in ATL*, when interpreted under the assumptions of imperfect information and imperfect recall, introduced in Section 2.

3.1. Bisimulation for ATL* with Imperfect Information and Imperfect Recall

We start with some auxiliary notions and definitions.

Partial strategies and outcome states. A *partial (uniform) strategy* for agent $i \in Ag$ is a partial function $\sigma : S \to Act$ such that for each $s, s' \in S$, if $s \sim_i s'$ then $\sigma(s) = \sigma(s')$. We denote the domain of the partial strategy σ as $dom(\sigma)$. Given a coalition $A \subseteq Ag$, a *partial strategy for A* is a tuple $(\sigma_i)_{i\in A}$ of partial strategies, one for each agent $i \in A$. The set of partial uniform strategies for A is denoted $PStr_A$. Given set $Q \subseteq S$ of states and coalition $A \subseteq Ag$, we denote by $PStr_A(Q)$ the set of partial uniform strategies whose domain is Q:

$$PStr_A(Q) = \left\{(\sigma_i)_{i\in A} \in PStr_A \mid dom(\sigma_i) = Q \text{ for all } i \in A\right\}$$

Additionally, given a (total or partial) strategy σ_A and a state $q \in dom(\sigma_A)$, define the set of *successor states of q by σ* as $succ(q, \sigma_A) = \{s \in S \mid q \xrightarrow{\sigma_A(q)} s\}$, and put $succ(\sigma_A) = succ(dom(\sigma_A), \sigma_A) = \bigcup_{s\in dom(\sigma_A)} succ(s, \sigma_A)$.

Strategy simulators. Let $\mathcal{G}, \mathcal{G}'$ be two iCGS. A *simulator of partial strategies for coalition A* from $\mathcal{G}$ into $\mathcal{G}'$ is a set ST of functions $ST_{Q,Q'} : PStr_A(Q) \to PStr_A(Q')$ for some subsets $Q \subseteq S$ and $Q' \subseteq S'$. Intuitively, every $ST_{Q,Q'}$ maps each partial strategy σ_A defined on set Q in the iCGS $\mathcal{G}$ into a "corresponding" strategy σ'_A defined on Q' in $\mathcal{G}'$. Typically, we will map strategies between the common knowledge neighborhoods of "bisimilar" states in $\mathcal{G}$ and $\mathcal{G}'$. We formalize this idea as follows. Let $R \subseteq S \times S'$ be some relation between states in $\mathcal{G}$ and $\mathcal{G}'$. A *simulator of partial strategies for coalition A with respect to relation R* is a family ST of functions $ST_{C_A(q),C'_A(q')} : PStr_A(C_A(q)) \to PStr_A(C'_A(q'))$ such that $q \in S$, $q' \in S'$, and qRq'. Note that, since the functions are indexed by equivalence classes of states, the following additional property is automatically guaranteed:

- for every $r \in S$, $r' \in S'$, if $r \in C_A(q)$, $r' \in C'_A(q')$, and rRr', then $ST_{C_A(q),C'_A(q')} = ST_{C_A(r),C'_A(r')}$.

Simulation and bisimulation. We can now present our notions of simulation and bisimulation on iCGS.

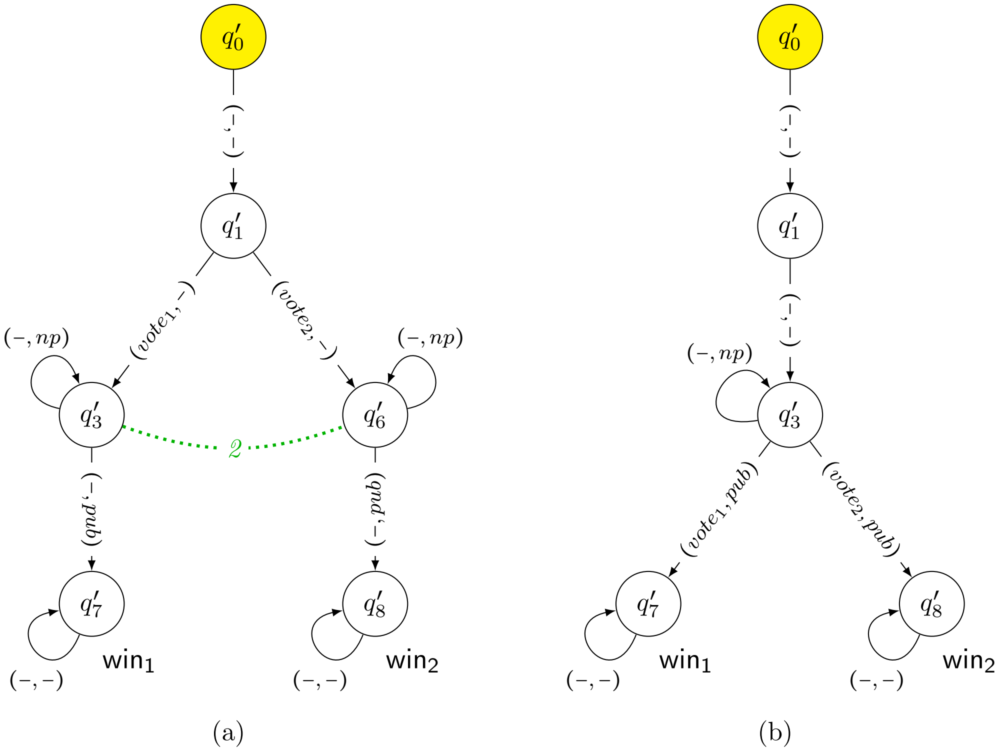
Definition 5 (Simulation). *Let $\mathcal{G} = \langle Ag, AP, S, s_0, Act, \{\sim_i\}_{i\in Ag}, d, \rightarrow, \pi\rangle$ and $\mathcal{G}' = \langle Ag, AP, S', s'_0, Act', \{\sim'_i\}_{i\in Ag}, d', \rightarrow', \pi'\rangle$ be two iCGS defined on the same sets Ag of agents and AP of atoms. Let $A \subseteq Ag$ be a coalition of agents. A relation $\Rrightarrow_A \subseteq S \times S'$ is a* simulation for A *iff*

1. *There exists a simulator ST of partial strategies for A w.r.t. $\Rrightarrow_A$, such that $q \Rrightarrow_A q'$ implies that:*
   (a) *$\pi(q) = \pi'(q')$;*
   (b) *for every $i \in A$, $r' \in S'$, if $q' \sim'_i r'$ then for some $r \in S$, $q \sim_i r$ and $r \Rrightarrow_A r'$.*
   (c) *For every states $r \in C_A(q)$, $r' \in C'_A(q')$ such that $r \Rrightarrow_A r'$, for every partial strategy $\sigma_A \in PStr_A(C_A(q))$, and every state $s' \in succ(r', ST(\sigma_A))$, there exists a state $s \in succ(r, \sigma_A)$ such that $s \Rrightarrow_A s'$.*
2. *If $q_1 \Rrightarrow_A q'$ and $q_2 \Rrightarrow_A q'$, then $C_A(q_1) = C_A(q_2)$.*

A relation $\Longleftrightarrow_A$ is a bisimulation *iff both $\Longleftrightarrow_A$ and its converse $\Longleftrightarrow_A^{-1} = \{(q', q) \mid q \Longleftrightarrow_A q'\}$ are simulations.*

By Def. 5 if state q' *simulates* q, i.e., $q \Rrightarrow_A q'$, then 1.(a) q and q' agree on the interpretation of atoms; 1.(b) q simulates the epistemic transitions from q', that is, information encoded in states is preserved by (bi)similarity; and 1.(c) for every partial strategy σ_A, defined on the common knowledge neighborhood $C_A(q)$, we are able to find some partial strategy $ST(\sigma_A)$ (the same for all states in $C_A(q)$) such that the transition relations $\xrightarrow{ST(\sigma_A)}$ and $\xrightarrow{\sigma_A}$ commute with the simulation relation $\Rrightarrow_A$. Moreover, (2) the simulation relation is injective when lifted to common knowledge neighborhoods.

Example 3. *Let us go back to the simple two-stage voting from Example 1 and Figure 1. We observe that the model is bisimilar for $A = \{2\}$ to the one in Figure 2(a). The bisimulation connects q_0 with q'_0; q_1 and q_2 with q'_1; q_3 and q_4 with q'_3; q_5 and q_6 with q'_6; q_7 with q'_7; and q_8 with q'_8. As we will see in Section 3.2, this implies that the abilities of the election official in both models must be exactly the same.*



(a)

(b)

Figure 2: (a) Even simpler model of 2-stage voting (from the point of view of the election official). (b) Yet simpler model of 2-stage voting.

The iCGS can be reduced even further, and still retain the same abilities of the singleton coalition $\{2\}$, see Figure 2(b). We leave it to the interested reader to find the bisimulation for $\{2\}$ between the two models.

Remark 6. *Technically, condition (2) in Def. 5 is required as is shown in the end of this section by a counterexample (cf. Subsection 3.3). The extra property (2) corrects the statement and proof of Theorem 9 in [48]. Notice that, for every bisimulation $\Longleftrightarrow_A$ defined on $\mathcal{G} = \mathcal{G}'$, condition (2) says that two bisimilar states must lie in the same common knowledge neighborhood, which is a natural constraint considering that two bisimilar states should satisfy the same formulas in ATL*.*

3.2. Preservation Theorem

In order to show that bisimilar states satisfy the same formulas in ATL*, we prove the following auxiliary result. Hereafter, runs $\lambda \in S^+$, $\lambda' \in S'^+$ are

A-bisimilar, or $\lambda \Longleftrightarrow_A \lambda'$ iff for every $i \geqslant 0$, $\lambda[i] \Longleftrightarrow_A \lambda'[i]$

Proposition 7. *If $q \Rrightarrow_A q'$ then for every strategy σ_A, there exists a strategy σ'_A such that*

(∗) for every run $\lambda' \in out_x^{\mathcal{G}'}(q', \sigma'_A)$, for $x \in \{subj, obj\}$, there exists an infinite run $\lambda \in out_x^{\mathcal{G}}(q, \sigma_A)$ such that $\lambda \Rrightarrow_A \lambda'$.

Proof. First of all we define the sequence $\big(dom^n(\sigma_A)\big)_{n\in\mathbb{N}}$, of sets of states in $\mathcal{G}$ such that $s \in dom^n(\sigma_A)$ iff s can be reached in at most n steps from $C_A(q)$ by applying actions compatible with strategy σ_A:

$$\begin{aligned} dom^0(\sigma_A) &= C_A(q) \\ dom^{n+1}(\sigma_A) &= dom^n(\sigma_A) \cup \bigcup \Big\{ C_A(r) \mid r \in \bigcup_{s \in dom^n(\sigma_A)} succ(s, \sigma_A) \Big\} \end{aligned}$$

Also, we denote by σ_A^n the partial strategy resulting from restricting σ_A to $dom^n(\sigma_A)$.

We now define inductively a sequence $(\overline{\sigma}_A^n)_{n\in\mathbb{N}}$ of partial strategies in $\mathcal{G}'$ such that $dom(\overline{\sigma}_A^n) \subseteq dom(\overline{\sigma}_A^{n+1})$ for every $n \in \mathbb{N}$. These partial strategies will be constructed using strategy σ_A and mapping ST from point (1) in Def. 5. The desired sequence of partial strategies $\overline{\sigma}_A^n$, for $n \geqslant 1$, is defined as follows:

1. $dom(\overline{\sigma}_A^0) = C'_A(q')$ and $dom(\overline{\sigma}_A^{n+1}) = dom(\overline{\sigma}_A^n) \cup \{r' \mid C'_A(r') \cap succ(\overline{\sigma}_A^n) \neq \varnothing\}$;
2. for all $r' \in dom(\overline{\sigma}_A^0)$, $\overline{\sigma}_A^0(r') = ST_{C_A(q), C'_A(q')}(\sigma_A^0)(r')$;
3. for all $r' \in dom(\overline{\sigma}_A^{n+1})$,

$$\overline{\sigma}_A^{n+1}(r') = \begin{cases} \overline{\sigma}_A^n(r') & \text{for } r' \in dom(\overline{\sigma}_A^n) \\ ST_{C_A(r), C'_A(r')}(\sigma_A^{n+1})(r') & \text{for } r' \notin dom(\overline{\sigma}_A^n), C'_A(r') \cap succ(\overline{\sigma}_A^n) \neq \varnothing, \\ & \text{and } r \text{ is any state in } \mathcal{G} \text{ s.t. } r \Rrightarrow_A r' \end{cases}$$

We give first a number of properties for this sequence of partial strategies. First note that, in the last line of the definition of $\overline{\sigma}_A^{n+1}(r')$, r can indeed be chosen arbitrarily, since, by point (2) in Def. 5, whenever $r_1 \Rrightarrow_A r'$ and $r_2 \Rrightarrow_A r'$ we must have $C_A(r_1) = C_A(r_2)$, which implies that $ST_{C_A(r_1), C'_A(r')} = ST_{C_A(r_2), C'_A(r')}$.

Further, if $C'_A(r') \cap dom(\overline{\sigma}_A^n) \neq \varnothing$ then $C'_A(r') \subseteq dom(\overline{\sigma}_A^n)$. This is trivial for $n = 0$. As for the induction step, if $C'_A(r') \cap dom(\overline{\sigma}_A^{n+1}) \neq \varnothing$, but $C'_A(r') \nsubseteq$

$dom(\overline{\sigma}^n_A)$, then by induction hypothesis we have $C'_A(r') \cap dom(\overline{\sigma}^n_A) = \emptyset$. Further, by definition of $\overline{\sigma}^{n+1}_A$ we obtain $C'_A(r') \cap succ(\overline{\sigma}^n_A) \neq \emptyset$, which is the case for all $r'' \in C'_A(r')$. Hence $C'_A(r') \subseteq dom(\overline{\sigma}^{n+1}_A)$.

We now show that, whenever we take some $u' \in C'_A(r') \cap succ(\overline{\sigma}^n_A) \neq \emptyset$ with $r' \notin dom(\overline{\sigma}^n_A)$, we have $\{u \in dom^{n+1}(\sigma_A) \mid u \Rrightarrow_A u'\} \neq \emptyset$ and, by the induction hypothesis $\overline{\sigma}^{n+1}_A$ is uniform over $dom(\overline{\sigma}^n_A)$. To see the former, whenever $u' \in C'_A(r') \cap succ(\overline{\sigma}^n_A)$, one can find $v' \in dom(\overline{\sigma}^n_A)$ s.t. $u' \in succ(v', \overline{\sigma}^n_A(v'))$. By definition of $\overline{\sigma}^n_A$, there exists $v \Rrightarrow_A v'$ s.t. $v \in dom(\sigma^n_A)$ and $\overline{\sigma}^n_A(v') = ST_{C_A(v), C'_A(v')}(\sigma^n_A)(v')$. From property 1.(c), this implies the existence of u with $u \Rrightarrow_A u'$ and $u \in succ(v, \sigma_A(v))$, which entails $u \in dom(\sigma^{n+1}_A)$.

We then prove by induction on n that $\overline{\sigma}^n_A$ is uniform. The case for $n = 0$ is immediate by application to σ^0_A of simulator ST of partial strategies. For $n > 0$, as noted above if $C'_A(r') \cap dom(\overline{\sigma}^n_A) \neq \emptyset$ for some r', then $C'_A(r') \subseteq dom(\overline{\sigma}^n_A)$. Therefore, the induction step only needs proof for $r'_1, r'_2 \in dom(\overline{\sigma}^{n+1}_A) \setminus dom(\overline{\sigma}^n_A)$. So assume, in this case, that $r'_1 \sim_i r'_2$ for some $i \in A$, hence $C'_A(r'_1) = C'_A(r'_2)$. Consider then some $r_1 \Rrightarrow_A r'_1$ and $r_2 \Rrightarrow_A r'_2$, which exist by the previous paragraph. Then condition 1.(b) in Def. 5 implies the existence of some r_3 such that $r_3 \sim_i r_1$ and $r_3 \Rrightarrow_A r'_2$, which then, by condition (2) implies $C_A(r_1) = C_A(r_3) = C_A(r_2)$, which in turn implies that $ST_{C_A(r_1), C'_A(r'_1)} = ST_{C_A(r_2), C'_A(r'_2)}$. Then $\overline{\sigma}^{n+1}_A(r'_1) = ST_{C_A(r_1), C'_A(r'_1)}(\sigma^{n+1}_A)(r'_1) = ST_{C_A(r_2), C'_A(r'_2)}(\sigma^{n+1}_A)(r'_2) = \overline{\sigma}^{n+1}_A(r'_2)$ as ST is a simulator of partial strategies which maps uniform strategies for A in $\mathcal{G}$ to uniform strategies for A in $\mathcal{G}'$.

As a result, the "limit" partial strategy $\overline{\sigma}_A = \bigcup\limits_{n \in \mathbb{N}} \overline{\sigma}^n_A$ defined as $\overline{\sigma}_A(r') = \overline{\sigma}^n_A(r')$ whenever $r' \in dom(\overline{\sigma}^n_A)$, is also uniform and has $dom(\overline{\sigma}_A) = succ(\overline{\sigma}_A)$. We then only need to transform it into a (total) uniform strategy by imposing a fixed action $a_0 \in Act$ wherever $\overline{\sigma}^n_A$ is undefined, that is, introducing the following uniform strategy σ'_A:

$$\sigma'_A(r') = \begin{cases} \overline{\sigma}_A(r') & \text{for } r' \in dom(\overline{\sigma}_A) \\ a_0 & \text{otherwise} \end{cases}$$

To prove property $(*)$ for the subjective semantics, consider a run $\lambda' \in out^{\mathcal{G}'}_{subj}(q', \sigma'_A)$. We build inductively the run λ as follows: for the base case, by condition 1.(b) in Def. 5 (and a short induction on the length of the indistinguishability path connecting q' with $\lambda'[0]$), we obtain a state $r \in C_A(q)$ such that $r \Rrightarrow_A \lambda'[0]$. Then, set $\lambda[0] := r$. For the inductive step, assume $\lambda[k]$ has been built, with $\lambda[j] \Rrightarrow_A \lambda'[j]$ for all $j \leqslant k$. Now, we have $\lambda'[k+1] \in$

$succ(\lambda'[k], \sigma'_A)$. By definition, $\sigma'_A(\lambda'[k]) = ST_{C_A(\lambda[k]), C'_A(\lambda'[k])}(\sigma_A)(\lambda'[k])$ since $\lambda[k] \Rightarrow_A \lambda'[k]$. This, by property 1.(c) implies the existence of some $u \in succ(\lambda[k], \sigma_A)$ such that $u \Rightarrow_A \lambda'[k+1]$. We then set $\lambda[k+1] := u$, and obtain the desired result. The same proof works for the objective semantics, by simply starting the induction with $\lambda[0] = q$. Property $(*)$ is then proved for both semantics. □

By using Proposition 7 we are finally able to prove the main preservation result of this paper.

Theorem 8. *Let $\mathcal{G}$ and $\mathcal{G}'$ be iCGS, $q \in S$ and $q' \in S'$ be states such that $q \Longleftrightarrow_A q'$, and $\lambda \in S^+$ and $\lambda' \in S'^+$ be runs such that $\lambda \Longleftrightarrow_A \lambda'$. Then, for every state A-formula φ and path A-formula ψ,*

$$(\mathcal{G}, q) \models \varphi \quad \textit{iff} \quad (\mathcal{G}', q') \models \varphi$$
$$(\mathcal{G}, \lambda) \models \psi \quad \textit{iff} \quad (\mathcal{G}', \lambda') \models \psi$$

Proof. The proof is by mutual induction on the structure of φ and ψ.

The case for propositional atoms is immediate as $(\mathcal{G}, q) \models p$ iff $p \in \pi(q)$, iff $p \in \pi'(q')$ by definition of bisimulation, iff $(\mathcal{G}', q') \models p$. The inductive cases for propositional connectives are also immediate.

For $\psi = \varphi$, suppose that $(\mathcal{G}, \lambda) \models \psi$, that is, $(\mathcal{G}, \lambda[0]) \models \varphi$. By assumption, $\lambda[0] \Longleftrightarrow_A \lambda'[0]$ as well, and by induction hypothesis $(\mathcal{G}', \lambda'[0]) \models \varphi$. Thus, $(\mathcal{G}', \lambda') \models \psi$.

For $\psi = X\psi'$, suppose that $(\mathcal{G}, \lambda) \models \psi$, that is, $(\mathcal{G}, \lambda_{\geqslant 1}]) \models \psi'$. By assumption, $\lambda_{\geqslant 1} \Longleftrightarrow_A \lambda'_{\geqslant 1}$ as well, and by induction hypothesis $(\mathcal{G}', \lambda'_{\geqslant 1}) \models \psi'$. Thus, $(\mathcal{G}', \lambda') \models \psi$. The inductive cases for $\psi = \psi' U \psi''$ and $\psi = \psi' R \psi''$ are similar.

For $\varphi = \langle\!\langle A \rangle\!\rangle \psi$, $(\mathcal{G}, q) \models \varphi$ iff that for some strategy σ_A, for all $\lambda \in out^{\mathcal{G}}_x(q, \sigma_A)$, $(\mathcal{G}, \lambda) \models \psi$. By Prop. 7, there exists strategy σ'_A s.t. for all $\lambda' \in out^{\mathcal{G}'}_x(q', \sigma'_A)$, there exists $\lambda \in out^{\mathcal{G}}_x(q, \sigma_A)$ s.t. $\lambda \Longleftrightarrow_A \lambda'$. By the induction hypothesis, $(\mathcal{G}, \lambda) \models \psi$ iff $(\mathcal{G}', \lambda') \models \psi$. Hence, $(\mathcal{G}', q') \models \varphi$. □

Corollary 9. *Let $\mathcal{G}$ and $\mathcal{G}'$ be iCGS, and $q \in S$, $q' \in S'$ be states such that $q \Longleftrightarrow_A q'$. Then, for every A-formula φ,*

$$(\mathcal{G}, q) \models \varphi \quad \textit{if and only if} \quad (\mathcal{G}', q') \models \varphi$$

By Theorem 8 we obtain that bisimilar states preserve the interpretation of ATL* formulas. More precisely, if states q and q' are A-bisimilar then they satisfy the same A-formulas. Observe that only A-formulas are normally preserved by A-bisimulations. This is already a feature of A-bisimulations for the case of perfect information [29, Theorem 6].

3.3. Discussion

In [48] we introduced a different notion of bisimulation for the ATL fragment only of ATL*. Specifically, the original Def. 6 of bisimulation in [48] differs from Def. 5 in that it lacks condition 2 (we refer to the paper for further details). Unfortunately, this weaker notion of bisimulation makes the preservation theorem false. In this paper we remedy the problem by requiring the injectiveness of the bisimulation relation w.r.t. common knowledge neighbourhoods, i.e., by including condition 2 in Def. 5.

To illustrate the issue consider the single-agent iCGS with states, actions, and transitions as illustrated in Fig. 3, and a relation $\Longleftrightarrow_A$ such that $q_i \Longleftrightarrow_A q'_i$ for $i \in \{0, 3\}$ and $q_i \Longleftrightarrow_A q'_{ij}$ for $i, j \in \{1, 2\}$, as indicated by the red lines in Fig. 3. A case-by-case analysis shows that $\Longleftrightarrow_A$ satisfies condition 1 in Def. 5, and therefore the two iCGS are bisimilar according to the notion presented in [48]. However, they do not satisfy condition 2. Indeed, we have that $q_1 \Longleftrightarrow_A q'_{1j}$, for $j \in \{1, 2\}$, but $C_1(q'_{11}) \neq C_1(q'_{12})$, and similarly for q_2, q'_{21}, q'_{22}. In particular, the iCGS $\mathcal{G}_1$ and $\mathcal{G}_2$ do not satisfy the same formulas: we have $(\mathcal{G}_1, q_0) \not\models \langle\langle 1\rangle\rangle Fp$ while $(\mathcal{G}_2, q_0) \models \langle\langle 1\rangle\rangle Fp$. Indeed, in $\mathcal{G}_2$ a memoryless strategy for agent 1 to achieve Fp is to choose action a in knowledge set $\{q'_{1,1}, q'_{2,1}\}$ and action b in $\{q'_{1,2}, q'_{2,2}\}$. Such a strategy is not transferable as a memoryless strategy into $\mathcal{G}_1$, as it would require one bit memory, that is, on the first visit of knowledge set $\{q_1, q_2\}$ agent 1 would play a, then b on the second visit. Hence, the notion of bisimulation introduced in [48] is too weak even to preserve formulas in the fragment ATL; while the present Def. 5 does preserve the whole ATL*.

3.4. Computational Complexity

Theorem 8 shows that two A-bisimilar iCGS are equivalent with respect to coalition A's strategic abilities. This can be useful when verifying abilities in realistic multi-agent systems. In such cases, the iCGS typically arises through some kind of product of local components, and suffers from state-space explosion as well as transition-space explosion. If there is a smaller bisimilar iCGS, one can use the latter as the input to the model checking procedure. Since the hardness of the model checking problem for ATL* with imperfect information is mostly related to the size of the model, using an equivalent reduced iCGS can turn an impossible task into a feasible one. An interesting question is therefore if one can automatically synthesise such reduced models, or at least check whether two given iCGS are bisimilar. We

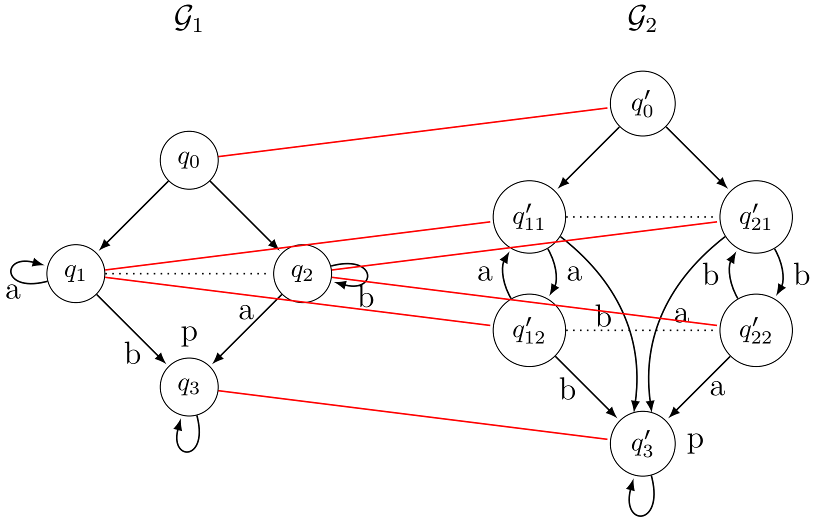


Figure 3: two iCGS satisfying only condition 1 in Def. 5.

briefly investigate the issue hereafter. We call a pair $(\mathcal{G}, s)$, for s in $\mathcal{G}$, a *pointed model*.

Theorem 10. *Checking A-bisimilarity of two pointed iCGS is in* $\mathbf{\Sigma^P_5}$.

Proof. Complexity-wise, the most problematic part of Def. 5 is the quantification over strategy simulators ST. Being a mapping from partial strategies to partial strategies, ST has exponential size w.r.t. the size of the model, which suggests at least exponential time for verifying bisimilarity. Fortunately, a closer look at the definition of strategy simulators, and at condition 2 in Def. 5, reveals that ST can be split into local mappings from mutually disjoint common knowledge neighbourhoods C_A in $\mathcal{G}$ to unique common knowledge neighbourhoods C'_A in $\mathcal{G}'$. Thus, the conceptual structure of Def. 5

can be (with a slight abuse of notation) summarized as:

$$\begin{aligned}
\mathit{bisimilar}((\mathcal{G},q),(\mathcal{G}',q')) \quad &\text{iff} \quad \exists(\Longleftrightarrow_A)\;\; \mathit{simul}((\mathcal{G},q),(\mathcal{G}',q'),\Longleftrightarrow_A) \\
&\qquad\quad \wedge\; \mathit{simul}((\mathcal{G}',q'),(\mathcal{G},q),\Longleftrightarrow_A) \;\wedge\; q \Longleftrightarrow_A q', \\
\mathit{simul}((\mathcal{G},q),(\mathcal{G}',q'),\Longleftrightarrow_A) \quad &\text{iff} \quad \forall(\mathcal{C}_A \in \mathcal{G}')\; \forall(\sigma_A \in \mathcal{C}_A)\exists(\sigma'_A \in \mathcal{C}'_A) \\
&\qquad\quad \forall(\hat{q} \in \mathcal{C}_A, \hat{q}' \in \mathcal{C}'_A, \hat{q} \Longleftrightarrow_A \hat{q}')\;\; \mathit{match}(\hat{q},\hat{q}') \\
&\qquad\quad \wedge\; \mathit{simulepist}(\hat{q},\hat{q}') \;\wedge\; \mathit{simultrans}(\mathcal{C}_A,\mathcal{C}'_A), \\
\mathit{match}(\hat{q},\hat{q}') \quad &\text{iff} \quad \pi(\hat{q}) = \pi(\hat{q}'), \\
\mathit{simulepist}(\hat{q},\hat{q}') \quad &\text{iff} \quad \forall(i \in A)\; \forall(\hat{q}' \sim_i r')\; \exists(r)\;\; \hat{q} \sim_i r,\; r \Longleftrightarrow_A r' \\
\mathit{simultrans}(\mathcal{C}_A,\mathcal{C}'_A) \quad &\text{iff} \quad \forall(r \in \mathcal{C}_A, r' \in \mathcal{C}'_A, r \Longleftrightarrow_A r')\; \forall(s' \in \mathit{succ}(r',\sigma'_A)) \\
&\qquad\quad \exists(s \in \mathit{succ}(r,\sigma_A))\;\; s \Longleftrightarrow_A s'.
\end{aligned}$$

Notice that all the structures quantified in the above expressions are of polynomial size with respect to the number of states, transitions, and epistemic links in $\mathcal{G}$ and $\mathcal{G}'$. Thus, it is relatively straightforward to observe, by a suitable translation to the QBF problem (i.e., satisfiability of Quantified Boolean Formulae) that:

1. checking $\mathit{simulepist}(\hat{q},\hat{q}')$ and $\mathit{simultrans}(\mathcal{C}_A,\mathcal{C}'_A)$ is in $\mathbf{\Pi_2^P}$,
2. checking $\mathit{simul}((\mathcal{G},q),(\mathcal{G}',q'),\Longleftrightarrow_A)$ is in $\mathbf{\Pi_4^P}$,
3. and thus, finally, checking $\mathit{bisimilar}((\mathcal{G},q),(\mathcal{G}',q'))$ is in $\mathbf{\Sigma_5^P}$.

□

We suspect that the upper bound is tight, as we do not see how any of the quantifier alternations, included in the above QBF translation, could be collapsed. For the moment, however, we only show the following, and leave the question of the exact complexity for future work.

Theorem 11. *Checking A-bisimilarity of two pointed iCGS is* **NP***-hard.*

Proof. We adapt the SAT reduction from [67, Proposition 11]. Let us first recall the idea of that reduction. Given a Boolean formula Φ in CNF, we build a 3-agent iCGS $\mathcal{G}_\Phi$ where each literal l from clause ψ in Φ is associated with a state q_l^ψ. Each agent has a different role in the construction: player 1 selects the literals to be satisfied (one literal per clause in Φ), and player 2 chooses the truth values for those literals. Player 3 "simulates" the common knowledge neighborhood for $\{1,2\}$, and thus ensures that a successful assignment must work for all the clauses in Φ.

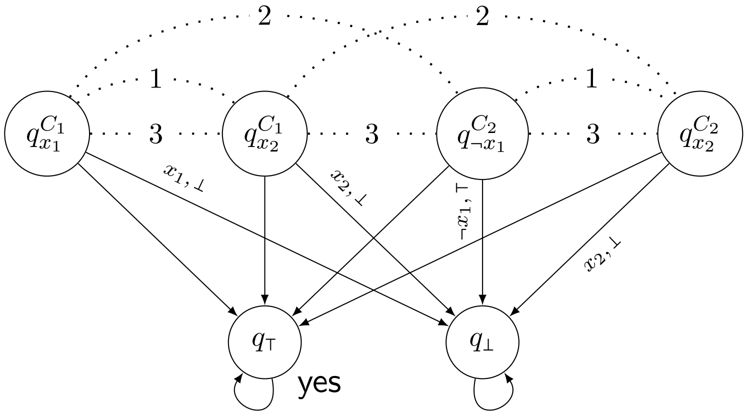


Figure 4: Model M_Φ for $\Phi \equiv C_1 \wedge C_2$ with $C_1 \equiv x_1 \vee x_2$ and $C_2 \equiv \neg x_1 \vee x_2$. Only transitions leading to $q_\bot$ are labeled; the other combinations of actions lead to $q_\top$.

Formally, at state q_l^ψ, player 1 indicates a literal from ψ, and player 2 decides on the valuation of the underlying Boolean variable. If 1 indicated a "wrong" literal $l' \neq l$ then the system proceeds to state $q_\top$ where proposition yes holds. The same happens if 1 indicated the "right" literal (l) and 2 selected the valuation that makes l true. Otherwise the system proceeds to the "sink" state $q_\bot$. Player 1 must select literals uniformly within clauses, which is imposed by fixing $q_l^\psi \sim_1 q_{l'}^{\psi'}$ iff $\psi = \psi'$. Player 2 is to select uniform valuations of variables, i.e., $q_l^\psi \sim_2 q_{l'}^{\psi'}$ iff $var(l) = var(l')$, where $var(l)$ is the variable contained in l. Finally, all states except $q_\top, q_\bot$ are indistinguishable for agent 3. An example of the construction is presented in Figure 4. Let q_0 be an arbitrary "literal" state, e.g., the one for the first literal in the first clause. Then, $\mathbf{SAT}(\Phi)$ iff $(\mathcal{G}_\Phi, q_0) \models \langle\!\langle 1,2,3 \rangle\!\rangle \mathrm{X}\mathsf{yes}$ according to the subjective semantics of **ATL** with imperfect information and imperfect recall.

Now, we construct iCGS $\mathcal{G}'_\Phi$ by adding action *skip*, available to agent 1 at every "literal" state, which always enforces a transition to $q_\top$ regardless of the actions selected by agents 2 and 3. In particular, $\mathcal{G}'_\Phi$ adds to $\mathcal{G}_\Phi$ a strategy for coalition $\{1,2,3\}$ whose only successor state from all the "literal" states is $q_\top$. Thus, $(\mathcal{G}_\Phi, q_0) \models \langle\!\langle 1,2,3 \rangle\!\rangle \mathrm{X}\mathsf{yes}$ iff $(\mathcal{G}_\Phi, q_0) \Longleftrightarrow_{\{1,2,3\}} (\mathcal{G}'_\Phi, q_0)$, which completes the reduction. □

In sum, the problem of deciding bisimilarity is easier than exponential time, but still far from practically automatizable. To obtain a reduction, one must propose the reduced model and the bisimulation according to one's intuition, and verify correctness of the bisimulation by hand. We will show an extensive example of how this can be done in Section 5.

4. Towards the Hennessy-Milner Property

In this section we show that the notion of bisimulation we introduced in Section 3 does not enjoy the Hennessy-Milner (HM) property, that is, some iCGS are logically equivalent (i.e., they satisfy the same formulas in ATL*) and yet they are not bisimilar. This is the case for both the subjective and objective variants of our semantics. More precisely, for $x \in \{sub, obj\}$, we say that states $q \in \mathcal{G}$ and $q' \in \mathcal{G}'$ are *A-equivalent*, or $q \approx^x_A q'$, iff for every A-formula φ, $(\mathcal{G}, q) \models_x \varphi$ iff $(\mathcal{G}, q') \models_x \varphi$. The iCGS $\mathcal{G}, \mathcal{G}'$ will be omitted as clear from the context. In other words, relation $\approx^x_A$ connects those states of two iCGS that satisfy the same formulas in ATL* that refer to coalition A.

In many logics, such as the Hennessy-Milner logic, CTL, CTL*, ATL and ATL* interpreted under perfect information [29], logical equivalence can be characterised in a local and efficient way by means of bisimulations. By contrast, the bisimulations for ATL* introduced in this paper are strictly weaker than logical equivalence $\approx^x_A$. In particular, in Section 4.1 we show that there are iCGS that satisfy the same formulas in ATL*, while not being bisimilar. Then, in Section 4.2 we present a necessary local condition that needs to be satisfied by $\approx^{subj}_A$. As we show, the condition is not sufficient however. Nevertheless, it is an important piece of the currently partially known puzzle about the characterisation power of bisimulations for ATL* under imperfect information.

In the rest of the section let $T \subseteq S$ and $R \subseteq S \times S'$. We define *the image* of T w.r.t. R:

$$Img(T, R) = \{q \in S' \mid \text{for some } q' \in T \ (q', q) \in R\}.$$

For each $q \in S$, let $E_A(q)$ denote the set of all states that are indistinguishable from q according to the "*everybody in A knows*" relation $\sim^E_A$.

4.1. Failure of the HM Property

We immediately state the failure of the HM property for our notion of bisimulation for ATL* under imperfect information.

Theorem 12. *For $x \in \{sub, obj\}$, there exists iCGS $\mathcal{G}, \mathcal{G}'$, and states q, q', such that $q \approx^x_{Ag} q'$ and $q \not\Leftrightarrow_{Ag} q'$.*

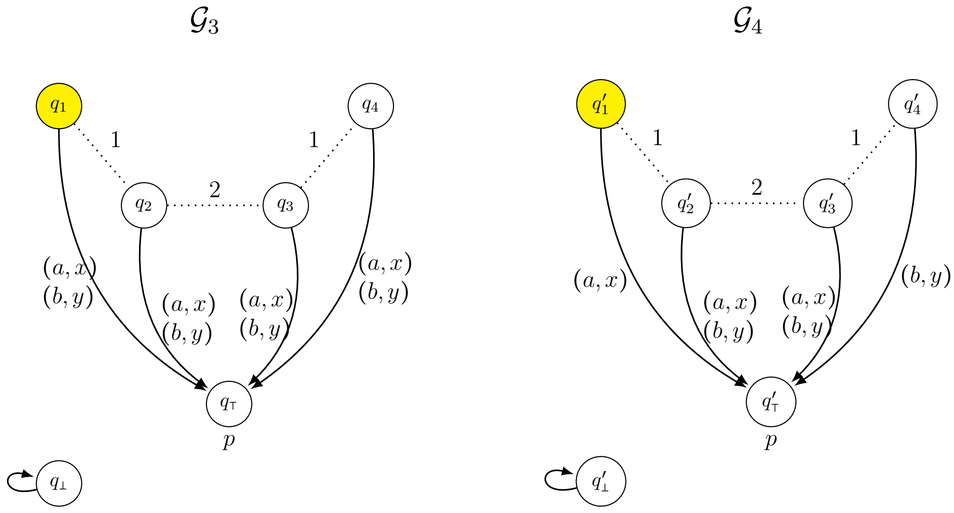


Figure 5: A counterexample to Hennessy-Milner property for subjective semantics. Available actions: $\{a, b, c\}$ for agent 1, $\{x, y, z\}$ for agent 2.

We prove Theorem 12 first for the subjective and then for the objective semantics.

Subjective Semantics. Consider the two-agent iCGS in Fig. 5. In each state agent 1 can execute actions $\{a, b, c\}$ while agent 2 can execute $\{x, y, z\}$. The transitions shown lead to $q_\top$ and $q'_\top$, while the omitted transitions lead to $q_\bot$ and $q'_\bot$, respectively. Via case-by-case analysis it can be proved that $q_i \approx^{subj}_{Ag} q_j$, $q'_i \approx^{subj}_{Ag} q'_j$, and $q_i \approx^{subj}_{Ag} q'_j$, for $i, j \in \{1, 2, 3, 4\}$, and also $q_\bot \approx^{subj}_{Ag} q'_\bot$, $q_\top \approx^{subj}_{Ag} q'_\top$. Therefore, $\mathcal{G}_3$ and $\mathcal{G}_4$ are Ag-equivalent, i.e., they satisfy the same Ag-formulas in ATL*. However, we show that there is no Ag-bisimulation between the two iGCS. In particular, for any $i, j \in \{1, 2, 3, 4\}$, state q_i cannot be Ag-bisimular with any state q'_j. In the proof we make use of the following lemma.

Lemma 13. *Let $A \subseteq Ag$. If $q \Longleftrightarrow_A q'$ then*

1. *for all $r' \in C_A(q')$, there exists $r \in C_A(q)$ such that $r \Rightarrow_A r'$;*
2. *for all $r \in C_A(q)$, there exists $r' \in C_A(q')$ such that $r \Rightarrow_A r'$.*

Proof. The thesis of the lemma follows immediately from the observation that $C_A(q') \subseteq Img(C_A(q), \Rightarrow_A)$, which in turn can be shown via straightforward induction on the length of the path w.r.t. relation $\sim^C_A$ that joins q' with r'. □

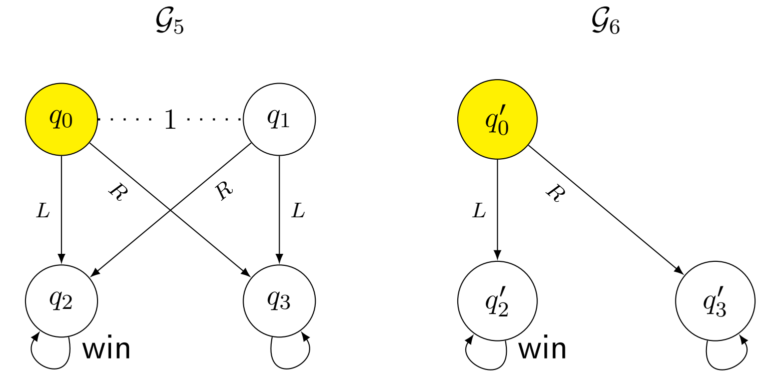


Figure 6: A counterexample to the Hennessy-Milner property for objective semantics.

Now, set $T = \{q_1, q_2, q_3, q_4\}$ and $T' = \{q'_1, q'_2, q'_3, q'_4\}$, and note that, for the partial strategy σ_{Ag} whose domain is T and is defined by $\sigma_{Ag}(q_i) = (a, x)$ for all $q_i \in T$, we have $succ(T, \sigma_{Ag}) = \{q_\top\}$. On the other hand, note that for any partial strategy σ'_{Ag} whose domain is T', we have $succ(T', \sigma'_{Ag}) = \{q'_\top, q'_\bot\}$.

So, if we assume the existence of a strategy simulator $ST : PStr_A(C_A(q_i)) \to PStr_A(C'_A(q'_i))$, then $succ(T', ST(\sigma^1_{Ag})) = \{q'_\bot, q'_\top\}$, which implies that there exists $q'_j \in T'$ such that $succ(q'_j, \sigma'_{Ag}) = q'_\bot$. On the other hand, by Lemma 13, there exist $q_i \in T$ such that $q_i \Rrightarrow_{Ag} q'_j$. But $succ(q_i, \sigma^1_{Ag}) = q_\top$, which is in contradiction with point 1.(c) in Def. 5, as it cannot be the case that $q_\top \Rrightarrow_{Ag} q'_\bot$.

Objective Semantics. Consider the single-agent iCGS in Fig. 6 in which $q_0 \not\approx^{subj}_{Ag} q'_0$. Observe first that, by symmetry, we have $q_0 \approx^{x}_{Ag} q_1$, for $x \in \{subj, obj\}$. Hence, the same two strategies (play R, play L) can be executed in q_0 and q_1. For each such strategy, its subtree starting from q_0 is isomorphic to the corresponding tree starting from q'_0, we thus have $q_0 \approx^{obj}_{Ag} q'_0$. In particular, we can establish that $q_i \approx^{obj}_{Ag} q'_i$, for all $i \in \{0, 2, 3\}$. On the other hand, as both $q_0 \approx^{obj}_{Ag} q'_0$ and $q_1 \approx^{obj}_{Ag} q'_0$, the simulator function $ST_{\{q_0,q_1\},\{q'_0\}}$ should satisfy $ST_{\{q_0,q_1\},\{q'_0\}}(L) = L$ and $ST_{\{q_0,q_1\},\{q'_0\}}(L) = R$, a contradiction. Hence, it is not the case that $q_0 \Longleftrightarrow_{Ag} q'_0$.

As a consequence of Theorem 12, our notion of bisimulation is sufficient for the preservation of formulas in ATL*, but it is not necessary. In particular, it does not enjoy the Hennessy-Milner property. In the following section we investigate this problem more closely.

4.2. Towards a Tight Characterization of ATL-Equivalence*

In this section we investigate necessary, rather than sufficient, conditions for logical equivalence. Specifically, we introduce a notion of pre-simulation and prove that it is implied by logical equivalence for the subjective variant of the semantics, while being strictly weaker.

Definition 14 (Pre-bisimulation)**.** *Let $\mathcal{G}$ and $\mathcal{G}'$ be two iCGS defined on the same sets Ag of agents and AP of atoms. Let $A \subseteq Ag$ be a coalition of agents. A relation $\rightsquigarrow_A \subseteq S \times S'$ is a* pre-simulation *iff there exists a simulator ST of partial strategies for A w.r.t. $\rightsquigarrow_A$, such that $q \rightsquigarrow_A q'$ implies that conditions (a) and (b) in Def. 5 hold and*

- *(c') For every states $r \in E_A(q)$, $r' \in E'_A(q')$ such that $r \rightsquigarrow_A r'$, for every partial uniform strategy $\sigma_A \in PStr_A(E_A(q))$, and every state $s' \in succ(r', ST(\sigma_A))$, there exists a state $s \in succ(r, \sigma_A)$ such that $s \rightsquigarrow_A s'$.*

A relation $\leftrightsquigarrow_A$ is called pre-bisimulation *if both $\leftrightsquigarrow_A$ and $\leftrightsquigarrow_A^{-1}$ are* pre-simulations*.*

The key difference between the pre-bisimulations in Def. 14 and the bisimulations in Def. 5 regards the domains of the partial uniform strategies mapped by strategy simulator ST: these are common knowledge neighbourhoods for bisimulation, and simple collective knowledge neighbourhoods for pre-bisimulations. Moreover, pre-bisimulation need not to satisfy condition (2) in Def. 5.

We can now prove the following result.

Theorem 15 (Necessary Conditions on $\approx_A^{subj}$)**.**
If $q \approx_A^{subj} q'$, then there exists a pre-bisimulation $\leftrightsquigarrow_A$ such that $q \leftrightsquigarrow_A q'$.

Proof. We prove that $\approx_A^{subj}$ is a pre-bisimulation. The proof is by contradiction. It is routine to prove that $\approx_A^{subj}$ satisfies points (a) and (b) in Def. 5, we therefore focus on point (c'). Let $q \approx_A^{subj} q'$ and to obtain a contradiction suppose that $\sigma_A \in PStr_A(E_A(q))$ is such that for every $\sigma'_A \in PStr_A(E_A(q'))$ there exists $r \in E_A(q)$, $r' \in E_A(q')$, and $s' \in succ(r', \sigma'_A)$ such that for every $s \in succ(r, \sigma_A)$, we have $s \not\approx_A^{subj} s'$. This in particular means that for each such s there exists a formula $\varphi_s^{\sigma'_A}$ in ATL* such that $(\mathcal{G}, s) \models_{subj} \varphi_s^{\sigma'_A}$ and $(\mathcal{G}', s') \models_{subj} \neg\varphi_s^{\sigma'_A}$. Now, define formula $\varphi_r^{\sigma'_A} = \bigvee_{s \in succ(r,\sigma_A)} \varphi_s^{\sigma'_A}$ and observe that $\varphi_r^{\sigma'_A}$ is true in the next step from r by using σ_A, whereas this is not the

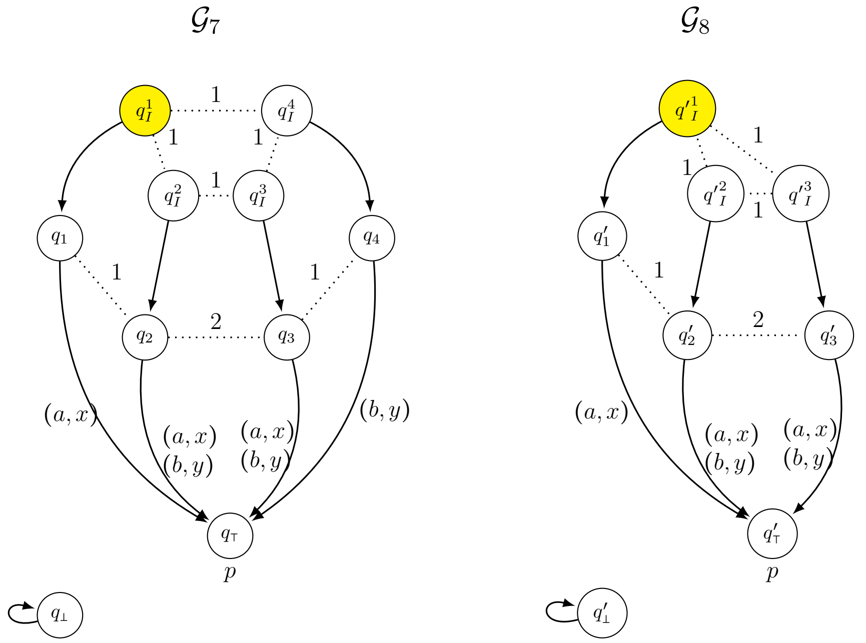


Figure 7: Pre-bisimulation does not imply logical equivalence.

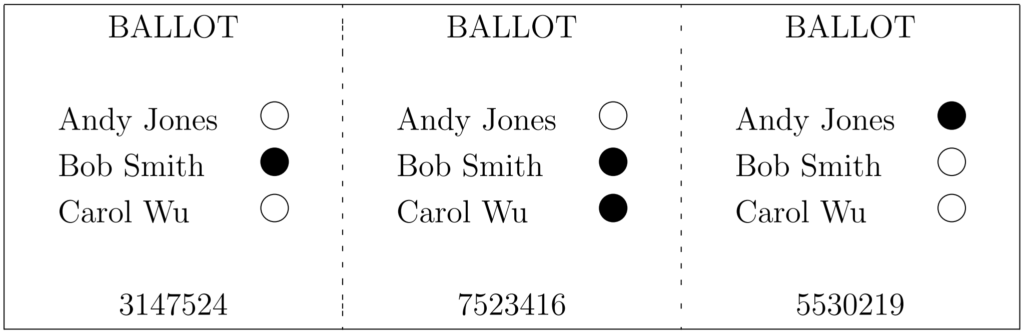
case for r'. Finally, let $\varphi = \bigwedge_{\sigma'_A \in PStr_A(E_A(q'))} \varphi_r^{\sigma'_A}$. To conclude, it suffices to observe that $(\mathcal{G}, q) \vDash_{subj} \langle\langle A \rangle\rangle X \varphi$ and $(\mathcal{G}', q') \vDash_{subj} \neg\langle\langle A \rangle\rangle X \varphi$, a contradiction with $q \approx_A^{subj} q'$. □

The following corollary to Theorem 15 deals with the case of objective semantics.

Corollary 16 (Necessary Condition on $\approx_A^{obj}$)**.** *If $q \approx_A^{obj} q'$, then for each uniform strategy σ_A there exists a uniform strategy σ'_A such that:*

$$succ(q', \sigma'_A) \subseteq Img(succ(q, \sigma_A), \approx_A^{obj})$$

It should be noted that pre-bisimilarity does not imply logical equivalence of two models. Consider the two-agent iCGS in Fig. 7. In the states that are 1-indistinguishable from q_I^1 (respectively, ${q'}_I^1$) the models are equipped with a single deterministic transition. In each of the remaining states agent 1 can execute actions $\{a, b, c\}$ and agent 2 can execute $\{x, y, z\}$. We assume that the transitions that are not depicted in the figure lead to $q_\bot$ or $q'_\bot$, respectively. Now, observe that $(\mathcal{G}_7, q_I^1) \vDash_{subj} \neg\langle\langle Ag \rangle\rangle F p$, as it is not possible



BALLOT	BALLOT	BALLOT
Andy Jones ○	Andy Jones ○	Andy Jones ●
Bob Smith ●	Bob Smith ●	Bob Smith ○
Carol Wu ○	Carol Wu ●	Carol Wu ○
3147524	7523416	5530219

Figure 8: ThreeBallot showing a vote for Bob Smith

to choose a strategies for both agents that select the same pair of actions over the set of successors of q_I^1 and leads to $q_\top$. On the other hand, such choice is possible for the successors of ${q'}_I^1$, namely it suffices to assign (a, x) to each ${q'}_I^i$, where $i \in \{1, 2, 3\}$, to see that $(\mathcal{G}_8, {q'}_I^1) \models_{subj} \langle\langle Ag\rangle\rangle Fp$. Finally, we can define a relation $\leftrightsquigarrow_A$ such that $q_i \leftrightsquigarrow_A q'_i$ and $q_I^i \leftrightsquigarrow_A {q'}_I^i$ for all $i \in \{1, 2, 3\}$, $q_\top \leftrightsquigarrow_A q'_\top$, $q_\bot \leftrightsquigarrow_A q'_\bot$, and $q_4 \leftrightsquigarrow_A q'_3$, $q_I^4 \leftrightsquigarrow_A {q'}_I^3$. A case-by-case check shows that $\leftrightsquigarrow_A$ is a pre-bisimulation. Specifically, conditions (a) and (b) in Def. 5 are immediate. As regards (c'), notice that for pre-simulations strategy simulators are defined on collective knowledge neighbourhoods rather than common knowledge ones. More specifically, $C_{Ag}(q_1) = \{q_1, q_2, q_3, q_4\}$ and $C_{Ag}(q'_1) = \{q_1, q_2, q_3\}$, while $E_{Ag}(q_1) = \{q_1, q_2, q_3\}$ and $E_{Ag}(q'_1) = \{q_1, q_2, q_3\}$. Then, while a strategy simulator can be defined on $E_{Ag}(q_1)$, $E_{Ag}(q'_1)$, so such simulator exists on $C_{Ag}(q_1)$, $C_{Ag}(q'_1)$.

5. Case Study: the ThreeBallot Voting Protocol

ThreeBallot [44, 45] is a voting protocol that strives to achieve some desirable properties, such as anonymity and verifiability of voting, without the use of cryptography. We provide a brief illustration of the protocol hereafter and refer to [44, 45] for further details. Each voter identifies herself at the poll site, and receives a “multi-ballot” paper to vote with. The multi-ballot consists of three vertical ribbons – identical except for ID numbers at the bottom (see Fig. 8, presented after [44]). The voter fills in the multi-ballot, separates the three ribbons, and casts them into the ballot box. The ballot box has the property, as usual, that it effectively scrambles the ballot order, destroying any indication of which triple of ballots originally went together, and what order ballots were cast in. To vote for a candidate, one must mark exactly two (arbitrary) bubbles on the row of the candidate. To not vote for a candidate, one must mark exactly one of the bubbles on the candidate's row (again, arbitrarily). In all the other cases the vote is invalid. The ballots are tallied by counting the number of bubbles marked for each candidate, and then subtracting the number of voters from the count.

While voting, the voter also receives a copy of one of her three ballots, and she can take it home. After the election has terminated, all the ballots are scanned and published on a web bulletin board. In consequence, the voter can check if her receipt matches a ballot listed on the bulletin board. If no ballot matches the receipt, the voter can file a complaint.

Since ThreeBallot is not a cryptographic protocol, it does not heavily rely on computers and counting can be done directly. Moreover, voters have no responsibility to ensure the integrity of cryptographic keys, and the security process in their vote is essentially the same as with traditional ballots.

Properties. ThreeBallot was proposed to provide several properties that reduce the possibility of electoral fraud. *Anonymity* (cf. e.g. [46]) requires that no agent should ever know how another voter voted, except in cases where it is inevitable, such as when all voters voted for the same candidate. Anonymity is important as it limits the opportunities of coercion and vote-buying. The latter kind of properties is usually captured by the notions of coercion-resistance and receipt-freeness [68, 69, 70]. In particular, *coercion-resistance* requires that the voter cannot reveal the value of her vote beyond doubt, even if she fully cooperates with the coercer. As a consequence, the coercer has no way of deciding whether to execute his threat (or, dually, pay for the vote). A preliminary formalization of coercion-resistance and receipt-freeness in ATL has been presented in [71].

Finally, *end-to-end voter verifiability* [34, 33] provides a way to verify the outcome of the election by allowing voters to audit the information published by the system. Typically, the focus is on individual verifiability: each voter should be able check if her vote has been taken into account and has not been altered.

5.1. iCGS Models

Here we present three iCGS models of the ThreeBallot voting protocol. All models have been specified in ISPL (Interpreted System Programming Language), the input language of the MCMAS model checker for multi-agent systems [15]. We do not model several aspects of the voting system: the ID of each ribbon, the copy of the ribbon which is given back to each voter after casting her ballot, the possibility for voters to verify the presence of the ribbon they are given back after voting. Moreover, we model a single attacker who is also a voter and, as such, follows the voting protocol and does not interact in any particular way with the other agents.

In the iCGS below, each agent is represented by means of her local variables and their evolution. The vote collector and bulletin board (BB) are modeled by the Environment agent (called Env). This agent posesses local variables modeling the fact that the voting process is open and the values of ribbons on the BB. These variables are observable by all voters, including the attacker. Env also owns private variables for collecting ribbons and can perform the three actions *stop*, *collect*, *nop* to close the voting process, to collect the votes, and finally to loop after the publication of the BB.

The agents representing voters have each a private variable representing their vote for each candidate, and they share three "ballot" variables with the environment Env. These variables represent the ribbons created by the "voting machine". Casting the vote is modeled by creating the three ribbons, consistently with the choice for each candidate. We assume that votes are immediately cast in the initial state. Being visible by Env, the values of the three ribbons are copied by Env onto the (variables represented on the) BB in a random order. Each agent has two actions: *vote*, by which the voter casts her vote, and *nop*, a non-voting or idle action. Action *vote* is enabled only in the initial state, *nope* is always enabled. All agent variables are never modified during the voting process.

In the first iCGS, denoted $\mathcal{G}_{tot}$, for each agent choice, all configurations of the three ribbons that are compatible with the agent's choice may occur. The communication between each agent and Env is entirely at Env's charge, who has direct access to agents' ribbons and copies them onto the BB. Also, copying is done randomly: Env chooses a not-yet-copied ribbon from some voter who has cast her vote (boolean variables are provided to help Env identify these situations) and copies it onto a free position on the BB.

With the second model, denoted $\mathcal{G}_{lex}$, we model a voting machine which sorts, according to the lexicographic order, the three ribbons produced for the agent's vote, and places the greatest one in the first "ballot" variable of the voter, the second greatest in the second variable, and the least in the third variable. Hence, for each choice of an agent, there are still several configurations of ribbons that are produced, but we no longer consider all permutations of a given configuration, but only a single representative.

Finally, we modify $\mathcal{G}_{lex}$ into a third model, in which Env no longer copies ribbons on the BB, but rather counts the votes for each candidate by peeping at the ”ballot” variables of each voter. This model is denoted $\mathcal{G}_{count}$.

More formally, in the case of $\mathcal{G}_{tot}$ for n voters and c candidates, each global state has the form $(vopen, pub, (ribb_\ell)_{1\leqslant\ell\leqslant 3n}, (ch_i, v_i)_{1\leqslant i\leqslant n}, (s_{ij})_{1\leqslant i\leqslant n, 1\leqslant j\leqslant 3})$ such that

1. The Boolean variable *vopen* is true when the vote is open, while *pub* signals that all ribbons of agents that have voted have been copied on the BB.
2. The local states for voter i is $(vopen, pub, ribb_1, ribb_2, ...ribb_{3n}, v_i, s_{i1}, s_{i2}, s_{i3})$. This means that each voter sees the BB during the process of copying ribbons. This is not harmful for anonymity since indistinguishability is state-based, which means agents do not remember their observations.
3. Integer $1 \leqslant ch_i \leqslant c$ specifies the choice of voter i.
4. Boolean v_i $(1 \leqslant i \leqslant n)$ registers whether agent i has voted.
5. Variables s_{ij} $(1 \leqslant j \leqslant 3)$ represent the three ”strips” of the ballot for voter i. They are shared between each agent and Env, who copies them onto the BB. Value range is $\{0, 2^c - 1\} \cup \{\bot\}$ with $s_{ij} = \bot$ denoting that strip s_{ij} has been copied on the BB.
6. Integer variables $ribb_\ell$ $(1 \leqslant \ell \leqslant 3n)$ represent the content of the BB. Value range is $\{0, 2^c - 1\} \cup \{\bot\}$, with $ribb_\ell = \bot$ denoting that no content of a strip has been copied into this ribbon on the BB.

Initial states are such that $vopen = true$, $v_i = false$ for all $i \leqslant n$, variables $ribb_\ell$ are set to the *undefined* value $\bot$ and, for variables s_{ij} we have the following rules modeling the creation of a triple of ribbons compatible with the choice of a candidate: for each voter i, let $b_{jk} = b^i_{jk}$ be the bit representing the bubble on the line corresponding to candidate k of the jth ballot of i's vote, as represented by the value of ch_i. A tuple $(b_{jk})_{1\leqslant j\leqslant 3, 1\leqslant k\leqslant c}$ is *compatible* with choice ch_i if the following holds:

1. if $ch_i = k$ then for some $p \leqslant 3$, $b_{pk} = 0$ and for all $p' \neq p$, $b_{p'k} = 1$
2. if $ch_i \neq k$ then for some $p \leqslant 3$, $b_{pk} = 1$ and for all $p' \neq p$, $b_{p'k} = 0$

Denote $B(ch_i)$ the set of bit tuples $(b_{jk})_{1\leqslant j\leqslant 3, 1\leqslant k\leqslant c}$ compatible with ch_i. Denote further by $R(ch_i)$ the transformation of these bit tuples into integer triples modeling the valid ballots compatible with choice ch_i:

$$R(ch_i) = \{(st_j)_{1\leqslant j\leqslant 3} \mid st_j = \sum_{1\leqslant k\leqslant c} b_{jk} \cdot 2^{k-1}, (b_{jk})_{1\leqslant j\leqslant 3, 1\leqslant k\leqslant c} \in B(ch_i)\}$$

As an example, valid triples of integers compatible with a vote for candidate 2, for $c = 2$, are all permutations of $(3,2,0)$ together with all permutations of $(2,2,1)$. Then $(s_{ij})_{1\leqslant j\leqslant 3} \in R(ch_i)$ for every $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant 3$.

Finally, in $\mathcal{G}_{tot}$ the protocol for the ThreeBallot system is given as follows:

1. actions *vote* and *nop* are available to all voters when $vopen = true$; otherwise, only *nop* is available.
2. *stop* and *nop* are available to Env when $vopen = true$.
3. *collect* and *publish* are available to Env when $vopen = false$.

Then, transitions are of the form:

$$(vopen, pub, (ribb_\ell)_{1\leqslant\ell\leqslant 3n}, (ch_i, v_i)_{1\leqslant i\leqslant n}, (s_{ij})_{1\leqslant i\leqslant n, 1\leqslant j\leqslant 3}) \xrightarrow{(a_e,a_1,a_2,\dots,a_n)} (vopen', pub', (ribb'_\ell)_{1\leqslant\ell\leqslant 3n}, (ch'_i, v'_i)_{1\leqslant i\leqslant n}, (s'_{ij})_{1\leqslant i\leqslant n, 1\leqslant j\leqslant 3})$$

where

1. $vopen' = false$ if $a_e = stop$ or $vopen = false$; $vopen' = true$ otherwise.
2. For $a_i = vote$, $v'_i = true$, and for $a_i = nop$, $v'_i = v_i$.
3. For $a_e = collect$ (and hence $a_i = nop$ for all i) we have the following:
   (a) there exists some subset of pairs $A \subseteq \{1,\dots,n\} \times \{1,2,3\}$ and a pair $(i_0, j_0) \notin A$ such that
      i. $s'_{ij} = s_{ij} = \perp$ for all $(i,j) \in A$
      ii. $s'_{i_0,j_0} = \perp$, $s_{i_0,j_0} \neq \perp$
      iii. $s'_{ij} = s_{ij} \neq \perp$ for all $(i,j) \notin A \cup \{(i_0,j_0)\}$;
   (b) there exists some $B \subseteq \{1,\dots,3n\}$ with $card(B) = card(A)$ and some integer $k \notin B$, $1 \leqslant k \leqslant 3n$ such that
      i. $ribb'_\ell = ribb_\ell \neq \perp$ for all $\ell \in B$
      ii. $ribb_k = \perp$, $ribb'_k = s_{i_0,j_0}$
      iii. $ribb_\ell = ribb'_\ell = \perp$ for all $\ell \notin B \cup \{k\}$.
4. Action $a_e = publish$ can only be executed when, for each i, either $s_{i1} = s_{i2} = s_{i3} = \perp$ or $v_i = false$, and its effect is to modify only $pub' = true$, all other variables remaining unchanged.

In the iCGS $\mathcal{G}_{lex}$ transitions are identical to the above, the only difference being in the initial states, more specifically in the configuration of variables s_{ij}. Intuitively, for each choice ch_i of voter i, we only keep those initial configurations in which (s_{i1}, s_{i2}, s_{i3}) is the maximal, in lexicographic order, among the encodings R_{ch_i} of ch_i. Formally, given a triple of integers $t = (t_1, t_2, t_3)$, we denote with $Perm(t)$ the set of all permutations of triple t. Then, the initial states in $\mathcal{G}_{lex}$ are the initial states of $\mathcal{G}_{tot}$ for which $(s_{i1}, s_{i2}, s_{i3}) = \max Perm\big((s_{i1}, s_{i2}, s_{i3})\big)$, the maximum being considered under the lexicographic order.

Finally, the iCGS $\mathcal{G}_{count}$ is similar to $\mathcal{G}_{lex}$ but all variables $ribb_\ell$ are replaced with c variables $(co_k)_{1 \leqslant k \leqslant c}$. The local states for agent i are then of the form $(vopen, pub = 0, v_i, s_{i1}, s_{i2}, s_{i3})$ and $(vopen, pub = 1, co_1, \ldots, co_c, v_i, s_{i1}, s_{i2}, s_{i3})$. The specification of transitions is then the same as for $\mathcal{G}_{tot}$, except for $a_e = collect$ and items 3.(b.i)-3.(b.iii) above (defining the updates of variables $ribb_\ell$), which are replaced by the following:

3.(b') For each $1 \leqslant k \leqslant c$, $co'_k = co_k + d_{i_0 j_0 k}$, where $d_{i_0 j_0 k}$ is the k-th least significant bit of $s_{i_0 j_0}$.

5.2. Coercion freeness and anonymity properties for the ThreeBallot

In this section we present the formulas that are of interest for the verification of ThreeBallot. We verify two types of formulas: a variant of coercion resistance [71] and a variant of anonymity. The coercion resistance property specifies the fact that the attacker att has no strategy by which he could know how agent i has voted ($i \neq att$):

$$\varphi_i = \langle\!\langle att \rangle\!\rangle F\Big((pub \wedge v_i) \rightarrow \bigvee_{1 \leqslant j \leqslant nc} K_{att}(j = ch_i)\Big)$$

Recall that, in our model the attacker is also a voter, which corresponds to situations in which a voter fully cooperates with the attacker. Additionnaly, as already noted in Section 2, the knowledge operator is definable in ATL with the subjective interpretation as $K_{att}\varphi = \langle\!\langle att \rangle\!\rangle \varphi U \varphi$.

The anonymity property does not require ATL but rather CTLK, the combination of branching-time temporal logic with epistemic logic of e.g. [72]. The AG operator from CTLK, whose meaning is that along all paths the target subformula holds globally, can be defined as usual in ATL as follows:

$$AG\varphi \equiv \langle\!\langle \varnothing \rangle\!\rangle G\varphi$$

Then, the anonymity property of interest for ThreeBallot is that any of the agents cannot know, at any time instant, the way another agent has voted:

$$\varphi_i^c = AG\big(\bigwedge_{1 \leqslant j \leqslant nc} \neg K_{att} p_{ch_i = j}\big)$$

Note that anonymity is not ensured when there are at most three voters, including the attacker (who may vote).

5.3. Bisimulations for $\mathcal{G}_{tot}$, $\mathcal{G}_{lex}$ and $\mathcal{G}_{count}$

The three iCGS defined in the previous section appear to be naturally related, in particular w.r.t. the properties pertaining to the attacker modifying the outcome of the vote or breaking anonymity. The interest in simplifying the model is that checking the coercion resistance property can be done faster and with less memory on $\mathcal{G}_{count}$ than on $\mathcal{G}_{lex}$, which, on its turn, requires less time and memory than $\mathcal{G}_{tot}$, as we will see in the last section on experimental results. In this section we show that the three models are bisimilar for the attacker, for the set of atomic propositions that refer only to choices of agents. Bisimulations formalize the "natural relation" between these iCGS and allows us to check coercion resistance on the smallest iCGS and then transfer the result to the two others, in particular to the original model $\mathcal{G}_{tot}$. Note that these bisimulations work because the properties do not refer to the status of the BB. For instance, these bisimulations would not be useful for simplifying systems for verifiability [37].

Formally, for each choice for an agent i to vote for candidate j, we introduce an atomic proposition $p_{ch_i=j}$, which holds true only in those states in which $ch_i = j$. Then, if we denote the attacker $att = n$ and $AP = \{p_{ch_i=j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant c\}$, we prove that the following relation is an $\{att\}$-bisimulation over AP between $\mathcal{G}_{tot}$ and $\mathcal{G}_{lex}$:

$$(vopen, pub, (ribb_\ell)_{1\leqslant \ell\leqslant 3n}, (ch_i, v_i)_{1\leqslant i\leqslant n}, (s_{ij}, a_{ij})_{1\leqslant i\leqslant n,1\leqslant j\leqslant 3}) \Longleftrightarrow^1_{\{att\}}$$
$$(vopen', pub', (ribb'_\ell)_{1\leqslant \ell\leqslant 3n}, (ch'_i, v'_i)_{1\leqslant i\leqslant n}, (s'_{ij}, a'_{ij})_{1\leqslant i\leqslant n,1\leqslant j\leqslant 3})$$

iff the following holds:

1. $vopen = vopen'$, $pub = pub'$, $v_i = v'_i$, $ch_i = ch'_i$ for all $1 \leqslant i \leqslant n$, $s_{att,j} = s'_{att,j}$ and $a_{att,j} = a'_{att,j}$ for all $1 \leqslant j \leqslant 3$ and $ribb_\ell = ribb'_\ell$ for all $1 \leqslant \ell \leqslant 3n$.
2. For every $1 \leqslant i \leqslant n$, if we denote b_{jk} the kth least significant bit of s_{ij} and b'_{jk} the kth bit of s'_{ij}, then both $(b_{jk})_{1\leqslant j\leqslant 3,1\leqslant k\leqslant c}, (b'_{jk})_{1\leqslant j\leqslant 3,1\leqslant k\leqslant c} \in B(ch_i)$.
3. Denote ρ_i the S_3-permutation of (s_{i1}, s_{i2}, s_{i3}) into $(s'_{i1}, s'_{i2}, s'_{i3})$, i.e. $s_{ij} = s'_{i\rho_i(j)}$. Also when $s_{ij} = s'_{ij} = \bot$ we put $\rho_i = id_{\{1,2,3\}}$. Then $a_{ij} = a'_{i\rho_i(j)}$ for all $1 \leqslant i \leqslant n-1, 1 \leqslant j \leqslant 3$.

Intuitively, item (3) above says that (b'_{jk}) is the largest, in lexicographic order, among all tuples in B_{ch_i} that are permutations of (b_{jk}).

For $\mathcal{G}_{lex}$ and $\mathcal{G}_{count}$, we consider the following $\{att\}$-bisimulation over AP:

$$(vopen, pub, (ribb_\ell)_{1\leqslant \ell\leqslant 3n}, (ch_i, v_i)_{1\leqslant i\leqslant n}, (s_{ij}, a_{ij})_{1\leqslant i\leqslant n,1\leqslant j\leqslant 3}) \Longleftrightarrow^2_{\{att\}}$$
$$(vopen', pub', (co_k)_{1\leqslant k\leqslant c}, (ch'_i, v'_i)_{1\leqslant i\leqslant n}, (s'_{ij}, a'_{ij})_{1\leqslant i\leqslant n,1\leqslant j\leqslant 3})$$

where:

1. $vopen = vopen'$, $pub = pub'$, $v_i = v'_i$, $ch_i = ch'_i$, $s_{ij} = s'_{ij}$ and $a_{ij} = a'_{ij}$ for all $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant 3$.
2. For every $1 \leqslant \ell \leqslant 3n$ and $1 \leqslant k \leqslant c$, if we denote $b_{\ell k}$ the kth least significant bit on the ribbon $ribb_\ell$, then:

$$co_k = \sum \left\{ b_{\ell k} \mid ribb_\ell \neq \bot, 1 \leqslant \ell \leqslant 3n \right\}$$

To prove that these relations are indeed alternating bisimulations, note that the condition 1.(a) is immediately satisfied as whenever $q \Longleftrightarrow^{\iota}_{\{att\}} q'$, for $\iota = 1, 2$, we must have that $(ch_i = j) \in q$ iff $(ch_i = j) \in q'$.

To prove properties 1.(b) and its dual for $\Longleftrightarrow^{1}_{\{att\}}$, consider states q, r in $\mathcal{G}_{tot}$ and $r \in \mathcal{G}_{lex}$ such that $q \Longleftrightarrow^{1}_{\{att\}} q'$ and $q \sim_{att} r$. Then it is the case that

$$\begin{aligned}
q &= (vopen, pub, (ribb_\ell)_{1\leqslant \ell \leqslant 3n}, (ch_i, v_i)_{1 \leqslant i \leqslant n}, (s_{ij}, a_{ij})_{1\leqslant i \leqslant n, 1 \leqslant j \leqslant 3})\\
q' &= (vopen, pub, (ribb_\ell)_{1\leqslant \ell \leqslant 3n}, (ch_i, v_i)_{1 \leqslant i \leqslant n}, (s'_{ij}, a'_{ij})_{1\leqslant i \leqslant n, 1 \leqslant j \leqslant 3})\\
r &= (vopen, pub, (ribb_\ell)_{1\leqslant \ell \leqslant 3n}, (\overline{ch}_i, \overline{v}_i)_{1 \leqslant i \leqslant n}, (\overline{s}_{ij}, \overline{a}_{ij})_{1\leqslant i \leqslant n, 1 \leqslant j \leqslant 3})
\end{aligned}$$

with q, q' related as per the definition of $\Longleftrightarrow^{1}_{\{att\}}$ above and $\overline{ch}_{att} = ch_{att}$, $\overline{s}_{att,j} = s'_{att,j}$ and $\overline{a}_{att,j} = a'_{att,j}$ for all $1 \leqslant j \leqslant 3$. Then set

$$r' = (vopen, pub, (ribb_\ell)_{1\leqslant \ell \leqslant 3n}, (ch'_i, v'_i)_{1 \leqslant i \leqslant n-1}, (\overline{ch}_{att}, \overline{v}_{att}), (s'_{ij}, a'_{ij})_{1\leqslant i \leqslant n-1, 1 \leqslant j \leqslant 3}, (\overline{s}_{att,j}, \overline{a}_{att,j})_{1 \leqslant j \leqslant 3})$$

and we obtain the desired result, that is, $q' \Longleftrightarrow^{1}_{\{att\}} r'$ and $r \sim_{att} r'$. The mirror argument works as well: given $q \Longleftrightarrow^{1}_{\{att\}} q'$ and $q' \sim_{att} r'$, we can find r such that $q \Longleftrightarrow^{1}_{\{att\}} r$ and $r \sim_{att} r'$.

Conditions 1.(b) and its dual for $\Longleftrightarrow^{2}_{\{att\}}$ can be proved similarly, by observing that, given $q, r \in \mathcal{G}_{lex}$, $q' \in \mathcal{G}_{count}$, such that $q \Longleftrightarrow^{2}_{\{att\}} q'$ and $q \sim_{att} r$, then it is the case that

$$\begin{aligned}
q &= (vopen, pub, (ribb_\ell)_{1\leqslant \ell \leqslant 3n}, (ch_i, v_i)_{1 \leqslant i \leqslant n}, (s_{ij}, a_{ij})_{1\leqslant i \leqslant n, 1 \leqslant j \leqslant 3})\\
q' &= (vopen, pub, (co_k)_{1\leqslant k \leqslant c}, (ch_i, v_i)_{1 \leqslant i \leqslant n}, (s'_{ij}, a'_{ij})_{1\leqslant i \leqslant n, 1 \leqslant j \leqslant 3})\\
r &= (vopen, pub, (ribb_\ell)_{1\leqslant \ell \leqslant 3n}, (\overline{ch}_i, \overline{v}_i)_{1 \leqslant i \leqslant n}, (\overline{s}_{ij}, \overline{a}_{ij})_{1\leqslant i \leqslant n, 1 \leqslant j \leqslant 3})
\end{aligned}$$

with the same relation between variables of q and r as above. Then set

$$r' = (vopen, pub, (co_k)_{1\leqslant k \leqslant c}, (ch'_i, v'_i)_{1 \leqslant i \leqslant n-1}, (\overline{ch}_{att}, \overline{v}_{att}), (s'_{ij}, a'_{ij})_{1\leqslant i \leqslant n-1, 1 \leqslant j \leqslant 3}, (\overline{s}_{att,j}, \overline{a}_{att,j})_{1 \leqslant j \leqslant 3})$$

and we obtain that $q' \Longleftrightarrow^2_{\{att\}} r'$ and $r \sim_{att} r'$.

Further, for conditions 1.(c) and its dual, notice first that for any state $q \in \mathcal{G}_{tot}$ or $q \in \mathcal{G}_{lex}$, $C_{att}(q)$ is the equivalence class of q w.r.t. $\sim_{att}$, that is, if $q = (vopen, pub, (ribb_\ell)_{1 \leqslant \ell \leqslant 3n}, (ch_i, v_i)_{1 \leqslant i \leqslant n}, (s_{ij}, a_{ij})_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant 3})$, then $C_{att}(q)$ includes all and only states where the local state for att is of the form $((ribb_\ell)_{1 \leqslant \ell \leqslant 3n}, ch_{att}, v_{att}, (s_{att,j}, a_{att,j})_{1 \leqslant j \leqslant 3})$. Similarly, for $q' \in \mathcal{G}_{count}$ with $q' = (vopen, pub, (co'_k)_{1 \leqslant k \leqslant c}, (ch'_i, v'_i)_{1 \leqslant i \leqslant n}, (s'_{ij}, a'_{ij})_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant 3})$, $C_{att}(q')$ includes all and only states where the local state for att is of the form $((co_k)_{1 \leqslant k \leqslant c}, ch_{att}, v_{att}, (s_{att,j}, a_{att,j})_{1 \leqslant j \leqslant 3})$.

Then, in all three iCGS, on each neighbourhood $C_{att}(q)$, only one or two partial strategies for att can be defined, depending on whether the vote is open or not. Therefore, we define the mapping $ST_{C^{\mathcal{G}_{tot}}_{att}(\cdot), C^{\mathcal{G}_{lex}}_{att}(\cdot)}$ to associate to each partial strategy prescribing *nop* to *att* in some $C^{\mathcal{G}_{tot}}_{att}(q)$ the quasi-identical strategy prescribing the same action in $C^{\mathcal{G}_{lex}}_{att}(q')$. Similarly, to the partial strategy prescribing *vote* to *att* in $C^{\mathcal{G}_{tot}}_{att}(q)$, $ST_{C^{\mathcal{G}_{tot}}_{att}(\cdot), C^{\mathcal{G}_{lex}}_{att}(\cdot)}$ associates the strategy prescribing *vote* in $C^{\mathcal{G}_{lex}}_{att}(q')$. The dual mapping ST' is defined similarly, and analogous definitions work for bisimulation $\Longleftrightarrow^2_{\{att\}}$. Notice also that these definitions satisfy the constraints on strategy simulators.

To prove condition 1.(c), consider first the strategy $vote_{att}$ prescribing *vote* for *att* on $C^{\mathcal{G}_{tot}}_{att}(q)$ and consider some state r' such that $q' \xrightarrow{vote_{att}} r'$. Since when voting is enabled, Env does not collect votes, r' has the same BB as q' and all booleans variables a_{ij} are false. Therefore, we can choose a state r that has the same values as r' for the local variables of all voters and the same BB as q. Then, we obtain $q' \xrightarrow{ST(vote_{att})} r'$ and $r \Longleftrightarrow^1_{\{att\}} r'$. A similar line of reasoning works for $\Longleftrightarrow^2_{\{att\}}$ as well. The same argument applies if the strategy for the attacker is nop_{att} and q and q' are states in which voting is open.

Consider now a state q in which voting is closed and the strategy $none_{att}$ prescribing action *nop* for *att* on $C^{\mathcal{G}_{tot}}_{att}(q)$ and consider again some state r' such that $q' \xrightarrow{nop_{att}} r'$. Then the only agent which executes a non-idle action on the above transition is Env, who copies one of the ribbons onto the BB. This transition can then be simulated in $\mathcal{G}_{tot}$ by copying the same ribbon (but which might be stored at a different position in q than in q') onto the BB. A mirror argument can be used to prove 1.(c) for ST'.

Thirdly, for proving condition 1.(c) for $q \Longleftrightarrow^2_{\{att\}} q'$, note that the same considerations above apply for the case of a transition from a state q' in which the voting is open. For the case of states q, q' where the voting is closed and hence only strategy nop_{att} is available to the attacker, we note that the only action that is compatible with $q \xrightarrow{nop_{att}} r$ in both models is Env collecting votes. This corresponds in $\mathcal{G}_{count}$ to an action in which Env counts votes, and hence we can find a state r that is an nop_{att}-compatible successor of q, has the same local states for voters as r', and in which each counter co_k keeps the sum of the bullets on kth line on the copied ribbons from r. This will ensure that $r \Longleftrightarrow^2_{\{att\}} r'$.

Finally, as regards condition (2), if $q_1 \Longleftrightarrow^1_{\{att\}} q'$ and $q_2 \Longleftrightarrow^1_{\{att\}} q'$, then the attacker must have made the same choice in q_1 and q_2 (possibly none), and her "ballot stripes" $s_{att,j}$ and "bookkeeping bits" $a_{att,j}$ must be the same in q_1 and q_2. But this entails the indistinguishability of q_1 and q_2 for the attacker, i.e., then $C_{att}(q_1) = C_{att}(q_2)$. The proof of (2) for $\Longleftrightarrow^2_{\{att\}}$ is similar.

To conclude, all iCGS $\mathcal{G}_{tot}$, $\mathcal{G}_{lex}$, and $\mathcal{G}_{count}$ are bisimilar for the attacker. We will make use of this fact in the next section on experimental results.

Remark 17. *Note that the two relations* $\Longleftrightarrow^1_{\{att\}}$ *and* $\Longleftrightarrow^2_{\{att\}}$ *are also bisimulations on models* $\mathcal{G}_{tot}, \mathcal{G}_{lex}$ *and* $\mathcal{G}_{count}$ *in the sense of (the symmetric variant of) Def. 5.1 in [73]. Therefore, applying [73, Lemma 5.2] both ways, any CTLK formula that is satisfied in any of the models, is satisfied in all of them. This justifies our use of CTLK operators in specification* φ_i^c *above.*

On the other hand, note that CTLK bisimulations in the sense of [73] do not preserve ATL formulas under imperfect information. Figure 9 provides a counterexample, with two 1-agent models that are timed epistemic bisimilar, but do not satisfy the same formulas in ATL.

6. Experimental Results

In this section, we exhibit the improvements in running time when checking the same properties over the three bisimilar models. The three models are checked with growing number of voters and candidates. For our experiments, we have used the latest version of MCMAS (1.3.0) [15]. Tests were made on a virtual machine running Ubuntu 16.04.1 LTS on a Dell PowerEdge R720 server with two Intel Xeon E5-2650 8 core processors at 2GHz, and 128 GB of RAM. In order to investigate certain non-intuitive results reported in Tables 1, 2, and 3, we re-ran the experiments on a different machine with a similar setup of 16-core Xeon E5620 2.4GHz processor with 64GB RAM.

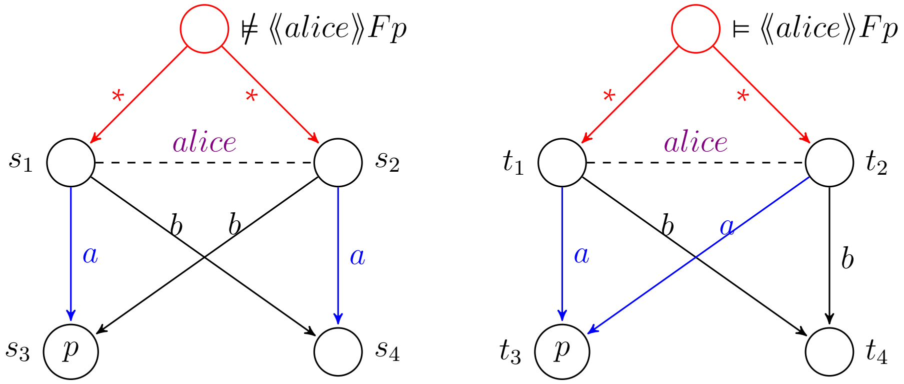

$\not\models \langle\langle alice\rangle\rangle Fp$
*
*
alice
s_1
s_2
b
b
a
a
s_3
p
s_4
$\models \langle\langle alice\rangle\rangle Fp$
*
*
alice
t_1
t_2
b
a
a
b
t_3
p
t_4

Figure 9: Two models that are (timed epistemic) bisimilar in the sense of [73], but do not satisfy the same formula in ATL.

The purported anomalies have shown to be stable over both the tests. In all the following tables, NA means a 2 hours timeout has been reached without obtaining any result. The C programs that were used to generate the ISPL files for the three ThreeBallot models, together with the generated ISPL files, are publicly available as [74].

MCMAS provides two options, `-atlk 2` or `-uniform`, for checking ATL formulas with uniform strategies, with some differences in the semantics of ATL formulas (`-uniform` is similar with *"irrevocable strategies"* of [31]). We observed that neither of these options were stable, and lead to a number of experiments ending with inconsistent results or MCMAS terminating abnormally (segfault on null-pointer assignment in one of the fixpoint computations related with ATL satisfiability). We refer the interested reader to [75, 76].

Table 1 reports the running times for a selection of $\mathcal{G}_{tot}$ and $\mathcal{G}_{lex}$ models. Note that several surprising data points are present. Namely, in the case of $(2v, 3c)$ the running time is smaller for $\mathcal{G}_{tot}$ than for $\mathcal{G}_{lex}$ despite considerably larger state space (but smaller BDD). Moreover in case of $(2v, 4c)$ the computations for $\mathcal{G}_{lex}$ timed out at the stage of computing the set of reachable states while the verification of $\mathcal{G}_{tot}$ completed successfully in an hour. Model checking the cases of $(3v, 2c)$ and $(4v, 2c)$ resulted in segmentation fault for both $\mathcal{G}_{tot}$ and $\mathcal{G}_{lex}$ on both the machines. Model checking the cases of $(3v, 3c)$ and $(4v, 3c)$ did not succeed in reasonable time for both models.

To address the segfault issues, we also checked the coercion resistance property with `-atlk 0` option, which utilizes ATL with perfect information.

	(# voters, # candidates)		
	(2v,2c)	(2v,3c)	(2v,4c)
G_{tot}	1.5 s $\|S\| \approx 5.9e + 06$ $\|BDD\| \approx 1.4e + 07$	12.8 s $\|S\| \approx 3.3e + 08$ $\|BDD\| \approx 5.5e + 07$	3714.0 s $\|S\| \approx 8.8e + 09$ $\|BDD\| \approx 4.0e + 08$
G_{lex}	1.3 s $\|S\| \approx 2.9e + 05$ $\|BDD\| \approx 1.5e + 07$	29.0 s $\|S\| \approx 1.1e + 07$ $\|BDD\| \approx 6.1e + 07$	NA

Table 1: MCMAS statistics for coercion freeness, $\mathcal{G}_{tot}$ and $\mathcal{G}_{lex}$ with the `-atlk 2` flag

This is nevertheless consistent with our theoretical setting since all tests for more than 3 candidates show that the formulas are false, and whenever a positive ATL formula is false under the perfect information semantics, it is also false under the imperfect information semantics, and hence preserved by alternating bisimulations. Tables 2 and 3 show the details of the verification of the only configurations for which MCMAS produces results in reasonable time (timeout = 2 hours) for, respectively, $\mathcal{G}_{tot}$ and $\mathcal{G}_{lex}$. Again, while the state-space of $\mathcal{G}_{lex}$ is consistently (and predictably) smaller than the state-space of $\mathcal{G}_{tot}$, the size of corresponding BDDs can increase. This is notable in particular in the case of $(2v, 3c)$ where model checking is twice faster for $\mathcal{G}_{tot}$ than for $\mathcal{G}_{lex}$. The case of $(2v, 4c)$ that terminated successfully for $\mathcal{G}_{tot}$ and timed-out for $\mathcal{G}_{lex}$ (the entry omitted from the table) is similar.

Our interpretation for the anomalies between state space, BDD size and running time is that the "ribbon" variables being more constrained in $\mathcal{G}_{lex}$ than in $\mathcal{G}_{tot}$, the default variable ordering produces larger BDDs for the more constrained model. On the other hand, the anomaly being present for both the `-atlk 0` and the `-atlk 2` flags, it does not seem to come from the difference in dealing with uniform strategies than with strategies with perfect information. Note that the same anomalies are present for the CTLK model-checking instances below.

The models $\mathcal{G}_{count}$ can be verified much faster with the `-atlk 2`, the number of reachable states and the BDD size decreasing substantially. Statistics are given in Table 4.

		# candidates		
		2c	3c	4c
# voters	2v	1.3 s $\lvert S\rvert \approx 5.9e+06$ $\lvert BDD\rvert \approx 1.2e+07$	13.3 s $\lvert S\rvert \approx 3.3e+08$ $\lvert BDD\rvert \approx 4.5e+07$	3558.5 s $\lvert S\rvert \approx 8.8e+09$ $\lvert BDD\rvert \approx 2.8e+08$
	3v	25.1 s $\lvert S\rvert \approx 2.9e+10$ $\lvert BDD\rvert \approx 5.9e+07$	NA	NA
	4v	1291.8 s $\lvert S\rvert \approx 1.7e+14$ $\lvert BDD\rvert \approx 3.0e+08$	NA	NA

Table 2: MCMAS statistics for coercion freeness, $\mathcal{G}_{tot}$ and `-atlk 0` flag

		# candidates	
		2c	3c
# voters	2v	1.1 s $\lvert S\rvert \approx 2.9e+05$ $\lvert BDD\rvert \approx 1.5e+07$	26.4 s $\lvert S\rvert \approx 1.1e+07$ $\lvert BDD\rvert \approx 5.4e+07$
	3v	10.4 s $\lvert S\rvert \approx 3.1e+08$ $\lvert BDD\rvert \approx 5.7e+07$	NA
	4v	297.9 s $\lvert S\rvert \approx 3.9e+11$ $\lvert BDD\rvert \approx 8.9e+07$	NA

Table 3: MCMAS statistics for coercion freeness, $\mathcal{G}_{lex}$ and `-atlk 0` flag

		# candidates					
		2c	3c	4c	5c	6c	7c
# voters	2v	0.1 s $\lvert S\rvert = 6.3e+03$ $\lvert BDD\rvert = 1.0e+07$	0.4 s $\lvert S\rvert = 1.7e+05$ $\lvert BDD\rvert = 1.2e+07$	1.5 s $\lvert S\rvert = 4.6e+06$ $\lvert BDD\rvert = 1.7e+07$	7.1 s $\lvert S\rvert = 1.2e+08$ $\lvert BDD\rvert = 3.9e+07$	94.2 s $\lvert S\rvert = 2.9e+09$ $\lvert BDD\rvert = 9.4e+07$	NA
	3v	0.5 s $\lvert S\rvert = 9.3e+04$ $\lvert BDD\rvert = 1.3e+07$	4.0 s $\lvert S\rvert = 6.2e+06$ $\lvert BDD\rvert = 2.9e+07$	110.0 s $\lvert S\rvert = 4.2e+08$ $\lvert BDD\rvert = 6.9e+07$	2917.8 s $\lvert S\rvert = 3.2e+10$ $\lvert BDD\rvert = 3.1e+08$	NA	
	4v	1.3 s $\lvert S\rvert = 7.8e+06$ $\lvert BDD\rvert = 1.6e+07$	32.7 s $\lvert S\rvert = 3.6e+08$ $\lvert BDD\rvert = 5.6e+07$	762.5 s $\lvert S\rvert = 1.6+12$ $\lvert BDD\rvert = 1.7e+08$	NA		
	5v	5.7 s $\lvert S\rvert = 1.7e+08$ $\lvert BDD\rvert = 1.9e+07$	341.2 s $\lvert S\rvert = 2.3e+11$ $\lvert BDD\rvert = 1.1e+08$	NA			
	6v	12.0 s $\lvert S\rvert = 3.3e+09$ $\lvert BDD\rvert = 4.2e+07$	NA				
	7v	38.8 s $\lvert S\rvert = 7.6e+10$ $\lvert BDD\rvert = 5.9e_07$	NA				
	8v	82.0 s $\lvert S\rvert = 8.7e+12$ $\lvert BDD\rvert = 5.9e+07$	NA				
	9v	139.5 s $\lvert S\rvert = 2.6e+14$ $\lvert BDD\rvert = 6.1e+07$	NA				

Table 4: MCMAS statistics for coercion freeness, $\mathcal{G}_{count}$ and `-atlk 2` flag

We also ran the same tests for the anonymity property, $\varphi_i^c = AG(\bigwedge_{1 \leqslant j \leqslant nc} \neg K_{att} p_{ch_i = j})$ The results are given in Tables 5, 6 and 7. Note the same anomalies of smaller state space but larger BDD size and running time between some $\mathcal{G}_{lex}$ models and the respective $\mathcal{G}_{tot}$ models.

		# candidates		
		2c	3c	4c
# voters	2v	1.1 s $\lvert S\rvert \approx 5.9e + 06$ $\lvert BDD\rvert \approx 1.2e + 07$	9.7 s $\lvert S\rvert \approx 3.3e + 08$ $\lvert BDD\rvert \approx 4.5e + 07$	3200.2 s $\lvert S\rvert \approx 8.8 + 09$ $\lvert BDD\rvert \approx 2.8e + 08$
	3v	14.0 s $\lvert S\rvert \approx 2.9e + 10$ $\lvert BDD\rvert \approx 5.4e + 07$	6455.9 s $\lvert S\rvert \approx 2.4e + 13$ $\lvert BDD\rvert \approx 2.4e + 08$	NA
	4v	440.0 s $\lvert S\rvert \approx 1.7e + 14$ $\lvert BDD\rvert \approx 1.1e + 08$	NA	

Table 5: MCMAS statistics for anonymity checking on $\mathcal{G}_{tot}$

		# candidates		
		2c	3c	4c
# voters	2v	0.9 s $\lvert S\rvert \approx 2.9e + 05$ $\lvert BDD\rvert \approx 1.3e + 07$	18.9 s $\lvert S\rvert \approx 1.1e + 07$ $\lvert BDD\rvert \approx 4.9e + 07$	NA
	3v	8.3 s $\lvert S\rvert \approx 3.1e + 08$ $\lvert BDD\rvert \approx 3.9e + 07$	7100.6 s $\lvert S\rvert \approx 1.5e + 11$ $\lvert BDD\rvert \approx 3.0e + 08$	NA
	4v	323.3 s $\lvert S\rvert \approx 4.0e + 11$ $\lvert BDD\rvert \approx 6.5e + 07$	NA	

Table 6: MCMAS statistics for anonymity checking on $\mathcal{G}_{lex}$

		# candidates					
		2c	3c	4c	5c	6c	7c
# voters	2v	0.1 s $\lvert S\rvert = 6.3e+03$ $\lvert BDD\rvert = 1.0e+07$	0.4 s $\lvert S\rvert = 1.6e+05$ $\lvert BDD\rvert = 1.2e+07$	1.6 s $\lvert S\rvert = 4.6e+06$ $\lvert BDD\rvert = 1.9e+07$	7.6 s $\lvert S\rvert = 1.2e+08$ $\lvert BDD\rvert = 5.1e+07$	103.6 s $\lvert S\rvert = 2.9e+09$ $\lvert BDD\rvert = 9.3e+07$	NA
	3v	0.5 s $\lvert S\rvert = 9.3e+04$ $\lvert BDD\rvert = 1.2e+07$	4.1 s $\lvert S\rvert = 6.2e+06$ $\lvert BDD\rvert = 2.9e+07$	114.2 s $\lvert S\rvert = 4.2e+08$ $\lvert BDD\rvert = 7.8e+07$	5617.9 $\lvert S\rvert = 3.2e+10$ $\lvert BDD\rvert = 4.2e+08$	NA	
	4v	1.3 s $\lvert S\rvert = 7.8e+06$ $\lvert BDD\rvert = 1.6e+07$	29.5 s $\lvert S\rvert = 3.6e+07$ $\lvert BDD\rvert = 5.4e+07$	898.8 s $\lvert S\rvert = 1.6+12$ $\lvert BDD\rvert = 2.0e+08$	NA		
	5v	5.7 s $\lvert S\rvert = 1.7e+08$ $\lvert BDD\rvert = 1.9e+07$	336.9 s $\lvert S\rvert = 2.3e+11$ $\lvert BDD\rvert = 1.1e+08$	NA			
	6v	12.1 s $\lvert S\rvert = 3.2e+09$ $\lvert BDD\rvert = 4.2e+07$	NA				
	7v	42.6 s $\lvert S\rvert = 7.6e+10$ $\lvert BDD\rvert = 5.3e+07$	NA				
	8v	73.2 s $\lvert S\rvert = 8.7e+12$ $\lvert BDD\rvert = 5.9e+07$	NA				
	9v	140.0 s $\lvert S\rvert = 2.6e+14$ $\lvert BDD\rvert = 6.0e+07$	NA				

Table 7: MCMAS statistics for anonymity checking on $\mathcal{G}_{count}$

7. Conclusions

In this paper we advance the state-of-the-art in model theory and verification for the strategy logic ATL* under imperfect information and imperfect recall. Specifically, we introduce a novel notion of (bi)simulation on imperfect information concurrent game structures that preserves the interpretation of formulas in ATL* both under the objective and subjective variants of the semantics (Theorem 8). Then, we apply this theoretical result to the verification of the ThreeBallot voting system, a relevant voting protocol without cryptography. We show how the ThreeBallot protocol can be captured within the framework of iCGS, and then provide successive abstractions that are provably bisimilar to the original iCGS for the ThreeBallot system. In particular, we have been able to model check the "smaller" bisimilar reductions of the ThreeBallot model, and then transfer the result to the original model in virtue of Theorem 8. As reported in the experimental results, the gains in terms of both time and memory resources are significant.

Future Work. We envisage several extensions of the present contribution. First, it is of interest to develop bisimulations for strategic properties of agents with perfect recall and bounded recall, as in many application domains agents do have some memory of past states and actions. Also for the verification of voting protocols, it is key to extend ATL* with epistemic modalities to express naturally properties of anonymity and confidentiality. We remarked that individual knowledge is expressible in the subjective semantics. However, no such result holds for the objective interpretation, nor common knowledge happens to be definable. Finally, we aim at automating and implementing the procedures described in this paper in a model checking tool for the formal verification of (electronic) voting protocols.

Acknowledgements. F. Belardinelli acknowledges the support of the ANR JCJC Project SVeDaS (ANR-16-CE40-0021). W. Jamroga acknowledges the support of the National Centre for Research and Development (NCBR), Poland, under the project VoteVerif (POLLUX-IV/1/2016).

[1] N. Bulling, J. Dix, W. Jamroga, Model checking logics of strategic ability: Complexity, in: Specification and Verification of Multi-agent Systems, Springer, 2010, pp. 125–159.

[2] N. Bulling, W. Jamroga, Comparing variants of strategic ability: how uncertainty and memory influence general properties of games, Autonomous Agents and Multi-Agent Systems 28 (3) (2014) 474–518.

[3] V. Goranko, W. Jamroga, Comparing semantics for logics of multi-agent systems, Synthese 139 (2) (2004) 241–280.

[4] R. Alur, T. A. Henzinger, O. Kupferman, Alternating-time temporal logic, in: Proceedings of the 38th IEEE Symposium on Foundations of Computer Science (FOCS97), IEEE Computer Society, 1997, pp. 100–109.

[5] R. Alur, T. A. Henzinger, O. Kupferman, Alternating-time temporal logic, Journal of the ACM 49 (5) (2002) 672–713.

[6] F. Laroussinie, N. Markey, Augmenting atl with strategy contexts, Information and Computation (245) (2015) 98–123.

[7] K. Chatterjee, T. Henzinger, N. Piterman, Strategy logic, in: Proceedings of the 18th International Conference on Concurrency Theory (CONCUR07), Vol. 4703, 2007, pp. 59–73.

[8] F. Mogavero, A. Murano, G. Perelli, M. Y. Vardi, Reasoning about strategies: On the model-checking problem, ACM Transactions in Computational Logic 15 (4) (2014) 34:1–34:47. doi:10.1145/2631917.
URL `http://doi.acm.org/10.1145/2631917`

[9] M. Pauly, A modal logic for coalitional power in games, Journal of Logic and Computation 12 (1) (2002) 149–166. doi:10.1093/logcom/12.1.149.
URL `http://dx.doi.org/10.1093/logcom/12.1.149`

[10] T. Ball, O. Kupferman, An abstraction-refinement framework for multi-agent systems, in: Proceedings of Logic in Computer Science (LICS), 2006, pp. 379–388. doi:10.1109/LICS.2006.10.

[11] W. van der Hoek, W. Jamroga, M. Wooldridge, A logic for strategic reasoning, in: Proceedings of the 4th international joint conference on Autonomous agents and multiagent systems (AAMAS05), ACM Press, New York, NY, USA, 2005, pp. 157–164.

[12] F. Mogavero, A. Murano, M. Vardi, Reasoning About Strategies, in: Proceedings of the 30th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS10), Vol. 8, Schloss Dagstuhl, 2010, pp. 133–144.

[13] P. Gammie, R. van der Meyden, MCK: Model checking the logic of knowledge, in: Proceedings of 16th International Conference on Computer Aided Verification (CAV04), Vol. 3114 of Lecture Notes in Computer Science, Springer, 2004, pp. 479–483.

[14] M. Kacprzak, W. Nabialek, A. Niewiadomski, W. Penczek, A. Pólrola, M. Szreter, B. Woźna, A. Zbrzezny, Verics 2007 - a model checker for knowledge and real-time, Fundamenta Informaticae 85 (1) (2008) 313–328.

[15] A. Lomuscio, H. Qu, F. Raimondi, MCMAS: A model checker for the verification of multi-agent systems, Software Tools for Technology TransferHttp://dx.doi.org/10.1007/s10009-015-0378-x. doi:10.1007/s10009-015-0378-x.
URL `http://dx.doi.org/10.1007/s10009-015-0378-x`

[16] W. Jamroga, J. Dix, Model checking abilities under incomplete information is indeed δ_p^2-complete, in: Proceedings of the 4th European Workshop on Multi-Agent Systems EUMAS'06, Citeseer, 2006, pp. 14–15.

[17] C. Dima, F. L. Tiplea, Model-checking ATL under imperfect information and perfect recall semantics is undecidable, CoRR abs/1102.4225.
URL `http://arxiv.org/abs/1102.4225`

[18] F. Belardinelli, A. Lomuscio, A. Murano, S. Rubin, Verification of multi-agent systems with imperfect information and public actions, in: 17, 2017, pp. 1268–1276.

[19] F. Belardinelli, A. Lomuscio, A. Murano, S. Rubin, Verification of broadcasting multi-agent systems against an epistemic strategy logic, in: IJCAI'17, 2017, pp. 91–97. doi:10.24963/ijcai.2017/14.
URL `https://doi.org/10.24963/ijcai.2017/14`

[20] P. Cermák, A. Lomuscio, F. Mogavero, A. Murano, Practical verification of multi-agent systems against slk specifications, Information and Computation 261 (Part) (2018) 588–614. doi:10.1016/j.ic.2017.09.011.
URL `https://doi.org/10.1016/j.ic.2017.09.011`

[21] W. Jamroga, M. Knapik, D. Kurpiewski, Fixpoint approximation of strategic abilities under imperfect information, in: K. Larson,

M. Winikoff, S. Das, E. H. Durfee (Eds.), Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017, ACM, 2017, pp. 1241–1249.
URL `http://dl.acm.org/citation.cfm?id=3091298`

[22] P. Cousot, R. Cousot, Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints, in: Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, 1977, pp. 238–252. doi:10.1145/512950.512973.

[23] E. M. Clarke, O. Grumberg, D. Long, Model checking and abstractions, ACM Transactions on Programming Languages and Systems 16 (5) (1994) 1512–1542.

[24] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, H. Veith, Counterexample-guided abstraction refinement, in: Proceedings of the 12th International Conference on Computer Aided Verification (CAV00), Vol. 1855 of Lecture Notes in Computer Science, Springer, 2000, pp. 154–169.

[25] L. de Alfaro, P. Godefroid, R. Jagadeesan, Three-valued abstractions of games: Uncertainty, but with precision, in: Proceedings of Logic in Computer Science (LICS), IEEE Computer Society, 2004, pp. 170–179.

[26] A. Lomuscio, J. Michaliszyn, Verification of multi-agent systems via predicate abstraction against ATLK specifications, in: Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2016, pp. 662–670.

[27] F. Belardinelli, A. Lomuscio, Agent-based abstractions for verifying alternating-time temporal logic with imperfect information, in: Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), ACM, 2017, pp. 1259–1267.

[28] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Vol. 53 of Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2001.

[29] R. Alur, T. A. Henzinger, O. Kupferman, M. Y. Vardi, Alternating refinement relations, in: In Proceedings of the Ninth International Con-

ference on Concurrency Theory (CONCUR'98), volume 1466 of LNCS, Springer-Verlag, 1998, pp. 163–178.

[30] U. Goltz, R. Kuiper, W. Penczek, Propositional temporal logics and equivalences, in: Proceedings of CONCUR '92, 1992, pp. 222–236. doi:10.1007/BFb0084794.

[31] T. Ågotnes, V. Goranko, W. Jamroga, Alternating-time temporal logics with irrevocable strategies, in: Proceedings of TARK XI, 2007, pp. 15–24.

[32] W. Jamroga, Logical Methods for Specification and Verification of Multi-Agent Systems, ICS PAS Publishing House, 2015.

[33] P. A. Ryan, The computer ate my vote, in: P. Boca, J. P. Bowen, J. Siddiqi (Eds.), Formal Methods: State of the Art and New Directions, Springer Verlag, 2009, Ch. 5, pp. 148–184.

[34] P. Y. A. Ryan, S. A. Schneider, V. Teague, End-to-end verifiability in voting systems, from theory to practice, IEEE Security & Privacy 13 (3) (2015) 59–62. doi:10.1109/MSP.2015.54.

[35] B. Beckert, R. Goré, C. Schürmann, T. Bormer, J. Wang, Verifying voting schemes, J. Inf. Secur. Appl. 19 (2) (2014) 115–129. doi:10.1016/j.jisa.2014.04.005.
URL `http://dx.doi.org/10.1016/j.jisa.2014.04.005`

[36] V. Cortier, Formal verification of e-voting: Solutions and challenges, ACM SIGLOG News 2 (1) (2015) 25–34. doi:10.1145/2728816.2728823.
URL `http://doi.acm.org/10.1145/2728816.2728823`

[37] S. Delaune, S. Kremer, M. Ryan, Verifying Privacy-Type Properties of Electronic Voting Protocols, Journal of Computer Security 17 (4) (2009) 435–487.

[38] C. A. R. Hoare, Communicating Sequential Processes, Commun. ACM 21 (8) (1978) 666–677.

[39] S. Schneider, A. Sidiropoulos, CSP and Anonymity, in: Proceedings of the 1996 European Symposium on Research in Computer Security (ESORICS'96), Vol. 1146 of Lecture Notes in Computer Science, Springer-Verlag, 1996, pp. 198–218.

[40] I. Cervesato, N. A. Durgin, P. Lincoln, J. C. Mitchell, A. Scedrov, A Meta-Notation for Protocol Analysis, in: Proceedings of the 12th IEEE Computer Security Foundations Workshop (CSFW'99), IEEE Computer Society, 1999, pp. 55–69.

[41] G. Denker, J. K. Millen, Modeling Group Communication Protocols Using Multiset Term Rewriting, Electr. Notes Theor. Comput. Sci. 71 (2002) 20–39.

[42] I. Boureanu, A. V. Jones, A. Lomuscio, Automatic Verification of Epistemic Specifications Under Convergent Equational Theories, in: Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'12), IFAAMAS, 2012, pp. 1141–1148.

[43] M. Tabatabaei, W. Jamroga, P. Y. A. Ryan, Expressing receipt-freeness and coercion-resistance in logics of strategic ability: Preliminary attempt, in: Proceedings of the 1st International Workshop on AI for Privacy and Security, PrAISe@ECAI 2016, ACM, 2016, pp. 1:1–1:8. doi:10.1145/2970030.2970039.

[44] R. L. Rivest, The ThreeBallot Voting System, http://theory.lcs.mit.edu/ rivest/Rivest-TheThreeBallotVotingSystem.pdf (October 2006).

[45] R. Rivest, W. Smith, Three voting protocols: ThreeBallot, VAV, and Twin, in: Proceedings of USENIX/ACCURATE Electronic Voting Technology Workshop (EVT), 2007.

[46] M. Moran, J. Heather, S. Schneider, Verifying anonymity in voting systems using csp, Formal Aspects of Computing 26 (1) (2014) 63–98. doi:10.1007/s00165-012-0268-x.
URL http://dx.doi.org/10.1007/s00165-012-0268-x

[47] M. Moran, J. Heather, S. Schneider, Automated anonymity verification of the ThreeBallot and VAV voting systems, Software & Systems Modeling 15 (4) (2016) 1049–1062. doi:10.1007/s10270-014-0445-x.
URL http://dx.doi.org/10.1007/s10270-014-0445-x

[48] F. Belardinelli, R. Condurache, C. Dima, W. Jamroga, A. V. Jones, Bisimulations for verifying strategic abilities with an application to

threeballot, in: Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS17), 2017.

[49] F. Belardinelli, C. Dima, A. Murano, Bisimulations for logics of strategies: A study in expressiveness and verification, in: M. Thielscher, F. Toni, F. Wolter (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018., AAAI Press, 2018, pp. 425–434.
URL `https://aaai.org/ocs/index.php/KR/KR18/paper/view/17992`

[50] M. Dastani, W. Jamroga, Reasoning about strategies of multi-agent programs, in: Proceedings of AAMAS2010, 2010, pp. 625–632.

[51] M. Melissen, Game theory and logic for non-repudiation protocols and attack analysis, Ph.D. thesis, University of Luxembourg (2013).

[52] Ştefan Ciobâcă, Verification and composition of security protocols with applications to electronic voting. (vérification et composition des protocoles de securité avec des applications aux protocoles de vote electronique), Ph.D. thesis, École normale supérieure de Cachan, France (2011).

[53] B. Meng, W. Huang, D. Wang, Automatic verification of remote internet voting protocol in symbolic model, JNW 6 (9) (2011) 1262–1271.

[54] S. Delaune, S. Kremer, M. Ryan, Verifying privacy-type properties of electronic voting protocols, Journal of Computer Security 17 (4) (2009) 435–487.

[55] H. DeYoung, C. Schürmann, Linear logical voting protocols, in: VoteID, E-Voting and Identity - Third International Conference, Tallinn, Estonia, Vol. 7187 of Lecture Notes in Computer Science, Springer, 2011, pp. 53–70.

[56] B. Beckert, R. Goré, C. Schürmann, Analysing vote counting algorithms via logic - and its application to the CADE election scheme, in: Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, Vol. 7898 of Lecture Notes in Computer Science, Springer, 2013, pp. 135–144.

[57] D. Pattinson, C. Schürmann, Vote counting as mathematical proof, in: AI 2015: Advances in Artificial Intelligence - 28th Australasian Joint Conference, Canberra, ACT, Australia, Vol. 9457 of Lecture Notes in Computer Science, Springer, 2015, pp. 464–475.

[58] A. Bruni, E. Drewsen, C. Schürmann, Towards a mechanized proof of selene receipt-freeness and vote-privacy, in: E-Vote-ID, Electronic Voting - Second International Joint Conference, Bregenz, Austria, Vol. 10615 of Lecture Notes in Computer Science, Springer, 2017, pp. 110–126.

[59] B. Beckert, R. Goré, C. Schürmann, On the specification and verification of voting schemes, in: Vote-ID, E-Voting and Identify - 4th International Conference, Guildford, UK, Vol. 7985 of Lecture Notes in Computer Science, Springer, 2013, pp. 25–40.

[60] B. Beckert, R. Goré, C. Schürmann, T. Bormer, J. Wang, Verifying voting schemes, J. Inf. Sec. Appl. 19 (2) (2014) 115–129.

[61] C. Schürmann, A risk-limiting audit in denmark: A pilot, in: E-Vote-ID, Electronic Voting - First International Joint Conference, Bregenz, Austria, Vol. 10141 of Lecture Notes in Computer Science, Springer, 2016, pp. 192–202.

[62] C. Schürmann, Certifying voting protocols, in: ITP, Interactive Theorem Proving - 4th International Conference, Rennes, France, Vol. 7998 of Lecture Notes in Computer Science, Springer, 2013, p. 17.

[63] V. Cortier, Formal verification of e-voting: solutions and challenges, SIGLOG News 2 (1) (2015) 25–34.

[64] W. Jamroga, M. Knapik, D. Kurpiewski, Model checking the SELENE e-voting protocol in multi-agent logics, in: Proceedings of the 3rd International Joint Conference on Electronic Voting (E-VOTE-ID), Vol. 11143 of Lecture Notes in Computer Science, Springer, 2018, pp. 100–116.

[65] W. Jamroga, W. van der Hoek, Agents that know how to play, Fundamenta Informaticae 62 (2004) 1–35.

[66] P. Y. Schobbens, Alternating-time logic with imperfect recall, Electronic Notes in Theoretical Computer Science 85 (2) (2004) 82–93.

[67] N. Bulling, W. Jamroga, Alternating epistemic mu-calculus, in: Proceedings of IJCAI-11, 2011, pp. 109–114.

[68] J. Benaloh, D. Tuinstra, Receipt-free secret-ballot elections, in: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, ACM, 1994, pp. 544–553.

[69] S. Delaune, S. Kremer, M. D. Ryan, Receipt-freeness: Formal definition and fault attacks, in: Proceedings of the Workshop Frontiers in Electronic Elections (FEE 2005), Milan, Italy, Citeseer, 2005.

[70] A. Juels, D. Catalano, M. Jakobsson, Coercion-resistant electronic elections, in: Proceedings of the 2005 ACM workshop on Privacy in the electronic society, ACM, 2005, pp. 61–70.

[71] M. Tabatabaei, W. Jamroga, P. Ryan, Expressing receipt-freeness and coercion-resistance in logics of strategic ability: Preliminary attempt, in: Proceedings of the 1st International Workshop on AI for Privacy and Security PrAISe 2016, ACM, 2016, pp. 1:1–1:8.
URL http://krak.ipipan.waw.pl/ wjamroga/papers/coercionInATL16praise.pdf

[72] A. Lomuscio, F. Raimondi, The complexity of model checking concurrent programs against CTLK specifications, in: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS06), ACM Press, 2006, pp. 548–550.

[73] M. Cohen, M. Dam, A. Lomuscio, F. Russo, Abstraction in model checking multi-agent systems, in: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '09, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2009, pp. 945–952.
URL http://dl.acm.org/citation.cfm?id=1558109.1558144

[74] F. Belardinelli, R. Condurache, C. Dima, M. Knapik, W. Jamroga, ThreeBallot ISPL Models, https://bitbucket.org/catadima/three-ballot-ispl-models/.

[75] W. van der Hoek, A. Lomuscio, M. Wooldridge, On the complexity of practical atl model checking knowledge, strategies, and games in multi-agent systems, in: Proceedings of the fifth international joint confer-

ence on Autonomous agents and multiagent systems (AAMAS06), ACM Press, 2006, pp. 201–208.

[76] S. Busard, C. Pecheur, H. Qu, F. Raimondi, Reasoning about memoryless strategies under partial observability and unconditional fairness constraints, Information and Computation 242 (2015) 128–156.

[77] J. van Eijck, S. Orzan, Epistemic verification of anonymity, Electr. Notes Theor. Comput. Sci. 168 (2007) 159–174. doi:10.1016/j.entcs.2006.08.026.
URL `http://dx.doi.org/10.1016/j.entcs.2006.08.026`

[78] R. Küsters, T. Truderung, A. Vogt, Verifiability, privacy, and coercion-resistance: New insights from a case study, in: 32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA, IEEE Computer Society, 2011, pp. 538–553. doi:10.1109/SP.2011.21.
URL `http://dx.doi.org/10.1109/SP.2011.21`

[79] R. Chadha, S. Kremer, A. Scedrov, Formal analysis of multi-party contract signing, J. Autom. Reasoning 36 (1-2) (2006) 39–83. doi:10.1007/s10817-005-9019-5.

[80] M. R. Neuhäußer, J. Katoen, Bisimulation and logical preservation for continuous-time Markov Decision Processes, in: CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings, Vol. 4703 of Lecture Notes in Computer Science, Springer, 2007, pp. 412–427. doi:10.1007/978-3-540-74407-8_28.