
eleventh
international
conference
on
autonomous
agents and
multiagent
systems
AAMAS
2012
4th- 8th June 2012
Valencia
W16
Workshop on
Logical
Aspects of
Multiagent
Systems
(LAMAS)

Proceedings of the 5th Workshop on Logical Aspects of Multi-Agent Systems LAMAS 2012

Satellite workshop of AAMAS 2012, June 5, 2012, Valencia, Spain

Edited by Valentin Goranko and Wojciech Jamroga

Preface

LAMAS is a scientific network spanning an interdisciplinary community of researchers working on logical aspects of MAS from the perspectives of logic, artificial intelligence, computer science, game theory, etc. The LAMAS workshop is the pivotal event of the network and it provides a platform for presentation, exchange, and publication of ideas in all these areas, including:

- Logical systems for specification, analysis, and reasoning about MAS
- Modeling MAS with logic-based models
- Deductive systems and decision procedures for logics for MAS
- Development, complexity analysis, and implementation of algorithmic methods for formal verification of MAS
- Logic-based tools for MAS
- Applications of logics in MAS

Programme Committee

Thomas Agotnes, University of Bergen
Natasha Alechina, University of Nottingham
Thomas Bolander, Technical University of Denmark
Nils Bulling, Clausthal University of Technology
Hans van Ditmarsch, University of Sevilla
Juergen Dix, Clausthal University of Technology
Barbara Dunin-Keplicz, University of Warsaw
Valentin Goranko, Technical University of Denmark
Andreas Herzig, University of Toulouse
Wojtek Jamroga, University of Luxembourg
Joao Leite, New University of Lisbon
Alessio Lomuscio, Imperial College London
John-Jules Meyer, Utrecht University
Wojciech Penczek, Polish Academy of Sciences, Warsaw
Ram Ramanujam, IMSc Chennai
Dirk Walther , Universidad Politcnica de Madrid

Additional reviewers:

Martin Slota, New University of Lisbon

Invited speakers:

Wiebe van der Hoek, University of Liverpool
Alessio Lomuscio, Imperial College London

Workshop organizers

The workshop is organized by Valentin Goranko, Technical University of Denmark, and Wojtek Jamroga, University of Luxembourg.

Contents

Local Properties in Modal Logic

Wiebe van der Hoek
(joint work with Hans van Ditmarsch and Barteld Kooi)

University of Liverpool

Abstract

In modal logic, when adding a syntactic property to an axiomatisation, this property will semantically become true in all models, in all situations, under all circumstances. For instance, adding a property like $K_a p \rightarrow K_b p$ (agent b knows at least what agent a knows) to an axiomatisation of some epistemic logic has as an effect that such a property becomes *globally* true, i.e., it will hold in all states, at all time points (in a temporal setting), after every action (in a dynamic setting) and after any communication (in an update setting), and every agent will know that it holds, it will even be common knowledge. We propose a way to express that a property like the above only needs to hold *locally*: it may hold in the actual state, but not in all states, and not all agents may know that it holds. We achieve this by adding relational atoms to the language that represent (implicitly) quantification over all formulas, as in $\forall p(K_a p \rightarrow K_b p)$. We show how this can be done for a rich class of modal logics and a variety of syntactic properties. We then study the epistemic logic enriched with the syntactic property 'knowing at least as much as' in more detail. We show that the enriched language is not preserved under bisimulations. We also demonstrate that adding public announcements to this enriched epistemic logic makes it more expressive, which is for instance not true for the 'standard' epistemic logic $S5$.

Verification of Artifact-Centric Multi-Agent Systems

Alessio Lomuscio
(joint work with F Belardinelli and F Patrizi)

Imperial College London

Abstract

Artifact-Centric systems are a particular kind of web-services where data feature prominently in the system description. The emphasis on data makes automata-based formalisms commonly used to model services insufficient and calls for the explicit representation of the evolution of the underlying databases. In this talk I will explore the verification problem for artifact-centric multi-agent systems, i.e., systems of agents interacting through artifact systems. I will point to the undecidability of the model checking problem of these systems when analysed against specifications based on first-order temporal-epistemic logic. I will then analyse conditions that enable us to obtain a decidable problem through finite abstractions that are bisimilar to a given model.

The talk is based on results published at IJCAI2011, ICSOC2011, and KR2012.

Embedding of Coalition Logic into a Normal Multi-Modal Logic

Thomas Ågotnes[1] and Natasha Alechina[2]

[1] Department of Information Science and Media Studies,
University of Bergen, Norway
thomas.agotnes@infomedia.uib.no
[2] School of Computer Science
University of Nottingham, UK
nza@cs.nott.ac.uk

1 Introduction

We define a satisfiability-preserving embedding of coalition logic into a fragment of a standard normal modal logic, namely multi-modal K with intersection. An advantage of standard, normal, modal logics is a well understood theoretical foundation and the availability of tools for automated verification and reasoning. Multi-modal K with intersection is a fragment of Boolean Modal Logic [3] which has been extensively studied (and implemented) as a variant of propositional dynamic logic with intersection and also by researchers in description logic (see for example [6]). Although other logics that are normal modal logics and/or have PDL-type operators and can express coalition operators have been studied recently [2, 5], these typically have non-standard syntactic operators and/or non-standard semantics. In this abstract we define a fragment of multi-modal K with intersection of modalities interpreted over models corresponding to game structures. This extended abstract is partly based on [1].

The idea of the embedding is as follows. Consider a formula ϕ of Coalition Logic (CL) [7]. As a side effect of the completeness proof for Alternating Time Temporal Logic (ATL) in [4], if ϕ has a concurrent game structure (CGS) model, then it has a CGS model where each agent has at most $k + 1$ actions, where k is the number of formulas of the form $[C]\psi$ or $\neg[C]\psi$ in $ecl(\phi)$, and $ecl(\phi)$ is the set of subformulas of ϕ closed under single negations and the condition that if $[C]\psi \in ecl(\phi)$, then $[C']\psi \in ecl(\phi)$ for all $C' \subseteq N$. We will refer to set of actions required for constructing a satisfying model for ϕ as Act^{ϕ}.

In this paper we use this fact ('bounded action property' of coalition logic) to provide an embedding of CL into a normal modal logic. This normal modal logic is based on K_n with intersection of modalities and some additional restrictions on models, explained below. We call it a logic of joint action.

The main result is as follows:

Theorem 1. *A CL formula ϕ is satisfiable iff $T^{Act^{\phi}}(\phi)$ has a joint action logic model.*

where $T^{Act^{\phi}}$ is defined as follows (where g is the number of agents mentioned in ϕ):

- $T^{Act^{\phi}}(p) = p$

- $T^{Act^\phi}(\neg\phi) = \neg T^{Act^\phi}(\phi)$
- $T^{Act^\phi}(\phi \wedge \psi) = T^{Act^\phi}(\phi) \wedge T^{Act^\phi}(\psi)$
- $T^{Act^\phi}([\{1,\ldots,m\}]\psi) \equiv \bigvee_{a_1,\ldots,a_m \in Act^\phi}($ $\bigvee_{a_{m+1},\ldots,a_g \in Act^\phi} \langle (1,a_1) \cap \ldots \cap (m,a_m) \cap (m+1,a_{m+1}) \cap \ldots \cap (g,a_g)\rangle \top \wedge$ $\bigwedge_{a_{m+1},\ldots,a_g \in Act^\phi} [(1,a_1) \cap \ldots \cap (m,a_m) \cap (m+1,a_{m+1}) \cap \ldots \cap (g,a_g)] T^{Act^\phi}(\psi))$

In the rest of this extended abstract, we define joint action logic and state some theorems about it.

2 Joint action logic

First we briefly define K_n (where n is the number of atomic modalities) with intersection of modalities.

First we define the language of $K_n^\cap$. Assume a set of primitive propositions Θ and actions A:

$$\phi ::= p \in \Theta \mid \neg\phi \mid \phi \wedge \phi \mid [\pi]\phi$$

$$\pi ::= a \in A \mid \pi \cap \pi$$

As usual, $\langle\pi\rangle\phi$ is defined as $\neg[\pi]\neg\phi$.

A $K_n^\cap$ model M is a tuple $\langle S, V, \{R_\pi : \pi \in \Pi\}\rangle$ where

- S is a set of *states*;
- $V : S \to 2^\Theta$ is a *valuation function*;
- For each $\pi \in \Pi$, $R_\pi \subseteq S \times S$
- $R_{\pi_1 \cap \pi_2} = R_{\pi_1} \cap R_{\pi_2}$ (INT)

The modality truth definition clause:

$$M, s \models [\pi]\phi \text{ iff } \forall (s,s') \in R_\pi, M, s' \models \phi$$

Now we impose additional conditions on $K_n^\cap$ models to define joint action models. Let Act be a finite set of actions and N a set of g agents. Define a set of atomic modalities as follows:

$$A = N \times Act$$

An atomic modality in A is an *individual action* A composite modality $\pi = \pi_1 \cap \pi_2$ is a *joint action*. Joint actions of the form $(1,a_1) \cap \ldots \cap (g,a_g)$ with one individual action for *every* agent in N will be called *complete (joint) actions*.

A $K_n^\cap$ model over A (where Act is finite) is a *joint action model* if it satisfies:

Seriality (SER) For any state s and agent i, at least one action is enabled in s for i (where a is enabled for i in s means that there is a state accessible from s by $R_{(i,a)}$).

Independent Choice (IC) For any state s, agents $C = \{i_1,\ldots,i_k\}$ and actions $a_1,\ldots,a_k \in Act$, if for every j a_j is enabled for i_j in s, then there is a state s' such that $(s,s') \in R_{(i_1,a_1)\cap\cdots\cap(i_k,a_k)}$.

Deterministic Joint Actions (DJA) For any complete joint action α and states s, s_1, s_2, $(s, s_1), (s, s_2) \in R_\alpha$ implies that $s_1 = s_2$.

Unique Joint Actions (UJA) For any complete joint actions α and β and states s, t, if $(s, t) \in R_\alpha \cap R_\beta$ then $\alpha = \beta$.

Theorem 2. *The logic of joint action models is completely axiomatised by the following set of axioms:*

K $[\pi](\phi \to \psi) \to ([\pi]\phi \to [\pi]\psi)$
A1 $\bigvee_{a \in Act} \langle (i, a) \rangle \top$
A2 $\langle \pi \rangle \phi \to \bigvee_{a \in Act} \langle \pi \cap (i, a) \rangle \phi$
A3 $\bigwedge_{i \in N} \langle (i, a_i) \rangle \top \to \langle (1, a_1) \cap \ldots \cap (g, a_g) \rangle \top$
A4 $\langle (1, a_1) \cap \cdots \cap (g, a_g) \rangle \phi \to [(1, a_1) \cap \ldots \cap (g, a_g)]\phi$
A5 $[\pi]\phi \to [\pi \cap \pi']\phi$
A6 $[(i, a) \cap (i, b)]\bot$ *when* $a \neq b$
MP *From* $\phi \to \psi$ *and* ϕ *infer* ψ
G *From* ϕ *infer* $[\pi]\phi$

Theorem 3. *The complexity of satisfiability problem of formulas in joint action models is PSPACE-complete.*

References

1. Ågotnes, T., Alechina, N.: Reasoning about joint action and coalitional ability in K_n with intersection. In: Leite, J., Torroni, P., Ågotnes, T., Boella, G., van der Torre, L. (eds.) Computational Logic in Multi-Agent Systems - 12th International Workshop (CLIMA XII, Proceedings. Lecture Notes in Computer Science, vol. 6814, pp. 139–156. Springer (2011)
2. Broersen, J., Herzig, A., Troquard, N.: A normal simulation of coalition logic and an epistemic extension. In: Samet, D. (ed.) Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2007), Brussels, Belgium, June 25-27, 2007. pp. 92–101 (2007)
3. Gargov, G., Passy, S.: A note on boolean modal logic. In: Mathematical Logic. Proc. of The Summer School and Conf. on Mathematical Logic "Heyting'88". pp. 311–321. Plenum Press, New York (1988)
4. Goranko, V., van Drimmelen, G.: Complete axiomatization and decidability of alternating-time temporal logic. Theor. Comput. Sci. 353(1-3), 93–117 (2006)
5. Lorini, E.: A dynamic logic of agency II: Deterministic DLA, coalition logic, and game theory. Journal of Logic, Language and Information 19, 327–351 (2010)
6. Lutz, C., Sattler, U.: The complexity of reasoning with boolean modal logics. In: Wolter, F., Wansing, H., de Rijke, M., Zakharyaschev, M. (eds.) Advances in Modal Logic. vol. 3, pp. 329–348. World Scientific (2002)
7. Pauly, M.: A modal logic for coalitional power in games. Journal of Logic and Computation 12(1), 149–166 (2002)

No big deal: introducing roles to reduce the size of ATL models

Sjur Dyrkolbotn[1], Piotr Kaźmierczak[2,3]*, Erik Parmann[1], and Truls Pedersen[3]

[1] Department of Informatics, University of Bergen
[2] Department of Computing, Mathematics and Physics, Bergen University College
[3] Department of Information Science and Media Studies, University of Bergen
sjur.dyrkolbotn@uib.no, phk@hib.no, erik.parmann@uib.no,
truls.pedersen@uib.no

Abstract. In the following paper we present a new semantics for the well-known strategic logic ATL. It is based on adding *roles* to concurrent game structures, that is at every state, each agent belongs to exactly one role, and the role specifies what actions are available to him at that state. We show advantages of the new semantics, analyze model checking complexity and prove equivalence between standard ATL semantics and our new approach.

1 Introduction

One of the most intensively studied [8, 9, 16] areas of research in the field of multi-agent systems are *strategic* or *cooperation* logics – formalisms that allow for reasoning about agents' strategies and behavior in a multi-agent setting. Two of the most known logics are Marc Pauly's Coalition Logic (CL) [10, 11] and Alur, Henzinger and Kupferman's Alternating-time Temporal Logic (ATL) [5], which can be considered a temporal extension of Coalition Logic. Both these logics gained much popularity and generated a 'zoo' of derivatives [15, 13, 1, 3, 2].

This popularity is in no small part due to relative high expressive power of both CL and ATL, but also due to low complexity of model checking problems for these respective logics. Model checking of Coalition Logic can be solved in polynomial time in the size of the model and the length of the formula [10]. It remains polynomial for ATL as well [5], which is considered a very good result. However, as investigated by Jamroga and Dix [7], in both cases the number of agents must be *fixed*. If it is not then model checking of ATL models represented as *alternating transition systems* is NP-complete, and if the models are represented as *concurrent game structures* (CGS) it becomes Σ_2^{P}-complete. Also, van der Hoek, Lomuscio and Wooldridge show [14] that complexity of model checking for ATL is sensitive to model representation. It is polynomial only if an *explicit* enumeration of *all* components of the model is assumed. For models represented in a *(simplified) reactive modules language* (RML) [4] complexity of

* Piotr Kaźmierczak's research was supported by the Research Council of Norway project 194521 (FORMGRID).

model checking for ATL becomes as hard as the satisfiability problem for this logic, namely EXPTIME [14].

We present an alternative semantics that interprets formulas of ordinary ATL over concurrent game structures with *roles*. As we describe in Section 2.1, such structures introduce an extra element – a set R of roles. Agents belonging to the same role are considered *homogeneous* in the sense that all consequences of their actions are captured by considering only the number of *votes* an action gets (one vote per agent). We give some examples that motivate our approach and prove equivalence with ATL based on concurrent game structures. We then discuss model checking, showing it to be of polynomial complexity in the size of models. This seems significant, since as long as the number of roles remain fixed, the size of our models does *not* grow exponentially in the number of players.

The structure of our paper is as follows. We present a revised formalism for ATL in Section 2, prove equivalence with the standard one in Section 3, discuss model checking results in Section 4 and conclude in Section 5.

2 Role-based semantics for ATL

The language of ordinary ATL is the following, as presented in [5]:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid \langle\!\langle A \rangle\!\rangle \bigcirc \phi \;\mid\; \langle\!\langle A \rangle\!\rangle \Box \phi \;\mid \langle\!\langle A \rangle\!\rangle \phi \mathcal{U} \phi$$

where p is propositional letter, and A is a coalition of agents. We follow standard abbreviations (e.g. $\langle\!\langle\ \rangle\!\rangle$ for $\langle\!\langle \emptyset \rangle\!\rangle$) and skip connectives that are derivable.

2.1 Concurrent Game Structures with Roles

In this section we will introduce *concurrent game structures with roles* (RCGS) and consider some examples. We will be using the notation $[n] = \{1, \ldots, n\}$, and we will let A^B denote the set of functions from B to A. We will often work with tuples $v = \langle v_1, \ldots, v_n \rangle$ and we will often view v as a function with domain $[n]$ and write $v(i)$ for v_i. We will do addition and subtraction on tuples of the same arity component by component, e.g. for $v = \langle v_1, \ldots, v_n \rangle, v' = \langle v'_1, \ldots, v'_n \rangle$, $v - v' = \langle v_1 - v'_1, \ldots, v_n - v'_n \rangle$. Given a function $f : A \times B \to C$ and $a \in A$, we will use f_a to denote the function $B \to C$ defined by $f_a(b) = f(a, b)$ for all $b \in B$.

Definition 2.1. *An* RCGS *is a tuple* $H = \langle \mathcal{A}, R, \mathcal{R}, Q, \Pi, \pi, \mathbb{A}, \delta \rangle$ *where:*

- $\mathcal{A}$ *is a non-empty set of players. In this text we assume* $\mathcal{A} = [n]$ *for some* $n \in \mathbb{N}$*, and we will reserve* n *to mean the number of agents.*
- Q *is the non-empty set of states.*
- R *is a non-empty set of roles. In this text we assume* $R = [i]$ *for some* $i \in \mathbb{N}$.
- $\mathcal{R} : Q \times R \to \wp(\mathcal{A})$*, such that for every* $q \in Q$ *we have*
  - *For all* $r, r' \in R$*, if* $r \neq r'$ *then* $\mathcal{R}(q, r) \cap \mathcal{R}(q, r') = \emptyset$
  - $\bigcup_{r \in R} \mathcal{R}(q, r) = \mathcal{A}$

For a coalition $A \subseteq \mathcal{A}$ we write $A_{r,q}$ for the agents in A which belong to role r at q, i.e. $A_{r,q} = \mathcal{R}(q, r) \cap A$.

- *Π is a set of propositional letters and $\pi : Q \to \wp(\Pi)$ maps states to the propositions true at that state.*
- *$\mathbb{A} : Q \times R \to \mathbb{N}^+$ is the number of available actions in a given state for a given role.*
- *For $\mathcal{A} = [n] = \{1, \dots n\}$, we say that the set of* complete *votes for a role r in a state q is $V_r(q) = \{v_{r,q} \in [n]^{[\mathbb{A}(q,r)]} \mid \sum_{1 \le a \le \mathbb{A}(q,r)} v_{r,q}(a) = |\mathcal{R}(q, r)|\}$, the set of functions from the available actions to the number of agents performing the action. The functions in this set account for the actions of* all *the agents. The set of* complete *profiles at q is $P(q) = \prod_{r \in R} V_r(q)$. For each $q \in Q$ we have a transition function at q, $\delta_q : P(q) \to Q$ defining a partial function $\delta : Q \times \bigcup_{q \in Q} P(q) \to Q$ such that for all $q \in Q$, $P \in P(q)$, $\delta(q, P) = \delta_q(P)$*

To illustrate how RCGS differs from an ordinary concurrent game structure, we provide some examples.

Example 2.1. We construct an example similar to the well-known train-controller scenario [5], but in contrast to the original, in our scenario there are n_t trains. Consider a turn-based synchronous game structure with roles $S_{train} = \langle \mathcal{A}, R, \mathcal{R}, Q, \Pi, \pi, \mathbb{A}, \delta \rangle$ where:

- $\mathcal{A} = \{1, \dots, n_t, n_t + 1\}$. There are n_t trains and one controller.
- $R = \{train, ctr\}$. There are two roles: one for trains and one for the controller.
- $Q = \{q_0, q_1, q_2, q_3\}$.
- $\mathcal{R}(q_i, train) = [n_t]$, and $\mathcal{R}(q_i, ctr) = \{n_t + 1\}$, for all $q_i \in Q$.
- $\Pi = \{out\_of\_gate, in\_gate, request, grant\}$
- $\pi(q_0) = \{out\_of\_gate\}$, $\pi(q_1) = \{out\_of\_gate, request\}$, $\pi(q_2) = \{out\_of\_gate, grant\}$, $\pi(q_3) = \{in\_gate\}$.
- $\mathbb{A}(q_0, train) = 2$, $\mathbb{A}(q_0, ctr) = 1$, $\mathbb{A}(q_1, train) = 1$, $\mathbb{A}(q_1, ctr) = 3$, $\mathbb{A}(q_2, train) = 2$, $\mathbb{A}(q_2, ctr) = 1$, $\mathbb{A}(q_3, train) = 1$, $\mathbb{A}(q_3, ctr) = 2$.
- and finally

$$
\begin{aligned}
&\delta(q_0, \langle (0, n_t), 1 \rangle) = \delta(q_1, \langle n_t, (1, 0, 0) \rangle) = \delta(q_2, \langle (0, n_t), 1 \rangle) \\
&\qquad = \delta(q_3, \langle (a, n_t - a), 1 \rangle) = q_0 && \text{where } 1 \le a \le n_t \\
&\delta(q_0, \langle (a, n_t - a), 1 \rangle) = \delta(q_1, \langle n_t, ((0, 1, 0)) \rangle) = q_1 && \text{where } 1 \le a \le n_t \\
&\delta(q_1, \langle n_t, (0, 0, 1) \rangle) = \delta(q_2, \langle (a, n_t - a), 1 \rangle) = q_2 && \text{where } 2 \le a \le n_t \\
&\delta(q_2, \langle (1, n_t - 1), 1 \rangle) = \delta(q_3, \langle (0, n_t), 1 \rangle) = q_3
\end{aligned}
$$

Figure 1 presents the example in a visual way. The model can be seen as a generalization of the classical train-controller example. In q_0 we stay in q_0 unless at least one train issues a request. In q_1 the controller behaves as before; it can postpone making a decision (staying in q_1), reject all requests (going to q_0), or accept the requests (going to q_2). In q_2 the trains can choose to enter the tunnel, but only one of them may do so; if nobody attempts to enter the grant is revoked (or relinquished), if more than one train attempts to enter we stay in q_2, and finally if (the trains reach an agreement and) only one train enters we go to q_3. In q_3 *any* train may decide that the train in the tunnel has to leave (returning to q_0), and the train in the tunnel *must* comply. This reflects the homogeneity among players in the trains role. The action of deciding to leave the tunnel is shared among all trains, and the train actually in the tunnel remains unidentified.

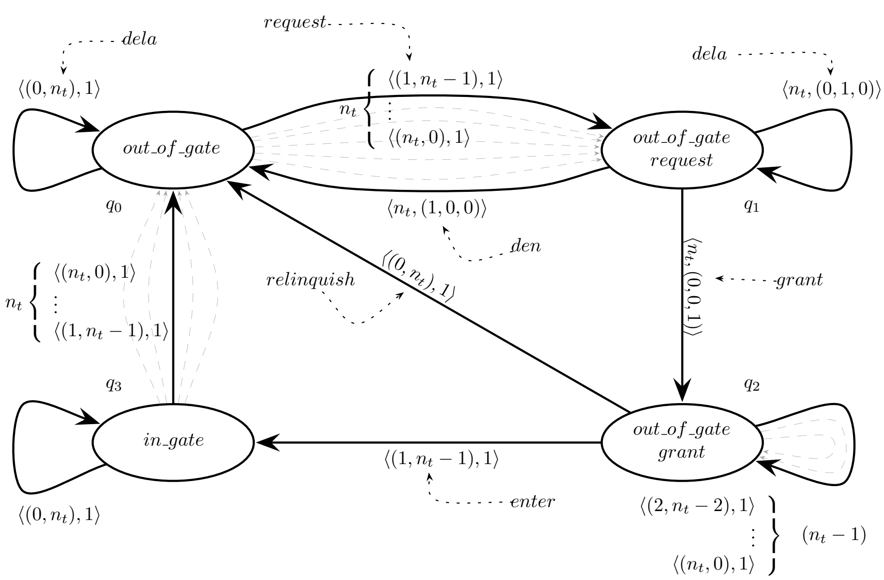


Fig. 1. Train controller model for n_t trains (similar to the one presented in [5]).

Notice that in the single-train case ($n_t = 1$), the train can not wait before entering the tunnel after being granted permission (*and* retain the permission). This could of course easily be avoided by adding another action. More importantly, in the case of several trains, the controller can not distinguish between the different trains, so permission must be granted to all or none. This is a consequence of the strict homogeneity in the model: not only are the agents homogeneous in terms of the actions available to them, we can not reasonably distinguish between them as long as they remain in the same role. Notice that this feature allow us to add any number of trains to the scenario without incurring more than a linear increase in the size of the model (total number of profiles). This would not be possible if we did not have roles. If the model above was to be rendered as a concurrent game structure, the number of possible ways in which trains could act would be exponential in all states where trains have to make a choice of what action to perform. This would be the case even if, as in the scenario above, almost all possible combinations of choices should be treated in the same way by the system.

Sometimes homogeneity is desirable. In our trains and controller example, for instance, homogeneity strongly *encourages* cooperation among trains; no one can enter the gate unless everyone agree, and everyone knows that whoever gets to enter *must* leave as soon as he is asked to. On the other hand, we notice that it is impossible for any train to enter the gate unless all trains cooperate. This might be overly restrictive. By adding more roles, however, we can amend this while still retaining many of the benefits of using roles.

Example 2.2. In the previous example all trains were equal before the controller; the controller could not distinguish between trains. We could grant the agents much more individual identity by simply adding one more role, and in this example we sketch the result of doing so. First we make n_t "copies" of the previous model sharing the state q_0. In Figure 2 we illustrate the resulting model for $n_t = 3$. In q_0 we let the trains vote for which train should be allowed to request permission to enter the tunnel. We assume majority voting, but we do not resolve ties. It means that if one train, x, gets more votes then all others we go to "his" state, q_1x. Otherwise we just loop on q_0. If we get to a q_1-state, the controller can grant or reject the request. Contrary to the previous example the controller now knows which train is being proposed. If the controller grants the request, the selected train is put in a privileged role and given the sole choice of what to do with the permission.

The model has grown, so the trains gain autonomy at a cost. Still, this cost is much less than the cost of modelling this scenario in a CGS. There, if each train is to have the option to "vote" for any train in q_0, each train must have n_t actions available. We would get $n_t^{n_t}$ edges leading out from q_0! In the RCGS model we get a substantially smaller degree, the following table summarizes the difference (formulas for counting the degree are explained and discussed further in section 4)

n_t:	3	4	5	6	...	n
CGS:	27	256	3125	46656	...	n^n
RCGS:	10	35	127	462	...	$\frac{(2n-1)!}{n!(n-1)!}$

Before we move on we introduce some more notation. Given a role $r \in R$, a state q and a coalition A, the set of A-votes for r at q is $V_r(q, A)$, defined as follows:

$$V_r(q, A) = \left\{ v \in [|A_{r,q}|]^{[\mathbb{A}(q,r)]} \;\middle|\; \sum_{a \in [\mathbb{A}(q,r)]} v(a) = |A_{r,q}| \right\}$$

The A-votes for r at q gives the possible ways agents in A that are in role r at q can vote. Given a state q and a coalition A, we define the set of A-profiles at q:

$$P(q, A) = \{\langle v_1, \ldots, v_{|R|}\rangle \mid 1 \le i \le |R| : v_i \in V_r(q, A)\}$$

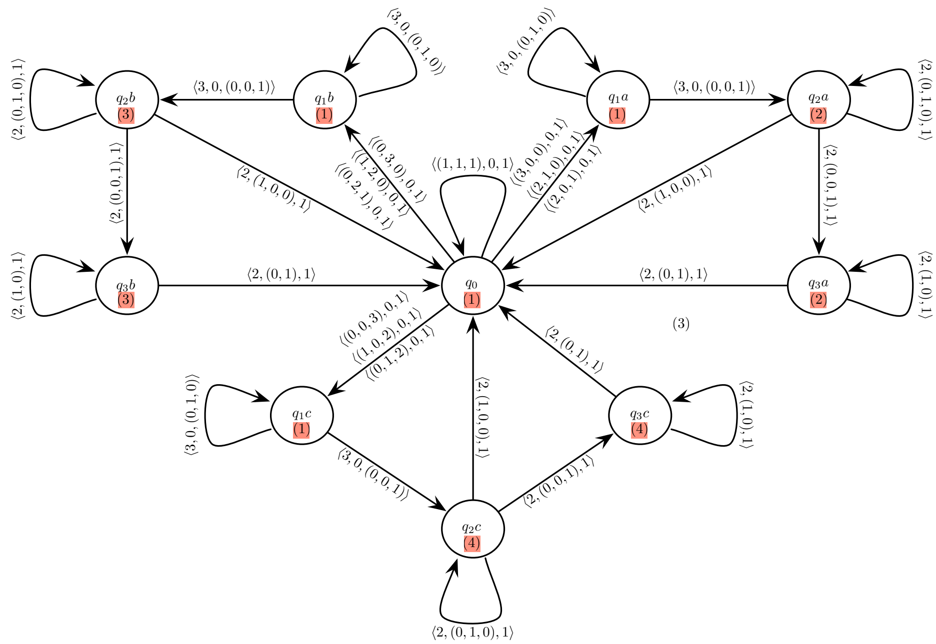


Fig. 2. The "autonomous trains" model for $n_t = 3$. Numbers in red indicate membership in roles: (1) $= \langle \mathcal{A}, \emptyset, \{ctr\}\rangle$, (2) $= \langle \mathcal{A}\setminus\{a\}, \{a\}, \{ctr\}\rangle$, (3) $= \langle \mathcal{A}\setminus\{b\}, \{b\}, \{ctr\}\rangle$, (4) $= \langle \mathcal{A} \setminus \{c\}, \{c\}, \{ctr\}\rangle$. Also, each q_1 state has a transition pointing at q_0 labeled $\langle 3, 0, (1, 0, 0)\rangle$ that was omitted from the picture for the sake of clarity.

When we say that a function $v : [\mathbb{A}(q, r)] \to [n]$ is a complete vote (for r at q), we mean that $v \in V_r(q, \mathcal{A})$. For any $v \in V_r(q, A)$ and $w \in V_r(q, B)$ we write $v \leq w$ iff for all $i \in [\mathbb{A}(q, r)]$ we have $v(i) \leq w(i)$. If $v \leq w$, we say that w *extends* v. If $F = \langle v_1, \ldots, v_R\rangle \in P(q, A)$ and $F' = \langle v'_1, \ldots, v'_R\rangle \in P(q, B)$ with $v_i \leq v'_i$ for every $1 \leq i \leq |R|$, we say that $F \leq F'$ and that F extends F'.

An A profile $F \in P(q, A)$ is a *complete* profile iff the sum of its components equal $|\mathcal{A}|$, i.e. $F \in P(q)$ iff $\left(\sum_{r\leq|R|} \sum_{a\in\mathbb{A}(q,r)} v(a)\right) = |\mathcal{A}|$ iff $A = \mathcal{A}$. Given a (partial) profile F' at a state q we write $ext(q, F)$ for the set of all complete profiles that extend F'.

Given two states $q, q' \in Q$, we say that q' is a *successor* of q if there is some $F \in P(q)$ such that $\delta(q, F) = q'$. A *computation* is an infinite sequence $\lambda = q_0q_1 \ldots$ of states such that for all positions $i \geq 0$, q_{i+1} is a successor of q_i. We follow the standard abbreviations, hence q-computation denotes a computation starting at q, and $\lambda[i]$, $\lambda[0, i]$ and $\lambda[i, \infty]$ denote the i-th state, the finite prefix $q_0q_1 \ldots q_i$ and the infinite suffix $q_iq_{i+1} \ldots$ of λ for any computation λ and its position $i \geq 0$. An A-*strategy* for $A \subseteq \mathcal{A}$ is a function $s_A : Q \to \bigcup_{q\in Q} P(q, A)$ such that $s_A(q) \in P(q, A)$ for all $q \in Q$. That is, s_A maps states to A-profiles at that state. The set of all A-strategies is denoted by $strat(A)$. If s is an $\mathcal{A}$-

strategy and we apply δ_q to $s(q)$, we obtain a unique new state $q' = \delta_q(s(q))$. Iterating, we get the *induced* computation $\lambda_{s,q} = q_0 q_1 \ldots$ such that $q = q_0$ and $\forall i \ \geq 0 : \delta_{q_i}(s(q_i)) = q_{i+1}$. Given two strategies s and s', we say that $s \leq s'$ iff $\forall q \in Q : s(q) \leq s'(q)$. Given an A-strategy s_A and a state q we get an associated *set* of computations $out(s_A, q)$. This is the set of all computations that can result when at any state, the players in A are voting in the way specified by s_A:
$out(s_A, q) = \{\lambda_{s,q} \mid s \text{ is an } \mathcal{A}\text{-strategy and } s \geq s_A\}$ It will also be useful to have access to the set of states that can result in the next step when $A \subseteq \mathcal{A}$ follows strategy s_A at state q, $succ(q, s_A) = \{q' \in Q \mid \exists F \in ext(q, s_A) : \delta(q, F) = q'\}$. Clearly, $q' \in succ(q, s_A)$ iff there is some $\lambda \in out(q, s_A)$ such that $q' = \lambda[0]$.

2.2 New Semantics for ATL

Definition 2.2. *Given a* RCGS *S and a state q in S, we define the satisfaction relation $\models$ inductively:*

$$
\begin{aligned}
&S, q \models p \text{ iff } p \in \pi(q) \\
&S, q \models \neg\phi \text{ iff not } S, q \models \phi \\
&S, q \models \phi \wedge \phi' \text{ iff } S, q \models \phi \text{ and } S, q \models \phi' \\
&S, q \models \langle\langle A \rangle\rangle \bigcirc \phi \text{ iff there is } s_A \in strat(A) \text{ such that} \\
&\qquad \text{for all } \lambda \in out(s_A, q), \text{ we have } S, \lambda[1] \models \phi \\
&S, q \models \langle\langle A \rangle\rangle \Box \phi \text{ iff there is } s_A \in strat(A) \text{ such that} \\
&\qquad \text{for all } \lambda \in out(s_A, q) \text{ we have } S, \lambda[i] \models \phi \text{ for all } i \geq 0 \\
&S, q \models \langle\langle A \rangle\rangle \phi \mathcal{U} \phi' \text{ iff there is } s_A \in strat(A) \text{ such that} \\
&\qquad \text{for all } \lambda \in out(s_A, q) \text{ we have } S, \lambda[i] \models \phi' \text{ and } S, \lambda[j] \models \phi \\
&\qquad \text{for some } i \geq 0 \text{ and for all } 0 \leq j < i
\end{aligned}
$$

3 Equivalence between RCGS and CGS

In this section we show that definition 2.2 provides an equivalent semantics for ATL. We do this by first giving a surjective function f that takes an RCGS and returns a CGS. Then we show that S and $f(S)$ satisfy the same ATL formulas.

Remember that a concurrent game structure is a tuple $\langle \mathcal{A}, Q, \Pi, \pi, d, \delta' \rangle$ where every element is defined as for an RCGS except $d : \mathcal{A} \times Q \to \mathbb{N}^+$ that maps agents and states to actions available at that state, and δ' that is a partial function from states and action tuples to states defined by $\delta'(q, t) = \delta'_q(t)$ where $\delta'_q : \prod_{a \in \mathcal{A}} [d_a(q)] \to Q$ is a transition function at q based on tuples of actions rather than profiles. The satisfaction relation for ATL based on CGSs can be defined exactly as in definition 2.2, the difference concerning only what counts as a strategy.

We refer to elements of $\prod_{a \in \mathcal{A}} [d_a(q)]$ as *complete* action tuples at q. A (memoryless) strategy for $a \in \mathcal{A}$ in a CGS M is a function $s_a : Q \to \mathbb{N}^+$ such that for all

$q \in Q$, $s_a(q) \in [d_a(q)]$ while a strategy for $A \subseteq \mathcal{A}$ is a list of strategies for all agents in A, $s_A = \langle s_{a_1}, s_{a_2}, \ldots, s_{a_{|A|}} \rangle$, for $A = \{a_1, a_2, \ldots, a_{|A|}\}$. We denote the set of strategies for $A \subseteq \mathcal{A}$ by $strat(A)$. When needed to distinguish between different structures we write $strat(S, A)$ to indicate that we are talking about the set of strategies for A in S.

We say that a complete action tuple at q, $t = \langle i_{a_1}, \ldots, i_{a_n} \rangle$ extends a strategy $s_A \in strat(A)$ if for all $a_j \in A$ we have $i_{a_j} = s_{a_j}(q)$. We denote the set of all complete action tuples at q extending s_A by $ext(q, s_A)$. For any state $q \in Q$ we have the set of all computations that comply with s_A:

$$\begin{aligned} out(q, s_A) = &\{\lambda = q_0 q_1 q_2 \ldots \\ &\qquad \mid q = q_0 \text{ and for all } i \in \mathbb{N} : \exists t \in ext(q_i, s_A),\ \delta(q, t) = q_{i+1}\} \end{aligned}$$

We define the set of s_A-successors at $q \in Q$:

$$succ(q, s_A) = \{q' \in Q \mid \exists t \in ext(q, s_A),\ \delta(q, t) = q'\}$$

When we need to make clear which structure we are talking about, we write $succ(S, q, s_A)$. Observe that $q' \in succ(q, s_A)$ iff $q' = \lambda[1]$ for some $\lambda \in out(q, s_A)$.

The translation function f from RCGS to CGS is defined as follows:

$$f\langle \mathcal{A}, R, \mathcal{R}, Q, \Pi, \pi, \mathbb{A}, \delta \rangle = \langle \mathcal{A}, Q, \Pi, \pi, d, \delta' \rangle$$

where:

$$\begin{aligned} d_a(q) &= \mathbb{A}(q, r) && \text{where } a \in \mathcal{R}(q, r) \\ \delta'(q, \alpha_1, \ldots, \alpha_n) &= \delta(q, v_1, \ldots, v_{|R|}) && \text{where for each role } r \\ v_r &= \langle |\{i \in \mathcal{R}(q, r) \mid \alpha_i = 1\}|, \ldots, |\{i \in \mathcal{R}(q, r) \mid \alpha_i = \mathbb{A}(q, r)\}| \rangle \end{aligned}$$

We can see straight away that f is surjective because for any CGS S' with n agents we could define a RCGS S with that many roles where each role contains exactly one agent. A vote for a role r, v_r, at q would then simply be a $d_a(q)$-tuple consisting of a single 1 (representing the agents chosen action) and otherwise zeros. It is easy to verify that $f(S) = S'$.

Given either a CGS or an RCGS S, we define the set of sets of states that a coalition A can *enforce* in the next state of the game:

$$force(S, q, A) = \{succ(q, s_A) \mid s_A \text{ is a strategy for } A \text{ in } S\}.$$

The first thing we do towards showing equivalence is to describe a surjective function $m : strat(f(S)) \to strat(S)$ mapping action tuples and strategies of $f(S)$ to profiles and strategies of S respectively. For all $A \subseteq \mathcal{A}$ and any action tuple for A at q, $t_q = \langle \alpha_{a_1}, \alpha_{a_2}, ..., \alpha_{a_{|A|}} \rangle$ with $1 \leq \alpha_{a_i} \leq d_{a_i}(q)$ for all $1 \leq i \leq |A|$, the A-profile $m(t_q)$ is defined in the following way:

$$m(t_q) = \langle v(t_q, 1), \ldots, v(t_q, |R|)\rangle \text{ where for all } 1 \leq r \leq |R| \text{ we have}$$
$$v(t_q, r) = \langle |\{a \in A_{r,q} \mid \alpha_a = 1\}|, \ldots, |\{a \in A_{r,q} \mid \alpha_a = \mathbb{A}(q, r)\}|\rangle$$

Thus the i-th component of $v(t_q, r)$ will be the number of agents from A in role r at q that perform action i.

Given a strategy s_A in $f(S)$ we define the strategy $m(s_A)$ for S by taking $m(s_A)(q) = m(s_A(q))$ for all $q \in Q$.

Surjectivity of m is helpful since it means that for every possible strategy that exists in the RCGS S, there is a corresponding one in $f(S)$. This in turn means that when we quantify over strategies in one of S and $f(S)$ we are implicitly also quantifying over strategies in the other. Showing equivalence, then, can be done by showing that these corresponding strategies have the same strength. Before we proceed, we give a proof of surjectivity of m.

Lemma 3.1. *For any* RCGS S *and any* $A \subseteq \mathcal{A}$*, the function* $m : strat(f(S), A) \to strat(S, A)$ *is surjective*

Proof. Let p_A be some strategy for A in S. We must show there is a strategy s_A in $f(S)$ such that $m(s_A) = p_A$. For all $q \in Q$, we must define $s_A(q)$ appropriately. Consider the profile $p_A(q) = \langle v_1, \ldots, v_{|R|}\rangle$ and note that by definition of a profile, all v_r for $1 \leq r \leq |R|$ are A-votes for r and that by definition of an A-vote, we have $\sum_{1 \leq i \leq \mathbb{A}(q,r)} v_r(i) = |A_{r,q}|$. Also, for all agents $a, a' \in A_{r,q}$ we know, by definition of f, that $d_a(q) = d_{a'}(q) = \mathbb{A}(q, r)$.

From this it follows that there are functions $\alpha : A \to \mathbb{N}^+$ such that for all $a \in A$, $\alpha(a) \in [d_a(q)]$ and $|\{a \in A_{r,q} \mid \alpha(a) = i\}| = v_r(i)$ for all $1 \leq i \leq \mathbb{A}(q, r)$, i.e.
$$v_r = \langle |\{a \in A_{r,q} | \alpha(a) = 1\}|, \ldots, |\{a \in A_{r,q} | \alpha(a) = \mathbb{A}(q, r)\}|\rangle$$
We choose some such α and $s_A = \langle \alpha(a_1), \ldots, \alpha(a_{|A|})\rangle$. Having defined s_A in this way, it is clear that $m(s_A) = p_A$.

Using the surjective function m we can prove the following lemma, showing that the "next time" strength of any coalition A is the same in S as it is in $f(S)$.

Lemma 3.2. *For any* RCGS S*, any state* $q \in Q$ *and any coalition* $A \subseteq \mathcal{A}$*, we have* $force(S, A, q) = force(f(S), A, q)$

Proof. By definition of $force$ and lemma 3.1 it is sufficient to show that for all $s_A \in strat(f(S), A)$, we have $succ(S, m(s_A), q) = succ(f(S), s_A, q)$. We show $\subseteq$ as follows: Assume that $q' \in force(S, m(s_A), q)$. Then there is some complete profile $P = \langle v_1, \ldots, v_{|R|}\rangle$, extending $m(s_A)(q)$, such that $\delta(q, P) = q'$. Let $m(s_A)(q) = \langle w_1, \ldots, w_{|R|}\rangle$ and form $P' = \langle v'_1, \ldots, v'_{|R|}\rangle$ defined by $v'_i = v_i - w_i$ for all $1 \leq i \leq |R|$. Then each v'_i is an $(\mathcal{A} \setminus A)$-vote for role i, meaning that the sum of entries in the tuple v'_i is $|(\mathcal{A} \setminus A)_{r,q}|$. This means that we can define a function $\alpha : \mathcal{A} \to \mathbb{N}^+$ such that for all $a \in \mathcal{A}$, $\alpha(a) \in [d_a(q)]$ and for all $a \in A$, $\alpha(a) = s_a(q)$ and for every $r \in R$ and every $a \in (\mathcal{A} \setminus A)$, and every $1 \leq j \leq \mathbb{A}(q,r)$, $|\{a \in (\mathcal{A} \setminus A)_{r,q} \mid \alpha(a) = j\}| = v'_r(j)$. Having defined α like this it follows by definition of m that for all $1 \leq j \leq \mathbb{A}(q,r)$, $|\{a \in A_{r,q} \mid \alpha(a) = j\}| = w_r(j)$. Then for all $r \in R$ and all $1 \leq j \leq \mathbb{A}(q,r)$ we have $|\{a \in \mathcal{R}(q,r) \mid \alpha(a) = j\}| = v_r(j)$. By definition of $f(S)$ it follows that $q' = \delta(q,P) = \delta'(q,\alpha)$ so that $q' \in force(f(S), s_A, q)$. We conclude that $force(S, f(s_A), q) \subseteq force(f(S), s_A, q)$. The direction $\supseteq$ follows easily from the definitions of m and f.

Given a structure S (with or without roles), and a formula ϕ, we define $true(S,\phi) = \{q \in Q \mid S, q \models \phi\}$. Equivalence of models S and $f(S)$ is now demonstrated by showing that the equivalence in next time strength established in lemma 3.2 suffices to conclude that $true(S,\phi) = true(f(S),\phi)$ for all ϕ.

Theorem 3.1. *For any* RCGS *S, any ϕ and any $q \in Q$, we have $S, q \models \phi$ iff $f(S), q \models_{CGS} \phi$*

Proof. We prove the theorem by showing that for all ϕ, we have $true(S,\phi) = true(f(S),\phi)$. We use induction on complexity of ϕ. The base case for atomic formulas and the inductive steps for Boolean connectives are trivial, while the case of $\langle\langle A\rangle\rangle \bigcirc \phi$ is a straightforward application of lemma 3.2. For the cases of $\langle\langle A\rangle\rangle\Box\phi$ and $\langle\langle A\rangle\rangle\phi\mathcal{U}\psi$ we rely on the following fixed point characterizations, which are well-known to hold for ATL, see for instance [6], and are also easily verified against definition 2.2:

$$\begin{aligned} \langle\langle A\rangle\rangle\Box\phi &\leftrightarrow \phi \wedge \langle\langle A\rangle\rangle \bigcirc \langle\langle A\rangle\rangle\Box\phi \\ \langle\langle A\rangle\rangle\phi_1\mathcal{U}\phi_2 &\leftrightarrow \phi_2 \vee (\phi_1 \wedge \langle\langle A\rangle\rangle \bigcirc \langle\langle A\rangle\rangle\phi_1\mathcal{U}\phi_2 \end{aligned} \tag{1}$$

We show the induction step for $\langle\langle A\rangle\rangle\Box\phi$, taking as induction hypothesis $true(S,\phi) = true(f(S),\phi)$. The first equivalence above identifies $Q' = true(S, \langle\langle A\rangle\rangle\Box\phi)$ as the maximal subset of Q such that ϕ is true at every state in Q' and such that A can enforce a state in Q' from every state in Q', i.e. such that $\forall q \in Q' : \exists Q'' \in force(q, A) : Q'' \subseteq Q'$. Notice that a unique such set always exists. This is clear since the union of two sets satisfying the two requirements will itself satisfy them (possibly the empty set). The first requirement, namely that ϕ is true at all states in Q', holds for S iff if holds for $f(S)$ by induction hypothesis. Lemma 3.2 states $force(S, q, A) = force(f(S), q, A)$, and this implies that also the second requirement holds in S iff it holds in $f(S)$. From this we conclude $true(S, \langle\langle A\rangle\rangle\Box\phi) = true(f(S), \langle\langle A\rangle\rangle\Box\phi)$ as desired. The case for $\langle\langle A\rangle\rangle\phi\mathcal{U}\psi$ is similar, using the second equivalence. □

4 Model checking and the size of models

We have already seen that using roles can lead to a dramatic decrease in the size of ATL-models. In this section we give a more formal account, first by investigating the size of models in terms of the number of roles, players and actions, then by an analysis of model checking ATL over concurrent game structures with roles.

Given a set of numbers $[a]$ and a number n, it is a well-known combinatorial fact that the number of ways in which to choose n elements from $[a]$, allowing repetitions, is $\frac{(n+(a-1))!}{n!(a-1)!}$. Furthermore, this number satisfies the following two inequalities:[4]

$$\tfrac{(n+(a-1))!}{n!(a-1)!} \leq a^n \ , \ \tfrac{(n+(a-1))!}{n!(a-1)!} \leq n^a \tag{2}$$

These two inequalities provide us with an upper bound on the *size* of RCGS models that makes it easy to compare their sizes to that of CGS models. Typically, the size of concurrent game structures is dominated by the size of the domain of the transition function. For an RCGS and a given state $q \in Q$ this is the number of complete profiles at q. To measure it, remember that every complete profile is an $|R|$-tuple of votes v_r, one for each role $r \in R$. It follows that $|P(q)|$ is the set of all possible combinations of votes for each role. Also remember that a vote v_r for $r \in R$ is an $\mathbb{A}(q,r)$-tuple such that the sum of entries is $|\mathcal{R}(q,r)|$. Equivalently, the vote v_r can be seen as the number of ways in which we can make $|\mathcal{R}(q,r)|$ choices, allowing repetitions, from a set of $\mathbb{A}(q,r)$ alternatives. Looking at it this way, we obtain:

$$|P(q)| = \prod_{r \in R} \frac{(|\mathcal{R}(q,r)| + (\mathbb{A}(q,r)-1))!}{|\mathcal{R}(q,r)|!(\mathbb{A}(q,r)-1))!}$$

We sum over all $q \in Q$ to obtain what we consider to be the size of an RCGS S. In light of equation 2, it follows that the size of S is upper bounded by both of the following expressions.

$$\mathcal{O}(\textstyle\sum_{q \in Q} \prod_{r \in R} |\mathcal{R}(q,r)|^{\mathbb{A}(q,r)}) \ , \ \mathcal{O}(\textstyle\sum_{q \in Q} \prod_{r \in R} \mathbb{A}(q,r)^{|\mathcal{R}(q,r)|}) \tag{3}$$

We observe that growth in the size of models is polynomial in $a = max_{q \in Q, r \in R}\mathbb{A}(r,q)$ if $n = \mathcal{A}$ and $|R|$ is fixed, and polynomial in $p = max_{q \in Q, r \in R}|\mathcal{R}(q,r)|$ if a and $|R|$ are fixed. This identifies a significant potential advantage arising from introducing roles to the semantics of ATL. The size of a CGS M, when measured in the same way, replacing complete profiles at q by complete action tuples at q, grows exponentially in the players whenever $d_a(q) > 1$ for each player a. We stress that we are *not* just counting the number of transitions in our models differently. We do have an additional parameter, the roles, but this is a genuinely new semantic construct that gives rise to genuinely different semantic structures. We show that it is possible to use them to give the semantics of ATL, but this does not mean that there is not more to be said about them. Particularly crucial is the question of model checking over RCGS models.

[4] If this is not clear, remember that n^a and a^n are the number of functions $[n]^{[a]}$ and $[a]^{[n]}$ respectively. It should not be hard to see that all ways in which to choose n elements from a induce non-intersecting sets of functions of both types

4.1 Model checking using roles

For strategic logics, checking satisfiability is usually non-tractable, and the question of model checking is often crucial in assessing the usefulness of different logics. For ATL there is a well known "standard" algorithm, see e.g. [5]. It does model checking in time linear in the length of the formula and the size of the model. The algorithm is based on the fixed point equation 1 from the proof of Theorem 3.1, so it will work also when model checking RCGS models. It is not clear, however, how the high level description should be implemented and, crucially, what the complexity will be in terms of the new parameters that arise.

Given a structure with roles, S, and a formula ϕ, the standard model checking algorithm returns the set $true(S, \phi)$, proceeding as detailed in algorithms 1 and 2.

Algorithm 1 $mcheck(S, \phi)$

---

if $\phi = p \in \Pi$ **then**
  return $\pi(p)$
if $\phi = \neg\psi$ **then**
  return $Q \setminus mcheck(S, \psi)$
if $\phi = \psi \wedge \psi'$ **then**
  return $mcheck(S, \psi) \cap mcheck(S, \psi')$
if $\phi = \langle\!\langle A \rangle\!\rangle \bigcirc \psi$ **then**
  return $\{q \mid enforce(S, q, A, mcheck(S, \psi))\}$
if $\phi = \langle\!\langle A \rangle\!\rangle \Box \psi$ **then**
  $Q_1 := Q$, $Q_2 := mcheck(S, \psi)$
  while $Q_1 \not\subseteq Q_2$ **do**
    $Q_1 := Q_2$, $Q_2 := \{q \in Q \mid enforce(S, A, q, Q_2)\} \cap Q_2$
    return Q_1
if $\phi = \langle\!\langle A \rangle\!\rangle \psi \mathcal{U} \psi'$ **then**
  $Q_1 := \emptyset$, $Q_2 = mcheck(S, \psi)$, $Q_3 = mcheck(S, \psi')$
  while $Q_3 \not\subseteq Q_1$ **do**
    $Q_1 := Q_1 \cup Q_3$, $Q_3 := \{q \in Q \mid enforce(S, A, q, Q_1)\} \cap Q_2$
    return Q_3

Given a structure S, a coalition A, a state $q \in Q$ and a set of states Q', the method $enforce$ answers true or false depending on whether or not A can enforce Q' from q. That is, it tells us if at q there is $Q'' \in force(q, A)$ such that $Q'' \subseteq Q'$. Given a fixed length formula and a fixed number of states, this step dominates the running time of $mcheck$ (algorithm 1). It is also the only part of the standard algorithm that behaves in a different way after addition of roles to the structures. It involves the following steps:[5]

[5] In implementations one would seek to take advantage of information collected by repeating calls to $enforce$ and not just do a Boolean check for every new instance in the way we do it here. This aspect is not crucial for our analysis, so we do not address it further

Algorithm 2 $enforce(S, A, q, Q')$

```
for F ∈ P(q, A) do
   p = true
   for F' ∈ ext(q, F) do
      if δ(q, F') ∉ Q' then
         p = false
   if p = true then
      return true
return false
```

For all profiles $F \in P(q, A)$ the algorithm runs through all complete profiles $F' \in P(q)$ that extend F. Over CGSs, given a coalition A and two action tuples $t = \langle \alpha_{a_1}, \alpha_{a_2}, \ldots, \alpha_{a_{|A|}} \rangle, t' = \langle \alpha'_{a_1}, \alpha'_{a_2}, \ldots, \alpha'_{a_{|A|}} \rangle$ for A at q, the sets of complete action tuples that extend t and t' respectively do not intersect. It follows that running through all such extensions for all possible action tuples for A at q is at most linear in the total number of complete action tuples at q. This is no longer the case for RCGS models. Given two profiles P, P' for A at q, there can be many shared extensions. In fact, P and P' can share exponentially many in terms of the number of players and actions available.[6] So, in general, running $enforce$ requires us to make several passes through the set of all complete profiles, and the complexity is no longer linear. Still, it is polynomial in the number of complete profiles, since for any coalition A and state q we have $|P(q, A)| \leq |P(q)|$, meaning that the complexity of $enforce$ is upper bounded by $|P(q)|^2$. It follows that model checking of ATL over concurrent game structures with roles is polynomial in the size of the model. We summarize this result.

Proposition 4.1. *Given a CGS S and a formula ϕ, $mcheck(S, \phi)$ takes time $\mathcal{O}(le^2)$ where l is the length of ϕ and $e = \sum_{q \in Q} P(q)$ is the total number of transitions in S*

Since model checking ATL over CGSs takes only linear time, $\mathcal{O}(le)$, adding roles apparently makes model checking harder. On the other hand, the *size* of CGS models can be bigger by an exponential factor, making model checking much easier after adding roles. In light of the bounds we have on the size of models, c.f. equation 3, we find that as long as the roles and the actions remain fixed, complexity of model checking is only polynomial in the number of agents. This is a potentially significant argument in favor of roles.

In practice, however, finding an optimal RCGS for a given CGS model M might be at least as difficult as model checking on M directly. It involves identifying the

[6] To see this, consider $P = \langle v_1, v_2 \ldots, v_{|R|} \rangle$ and $P' = \langle v'_1, v'_2, \ldots, v'_{|R|} \rangle$. Each $v_r, v'_r \in V_A(q, r)$ sums to $\Sigma_{1 \leq j \leq \mathbb{A}(q,r)} v_i(j) = |A_{q,r}|$. Then form a complete profile $P'' = \langle v''_1, v''_2, \ldots, v''_{|R|} \rangle$ at q such that for all $1 \leq r \leq |R|$ and all $1 \leq j \leq \mathbb{A}(q, r)$ we have $v''_r(j) = max(v_r(j), v'_r(j))$. Then, if it exists, choose a coalition A' such that $|A'_{r,q}| = \Sigma_{1 \leq j \leq \mathbb{A}(q,r)} v''_r(j)$. It is clear that the number of complete profiles that extends both v and v' is equal to the number of *all* $\mathcal{A} \setminus A'$-profiles at q.

structure from $f^{-}(M)$ that has the minimum number of roles. In general, one cannot expect this task to have sub-linear complexity in the size of M.[7] Roles should be used at the modelling stage, as they give the modeller an opportunity for exploiting homogeneity in the system under consideration. We think that it is reasonable to hypothesize that in practice, most large scale systems that lends themselves well to modelling by ATL do so precisely because they exhibit significant homogeneity. If not, identifying an accurate ATL model of the system, and model checking it, seems unlikely to be tractable at all.

The question arises as to whether or not using an RCGS is *always* the best choice, or if there are situations when the losses incurred in the complexity of model checking outweigh the gains we make in terms of the size of models. A general investigation of this in terms of how fixing or bounding the number of roles affect membership in complexity classes is left for future work. Here, we conclude with the following proposition which states that as long we use the standard algorithm, model checking any CGS M can be done at least as quickly by model checking an *arbitrary* $S \in f^{-}(M)$.

Proposition 4.2. *Given any* CGS*-model M and any formula ϕ, let $c(mcheck(M,\phi))$ denote the complexity of running $mcheck(M,\phi)$. We have, for all $S \in f^{-}(M)$, that complexity of running $mcheck(S,\phi)$ is $\mathcal{O}(c(mcheck(M,\phi))$*

Proof. It is clear that for any $S \in f^{-}(M)$, running $mcheck(S,\phi)$ and $mcheck(M,\phi)$, a difference in overall complexity can arise only from a difference in the complexity of $enforce$. So we compare the complexity of $enforce(S,A,q,Q'')$ and $enforce(M,A,q,Q'')$ for some arbitrary $q \in Q$, $Q'' \subseteq Q$. The complexity in both cases involves passing through all complete extensions of all strategies for A at q. The sizes of these sets are can be compared as follows, the first inequality is an instance of equation 2 and the equalities follow from definition of f and the fact that $M = f(S)$.

$$\prod_{r \in R}\left(\frac{(|A_{r,q}| + (\mathbb{A}(r,q)-1))!}{|A_{r,q}|!(\mathbb{A}(r,q)-1)!}\right) \times \prod_{r\in R}\left(\frac{((|\mathcal{R}(r,q)| - |A_{r,q}|) + (\mathbb{A}(r,q)-1))!}{(|\mathcal{R}(r,q)| - |A_{r,q}|)!(\mathbb{A}(r,q)-1)!}\right)$$
$$\leq \left(\prod_{r\in R}\mathbb{A}(r,q)^{|A_{r,q}|} \times \prod_{r\in R}\mathbb{A}(r,q)^{|\mathcal{R}(r,q)|-|A_{r,q}|}\right)$$
$$= \prod_{r\in R}\left(\prod_{a\in A_{r,q}}\mathbb{A}(r,q)\right) \times \prod_{r\in R}\left(\prod_{a\in\mathcal{R}(a,r)\setminus A_{r,q}}\mathbb{A}(r,q)\right)$$
$$= \left(\prod_{a\in A} d_a(q) \times \prod_{a\in\mathcal{A}\setminus A} d_a(q)\right) = \prod_{a\in\mathcal{A}} d_a(q)$$

[7] Although in many practical cases, when models are given in some compressed form, the situation might be such that it is possible. The question of how to efficiently find small RCGS-models will be investigated in future work.

We started with the number of profiles (transitions) we need to inspect when running $enforce$ on S at q, and ended with the number of action tuples (transitions) we need to inspect when running $enforce$ on $M = f(S)$. Since we showed the first to be smaller or equal to the latter and the execution of all other elements of *mcheck* are identical between S and M, the claim follows.

5 Conclusions, related and future work

In this paper we have described a new type of semantics for the strategic logic ATL. We have provided motivational examples and argued that although in principle model checking ATL interpreted over concurrent game structures with roles is harder than the standard approach, it is still polynomial and generates exponentially smaller models. We believe this provides conclusive evidence that concurrent game structures with roles are an interesting semantics for ATL, and should be investigated further.

Relating our work to ideas already present in the literature we find it somewhat similar to the concept of exploiting symmetry in model checking, as investigated by Sistla and Godefroid [12]. Our approach is however different, since we we only look at agent symmetries in ATL. When it comes to work related directly to strategic logics, we find no similar ideas present, hence concluding that our approach is indeed novel.

For future work we plan on investigating the homogeneous aspect of our 'roles' in more depth. We are currently working on a derivative of ATL with a different language that will fully exploit the role based semantics.

Acknowledgements We thank anonymous reviewers of LAMAS 2012 and Pål Grønås Drange for helpful comments.

References

1. Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. Quantified coalition logic. *Synthese (Special Section on Knowledge, Rationality and Action)*, 165(2):269–294, 2008.
2. Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. Reasoning about coalitional games. *Artificial Intelligence*, 173(1):45–79, 2009.
3. Natasha Alechina, Brian Logan, Hoang Nga Nguyen, and Abdur Rakib. Logic for coalitions with bounded resources. *Journal of Logic and Computation*, 21(6):907–937, 2011.
4. Rajeev Alur and Thomas A. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(11):7–48, July 1999.
5. Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM (JACM)*, 49(5):672–713, 2002.
6. Wojciech Jamroga. Easy yet hard: Model checking strategies of agents. In Michael Fisher, Fariba Sadri, and Michael Thielscher, editors, *Computational Logic in Multi-Agent Systems*, pages 1–12. Springer-Verlag, Berlin, Heidelberg, 2009.

7. Wojciech Jamroga and Jürgen Dix. Do agents make model checking explode (computationally)? In M. Pechoucek, P. Petta, and L. Z. Varga, editors, *Multi-Agent Systems and Applications IV (LNAI Volume 3690)*, 2005.
8. Wojciech Jamroga and Wiebe van der Hoek. Agents that Know How to Play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.
9. Magdalena Kacprzak and Wojciech Penczek. A sat-based approach to unbounded model checking for alternating-time temporal epistemic logic. *Synthese*, 142(2):203–227, November 2004.
10. Marc Pauly. *Logic for Social Software*. ILLC dissertation series 2001-10, University of Amsterdam, 2001.
11. Marc Pauly. A Modal Logic for Coalitional Power in Games. *Journal of Logic and Computation*, 12(1):149–166, February 2002.
12. A. Prasad Sistla and Patrice Godefroid. Symmetry and reduced symmetry in model checking. *ACM Trans. Program. Lang. Syst.*, 26(4):702–734, July 2004.
13. Wiebe van der Hoek, Wojciech Jamroga, and Michael Wooldridge. A logic for strategic reasoning. In *Proceedings of AAMAS'05*, pages 157–164, 2005.
14. Wiebe van der Hoek, Alessio Lomuscio, and Michael Wooldridge. On the complexity of practical atl model checking. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 201–208. ACM, 2006.
15. Wiebe van der Hoek and Michael Wooldridge. Time, knowledge, and cooperation: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.
16. Wiebe van der Hoek and Michael Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 164(1-2):81–119, May 2005.

Decision support for extensive form negotiation games

Sujata Ghosh[1] *, Thiri Haymar Kyaw[2] and Rineke Verbrugge[2]

[1] Indian Statistical Institute, Chennai Centre
SETS Campus, MGR Film City Road
Chennai 600113, India.
sujata@isichennai.res.in
[2] Department of Artificial Intelligence
University of Groningen
The Netherlands
thirihaymarkyaw@gmail.com, rineke@ai.rug.nl

Abstract. This paper presents an implementation tool NEGEXT for finding individual and group strategies to achieve certain goals while playing extensive form negotiation games. We consider sequential and parallel combinations of such games also. NEGEXT is used as a model-checking tool which investigates the existence of strategies in negotiation situations. Thus it may aid students of negotiation in their understanding of extensive game-form negotiation trees and their combinations, as well as in their learning to construct individual and group strategies.

1 Introduction

Negotiation may be found everywhere: From mundane conversations between partners about who will fetch the children from school and who will cook dinner, to the sale of an apartment whilst the seller is trying to hide from the buyer that she has bought a new house already, and to fully-fledged international multi-party multi-issue negotiations about climate control, and so forth. Negotiation is a complex skill, and one that is not learnt easily. Thus, many negotiations are broken off, even when they have potential for a win-win solution. Moreover, in many negotiations that do result in an agreement, one or more participants "leave money on the table": they could have done better for themselves [1]. Thus, it is no wonder that several scientific fields have made contributions to analyzing, formalizing, and supporting negotiation.

Kuhn [2] highlighted the importance of using extensive form games in modeling negotiation situations in an objective way, by focusing on the temporal and dynamic nature of the negotiations. Bargaining, one of the main forms of negotiation, has been modeled in various situations and in various forms, for example, the alternating-offers model [3], private information models [4], and legislatures [5]. An overview of different models of conflict resolution can be found

* This research was supported by the Netherlands Organisation of Scientific Research grant 600.065.120.08N201

in [6]. For a general overview on negotiation games, see [7]. Game-theoretic models of important real-life negotiations can be found in [8,9] and in the numerous case-studies published by the Harvard Project on Negotiation.

Researchers in multi-agent systems investigate many aspects of negotiations: some design negotiation mechanisms [10], some analyze negotiation as a form of dialogue [11], whilst others build software to simulate and support negotiation [12,13]. In recent years, the logical perspective has started to shed light on negotiation [14,15]. In summary, the fast-growing body of research on negotiation provides varied sophisticated models for negotiations.

This paper reports the development of a simple tool, NEGEXT (`http://www.ai.rug.nl/~sujata/negext.html`), written in the platform-independent Java language. NEGEXT has been constructed to aid students of negotiation. This toolkit will aid in understanding how to combine negotiations, and in planning one's strategic moves in interaction situations when the opponents' possible moves can be approximated. Even though some visualization tools for extensive form game trees already exist,[3] as well as software for negotiation support,[4] we believe we are the first to make a tree-based negotiation toolkit that incorporates the possibility of representing learning from game to game, by sequential and parallel composition (cf. [16]). Moreover, the toolkit has a model-checking component which computes whether and how an individual or a specific coalition can achieve a given objective.

The rest of the paper is structured as follows. Two negotiation situations are described in Section 2, where they are modeled as perfect information extensive form games and their combinations. Section 3 presents the toolkit NEGEXT. Section 4 describes runs of the toolkit on examples described in Section 2. Finally, pointers for further development of the toolkit are provided in Section 5.

2 Analyzing negotiation situations using game trees

A finite extensive form game can be represented by a finite tree where the nodes correspond to players' positions and edges correspond to moves of the players. The terminal nodes of the tree are the end-points of the game, which are generally termed as leaves of the tree. A strategy for a player i is a subtree of the finite game tree, which consists of single edges from player i nodes, and all possible edges from the other players' nodes. A strategy for a group of players K is a subtree consisting of single edges from player k's nodes, where $k \in K$, and all possible edges from player k''s nodes, where $k' \notin K$. Let us briefly describe the notion of sequential and parallel combinations of extensive form games and players' strategies in such games, providing an application of each in negotiations.

2.1 Sequential composition

Two extensive form games are *sequentially composed* by plugging in the second game at each leaf node of the first game, such that each leaf node of the first

[3] See for example `http://www.gametheory.net/Mike/applets/ExtensiveForm/ExtensiveForm.html` and `http://www.gambit-project.org`

[4] See for example `http://www.negotiationtool.com/`

game becomes the root node of the second game [16]. One can extend this idea to define sequential composition of a game with a set of games. In this case, at each leaf node of the first game, some game from the given set of games is plugged in.

Story-I: Sequential cooperation between two companies Suppose that two Research & Development companies on biotechnology, Biocon and Wockhard, have just entered into a joint project concerning the discovery of biomarkers. They need to hire two types of specialists, a genomic expert and a proteomic expert, one specialist per company. Unfortunately the two companies did not discuss beforehand which of them should hire which type of expert. The hiring process can be represented by the simple game trees in Figure 1, where action "hire g" stands for "hire a genomic expert" and action "hire p" stands for "hire a proteomic expert". The propositional atoms have the following meanings:

g_B: genomics is covered by Biocon	g_W: genomics is covered by Wockhard
p_B: proteomics is covered by Biocon	p_W: proteomics is covered by Wockhard

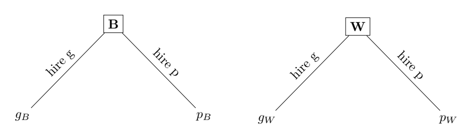


Fig. 1. Hiring game trees

Now Wockhard notices that it is a good idea to await Biocon's hiring decision. In order to analyze the different possibilities, they combine the two games sequentially (Figure 2), plugging in the right game of Figure 1 at each leaf node of the left game of Figure 1. Figure 2 clearly shows that hiring one expert after another, in a perfect information game, is much better than the imperfect information game represented by the two game trees in Figure 1. The second company *learns* who was hired by the first company, and hires someone with complementary expertise. The two 'middle branches' in Figure 2 provide win-win solutions.

2.2 Parallel composition

An *interleaving parallel combination* of two extensive form games, as defined in [16], gives rise to a set of games. The main idea of the interleaving parallel operator is as follows: A play of such a game basically moves from one game to the other. One player can move in one of the games, and in the next instant some other player or even the same player may move in the other game. A parallel game takes care of such interleaving. The different orders in the way moves of the players can be interleaved give rise to different games in a parallel

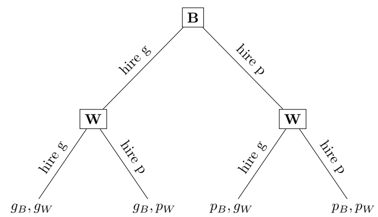


Fig. 2. Sequentially hiring game tree

combination of two games. Interleaving parallel games allows for copy-cat moves, and in general enables the transfer of strategies in between games. For the formal details, see [16].

Story-II: Analysis of job application in lockstep synchrony This example has been inspired by the trick regarding *how not to lose while playing chess with a grandmaster* [16], but with additional aspects of translation. Suppose that Prof. Flitwick (FL) applied for a position in the research group of Prof. Sprout (SP), and he turns out to be the favored candidate. Job negotiations N_1 between Flitwick and Sprout are on the verge of starting. In the same period, Prof. Sprout herself applied for a professorship in the department of Prof. Quirrell (QR), and has just been selected by Prof. Quirrell. The time arises to discuss the particulars of this position as well, in negotiation N_2.

Prof. Sprout is not a very savvy negotiator herself, but she knows that both Prof. Flitwick and Prof. Quirrell are. Moreover, Prof. Sprout notices that the issues of negotiation are quite similar in both cases: The prospective employer can offer either a much higher salary than the standard one, or a standard salary. On the employee's turn, he or she can choose to offer teaching subjects of their own choice, or whatever the department demands (Figure 3).

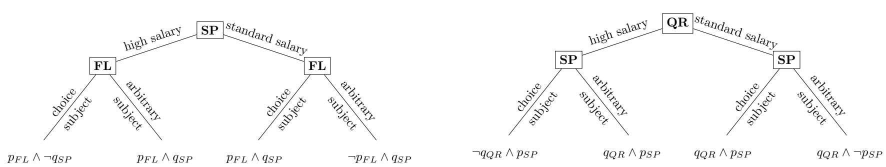


Fig. 3. Appointment game trees for negotiation N_1 (left) and negotiation N_2 (right)

The main pay-offs of the negotiators can be given in terms of propositional letters as follows, where:

- p_i: the new employee i gets more satisfaction in the job (for i is FL or SP);
- q_j: the new employer j is happy with the terms (where j is SP or QR).

In both negotiations, the goal of the new employee can be formulated conditionally: either she procures a higher salary, and then he or she is ready to do whatever he or she is asked; or, if she gets a standard salary, then she would like to teach her favorite subjects only. The goal of the employer would be to have a settlement favorable for the group, i.e. favoring paying a standard salary and / or the new employee teaching according to demand. The salaries and subjects in question are different in both negotiations, but there is a reasonable one-one map of possible offers from one negotiation to the other, represented by the actions of offering "high salary" versus "low salary" and "choice subject" versus "arbitrary subject". The general idea for Prof. Sprout is to be a copy-cat:

- wait for QR to make an offer in negotiation N_2 about the salary;
- translate that offer to the terms of negotiation N_1, and propose a similar offer to FL;
- await the counteroffer from FL about subjects to teach, translate it to the terms of negotiation N_2, and make that offer to QR.

To model this, one needs to compose the trees in Figure 3 in a parallel fashion with interleaving moves, see Figure 4. It is clear that winning proposals are possible in both subtrees, namely on the two 'middle branches' of each, and therefore also in the parallel copy-cat negotiation.

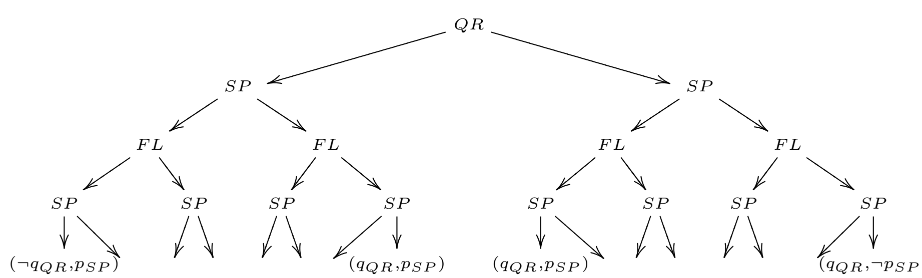


Fig. 4. A parallel game tree

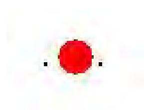
3 NEGEXT toolkit

We have developed the NEGEXT toolkit to aid students in strategic interactions during negotiations, with negotiations represented as extensive form games and their sequential and parallel combinations. NEGEXT has been written in Java version 1.7.0_02 using the Eclipse editor. It can run on any system that has a Java Virtual Machine (JVM) or Java-enabled web browsers. NEGEXT uses an applet to display the graphical user interface. The user can draw game trees using a menu of input nodes. NEGEXT can also generate the sequential combination of two game trees and all possible parallel combinations. Another feature of the software is that for both individual players and coalitions, it can check whether a strategy to achieve a proposition exists, and if so, point to such a winning strategy in the tree. We used a modified form of binary tree data structure for organizing the tree nodes.

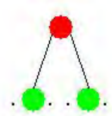
The main frame of the program consists of two panels. The left panel is the menu panel where the user can make choices for drawing input trees in a step-by-step fashion. The right panel has two canvas areas for displaying the game trees (Figure 6).

3.1 Drawing a tree using NEGEXT

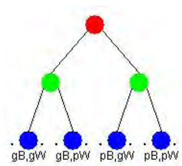
For negotiation problems that can be represented as extensive form games, one can draw a tree using NEGEXT. The user draws a tree from the root node to the terminal leaf nodes, one after another. For a current node, he needs to define the current level of the tree, the parent node, and the player whose turn it is. For the example described in Figure 2, the user may draw the tree as follows:



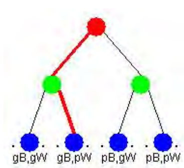
Level 0 for root node and no need to define the parent (as it has none). Select player i on the left panel (i =1 to 10). Players are represented by colors; in this case red represents Biocon.



Level 1 nodes: no need to define the parent (default parent is root node). Select the player i (i =1 to 10) for the left child and right child nodes; in this case both are green, representing Wockhard.



Level 2: terminal nodes with blue color; parent = 1 (level 1, left child) or parent = 2 (level 1, right child). Define the proposition and add the terminal nodes one after another.



Checking strategy for the coalition of player 1 and player 2 and for formula $g_B \wedge p_W$. The red line shows their joint strategy.

3.2 Checking for strategies that achieve goals in NEGEXT

After the user has drawn a tree, NEGEXT can check whether a strategy exists for either one player or a coalition of players by applying Algorithm 1. The basic idea behind the algorithm for finding strategies for an individual player i is to observe every edge from the root to the terminal nodes, considering a single edge from any node representing this player's turn and all possible edges from the nodes representing other players' turns. If there is no alternate strategy for a different player that prevents the player under consideration from having a winning strategy, then there exists a strategy for this player to achieve the proposition (see **Algorithm 1**).

Algorithm 1 Algorithm1 Finding strategy for Player(s)

```
Input: A single player i or set of players Sp, Formula φ
for all TreeNode do
  if IsTerminalNode(TreeNode) and IsTrue(Formula φ, Node TreeNode)
  then
    if Input of Player == Player i then
      if ParentOf(Node TreeNode) == Player i then
        if IsRoot(Player i) then
          Print Player i has a winning strategy for Formula;
        else
          Print Player i has no winning strategy for Formula;
        end if
      else if (ParentOf(ParentOf(Node TreeNode) )) == Player i then
        Check both left child and right child of ParentOf(Node TreeNode);
        if (Check is OK) then
          Print Player i has a winning strategy for Formula;
        else
          Print Player i has no winning strategy for Formula;
        end if
      end if
    else if Input of Player == set of players Sp then
      while (ParentOf(Node TreeNode) != null) do
        if ParentOf(Node TreeNode) == Player i ϵ Sp then
          Sp = Sp \ {Player i};
          Node = (ParentOf(Node TreeNode));
        else
          Print set of players Sp has no winning strategy for Formula.
        end if
      end while
      if Sp == ∅ then
        Print set of players Sp has a winning strategy for Formula
      end if
    end if
  end if
end for
```

For the user to find a strategy to achieve a particular formula, he needs to input the player's name ("player 1") or the names of the coalition of players ("player 1, player 2") and input the goal formula corresponding to a proposition at a terminal node. Then NEGEXT checks all tree nodes starting from the root to the terminal nodes: If a proposition of any terminal node t matches with the input formula, NEGEXT checks the parent node of t. If the parent node of t is matched with player i and it is also the root node, then player i has an individual strategy for this proposition. For example, in the left part of Figure

5, player 1 (red) has a strategy to achieve formula $\neg p$. If it is a node at level 1 at which it is another's turn, then the algorithm needs to check whether all its children have p. For example, in the right part of Figure 5, player 1 (red) has a strategy to achieve formula p. Otherwise, player i has no strategy to achieve this proposition.

To find a group strategy for a formula, NEGEXT also checks all tree nodes starting from the root to the terminal nodes. If a proposition of any terminal node t matches with the input formula, NEGEXT checks the parent node of t recursively up to the root. If all predecessor nodes of t are matched with all players in the group, then the group has a strategy for the formula, otherwise it does not. In the toolkit, the input tree is shown in the left canvas area, and the output tree with a strategy path (red line) is shown in the right canvas area (Figure 6).

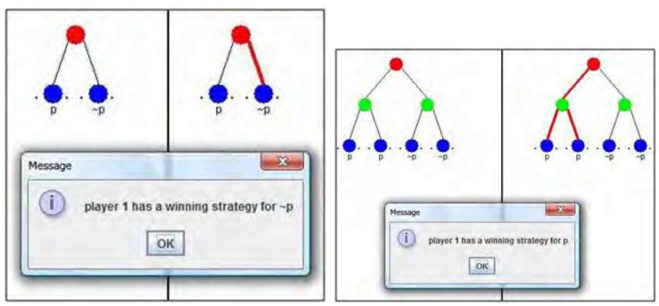


Fig. 5. Strategy checking for an individual player

4 Sequential and parallel combination in NEGEXT

Let us consider how NEGEXT can be used to analyze the stories in Subsections 2.1 and 2.2 which involve combinations of trees. For sequentially combining the trees of Subsection 2.1, NEGEXT first takes as inputs the different trees to combine, one after another, and then gives the sequential combination tree in a separate window (Figure 7). To combine the trees sequentially, NEGEXT first traverses all nodes of tree 1 from the root to the leaf nodes, and then concatenates tree 2 to all leaf nodes of tree 1. Suppose the user asks the system to find one possible strategy for achieving $(g_B \wedge p_W)$ by the coalition $\{B, W\}$, depicted by the red and green players in the screenshots. If company B chooses a genomic expert and company W chooses a proteomic expert, then they can achieve $(g_B \wedge p_W)$, as shown in Figure 7.

Now let us consider the parallel combination of trees, corresponding to the situation described in Section 2.2. As in the previous case, the different trees that are to be combined are taken as separate inputs, one after another, as given

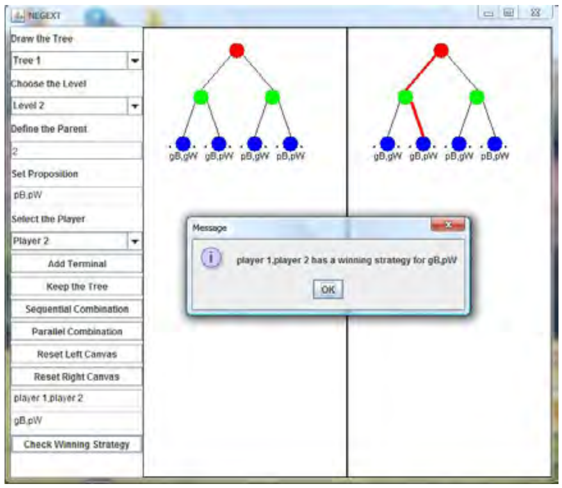


Fig. 6. Strategy checking for a group of players

in Figure 8. Here, agents QR, SP and FL are represented by pink, red, and green, respectively. Let us define abbreviations as follows: a for $p_{FL} \wedge \neg q_{SP}$; b for $p_{FL} \wedge q_{SP}$; c for $\neg p_{FL} \wedge q_{SP}$; d for $\neg q_{QR} \wedge p_{SP}$; e for $q_{QR} \wedge p_{SP}$; and f for $q_{QR} \wedge \neg p_{SP}$.

Before combining these trees in an interleaving way, we should note here that the parallel combination of two trees will give rise to a bunch of possible trees. These trees will appear in an enumeration, from which the user selects one combination as the final tree. In this case, the particular tree as depicted by the story is given in Figure 8. Figure 9 presents a copy-cat strategy that the common red player (Sprout) can follow in order to end up in a winning situation in the parallel game. Note that formally this strategy is a $\langle QR, SP, FL\rangle$-strategy, which may have been elicited by the user's question as to whether the set $\{QR, SP, FL\}$ can achieve the goal $p_{FL} \wedge p_{SP} \wedge q_{QR} \wedge q_{SP}$.Thus, using NEGEXT enables students to see clearly how players QR, FL, and SP can jointly achieve an intuitive goal.

5 Conclusions and future work

In this work we have presented a toolkit to represent negotiations which span over a finite time, and we consider the actions of the negotiators one after another in response to each other. Two examples have been provided to advocate the fact that many real-life negotiations can be aptly described by perfect information extensive form games. The NEGEXT toolkit can help students of negotiation to learn how to respond in order to achieve their goals, including situations where it is not easy to compute the optimal response. The current version of the toolkit has not been tested for learnability and usability yet, so a first step in future research will be to improve it on the basis of a usability study, in which

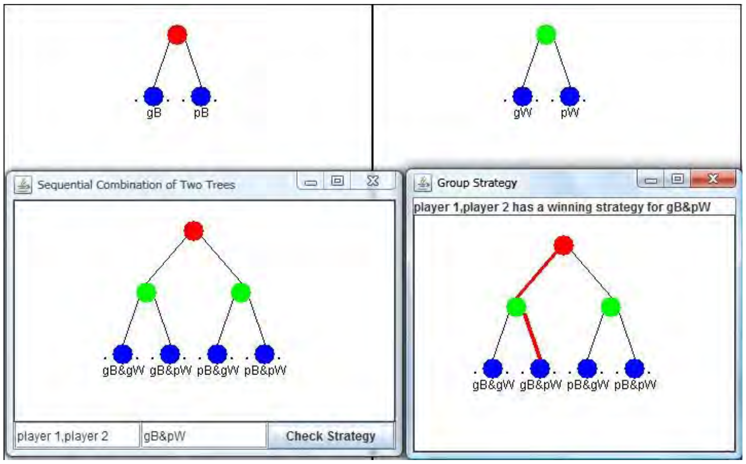


Fig. 7. Sequential combination and strategy checking

subjects will be asked to use the tool in order to find strategies given particular negotiation trees.

We used a binary tree data structure for drawing trees in the NEGEXT toolkit, implemented on a pure Java applet. In real life, players often have more than two options. In addition, current NEGEXT still allows only at most level 2 trees (root plus intermediate level plus leaves) for sequential or parallel combination, because of the node placing in the current graphical user interface. We aim to relax both restrictions in future work so that NEGEXT will be able to represent various types of branching at different nodes, as well as deeper trees.

In parallel combination, NEGEXT presents all possible parallel combinations to the user. NEGEXT allows the user to click the "Parallel Combination" button repeatedly in order to view all parallel combined trees in different interleaving ways. If the user wants to know of only an optimally combined tree, NEGEXT may confuse him. Solving this problem is also left for future work.

Note that while defining parallel combination of trees we only considered interleaving moves in between trees. We plan to incorporate simultaneous moves as well, bringing NEGEXT closer to the spirit of concurrent games, which have been used extensively in the context of alternating-time temporal logic [17].
The current version of NEGEXT is restricted to perfect information situations. However, in many real-life negotiations, the information dilemma looms large: Which aspects to make common knowledge and which aspects to keep secret or to divulge to only a select subset of co-players? For example, Raiffa distinguishes negotiation styles with "full or partial open truthful exchange" [1]. Such aspects of imperfect or incomplete information cause far-reaching asymmetries between parties, sometimes with grave consequences [18]. It will be future work to extend NEGEXT so that incomplete, imperfect, and asymmetric information can be incorporated in its tree representations and its strategic advice.

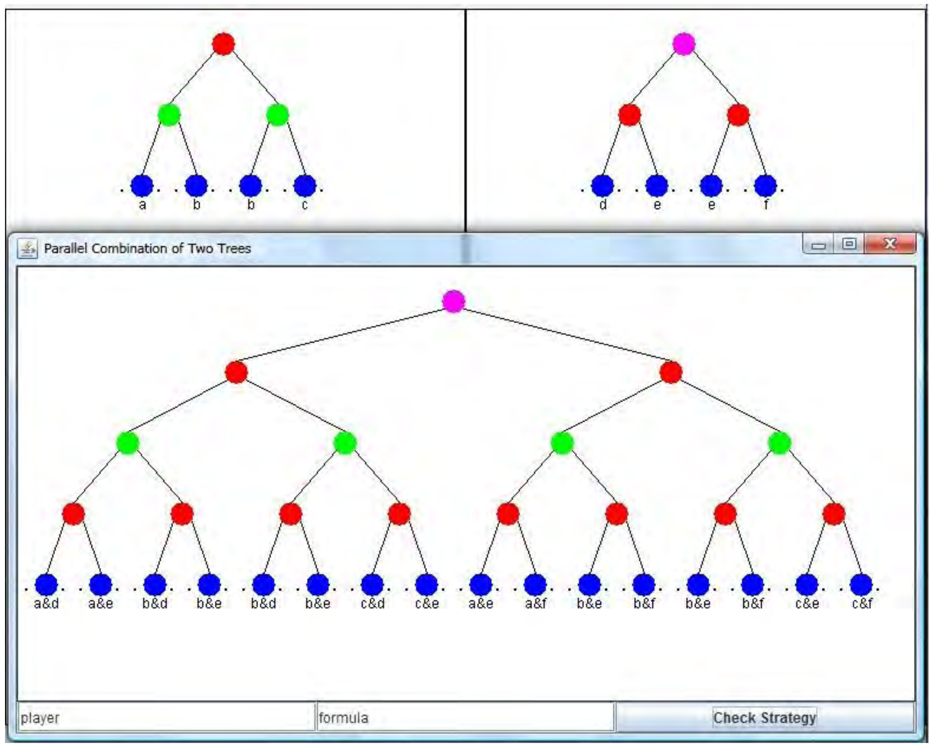


Fig. 8. An interleaving parallel combination

References

1. Raiffa, H., Richardson, J., Metcalfe, D.: Negotiation Analysis: The Science and Art of Collaborative Decision Making. Belknap Press of Harvard Univ. Press, Cambridge (MA) (2002)
2. Kuhn, H.: Game theory and models of negotiation. The Journal of Conflict Resolution **6**(1) (1962) 1–4
3. Rubinstein, A.: Perfect equilibrium in a bargaining model. Econometrica **50** (1982) 97–109
4. Kennan, J., Wilson, R.: Bargaining with private information. Journal of Economic Literature **31**(1) (March 1993) 45–104
5. Baron, D.P., Ferejohn, J.A.: Bargaining in legislatures. The American Political Science Review **83**(4) (1989) 1181–1206
6. Arrow, K., others, eds.: Barriers to Conflict Resolution. In Arrow, K., others, eds.: Barriers to Conflict Resolution, New York, Norton (1995)
7. Brams, S.J.: Negotiation Games: Applying Game Theory to Bargaining and Arbitration. Routledge, London (2003)
8. Birner, R.: Analyzing negotiation approaches in natural resource management: A case study of crop-livestock conflicts in Sri Lanka. In: Proceedings of the 25th In-

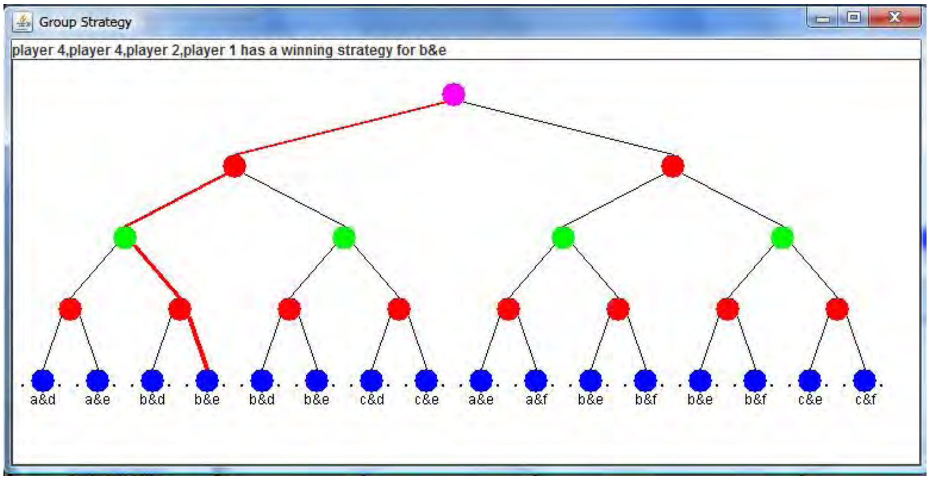


Fig. 9. A group strategy for combined tree

ternational Conference of Agricultural Economists (IAAE), Durban, International Association of Agricultural Economists (2003)

9. Forgó, F., Fulop, J., Prill, M.: Game theoretic models for climate change negotiations. European Journal of Operational Research **160**(1) (2005) 252–267
10. Rosenschein, J., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. MIT-Press, Cambridge (MA) (1994)
11. Parsons, S., Wooldridge, M., Jennings, N.: Agents that reason and negotiate by arguing. Journal of Logic and Computation **8** (1998) 261–292
12. Lin, R., Kraus, S., Wilkenfeld, J., Barry, J.: Negotiating with bounded rational agents in environments with incomplete information using an automated agent. Artificial Intelligence Journal **172**(6-7) (2008) 823–851
13. Hindriks, K., Jonker, C., Tykhonov, D.: Towards an open negotiation architecture for heterogeneous agents. In Klusch, M., others, eds.: Cooperative Information Agents XII. Volume 5180 of LNCS. Springer, Berlin (2008) 264–279
14. Endriss, U., Pacuit, E.: Modal logics of negotiation and preference. In Fisher, M., others, eds.: Logics in Artificial Intelligence. Volume 4160 of LNCS. Springer, Berlin (2006) 138–150
15. Sadri, F., Toni, F., Torroni, P.: Logic agents, dialogues and negotiation: An abductive approach. In: Proceedings AISB'01 Convention, AISB (2001)
16. Ghosh, S., Ramanujam, R., Simon, S.E.: Playing extensive form games in parallel. In Dix, J., others, eds.: Proceedings 11th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA XI). Volume 6245 of LNCS., Berlin, Springer (2010) 153–170
17. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. J. ACM **49** (2002) 672–713
18. Oakman, J.: The Camp David Accords: A case study on international negotiation. Technical report, Princeton University, Woodrow Wilson School of Public and International Affairs (2002)

Modal Logics for Argumentation

Cristian Gratie[1], John-Jules Ch. Meyer[2], and Adina Magda Florea[1]

[1] AI-MAS Group, Computer Science Department
University "Politehnica", Bucharest, Romania
cristian.gratie@cs.pub.ro, adina.florea@cs.pub.ro
[2] Intelligent Systems Group, Computer Science Department
Utrecht University, Utrecht, The Netherlands
jj@cs.uu.nl

Abstract. Previous research has linked modal logic and argumentation by providing formulas that describe extension-based semantics. This paper presents ongoing work on the same topic and proposes several research directions. More precisely, we aim to describe other argumentation semantics, with the use of multimodal logics such as Boolean Modal Logic and Propositional Dynamic Logic. Also, we aim to investigate how argumentation can be useful for modal logic.

Keywords: modal logic, argumentation, extension-based semantics

1 Introduction

Argumentation and modal logic were brought together in the work of Grossi [1]. A link between the two is appealing because, once established, it allows for porting theoretical results and algorithms from one domain to the other. While the foundations have been laid, we argue that there is still a lot that we don't know about mixing argumentation and modal logics.

This paper proposes several research directions for extending Grossi's work. We present results that we have already obtained and also the intuition behind considering other modal logics for certain argumentation semantics.

In the following section we provide a minimal argumentation background, while Section 3 discusses a part of Grossi's work together with the intuition behind using modal logic for argumentation. The main results and research directions we propose are presented in Section 4. The paper ends with conclusions in Section 5.

2 Argumentation Background

This section aims to provide a basic background on abstract argumentation, so that the context and the goals of our research can be better understood. We start by introducing argumentation frameworks as proposed by Dung [2] in 1995, with a slightly different notation.

Definition 1. *An* argumentation framework *is a pair* $F = (\mathcal{A}, \mathcal{R})$*, where* $\mathcal{A}$ *is a set of arguments and* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary attack relation on* $\mathcal{A}$*. We say that an argument* a attacks *another argument* b *and we write this as* $a \rightarrow b$ *iff* $(a, b) \in \mathcal{R}$*. Otherwise,* a does not attack b *and we write* $a \nrightarrow b$*. Also, we say that a set of arguments* S *attacks an argument* a *iff* S *contains an attacker of* a*. A set of arguments* S defends *an argument* a *(or, alternatively,* a *is* acceptable with respect to S*) iff* S *attacks all the attackers of* a*.*

Given a set of arguments and the attacks between them, one must be able to identify the arguments that are acceptable. Usually, several sets of acceptable arguments (extensions) can be identified. In the argumentation literature a method that provides the extensions of an argumentation framework (based on their properties, computed by an algorithm or using some other approach) is referred to as argumentation semantics. Several such semantics were defined, starting with those introduced by Dung in [2], which are listed in Definition 2.

Definition 2. *Let* $F = (\mathcal{A}, \mathcal{R})$ *be an arbitrary argumentation framework and let* S *be a set of arguments.*

- S *is* conflict-free *(*$\mathcal{CF}$*) iff* S *does not attack any of its arguments.*
- S *is* admissible *(*$\mathcal{AS}$*) iff* S *is conflict-free and* S *defends all its arguments.*
- S *is a* complete extension *(*$\mathcal{CO}$*) iff* S *is admissible and* S *contains all the arguments it defends.*
- S *is a* stable extension *(*$\mathcal{ST}$*) iff* S *is conflict-free and* S *attacks all the arguments it does not contain.*
- S *is a* preferred extension *(*$\mathcal{PR}$*) iff* S *is a maximal (with respect to set inclusion) admissible set.*
- S *is the* grounded extension *(*$\mathcal{GR}$*) of* F *iff* S *is the least (with respect to set inclusion) complete extension.*

For an argumentation semantics $\mathcal{S}em$ *we will use* $\mathcal{E}_{\mathcal{S}em}$ *to denote the set of all extensions prescribed by it, for example* $\mathcal{E}_{\mathcal{CO}}(F)$ *stands for all the complete extensions of* F*.*

Several additional semantics were defined in the literature, such as: ideal [3], semi-stable [4], eager [5], resolution-based grounded [6] and $CF2$ semantics [7]. We will refer some of them later on and only provide the distinguishing features that link them to modal logic. For further details, the reader is encouraged to consult the referred papers.

3 Argumentation and Modal Logic

This section links argumentation and modal logic based on the work of Grossi [1]. A minimal background of modal logic is assumed; a more detailed account of modal logics can be found in Blackburn et al [8].

We consider the basic modal language $ML(\Diamond)$, constructed in the usual way with $\top, \bot, \neg, \wedge, \vee, \Diamond, \Box$ and a set of proposition symbols $\mathcal{P}rop$. A Kripke model

$\mathfrak{M} = (W, R, V)$ is given by the set of worlds (or states) W, an accessibility relation $R \subseteq W \times W$ and a valuation function $V : \mathcal{P}rop \to 2^W$, where $V(p) \subseteq W$ is the set of worlds where p holds. Satisfiability of modal formulas (ϕ, ψ) at a given world w of a model $\mathfrak{M}$ is recursively defined as follows:

$$\begin{cases} \mathfrak{M}, w \nVdash \bot \\ \mathfrak{M}, w \Vdash p \Leftrightarrow w \in V(p) \\ \mathfrak{M}, w \Vdash \neg\phi \Leftrightarrow \mathfrak{M}, w \nVdash \phi \\ \mathfrak{M}, w \Vdash \phi \vee \psi \Leftrightarrow (\mathfrak{M}, w \Vdash \phi) \vee (\mathfrak{M}, w \Vdash \psi) \\ \mathfrak{M}, w \Vdash \Diamond\phi \Leftrightarrow \exists w'(w' \in W \wedge (w, w') \in R \wedge \mathfrak{M}, w' \Vdash \phi) \end{cases}$$

It is sometimes convenient to extend the valuation function V of a model from proposition symbols to arbitrary formulas so that $V(\phi)$ always gives the set of worlds where ϕ is true: $V(\phi) = \{w \in W \mid \mathfrak{M}, w \Vdash \phi\}$.

Now, given an argumentation framework $F = (\mathcal{A}, \mathcal{R})$ we can see the arguments as worlds and use the converse of the attack relation as an accessibility relation. The converse makes more sense because the argumentation semantics are usually defined by looking at the attackers of an argument rather than the arguments it attacks. With these observations, we can add a valuation function V and convert the argumentation framework to a Kripke model $\mathfrak{M} = (\mathcal{A}, \mathcal{R}^-, V)$.

Given an argumentation semantics $\mathcal{S}em$, we will say that the modal formula $\mathcal{S}em(x)$ describes $\mathcal{S}em$ iff the following relation holds for any modal formula ϕ:

$$(\mathcal{A}, \mathcal{R}^-, V), a \Vdash \mathcal{S}em(\phi) \Leftrightarrow V(\phi) \in \mathcal{E}_{\mathcal{S}em}(F)$$

where $\mathcal{S}em(\phi)$ denotes the formula obtained by replacing all occurrences of the formal variable x with ϕ. In words, the modal description $\mathcal{S}em(x)$ holds whenever the valuation of its argument corresponds to an extension prescribed by the argumentation semantics $\mathcal{S}em$.

Since the truth value of $\mathcal{S}em(\phi)$ does not depend on the world (argument) a where it is evaluated, we need a modal language that can express this, so we will add the global modalities $\mathbf{A}$ and $\mathbf{E}$ to $ML(\Diamond)$ and get the global modal language $ML(\Diamond, \mathbf{E})$. Satisfiability is extended using the following relations:

$$\begin{aligned} &\mathfrak{M}, w \Vdash \mathbf{E}\phi \Leftrightarrow \exists w'(w' \in W \wedge \mathfrak{M}, w' \Vdash \phi) \\ &\mathfrak{M}, w \Vdash \mathbf{A}\phi \Leftrightarrow \forall w'(w' \in W \to \mathfrak{M}, w' \Vdash \phi) \end{aligned}$$

In words, $\mathbf{A}\phi$ holds whenever ϕ holds in all worlds, whereas $\mathbf{E}\phi$ holds whenever ϕ holds in at least one world. For more details about the global modality see Chapter 7 of Blackburn et al [8]. In the work of Grossi [1], the following formulas are shown to describe the corresponding argumentation semantics.

$$\begin{aligned} &\mathcal{CF}(x) = \mathbf{A}(x \to \neg \Diamond x) \\ &\mathcal{AS}(x) = \mathbf{A}(x \to \neg \Diamond x \wedge \Box \Diamond x) \\ &\mathcal{CO}(x) = \mathbf{A}((x \to \neg \Diamond x) \wedge (x \leftrightarrow \Box \Diamond x)) \\ &\mathcal{ST}(x) = \mathbf{A}(x \leftrightarrow \neg \Diamond x) \end{aligned}$$

We present the intuition behind the first formula. We have that x is true for an argument a iff $a \in V(x)$, while $\Diamond x$ is true iff a has an attacker in $V(x)$. So, if

a is in $V(x)$, no attacker of a is in $V(x)$, exactly what conflict-free sets require. For a more detailed explanation please see [1].

4 Further Use of Modal Logic in Argumentation

In this section we describe the main goals of our current work on the use of modal logic for argumentation. We discuss several aspects that we consider important for extending the work of Grossi [1].

Consider and formalize impossibility results. It is suggested in [1] that the grounded and preferred semantics cannot be described within global modal logic. A bisimulation proof is provided for the preferred semantics, but the grounded semantics is just described within modal calculus. In our previous work, we have provided an even stronger impossibility result, showing that the only argumentation semantics that satisfies a small set of very reasonable constraints and can at the same time be described with a global modal formula is the complete semantics. The result covers the grounded and preferred semantics, but also newer proposals from the literature. The direct implication of this result is that further use of modal logic for argumentation (especially for the more interesting argumentation semantics) requires extending the basic modal language with more than the global modalities.

Consider other modal logics. In his work, Grossi provides a description for the grounded semantics, using modal mu calculus. There is, however, no indication whether other modal languages, less expressive than mu calculus, can describe it as well. Of the modal logics that lie in between global modal logic and mu calculus with respect to expressiveness, our main focus is on multimodal logics that contain operators on modalities, such as Propositional Dynamic Logic (PDL) and Boolean Modal Logic (BML).

BML modalities consider boolean operators on modalities, where negation corresponds to the complement of the accessibility relation, conjunction to the intersection and disjunction to the union of the accessibility relations. PDL, on the other hand, uses modalities to talk about programs, the basic modalities corresponding to atomic actions. The relevant operators are: composition (corresponds to the composition of the accessibility relations), iteration (corresponds to the transitive closure of the accessibility relation) and choice (which is in fact the same as disjunction from BML). PDL and BML are presented in Blackburn et al [8], in Chapter 1, respectively Chapter 7.

Consider the other argumentation semantics. The argumentation literature contains several other interesting argumentation semantics aside from those proposed by Dung. Since most of them are already covered by our impossibility result, we need something more expressive than global modal logic, so we expect the description to come from a multi-modal logic. For example, the weak reinstatement principle [9] requires one to consider both the attacking and the attacked arguments, so it needs the converse operator for the modalities; mutual attacks, used in the definition of resolution-based semantics [6], can be identified using the conjunction and the converse; indirect attacks, used for prudent semantics [10], can be described using iteration and composition. A combination of iteration, converse and conjunction (at least) is needed for capturing SCC-recursiveness [7].

Add argumentation-based operators to modal logics that cannot express them. Instead of focusing only on what modal logic can do for argumentation, it might be interesting to check whether the use of argumentation-based operators with modal logics gives rise to useful and well behaved (axiomatizable, decidable) logics.

5 Conclusions

In this paper we have provided a brief but intuitive presentation of the use of modal logic for argumentation. We have also shown that existing work can be continued in several directions that can benefit both the argumentation and the modal logic communities. We have also presented partial results and intuition on choosing certain modality operators for describing several argumentation semantics proposed in the literature.

References

1. Grossi, D.: On the logic of argumentation theory. In W. van der Hoek, G. Kaminka, Y. Lesperance, M.L., Sandip, S., eds.: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), IFAAMAS (2010) 409–416
2. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2) (1995) 321–357
3. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artificial Intelligence **171**(10-15) (July 2007) 642–674
4. Caminada, M.: Semi-stable semantics. In Dunne, P.E., Bench-Capon, T.J.M., eds.: Computational Models of Argument; Proceedings of COMMA 2006, IOS Press (2006) 121–130
5. Caminada, M.: Comparing two unique extension semantics for formal argumentation: Ideal and eager. In: Proceedings of the 19th Benelux Conference on Artificial Intelligence (BNAIC 2007). (2007) 81–87
6. Baroni, P., Giacomin, M.: Resolution-based argumentation semantics. In: Proceedings of the 2nd Conference on Computational Models of Argument (COMMA 2008), Tolouse, France (2008) 25–36
7. Baroni, P., Giacomin, M., Guida, G.: Scc-recursiveness: a general schema for argumentation semantics. Artificial Intelligence **168**(1-2) (October 2005) 162–210
8. Blackburn, P., De Rijke, M., Venema, Y.: Modal Logic. Volume 27 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge (2001)
9. Baroni, P., Giacomin, M.: On principle-based evaluation of extension-based argumentation semantics. Artificial Intelligence **171**(10-15) (July 2007) 675–700
10. Coste-Marquis, S., Devred, C., Marquis, P.: Prudent semantics for argumentation frameworks. In: Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005), Hong-Kong, China (2005) 568–572

Questions about Voting Rules, With Some Answers

Jan van Eijck and Floor Sietsma

CWI and ILLC, Amsterdam

Abstract. We raise questions about voting rules, and provide some of the answers. The method is to define a number of new formal properties of voting rules, and use these for classification and analysis. The aim is to get a better perspective on vices and virtues of individual voting rules.

Keywords: Voting rules, collective choice, multi-agent decision making.

1 Ballots, Profiles, Voting Rules

Voting is the process of selecting an item or a set of items from a finite set A of alternatives, on the basis of the stated preferences of a set of voters. See [3] for a detailed account. By calling the voters agents, voting can be seen as a form of multi-agent decision making.

A ballot is a linear ordering of A. Let $\mathbf{ord}(A)$ be the set of all ballots on A. We assume that the preferences of a voter are represented by a ballot. A profile is a vector of ballots, one for each voter. We assume voter anonymity, so it does not matter which voter has which ballot. The only thing that matters is the number of voters holding a certain ballot. Under this assumption voting profiles can be represented as mappings from ballots to non-negative integers. Another way to say this is by saying that our profiles are *quantified*, and the voting rules to be introduced below act like quantifiers: they calculate an outcome based on numbers (of voters holding certain ballots).

We will use $\mathbf{P}, \mathbf{Q}$ to range over profiles, and $\mathbf{b}, \mathbf{b}'$ to range over ballots.

Profiles can be represented as lists of non-negative integers, where the length of the list equals $m!$, with m the number of alternatives. The size of a profile is equal to the length of its ballots. If a profile $\mathbf{P}$ has size m, this means that its alternative set A has $|A| = m$.

If $\mathbf{b}$ is a ballot and $\mathbf{P}$ a profile, we use $\mathbf{P}(\mathbf{b})$ for the number of voters with ballot $\mathbf{b}$ in $\mathbf{P}$.

For example, assume the set of alternatives A equals $\{a, b, c\}$. Then the ballot that has a in first position, b is second position, and c in third position is abc. The following represents the profile $\mathbf{P}$ with $\mathbf{P}(abc) = 2$ (two voters hold ballot abc), $\mathbf{P}(bca) = 6$ (six voters hold ballot bca), and so on:

$$(abc, 2), (bca, 6), (cab, 0), (acb, 4), (cba, 0), (bac, 2).$$

Profiles can be normalized by dividing with the gcd of the list of all nonzero vote numbers. If $\mathbf{P}$ is a profile, we use $\mathbf{P}^\circ$ for the normalized form of the profile. The normalized form of the above example profile is:

$$(abc, 1), (bca, 3), (cab, 0), (acb, 2), (cba, 0), (bac, 1).$$

Definition 1. *An (anonymous) voting rule V for set of alternatives A is a function from A-profiles to $\mathcal{P}^+(A)$ (the set of non-empty subsets of A). A voting rule V is* resolute *if V maps every profile to a singleton set. If $V(\mathbf{P}) = B$, then the members of B are called the winners of $\mathbf{P}$ under V.*

Anonymity means that all voters are treated equally. This is built into our framework because we take profiles to be given by numbers of voters for each ballot.

If $\mathbf{P}$ is a profile for A, and π is a permutation of A, then $\mathbf{P}^\pi$ is the result of replacing x by $\pi(x)$ everywhere in $\mathbf{P}$. If $B \subseteq A$, then $\pi(B) = \{\pi(x) \mid x \in B\}$.

Definition 2. *A voting rule V is* neutral *if for every profile $\mathbf{P}$ and for every permutation π of the set A of alternatives,*

$$V(\mathbf{P}^\pi) = \pi(V(\mathbf{P})).$$

Neutrality means that all alternatives are treated equally.

Definition 3. *A voting rule V is* normal *if it holds for every profile $\mathbf{P}$ that $V(\mathbf{P}) = V(\mathbf{P}^\circ)$.*

Proposition 1. *There are anonymous and neutral voting rules that are not normal.*

Proof. Let V_k be given by $x \in V_k(\mathbf{P})$ if at least k voters have x at the top of their ballots. Then V_k is anonymous and neutral, but V_k is not normal. □

Question 1. Characterize the normal voting rules.

A scoring vector for ballots of size m is a list of non-negative integers $(w_0, \ldots, w_{m-1})$ satisfying $w_i \geq w_{i+1}$. The number w_i indicates the weight of position i in the ballot. The plurality rule has scoring vector $(1, 0, \cdots, 0)$. The anti-plurality rule (or: veto rule) has scoring vector $(1, \cdots, 1, 0)$. The Borda rule (see [2]) has scoring vector $(m-1, m-2, \cdots, 1, 0)$. The trivial voting rule that always returns the set of all alternatives has scoring vector $(0, \ldots, 0)$.

Every scoring vector w determines a voting rule S_w by means of:

$$S_w(\mathbf{P}) = \{x \in A \mid x \text{ has maximal } w\text{-scores in } \mathbf{P}\}.$$

For any scoring vector $w = (w_0, \ldots, w_{m-1})$, let w° be the result of dividing out common factors in $(w_0 - w_{m-1}, \ldots, w_{m-2} - w_{m-1}, 0)$. Call w° the normalization of w.

Proposition 2. *Scoring vector normalization does not affect the set of winners: for all* $\mathbf{P}$ *and all scoring vectors* w *it holds that* $S_w(\mathbf{P}) = S_{w^\circ}(\mathbf{P})$.

Proof. Let $(w_1, \ldots, w_{m-1})$ be a scoring vector. If x is a winner under this vector for profile $\mathbf{P}$, this means that the score N of x for $\mathbf{P}$ is maximal among the scores, i.e., greater than or equal to the score M of any alternative $y \neq x$. Scoring for the vector $(w_1 - w_{m-1}, \ldots, w_{m-2} - w_{m-1}, 0)$ give scores $N - kmw_{m-1}$ and $M - kmw_{m-1}$, so the score of x is still maximal. In the other direction, the scores change by adding a constant, so winners are also preserved.

Next, compare $(w_1, \ldots, w_{m-1})$ and $(w_1 K, \ldots, w_{m-1} K)$, with $K > 1$. Scores M and N for x and y under $(w_1, \ldots, w_{m-1})$ change into MK and NK. Since $M > N$ iff $MK > NK$, winners are not affected in either direction. □

Absolute majority is the voting rule that selects an alternative with more than 50 % of the votes as winner, and returns the whole set of alternatives otherwise. This is not the same as plurality, which selects an alternative that has the maximum number of votes as winner, regardless of whether more than half of the voters voted like this or not. Unanimity: if all voters have an alternative a at the top of their ballots then a is the winner, otherwise all alternatives tie for a win. Near-unanimity: if all but at most one of the voters have an alternative a at the top of their ballots then a is the winner, otherwise all alternatives tie for a win.

In the examples below we also use the Condorcet rule. the Copeland rule and the Hare rule. Here are the definitions (see also [15]).

A Condorcet winner is an alternative that beats every other alternative in pairwise contests. An alternative x beats another alternative y in a one-to-one contest if more than half of the voters prefer x to y. The Condorcet voting rule (proposed in 1785 by the marquis of Condorcet in [6]) selects the Condorcet winner if it exists, and the set of all alternatives otherwise. The Copeland voting rule [7] selects the alternative that maximizes the difference between the number of won and lost pairwise majority contests. The voting rule of single transferable vote, also known as the Hare rule (see [9]; the rule is also described by John Stuart Mill, with an attribution to Thomas Hare, in [11]), works as follows. If one of the candidates gets an absolute majority, that candidate wins. Otherwise prune the candidate(s) who is/are ranked first by the fewest number of voters from the profile, and repeat.

2 Profile Restriction

Profile restriction is computing a new profile for a subset of the alternative set of the original profile. The relative preferences of the voters in the new profile should remain unchanged.

If $B \subseteq A$, we use $\mathbf{P}^B$ for the result of restricting $\mathbf{P}$ to B. Formally, let $\mathbf{b} \sim_B \mathbf{b}'$ if the ballots $\mathbf{b}$ and $\mathbf{b}'$ become the same after restriction to the set B. Then $\mathbf{P}^B$ is given by

$$\mathbf{P}^B(\mathbf{b}) = \sum\{\mathbf{P}(\mathbf{b}') \mid \mathbf{b}' \in \mathbf{ord}(A), \mathbf{b} \sim_B \mathbf{b}'\}.$$

For example, let $\mathbf{P}$ be the following profile:

$$(abc, 1), (bca, 2), (cab, 0), (acb, 3), (cba, 0), (bac, 2).$$

Then the restriction of $\mathbf{P}$ to $\{a, b\}$ is given by

$$(ab, 4), (ba, 4),$$

the restriction of $\mathbf{P}$ to $\{a, c\}$ is given by

$$(ac, 6), (ca, 2),$$

and the restriction of $\mathbf{P}$ to $\{b, c\}$ is given by

$$(bc, 5), (cb, 3).$$

Definition 4. *A voting rule V is invariant for restriction if it holds for every $B \subseteq A$ and every profile $\mathbf{P}$ that*

$$V(\mathbf{P}) \neq A \textit{ and } V(\mathbf{P}) \cap B \neq \emptyset \textit{ imply } V(\mathbf{P}) \cap B = V(\mathbf{P}^B).$$

Note: Invariance for restriction can be viewed as a strengthening of a property that is known as Chernoff's condition [4], or as Sen's property alpha [14], or as Arrow's principle of invariance for irrelevant alternatives [1], applied to voting rules. A voting rule V satisfies this condition if winners in a subset B of the set of all alternatives remain winners if the choice is limited to B. In our terminology: if $V(\mathbf{P}) \cap B \subseteq V(\mathbf{P}^B)$.

Proposition 3. *The Hare rule and the Copeland rule are not invariant for restriction.*

Proof. For the Hare rule, consider the following profile $\mathbf{P}$ (ballots that are not mentioned get 0 votes):

$$(abc, 3), (bca, 2), (cab, 2).$$

If V is the Hare rule we get $V(\mathbf{P}) = \{a\}$. The restricted profile $\mathbf{P}^{\{a,c\}}$ looks like this:

$$(ac, 3), (ca, 4).$$

This gives $V(\mathbf{P}^{\{a,c\}}) = \{c\}$.

For the Copeland rule, consider the following profile:

$$(bacde, 1), (acdeb, 1), (debac, 1).$$

Under the Copeland rule, this is a win for a. Next, restrict the profile to $\{a, b, c\}$. This gives:

$$(bac, 2), (acb, 1).$$

Now b is the Copeland winner. □

Theorem 1. *The Condorcet rule is invariant for restriction.*

Proof. If there are no Condorcet winners then there is nothing to prove. A winner in the contest between a and b in $\mathbf{P}$ is still a winner in a contest between a and b in $\mathbf{P}^B$ for any B with $\{a, b\} \subseteq B$, and vice versa. □

Theorem 2. *Positional scoring rules with weights* $(w_0, ..., w_{m-1})$ *such that* $w_0 > w_{m-1}$ *are not invariant for restriction.*

Proof. Consider the following profile P consisting of 3 ballots of 7 voters:

$$(abc, 3), (bca, 2), (cab, 2).$$

Suppose the scoring rule gives weights (w_0, w_1, w_2) to the three positions. Then the scores of the candidates are as follows:

$$\begin{aligned} a &: 3w_0 + 2w_1 + 2w_2 \\ b &: 2w_0 + 3w_1 + 2w_2 \\ c &: 2w_0 + 2w_1 + 3w_2 \end{aligned}$$

Note that the difference in score between a and c is exactly $w_0 - w_2$. Since by assumption $w_0 > w_2$, the score of a is larger than that of c. This means that the set of winners is either $V(P) = \{a\}$ or $V(P) = \{a, b\}$. Now let us remove b from the set of candidates. In both cases, the intersection of $V(\mathbf{P})$ with the set of remaining candidates is $\{a\}$. The profile that remains after removing b is the following:

$$(ac, 3), (ca, 4).$$

Now since there is a different number of candidates, the scoring rule may give different weights to the positions. Suppose the weights are (v_0, v_1). Then the scores of the candidates are as follows:

$$\begin{aligned} a &: 3v_0 + 4v_1 \\ c &: 4v_0 + 3v_1 \end{aligned}$$

By assumption $v_0 > v_1$, so c wins the election. Because $V(\mathbf{P}) \cap \{a, c\} = \{a\}$, this shows that V is not invariant for restriction. $\square$

Question 2. Characterize the voting rules that are invariant for restriction.

Question 3. Is invariance for restriction a desirable property for a voting rule to have, or not?

The last question may seem a bit vague, but in any case, here are some relevant observations. Notice that restriction destroys information. If there are m alternatives and k voters then there are $m!$ possible ballots. The number of integer solutions for

$$x_1 + \cdots + x_n = k$$

under the condition that $x_i \geq 0$ for all $i = 1, \ldots, n$ is $\binom{n+k-1}{k}$ [10, Proposition 1.5]. Thus, for m alternatives and k voters there are

$$\binom{m! + k - 1}{k}$$

possible profiles. There are m ways to prune away one alternative. After pruning, there are $(m-1)!$ possible ballots, which leaves

$$\binom{(m-1)! + k - 1}{k}$$

profiles. All in all this gives

$$m\binom{(m-1)! + k - 1}{k}$$

possibilities.

To put these outcomes in perspective, here are some calculations for $m = 4$ and $k = 10$:

$$\binom{4! + 10 - 1}{10} = 92561040.$$

$$4\binom{3! + 10 - 1}{10} = 12012.$$

To see that the information destruction is vast, consider the case where the pruning process leaves only pairs. m alternatives give $m(m-1)$ pairs, so after pair pruning there are only $m(m-1)(k+1)$ possibilities left, since there are $k+1$ ways to split k into non-negative integers k_1, k_2 with $k_1 + k_2 = k$. For 4 alternatives and 10 voters, this reduces the number of possibilities from 92561040 to 132.

3 Profile Addition, Additivity of Voting Rules

Intuitively, we can merge two elections into a single election, by adding the numbers of votes for the various ballots. Call this operation $\oplus$. Note that the two operand profiles have to be of the same size (i.e., over the same set of alternatives). Note also that $(\mathbf{P} \oplus \mathbf{P})^\circ = \mathbf{P}^\circ$.

Definition 5. *A voting rule V is* additive *if it holds for all m-profiles* $\mathbf{P}$ *and* $\mathbf{Q}$ *that $V(\mathbf{P}) \cap V(\mathbf{Q}) \subseteq V(\mathbf{P} \oplus \mathbf{Q})$. Or in words: V is additive if winners of two separate elections concerning the same set of alternatives remain winners if the elections are merged.*

The following definition is from [16].

Definition 6. *A voting rule V is* consistent *if it holds for all m-profiles* $\mathbf{P}$ *and* $\mathbf{Q}$ *that $V(\mathbf{P}) \cap V(\mathbf{Q}) \neq \emptyset$ implies $V(\mathbf{P}) \cap V(\mathbf{Q}) = V(\mathbf{P} \oplus \mathbf{Q})$.*

Clearly, every consistent rule is additive, but the property of additivity is weaker than the property of consistency: see Proposition 8 below.

The requirement of additivity seems entirely reasonable. Still, there are respectable voting rules that do not satify it.

Proposition 4. *The Condorcet rule is not additive.*

Proof. Consider the following two profiles $\mathbf{P}$ and $\mathbf{Q}$:

$$(abc, 3), (bca, 0), (cab, 0), (acb, 0), (cba, 0), (bac, 2),$$

$$(abc, 1), (bca, 1), (cab, 1), (acb, 3), (cba, 3), (bac, 3).$$

The first of these has Condorcet winner a, the second has no Condorcet winner. So $V(\mathbf{P}) = \{a\}$ and $V(\mathbf{Q}) = \{a, b, c\}$, and therefore $V(\mathbf{P}) \cap V(\mathbf{Q}) = \{a\}$. Their sum is:

$$(abc, 4), (bca, 1), (cab, 1), (acb, 3), (cba, 3), (bac, 5).$$

The Condorcet winner of this sum is b. □

A voting rule satisfies the *Condorcet Criterion* if it always elects the Condorcet winner if there is one. The above proposition should worry anyone who thinks of the Condorcet criterion as a benchmark for voting rule quality.

Proposition 5. *The Hare rule is not additive.*

Proof. Consider the following two profiles (ballots that are not mentioned have no voters):

$$\mathbf{P} = \{(abcd, 5), (bacd, 6), (cabd, 2), (dabc, 10)\}.$$

$$\mathbf{Q} = \{(abcd, 4), (bacd, 4), (cabd, 8), (dabc, 2)\}.$$

If V is the Hare rule, then $V(\mathbf{P}) = V(\mathbf{Q}) = \{a\}$, and $V(\mathbf{P} \oplus \mathbf{Q}) = \{b\}$. □

Question 4. Is the Copeland rule additive?

Proposition 6. *The majority, unanimity and near-unanymity rules are additive.*

Proof. Suppose $\mathbf{P}$ and $\mathbf{Q}$ are m-profiles, V is the majority rule, and $a \in V(\mathbf{P}) \cap V(\mathbf{Q})$. Let $\mathbf{P}$ have N voters and $\mathbf{Q}$ have M voters. Then either no $x \in A$ has an absolute majority, or more than $N/2$ ballots in $\mathbf{P}$ have a in first position. Similarly, either no $x \in A$ has an absolute majority in $\mathbf{Q}$, or more than $M/2$ ballots in $\mathbf{Q}$ have a in first position. It follows that either no $x \in A$ has an absolute majority in $\mathbf{P} \oplus \mathbf{Q}$, in which case $a \in V(\mathbf{P} \oplus \mathbf{Q}) = A$, or $(N + M)/2$ ballots in $\mathbf{P} \oplus \mathbf{Q}$ have a in first position, i.e., a is the majority winner in $\mathbf{P} \oplus \mathbf{Q}$.

Same reasoning for the unanimity and near-unanymity rule. □

Proposition 7. *The near-unanymity rule is not consistent.*

Proof. Let V be the near-unanimity rule and let $\mathbf{P}$ be the following profile:

$$(ab, 2), (ba, 1).$$

Then $V(\mathbf{P}) = \{a\}$ and $V(\mathbf{P} \oplus \mathbf{P}) = \{a, b\}$. This shows that V is not consistent. □

Proposition 8. *Additivity does not imply consistency.*

Proof. Immediate from Propositions 6 and 7. □

Theorem 3. *Every positional voting rule is additive.*

Proof. Let V be a positional voting rule, and let $\mathbf{P}$, $\mathbf{Q}$ be a pair of m-profiles, for some m. Suppose $a \in V(\mathbf{P}) \cap V(\mathbf{Q})$. We have to show that $a \in V(\mathbf{P} \oplus \mathbf{Q})$. But this is immediate from the fact that if the score of a is maximal in $\mathbf{P}$ and $\mathbf{Q}$, it is also maximal in $\mathbf{P} \oplus \mathbf{Q}$. □

Question 5. Can we prove an if and only if for additivity?

4 Cycles, Reduction

Definition 7. *A permutation of alternatives π on $A = \{a_0, \ldots, a_{m-1}\}$ is a* full cycle *if π can be given as $a_0 = \pi^0(a_0) \mapsto \pi(a_0) \mapsto \pi^2(a_0) \mapsto \cdots \mapsto \pi^{m-1}(a_0)$, with the $\pi^i(a_0)$ all different.*

Any full cycle on A can be considered as a linear ordering on A with a_0 as least element, and vice versa. Thus, there are $(m-1)!$ full cycles on $\{a_0, \ldots, a_{m-1}\}$.

Customary notation for full cycles π on a list of m elements is to give the list:

$$(a_0, \pi(a_0), \pi^2(a_0), \cdots, \pi^{m-1}(a_0)).$$

For example, the full cycle in the following picture can be given as $(adbc)$.

$$\begin{tikzcd} a \arrow[r] & d \arrow[d] \\ c \arrow[u] & b \arrow[l] \end{tikzcd}$$

So cycles can also be represented as ballots. Moreover, cycles can be used to classify ballots. Two ballots $\mathbf{b}$ and $\mathbf{b}'$ are in the same ballot cycle if there is a full cycle π on A and a number k such that π^k maps $\mathbf{b}$ to $\mathbf{b}'$. If the ballot size is m, then each ballot is part of a cycle of m ballots.

For example, the ballot *abcd* is part of the following cycle:

$$abcd, bcda, cdab, dabc.$$

The following definition is from Saari [13].

Definition 8. *A profile is* reduced *if each cycle in the profile contains a ballot with no voters.*

Example 1. The profile

$$(abc, 3), (bca, 1), (cab, 0), (acb, 2), (cba, 0), (bac, 2)$$

is reduced.

Explanation: there are two cycles, $\{abc, bca, cab\}$ and $\{acb, cba, bac\}$, and both have a ballot with no voters.

Definition 9. *A profile is* balanced *if each cycle in the profile is such that each ballot in the cycle has the same number of voters. Use* $\mathbf{B}$ *for balanced profiles.*

Example 2. The profile

$$(abc, 1), (bca, 1), (cab, 1), (acb, 3), (cba, 3), (bac, 3)$$

is balanced.

Proposition 9. *For every profile* $\mathbf{P}$ *there exist a reduced* $\mathbf{Q}$ *and a balanced* $\mathbf{B}$ *such that* $\mathbf{P} = \mathbf{Q} \oplus \mathbf{B}$.

Definition 10. *If* $\mathbf{P} = \mathbf{Q} \oplus \mathbf{B}$*, as in Proposition 9, then call* $\mathbf{B}$ *the* surplus *of* $\mathbf{P}$ *and* $\mathbf{Q}$ *the reduced form of* $\mathbf{P}$*. Use* $\mathbf{P}^r$ *for the reduced form of* $\mathbf{P}$.

Proposition 10. *A profile* $\mathbf{P}$ *is both balanced and reduced iff* $\mathbf{P}$ *has no voters.*

Definition 11. *Call the operation of subtracting a balanced profile from* $\mathbf{P}$ reduction*. Call the operation of adding a balanced profile to* $\mathbf{P}$ dilution*.*

Here is an obvious **algorithm** for putting a profile $\mathbf{P}$ in reduced form:
For each cycle π of $\mathbf{P}$, let the minimum of the vote numbers in that cycle be k. Subtract k from every vote number in the cycle.

The surplus of a profile indicates by how much the profile can be reduced.

Example 3. The surplus of the profile

$$(abc, 4), (bca, 2), (cab, 1), (acb, 3), (cba, 3), (bac, 6)$$

is the profile

$$(abc, 1), (bca, 1), (cab, 1), (acb, 3), (cba, 3), (bac, 3).$$

Example 4. The reduced form of the profile

$$(abc, 4), (bca, 2), (cab, 1), (acb, 3), (cba, 3), (bac, 6)$$

is the profile

$$(abc, 3), (bca, 1), (cab, 0), (acb, 0), (cba, 0), (bac, 3).$$

Theorem 4. *Any anonymous and neutral voting rule maps a balanced profile to the set of all alternatives.*

Proof. Let $\mathbf{P}$ be a balanced profile for A. Let V be an anonymous and neutral voting rule. We must prove that $V(\mathbf{P}) = A$.

Suppose not, i.e., suppose there is some $b \notin V(\mathbf{P})$. There also is some $a \in V(\mathbf{P})$, for $V(\mathbf{P}) \neq \emptyset$.

Let σ be any permutation of A that satisfies $\sigma(a) = b$.

Observe that each cycle will remain a cycle under the permutation σ. Therefore, because of anonymity and the fact that $\mathbf{P}$ is balanced: $\mathbf{P}^{\sigma} = \mathbf{P}$. Because of neutrality $V(\mathbf{P}^{\sigma}) = \sigma(V(\mathbf{P}))$, and therefore $b = \sigma(a) \in V(\mathbf{P}^{\sigma}) = V(\mathbf{P})$, and contradiction. □

Theorem 5. *If $|A| = m$ then the number of voters in any balanced profile for A is a multiple of m.*

Proof. Each cycle in an m-profile has m elements. There are $(m-1)!$ cycles. Let cycle i have k_i voters. Then all in all we have $m \sum_{i=1}^{(m-1)!} k_i$ voters. □

Definition 12. *A voting rule V* is safe for dilution *if it holds for all profiles $\mathbf{P}$ and balanced profiles $\mathbf{B}$ that $V(\mathbf{P}) \supseteq V(\mathbf{P} \oplus \mathbf{B})$.*

Safety for dilution means that dilution does not introduce new winners.

Definition 13. *A voting rule V* is safe for reduction *if it holds for all profiles $\mathbf{P}$ and balanced profiles $\mathbf{B}$ that $V(\mathbf{P}) \subseteq V(\mathbf{P} \oplus \mathbf{B})$.*

Safety for reduction means that reduction does not introduce new winners.

Theorem 6. *Any anonymous, neutral and additive voting rule is safe for reduction.*

Proof. Assume V is anonymous and neutral. Then $V(\mathbf{B})$ equals the set of all alternatives. By additivity we have:

$$V(\mathbf{P}) = V(\mathbf{P}) \cap V(\mathbf{B}) \subseteq V(\mathbf{P} \oplus \mathbf{B}).$$

□

Proposition 11. *The Condorcet rule is neither safe for reduction nor safe for dilution.*

Proof. Consider the profile:

$$(abc, 1), (bac, 3), (bca, 1), (acb, 5), (cab, 4), (cba, 3).$$

The Condorcet winner for this profile is a. The reduced form of this is:

$$(abc, 0), (bac, 0), (bca, 0), (acb, 2), (cab, 3), (cba, 0).$$

The Condorcet winner for the reduced profile is c. □

Proposition 12. *The absolute majority rule is safe for reduction, but not safe for dilution.*

Proof. The example from Proposition 11 works here as well. In the reduced profile

$$(abc, 0), (bac, 0), (bca, 0), (acb, 2), (cab, 3), (cba, 0)$$

there is an absolute majority for c. Dilute this profile with

$$(abc, 1), (bac, 1), (bca, 1), (acb, 3), (cab, 3), (cba, 3).$$

There is no absolute majority in the diluted profile

$$(abc, 1), (bac, 3), (bca, 1), (acb, 5), (cab, 4), (cba, 3).$$

□

Theorem 7. *Any voting rule V with positional scoring will assign to every alternative in a balanced profile* **B** *the same score.*

Proof. Let **B** be a balanced m-profile. Then there are $(m-1)!$ cycles, and there are k_i voters in each ballot in the i-th cycle. Let V be a positional voting rule with $(x_0, \cdots, x_{m-1})$ as its scoring vector. Let π_i be an arbitrary cycle of **P**, let a be an arbitrary alternative, and let j be an arbitrary position (i.e., $0 \leq j < m$). Then the score for a for this position in the cycle under the voting rule is given by $k_i x_j$, for a occurs in this position exactly once in the cycle. Summing over the cycles, we get that a collects the following score in **B**:

$$\sum_{i=1}^{(m-1)!} k_i x_j.$$

Summing over the positions, we see that a collects the score:

$$\sum_{j=0}^{m-1} \sum_{i=1}^{(m-1)!} k_i x_j.$$

Since a was arbitrary, every alternative collects this same score. □

Theorem 8. *Any voting rule V with positional scoring is safe for reduction and safe for dilution.*

Proof. Let **P** be an m-profile, and let **B** be a balanced m-profile.

Since **B** is balanced, it follows from the previous Theorem that the scores for the alternatives under V for **P** can be computed from those for $\mathbf{P} \oplus \mathbf{B}$ by subtracting a constant c from each score, and vice versa, by adding a constant c to each score. These subtractions and additions do not affect the outcome of V. □

Question 6. Does the converse hold as well? If a voting rule is safe in both directions, does it follow that it is positional?

If this is too difficult to answer, the following questions may be easier:

Question 7. If a voting rule is safe in both directions, does it follow that it is additive?

Question 8. If a voting rule is safe in both directions, does it follow that it is consistent?

Notice that for all voting rules V that are not invariant under reduction, the derived voting rule V^r defined by $V^r(\mathbf{P}) = V(\mathbf{P}^r)$ is different from V. Also, for any voting rule V, the derived voting rule V^r is invariant for reduction and dilution by definition.

Question 9. What are the formal properties of the Condorcet^r rule?

Question 10. Are there non-positional voting rules V with the property that V^r is positional?

5 Strategizing

Strategizing is replacing a ballot $\mathbf{b}$ by a different one, $\mathbf{b}'$, in the hope or expectation to get a better outcome (where better is "closer to $\mathbf{b}$" in some sense).

As is explained in [15], there are many ways to interpret 'better'. One way is that X is better than Y if X weakly dominates Y, that is if every $x \in X$ is at least as good as every $y \in Y$ and some $x \in X$ is better than some $y \in Y$. Formally:

Definition 14. *If $X, Y \subseteq A$ $X \neq \emptyset$, $Y \neq \emptyset$, and $\mathbf{b} \in \mathbf{ord}(A)$, then $X >_{\mathbf{b}} Y$ if $\forall x \in X \forall y \in Y$: $x = y$ or x is above y in $\mathbf{b}$, and $\exists x \in X \exists y \in Y$: x is above y in $\mathbf{b}$.*

Let $\mathbf{P} \sim_i \mathbf{P}'$ express that $\mathbf{P}$ and $\mathbf{P}'$ differ only in the ballot of voter i.

Definition 15. *A voting rule is* strategy-proof *if $\mathbf{P} \sim_i \mathbf{P}'$ implies $V(\mathbf{P}) \geq_i V(\mathbf{P}')$, where $\geq_i$ expresses 'betterness' according to the i-ballot in $\mathbf{P}$.*

Note: the following definition does not assume voter anonymity. The definition uses $\mathbf{P}^{-i}$ for the result of removing the ballot of voter i from profile $\mathbf{P}$.

Definition 16. *A voting rule V is* monotone *if for any profile $\mathbf{P}$ and any alternative $a \in V(P)$, if $\mathbf{b}'_i$ is a new ballot for some voter i that results from moving a up in the ranking of i and not changing the order between the other alternatives, then $a \in V(\mathbf{P}^{-i} \cup \mathbf{b}'_i)$.*

Definition 17. *A voting rule is* resolute *if* $V(\mathbf{P})$ *is a singleton for any profile* $\mathbf{P}$.

Theorem 9. *Any resolute voting rule that is monotone and invariant for restriction is strategy-proof.*

Proof. Take some resolute voting rule V, profile $\mathbf{P}$ and voter i. Suppose $V(\mathbf{P}) = \{a\}$. Then for any other candidate c, $V(P^{\{a,c\}}) = \{a\}$ by winner preservation under restriction. Suppose i can strategize by submitting some dishonest ballot $\mathbf{b}'_i$ in order to elect some candidate b such that $b >_i a$.

Let $V(\mathbf{P}^{-i} \cup \mathbf{b}'_i) = \{b\}$. It is possible that $a >'_i b$. By monotonicity, if we construct the ballot $\mathbf{b}''_i$ by moving b up until $b >''_i a$ then $V(P^{-i} \cup \mathbf{b}''_i) = \{b\}$. By winner preservation under restriction, $V(\mathbf{P}^{\{a,b\}}) = a$. But because $b >''_i a$, $(\mathbf{P}^{-i} \cup \mathbf{b}''_i)^{\{a,b\}} = \mathbf{P}^{\{a,b\}}$ so $V((\mathbf{P}^{-i} \cup \mathbf{b}''_i)^{\{a,b\}}) = V(\mathbf{P}^{\{a,b\}}) = \{a\}$. This contradicts our assumption that $b >_i a$, so strategizing is not possible. □

The concept of weak domination is borrowed from game theory (see, e.g., [12]). As Taylor [15, p. 39] remarks:

> In point of fact, an election can be thought of as a game in which a strategy for a player (voter) is a choice of ballot, and the outcome of the game is the set of winners in the election.

To formalize this, let a ballot vector for A be a list of A-ballots $(\mathbf{b}_0, \ldots, \mathbf{b}_{n-1})$. We assume that a ballot vector represents the true ballots of voters $\{0, \ldots, n-1\}$, in the sense that $\mathbf{b}_i$ represents the true preferences of voter i.

Define a payoff function in terms of $\geq_{\mathbf{b}}$ from Definition 14, as follows.

Definition 18. $payoff(\mathbf{b}, X) = |\{Y \mid Y \in \mathcal{P}^+(A), X >_{\mathbf{b}} Y\}|$.

Thus, the payoff of a voting outcome X, given a ballot $\mathbf{b}$ serving as a point of reference, is the size of the set of possible voting outcomes that are strictly worse than X.

This payoff function can be used to define the value of a move for a player with true ballot $\mathbf{b}$, as follows:

Definition 19. $move(V, \mathbf{P}, \mathbf{b}, \mathbf{b}') = payoff(\mathbf{b}, V(\mathbf{P}'))$, *where* $\mathbf{P}'$ *is the result of adding ballot* $\mathbf{b}'$ *to* $\mathbf{P}$.

The game for voting rule V and ballot vector $(\mathbf{b}_0, \ldots, \mathbf{b}_{n-1})$ is now given in terms of the move function, as follows.

Definition 20. *Assume* $\mathbf{P}$ *is some profile for* $n-1$ *voters. Then*

$$Game(V, (\mathbf{b}_0, \ldots, \mathbf{b}_{n-1}), \mathbf{P}, i) = \{(\mathbf{b}, move(V, \mathbf{P}, \mathbf{b}_i, \mathbf{b}) \mid \mathbf{b} \in \mathbf{ord}(A)\}.$$

Thus, we see that a voting rule together with a ballot vector determines an n-player game Game$(V, (\mathbf{b}_0, \ldots, \mathbf{b}_{n-1}))$, where each voter has a choice between the members of $\mathbf{ord}(A)$ (the possible ballots), and where the payoff for player i for a profile $\mathbf{P}$ for $n-1$ voters, and for (cast) ballot $\mathbf{b}$ is given by move$(V, \mathbf{P}, \mathbf{b}_i, \mathbf{b})$.

Clearly, a voting rule V is strategy-proof iff it holds for each ballot vector $(\mathbf{b}_0, \ldots, \mathbf{b}_{n-1})$ that the profile $\mathbf{P}$ corresponding to vector $(\mathbf{b}_0, \ldots, \mathbf{b}_{n-1})$ is a Nash equilibrium for Game$(V, (\mathbf{b}_0, \ldots, \mathbf{b}_{n-1}))$. But we can take a more general perspective:

Question 11. Characterize the ballot vectors for which Game$(V, (\mathbf{b}_0, \ldots, \mathbf{b}_{n-1}))$ (for given V) has nontrivial pure Nash equilibria.

The following proposition shows that there are many trivial Nash equilibria.

Proposition 13. *Let V be a voting rule and $\mathbf{P}$ a profile. If for all i and $\mathbf{P}'$ with $\mathbf{P} \sim_i \mathbf{P}'$ it holds that $V(\mathbf{P}) = V(\mathbf{P}')$, then $\mathbf{P}$ is a Nash equilibrium for Game$(V, (\mathbf{b}_0, \ldots, \mathbf{b}_{n-1}))$, for any ballot vector $(\mathbf{b}_0, \ldots, \mathbf{b}_{n-1})$.*

Proof. No voter has an incentive to deviate from his ballot in $\mathbf{P}$, as it makes no difference for the outcome. □

Players who realize they have lost the game have no incentive to strategize. Similarly for players who realize they have won the game. If all players know they are in one of these two categories, no strategizing will occur. Compare also [5] for a first analysis of the crucial role of knowledge in strategic voting.

Question 12. Analyze the abstention game for a voting rule and a ballot vector, where each player has the choice between casting his true ballot or abstaining from the vote. A voting rule V is abstention-proof if it holds for each ballot vector $(\mathbf{b}_0, \ldots, \mathbf{b}_{n-1})$ that the profile corresponding to that vector is a Nash equilibrium for the abstention game for V and $(\mathbf{b}_0, \ldots, \mathbf{b}_{n-1})$. Characterize the voting rules that are abstention-proof.

6 Conclusion and Further Research

We have introduced a number of concepts to classify and analyze voting rules: invariance for restriction, additivity, safety for dilution, safety for reduction. We have demonstrated the use of these concepts by proving some new results about voting rules. Further clarification of relations between voting rules will no doubt result from finding answers to the list of questions we have left open. Answering the list of questions we have raised (or in some cases, finding the answers in the literature) is future work.

We have an implemented system for voting with anonymous voting rules that we used for checking a number of the factual propositions in this paper. The present version of the software implements strategizing, under the assumption that the rest of the profile is known to the strategizer. Our intention is to extend this implementation to an epistemic model checker for voting under partial uncertainty about the profile. The software is available on the internet as a literate Haskell program [8].

Acknowledgement Thanks to Krzysztof Apt, Ulle Endriss and Sunil Simon for enlightening discussions on the topic of this paper. The participants of the Lorentz workshop on Modeling Strategic Reasoning (Leiden, Feb 20–24, 2012) and three anonymous LAMAS 2012 reviewers also gave useful feedback.

References

1. K. Arrow. *Social Choice and Individual Values*. Wiley, New York, 1951, second edition: 1963.
2. J.-C. de Borda. *Mémoire sur les élections au scrutin*. Histoire de l'Académie Royale des Sciences, Paris, 1781.
3. Steven J. Brams and Peter C. Fishburn. Voting procedures. In K.J. Arrow, A.K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume I, chapter 4. Elsevier, 2002.
4. H. Chernoff. Rational selection of decision functions. *Econometrica*, 22:422–443, 1951.
5. Samir Chopra, Eric Pacuit, and Rohit Parikh. Knowledge-theoretic properties of strategic voting. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA-2004)*, pages 18–30, 2004.
6. M. le Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Imprimerie Royale, Paris, 1785.
7. A.H. Copeland. A "reasonable" social welfare function. Seminar on Mathematics in Social Sciences, 1951.
8. Jan van Eijck and Floor Sietsma. Basic voting theory. Literate Haskell Program, available from `www.cwi.nl/~jve/software/voting/`, 2012.
9. T. Hare. *The Election of Representatives, Parliamentary and Municipal: A Treatise*. Longman, Green, London, 1861.
10. Stasys Jukna. *Extremal Combinatorics, with Applications in Computer Science — Second Edition*. Texts in Theoretical Computer Science. Springer, 2011.
11. John Stuart Mill. *Considerations of a Representative Government*. Parker, Son, and Bourn, London, 1861. Electronically available from Project Gutenberg.
12. Martin J. Osborne. *An Introduction to Game Theory*. Oxford University Press, New York, Oxford, 2004.
13. D.G. Saari. *Basic Geometry of Voting*. Springer, 1995.
14. Amartya Sen. *Collective Choice and Social Welfare*. Holden-Day, 1970.
15. Alan D. Taylor. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, 2005.
16. H.P. Young. Social choice scoring functions. *SIAM Journal on Applied Mathematics*, 28(4):824–836, 1975.

Public Announcements are Exponentially More Succinct than "Everybody Knows" and "Somebody Knows"

Wiebe van der Hoek and Petar Iliev

Department of Computer Science, University of Liverpool,
Ashton Building, Ashton Street, Liverpool, L69 3BX, UK
{wiebe,pvi}@liverpool.ac.uk

Abstract. Formula-Size games are generalisation of Ehrenfeucht - Fraisse games that are used to give bounds on the length of first-order formulas rather than their quantifier depth. Using a suitable version of these games, we prove that the public announcement logic PAL is exponentially more succinct than both the logics ELS (basic epistemic logic extended with an operator for "somebody knows") and ELE (basic epistemic logic extended with an operator for "everybody knows").
Keywords: epistemic logic, artificial intelligence, multi-agent systems, finite model theory of modal logics, formula-size games

1 Introduction

Consider the following general question. Let L_1 and L_2 be two formalisms that express the same class of "properties" or "situations". Is one of the formalisms representationally more succinct (allowing for a more "economical" representation of information) than the other and by how much? A famous instance of this question is whether there is a family of Boolean functions for which boolean circuits can be exponentially more succinct than boolean formulas. It is widely believed that the answer is beyond our current mathematical knowledge and techniques. On a more positive note, there are some well-known instances of the question that have been solved, e.g. CTL^+ is exponentially more succinct than CTL (see [9] and [1]); the four-variable fragment of first-order logic is exponentially more succinct than the three-variable one (see [6]); first-order logic is non-elementary more succinct than LTL (see [8]).

Here we prove that the logic PAL (basic epistemic logic EL extended with an operator for "public announcements") is exponentially more succinct than the logic ELS (i.e., EL extended with an operator for "somebody knows") and the logic ELE (i.e., EL extended with an operator for "everybody knows"). Hence, there are semantic properties that are more economically expressed by using the "public announcement" operator, than by using the *E* or *S* operator. One disadvantage of our proof is that the commonly accepted semantics of epistemic logic is given via models in which every relation is a relation of equivalence. This is not true for the models we use. We intend to solve this technically more complicated problem in a future paper.

Intuitively, a public announcement formula $[\phi]\psi$ is true at a given point *w* in some model *M* if the truth of ϕ in *w* implies that ψ remains true at *w* when we remove from *M* all points that do not satisfy ϕ. It is well known that every such formula is equivalent to an EL formula.

A formula of the form $E_{\{a,\ldots,m\}}\phi$ (intuitively, everybody in the group of agents $\{a,\ldots,m\}$ knows ϕ) is just an abbreviation of the EL formula $K_a\phi \wedge \ldots \wedge K_m\phi$ (intuitively, a knows ϕ and b knows ϕ, etc). In the same way, a formula of the form $S_{\{a,\ldots,m\}}\phi$ (intuitively, somebody in the group of agents $\{a,\ldots,m\}$ knows ϕ) is just an abbreviation of the EL formula $K_a\phi \vee \ldots \vee K_m\phi$ (intuitively, at least one agent in the group knows ϕ). It is obvious that adding formulas of the form $E_{\{a,\ldots,m\}}\phi$ to EL does not increase its expressive power; what is more, the computational complexity of the satisfiability problem for ELE is the same as that for EL [7]. In the same way, ELS is not more expressive than EL. It is known, however, that PAL, ELE and ELS are exponentially more succinct than EL. We invate the reader to consult [7] and [4] for proofs of these results. This shows that abbreviations are not just "syntactic sugar" but a genuinely powerful tool. Proving that PAL is exponentially more succinct than ELS and ELE is a further continuation of the line of work started in [4] and [7].

We would like to stress that despite the fact that we can translate every ELE, ELS and PAL formula into an equivalent EL formula, it is by no means obvious that our translation is optimal. In particular, it may seem "trivial" that deep nesting of the E operator, as in the formula $\psi = E_{\{a,b\}}E_{\{a,b\}}p$, "inevitably" leads to an exponential blow-up in the length of any equivalent EL formula but this "intuition" is supported only by the fact that we define ψ as an abbreviation of the formula $\phi = K_a(K_ap \wedge K_bp) \wedge K_b(K_ap \wedge K_bp)$. However, *it does not follow* that there is no "considerably" shorter EL formula θ that is equivalent to ψ especially if we impose specific conditions on our semantics. A proof that there is no such θ on $S5$ models can be found in [4].

As far as we know, the first *systematic* study of knowledge representation formalisms in terms of succinctness is [5]. We would like to stress that many of the results in this important paper are based on unproven computational complexity conjectures, while the results presented here are absolute in the sense that we do not rely on such conjectures.

2 Preliminaries

In this section we fix the main definitions and provide the technica tools we use.

2.1 Epistemic logic, Public announcements, Everybody knows, and Somebody knows

Definition 1 (PAL with Everybody knows and Somebody knows). *The signature of public announcement logic with operators for "everybody knows" and "somebody knows" ($ESPAL_n^m$) is a pair $S = \{P, I\}$, where $P = \{p_0, p_1, \ldots, p_n\}$ is a finite set of propostional symbols and $Ag = \{a_1, \ldots, a_m\}$ is a finite set of agent names. Let $\mathcal{P}Ag$ be the set of nonempty subsets of Ag. The formulas of $ESPAL_n^m$ are built according to the rule:*

$$\psi := \bot \mid \top \mid p \in P \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid K_a\psi \mid M_a\psi \mid S_\Gamma\psi \mid E_\Gamma\psi, \mid [\psi]\psi,$$

where $a \in Ag$ and $\Gamma \in \mathcal{P}Ag$.

A formula $K_a\varphi$ is read as 'agent a knows φ' and $M_a\varphi = \neg K_a\neg\varphi$ is read as 'given the information to a, he cannot rule out φ', or, briefly, 'agent a considers φ possible'.

Definition 2 **(PAL_n^m, $ELES_n^m$, ELE_n^m, ELS_n^m, and EL_n^m).** *The formulas of the logic PAL_n^m are the formulas of $ESPAL_n^m$ with the exception of all formulas of the form S_Γ and E_Γ. The formulas of the logic $ELES_n^m$ are the formulas of the logic $ESPAL_n^m$ with the exception of all formulas of the form $[\psi_1]\psi_2$. The formulas of the logic ELE_n^m are the formulas of the logic $ESPAL_n^m$ with the exception of all formulas of the form $[\psi_1]\psi_2$ and $S_\Gamma\psi$. The formulas of ELS_n^m are the formulas of $ESPAL_n^m$ with the exception of all formulas of the form $[\psi_1]\psi_2$ and $E_\Gamma\psi$. Finally, the formulas of epistemic logic EL_n^m are the formulas of $ESPAL_n^m$ with no formulas of the forms $E_\Gamma\psi$, $S_\Gamma\psi$, and $[\psi_1]\psi_2$ allowed.*

Next we define the length of a formula φ (denoted $|\varphi|$). Usually, $|\varphi|$ is defined as the length of the binary string that encodes φ given some binary encoding of the formulas of $ESPAL_n^m$. Such precission, however, is irrelevant for our purposes and we define formula length as follows.

Definition 3. *The length of an $ESPAL_n^m$ formula φ, denoted $|\varphi|$, is defined by induction on the structure of φ.*

(φ is either a propositional symbol $p \in P$, or $\top$, or $\bot$): $|\varphi| = 1$;
(φ is $\neg\psi$): $|\varphi| = 1 + |\psi|$;
(φ is either $\psi_1 \wedge \psi_2$ or $\psi_1 \vee \psi_2$) : $|\varphi| = 1 + |\psi_1| + |\psi_2|$;
(φ is either $K_a\psi$, or $M_a\psi$, or $S_\Gamma\psi$, or $E_\Gamma\psi$): $|\varphi| = 1 + |\psi|$;
(φ is $[\psi_1]\psi_2$): $|\varphi| = |\psi_1| + |\psi_2|$.

We give the usual Kripke semantics for $ESPAL_n^m$ (see for example [2]).

Definition 4 **(Model).** *A model for the signature $S = \{P, Ag\}$ is a triple $M = \langle W, R, V\rangle$, where*

- *W is a set of points;*
- *$R : Ag \to 2^{W\times W}$ is a function that assigns a binary relation $R(a)$ on W to every $a \in Ag$. We write wR_av for $(w, v) \in R(a)$ and say that v can be reached from w in one a-step.*
- *$V : P \to 2^W$ is a function that assigns a subset $V(p) \subseteq W$ to every $p \in P$.*

*A model $M = \langle W, R, V\rangle$ is said to be **finite** if W is finite. Given a model $M = \langle W, R, V\rangle$, a **pointed model** is a pair (M, w), where $w \in W$. Sets of pointed models are denoted $\mathbb{M}$, $\mathbb{N}$, $\mathbb{M}_1$, $\mathbb{N}_1$, $\mathbb{M}_2$, $\mathbb{N}_2$, etc.*

The notion "formula φ is true in a pointed model (M, w)" is defined in the well-known way (see for example [3]). In particular:

$$
\begin{array}{ll}
(M, w) \models M_a\psi & \text{iff there is a } v \in W \text{ such that } wR_av \text{ and } (M, v) \models \psi; \\
(M, w) \models K_a\psi & \text{iff } (M, v) \models \psi \text{ for all } v \in W \text{ such that } wR_av; \\
(M, w) \models E_\Gamma\psi & \text{iff } (M, w) \models \bigwedge_{a\in\Gamma} K_a\psi; \\
(M, w) \models S_\Gamma\psi & \text{iff } (M, w) \models \bigvee_{a\in\Gamma} K_a\psi. \\
(M, w) \models [\psi_1]\psi_2 & \text{iff If } (M, w) \models \psi_1 \text{ then } (M|_{\psi_1}, w) \models \psi_2.
\end{array}
$$

where $M|_{\psi_1}$ is the model obtained from M by removing all points that do not satisfy ψ_1.

Given this semantics, it is obvious that ELE_n^m and ELS_n^m are just extensions by definition of the logic EL_n^m and, if $\Gamma = \{a\}$, then both $E_\Gamma\psi$ and $S_\Gamma\psi$ are the formula $K_a\psi$[1]. What is more, every PAL_n^m formula can be rewritten into an equivalent EL_n^m formula using the rules below ([3]).

$$\begin{array}{lcr} [\psi]p & \leftrightarrow & \psi \to p, \\ [\psi](\phi_1 \wedge \phi_2) & \leftrightarrow & [\psi]\phi_1 \wedge [\psi]\phi_2 \\ [\psi]\neg\phi & \leftrightarrow & \psi \to \neg[\psi]\phi \\ [\psi]K_a\phi & \leftrightarrow & \psi \to K_a[\psi]\phi \\ [\psi_1][\psi_2]\phi & \leftrightarrow & [\psi_1 \wedge [\psi_1]\psi_2]\phi \end{array} \tag{1}$$

From now on, if (M, w) is a pointed model, where $M = \langle W, R, V\rangle$, we write $v \in M$ instead of $v \in W$; all models and all sets of pointed models are finite. We write $\mathbb{M} \models \varphi$ to mean that for all $(M, w) \in \mathbb{M}$, $(M, w) \models \varphi$. Note that if $\mathbb{M} = \emptyset$, then for every $ESPAL_n^m$ formula φ, it is trivially true that $\mathbb{M} \models \varphi$. We are going to use the well-known fact that if two pointed models (M, w) and (N, v) are bisimilar, then for every $ELES_n^m$ formula φ, $(M, w) \models \varphi$ if and only if $(N, v) \models \varphi$ (see [2]).

2.2 Formula-Size Games

Formula-Size games were first introduced in [1]. Our version is as follows.

Definition 5 (Formula-Size Game). *The rules of the one-person (called Spoiler) formula size game (FSG) are the following. The game is played on a tree, where each node is labeled with a pair $\langle \mathbb{M} \circ \mathbb{N}\rangle$ such that $\mathbb{M}$ and $\mathbb{N}$ are finite sets of finite pointed models. At each step, a node is labeled with one of the symbols from the set $\Sigma = \{\top, \bot, p, \neg, \vee, \wedge, M_i, K_i, E_\Gamma, S_\Gamma\}$ and either it is closed or at most two new nodes are added. Let a node $\langle \mathbb{M} \circ \mathbb{N}\rangle$ be given. Spoiler can make the following moves at this node:*

$\bot$-move *This can be played only if $\mathbb{M} = \emptyset$. When Spoiler plays this move, the leaf $\langle \mathbb{M} \circ \mathbb{N}\rangle$ is closed and labeled with the symbol $\bot$.*

$\top$-move *This can be played only if $\mathbb{N} = \emptyset$. When Spoiler plays this move, the node is closed and labeled with the symbol $\top$.*

atomic-move *Spoiler chooses a propositional variable $p \in P$ such that $\mathbb{M} \models p$, and $\mathbb{N} \models \neg p$. Then the node is closed and labeled with the symbol p.*

not-move *Spoiler labels the node with the symbol $\neg$ and adds one new node denoted $\langle \mathbb{N} \circ \mathbb{M}\rangle$ as a successor to the node $\langle \mathbb{M} \circ \mathbb{N}\rangle$.*

and-move *Spoiler labels the node with the symbol $\wedge$ and splits $\mathbb{N}$ in two sets $\mathbb{N} = \mathbb{N}_1 \cup \mathbb{N}_2$. Two new nodes are added to the tree as successors to $\langle \mathbb{M} \circ \mathbb{N}\rangle$, namely $\langle \mathbb{M} \circ \mathbb{N}_1\rangle$ and $\langle \mathbb{M} \circ \mathbb{N}_2\rangle$.*

or-move *Spoiler labels the node with the symbol $\vee$ and splits $\mathbb{M}$ in two sets $\mathbb{M} = \mathbb{M}_1 \cup \mathbb{M}_2$. Two new nodes are added to the tree as successors to the node $\langle \mathbb{M} \circ \mathbb{N}\rangle$, namely $\langle \mathbb{M}_1 \circ \mathbb{N}\rangle$ and $\langle \mathbb{M}_2 \circ \mathbb{N}\rangle$.*

[1] Hence, in what follows, we assume that the set Γ contains at least two indices.

K_i**-move** *Spoiler labels the node with the symbol K_i and for each pointed model $(N,v) \in \mathbb{N}$, he chooses a model (N,v') such that vR_iv'. All those choices are collected in $\mathbb{N}_1$. Then, for each pointed model $(M,w) \in \mathbb{M}$, Spoiler choses all the possible pointed models (M,w') such that wR_iw' and collects them in $\mathbb{M}_1$. If for some (M,w), the point w does not have an R_i successor, nothing is added to $\mathbb{M}_1$ for this model. The node $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$ is added as a successor of $\langle \mathbb{M} \circ \mathbb{N} \rangle$.*

M_i**-move** *Spoiler labels the node with the symbol M_i and for each pointed model $(M,w) \in \mathbb{M}$, he chooses a model (M,w') such that wR_iw'. All those choices are collected in $\mathbb{M}_1$.Then, for each pointed model $(N,v) \in \mathbb{N}$, Spoiler chooses all the possible pointed models (N,v') such that vR_iv' and collects them in $\mathbb{N}_1$. If for some (N,v), the point v does not have an R_i successor, nothing is added to $\mathbb{N}_1$ for this model. A new node $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$ is added as a successor of $\langle \mathbb{M} \circ \mathbb{N} \rangle$.*

E_Γ**-move** *Spoiler choses a subset $\Gamma \subseteq Ag$, then labels the node with the symbol E_Γ and for each pointed model $(N,w) \in \mathbb{N}$ he choses an agent $i \in \Gamma$ and a model (N,v) such that wR_iv. Let $\mathbb{N}_1$ be the set of all such models. For each model $(M,s) \in \mathbb{M}$ and each agent $j \in \Gamma$, Spoiler choses all the possible models (M,t) such that sR_jt. If for some (M,s), the point s does not have an R_j successor, no model is chosen for (M, s) and the agent j . Let $\mathbb{M}_1$ be the set of all such models. The node $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$ is added as a successor of the node $\langle \mathbb{M} \circ \mathbb{N} \rangle$.*

S_Γ**-move** *Spoiler choses a subset $\Gamma \subseteq Ag$, then labels the node with the symbol S_Γ and, for each pointed model $(N,w) \in \mathbb{N}$ and every agent $i \in \Gamma$, he choses a model (N,v) such that wR_iv. Let $\mathbb{N}_1$ be the set of all such models. For each model $(M,s) \in \mathbb{M}$, Spoiler choses an agent $j \in \Gamma$, and all the possible models (M,t) such that sR_jt. If for some (M,s), the point s does not have an R_j successor, no model is chosen for (M, s) and the agent j . Let $\mathbb{M}_1$ be the set of all such models. The node $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$ is added as a successor of the node $\langle \mathbb{M} \circ \mathbb{N} \rangle$.*

And-moves and or-moves are collectively called *splitting* moves. Splitting moves and the not-move are called Boolean moves. M_i-moves and K_i-moves are called *agent*-moves. In a node $\eta = \langle \mathbb{M} \circ \mathbb{N} \rangle$, we call $\mathbb{M}$ *the models on the left* and $\mathbb{N}$ *the models on the right*. Formula-size games in which no S_Γ and E_Γ moves are allowed are called *EL*-games. Games with no S_Γ moves allowed are called *ELE*-games and games in which no E_Γ moves are allowed are called *ELS*-games.

Definition 6 (FSG Winning Condition). *We say that Spoiler wins the FSG starting at $\langle \mathbb{M} \circ \mathbb{N} \rangle$ in n moves iff there is a game tree T with root $\langle \mathbb{M} \circ \mathbb{N} \rangle$ and precisely n nodes such that every leaf of T is closed.*

The next theorem connects the formula size games with the length of formulas of $ELES^m_n$. A proof can be found in [4].

Theorem 1. *Spoiler can win the FSG starting at $\langle \mathbb{M} \circ \mathbb{N} \rangle$ in n moves iff there is an $ELES^m_n$ formula ψ that is true in all the models in $\mathbb{M}$ and false in all the models in $\mathbb{N}$ and for which $|\psi| \leq n$.*

Example 1. Figure 1 shows a 4-round FSG starting at a node with two models on each side: $\langle (M_1,s_1),(M_2,s_2) \circ (M_3,s_3),(M_4,s_4) \rangle$. Only the atoms true at a given state are shown. There is only one agent. The big circles represent the nodes in the game tree.

Black circles designate closed leaves. The current points in the models are also black. Spoiler starts by playing an *and*-move. He exploits the fact that in one of the models on the right q is false in the current state, whereas in the other the current world has no successor. Spoiler can close the left branch by playing an atomic move q. In the right branch, Spoiler plays an M-move taking advantage of the fact that s_4 has no successor in M_4. Hence, there is no model on the right in the next node. Therefore, Spoiler plays a $\top$-move and wins the game. His strategy is *induced* by the formula $q \wedge M\top$.

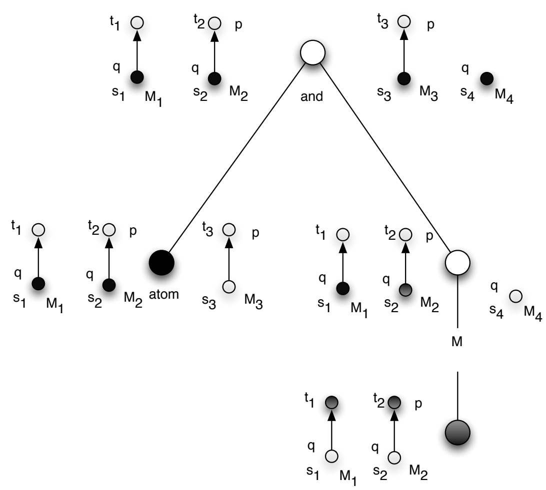


Fig. 1. The 4-round FSG starting in node $\langle\{(M_1, s_1), (M_2, s_2)\} \circ \{(M_3, s_3), (M_4, s_4)\}\rangle$.

Definition 7 (Isomorphism of branches). *A branch in a closed game tree is any path leading from the root of the tree to a closed leaf.*

1. *Let two closed game trees T_1 and T_2 be given and let the branch B_1, consisting of the nodes $n_0, n_1, ..., n_k$, and the branch B_2, consisting of the nodes $t_1, t_2, ...t_l$ (where the nodes in each branch have been numbered in increasing order starting from the root of the tree), belong to T_1 and T_2 respectively. We call B_1 and B_2 isomorphic, and write $B_1 \cong B_2$, iff $k = l$ and the symbols from the set $\Sigma = \{\top, \bot, p, \neg, \vee, \wedge, M_i, K_i\}$ labelling the nodes n_j, t_j $(0 \leq j \leq k)$ are the same.*
2. *Let $Ag(B)$ be the sequence of agent names obtained from B in such a way that they represent the agent moves along B, when traversing the branch from the root to the closed leaf. For instance, the branches B_1 (left) and B_2 (right) of the game tree of Figure 1 satisfy $Ag(B_1) = \epsilon^2$ and $Ag(B_2) = a$.*

[2] We use ϵ to denote the empty word.

Intuitively, isomorphism of branches means that the branches are equally long and look the same provided that we do not take into account the sets of pointed models labelling the respective nodes n_j, t_j $(0 \leq j \leq k)$. It is obvious that if a game tree T has two non - isomorphic branches B_1 and B_2, then T has at least two different branches. The word $Ag(B)$ encodes the agent-moves played along the branch B starting from the root.

Definition 8 (Isomorphism of game-trees). *Let $Br(T)$ be the set of branches of the game tree T. We call two closed game-trees T_1 and T_2 isomorphic, and write $T_1 \cong T_2$, iff there is a bijective function $f : Br(T_1) \rightarrow Br(T_2)$ such that for all branches $B \in Br(T_1)$, $f(B) \cong B$.*

The reader may think about this notion in the following way. If we have two closed game-trees that look identical provided that we do not take into account the sets of pointed models labelling each node, then these trees are isomorphic. It is easy to see that if $T_1 \cong T_2$, then T_1, T_2 are both parse trees of the same EL formula.

Lemma 1 (Properties of FSG). *For any FSG starting at a node $\langle \mathbb{M} \circ \mathbb{N} \rangle$.*

1. *If there are two bisimilar models $(M, w) \in \mathbb{M}$ and $(N, v) \in \mathbb{N}$, then Spoiler cannot win this game.*
2. *If $\mathbb{M} = \emptyset$ and $\mathbb{N} \neq \emptyset$, then Spoiler can win this game by playing according to any formula ϕ such that $\mathbb{N} \models \neg\phi$.*
3. *If $\mathbb{M} \neq \emptyset$ and $\mathbb{N} = \emptyset$, then Spoiler can win this game by playing according to any formula ϕ such that $\mathbb{M} \models \phi$.*
4. *If $\mathbb{M} = \mathbb{N} = \emptyset$, then Spoiler can win this game in any number of steps $n \geq 1$.*
5. *If Spoiler can win this game in n moves, then Spoiler can also win the FSG starting at $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$ in n moves, for all $\mathbb{M}_1 \subseteq \mathbb{M}$, $\mathbb{N}_1 \subseteq \mathbb{N}$; what is more, every closed game tree for the FSG starting at the node $\langle \mathbb{M} \circ \mathbb{N} \rangle$ is isomorphic to a closed game tree for the FSG starting at the node $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$.*
6. *Suppose that Spoiler can win this game in n moves. Let $\mathbb{M}_1 \subseteq \mathbb{M}$, $\mathbb{N}_1 \subseteq \mathbb{N}$ and let k be the smallest possible number of moves that Spoiler needs to win the FSG starting at $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$, then $k \leq n$.*
7. *Suppose that Spoiler can win this game. Let $\mathbb{M}_1, \mathbb{M}_2, \ldots, \mathbb{M}_k$ be subsets of $\mathbb{M}$ and $\mathbb{N}_1$, $\mathbb{N}_2, \ldots, \mathbb{N}_k$ be subsets of $\mathbb{N}$. If for every k winning games for Spoiler starting at $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$, $\langle \mathbb{M}_2 \circ \mathbb{N}_2 \rangle, \ldots, \langle \mathbb{M}_k \circ \mathbb{N}_k \rangle$ resulting in closed game trees T_1, $T_2, \ldots, T_k$ we can find branches B_1, $B_2, \ldots, B_k$ respectively such that $B_i \ncong B_j$ for all $i \neq j$, then every closed game tree for the FSG starting at $\langle \mathbb{M} \circ \mathbb{N} \rangle$ contains at least k different branches.*
8. *For any two branches B_1 and B_2, if $Ag(B_1) \neq Ag(B_2)$, then $B_1 \ncong B_2$.*

Proof.

1. Immediate from Theorem 1 (bisimilar models satisfy the same formulas [2]).
2. Let $\mathbb{N} \models \neg\psi$. Since $\mathbb{M} = \emptyset$, we have $\mathbb{M} \models \psi$. The result follows from Theorem 1.
3. Let $\mathbb{M} \models \psi$. Since $\mathbb{N} = \emptyset$, it is trivially true that $\mathbb{N} \models \neg\psi$. The result follows.
4. Let $\mathbb{M} = \mathbb{N} = \emptyset$. Then, for any formula ψ, $\mathbb{M} \models \psi$ and $\mathbb{N} \models \neg\psi$. Therefore, if $|\psi| = n$, then Spoiler can win the FSG starting at $\langle M \circ N \rangle$ in n moves. Obviously, for any $n \geq 1$, we can find a formula of length n, e.g., $\neg^{n-1}p$, i. e., $n-1$ occurrences of $\neg$ followed by the propositional symbol p.

5. Suppose that Spoiler can win the FSG starting at $\langle \mathbb{M} \circ \mathbb{N} \rangle$ in n moves. Then, there is a closed game tree T that is a parse tree of a formula ψ of length n such that $\mathbb{M} \models \psi$ and $\mathbb{N} \models \neg\psi$. Hence, Spoiler can win the FSG starting at $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$ in n moves by playing according to ψ. It is obvious that the resulting closed game tree T_1 with root $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$ is isomorphic to T.
6. Suppose that $k > n$. It follows immediately from 5 that Spoiler can win the FSG starting at $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$ in n moves. Therefore, k is not the minimal number of moves that Spoiler needs to win the FSG starting at $\langle \mathbb{M}_1 \circ \mathbb{N}_1 \rangle$.
7. Since Spoiler can win the *FSG* starting at $\langle \mathbb{M} \circ \mathbb{N} \rangle$, it follows from 5 that the resulting closed game tree T is isomorphic to a closed game tree T_i for each *FSG* starting at $\langle \mathbb{M}_i \circ \mathbb{N}_i \rangle$, $1 \leq i \leq k$. According to our assumption there are branches $B_1, \ldots, B_k$ in $T_1, \ldots, T_k$ respectively, such that $B_i \not\cong B_j$, $i \neq j$. Hence, there are branches $B'_1, \ldots, B'_k$ in T such that $B'_i \cong B_i$. Therefore, T has at least k different branches.
8. Immediate from the definition of $B_1 \cong B_2$.

2.3 Succinctness

We say that logic L_1 is at least as expressive as L_2 on the class of models $\mathbb{C}$, written $\mathrm{L}_2 \leq_{\mathbb{C}}^{\text{expr}} \mathrm{L}_1$, if for every $\varphi_2 \in \mathrm{L}_2$ there is a formula φ in L_1 such that $\mathbb{C} \models \varphi_1 \leftrightarrow \varphi_2$.

Following [6], our formal definition of the term succinctness is:

Definition 9 (Succinctness). *Let L_1, L_2 be two logics. Let $\mathbb{C}$ be a class of models such that $\mathrm{L}_1 \leq_{\mathbb{C}}^{expr} \mathrm{L}_2$. Let $\mathcal{F}$ be a class of functions $f : \mathbb{N} \to \mathbb{R}$. We say that L_1 is $\mathcal{F}$-succinct in L_2 on $\mathbb{C}$, and write $\mathrm{L}_1 \leq_{\mathbb{C}}^{\mathcal{F}} \mathrm{L}_2$, iff there is a function $f \in \mathcal{F}$ such that for every L_1-formula ϕ_1, there is an L_2-formula ϕ_2 for which the following is true:*

- $\mathbb{C} \models \phi_1 \leftrightarrow \phi_2$;
- $|\phi_2| \leq f(|\phi_1|)$.

Intuitively, this means that $\mathcal{F}$ gives an upper bound on the size of L_2 formulas needed to express all of L_1 on $\mathbb{C}$. It is worth mentioning that the function $f \in \mathcal{F}$ bounding the size of L_2 formulas need not be computable.

Using this definition, when we say that L_1 *is exponentially more succinct than* L_2 on $\mathbb{C}$, we mean $\mathrm{L}_1 \leq_{\mathbb{C}}^{\text{expr}} \mathrm{L}_2$ and $\mathrm{L}_1 \not\leq_{\mathbb{C}}^{SUBEXP} \mathrm{L}_2$, i.e., the length of formulas of L_2 expressing all of L_1 on $\mathbb{C}$ cannot be bounded from above by a sub-exponential function.

3 Main Results

In order to prove that for two logics L_1 and L_2 we have $\mathrm{L}_1 \not\leq_{\mathbb{C}}^{SUBEXP} \mathrm{L}_2$, it is enough to show that there are two infinite sequences of formulas $\phi_1, \phi_2, \ldots$ in L_1 and $\chi_1, \chi_2, \ldots$ in L_2, and rational numbers k and t such that

1. $|\phi_n| = kn + t$;
2. $|\chi_n| \geq 2^i$;
3. χ_n is the shortest formula in L_2 such that $\mathbb{C} \models \phi_n \leftrightarrow \chi_n$.

Definition 10. *Let $S = \{P, Ag\}$ be a signature where P contains at least one propositional symbol p and Ag contains at least two agent names a and b and let $\Gamma = \{a, b\}$. For every $n \geq 1$, the PAL^m_n formulas φ_n and θ_n, and the EL^m_n formulas ψ_n and χ_n are defined as follows.*

Table 1. Formulas

$\phi_1 := (M_a p \vee M_b p)$,	$\psi_1 := (M_a p \vee M_b p)$;
$\phi_n := \langle \phi_{n-1} \rangle (M_a p \vee M_b p)$	$\psi_n := \psi_{n-1} \wedge (M_a(p \wedge \psi_{n-1}) \vee M_b(p \wedge \psi_{n-1})), n > 1.$
$\theta_1 := (K_a p \vee K_b p)$,	$\chi_1 := (K_a p \vee K_b p)$;
$\theta_n := \langle \theta_{n-1} \rangle (K_a p \vee K_b p)$	$\chi_n := \chi_{n-1} \wedge (K_a(p \wedge \chi_{n-1}) \vee K_b(p \wedge \chi_{n-1})), n > 1.$

Using the rewriting rules for PAL^m_n formulas, it is easy to see that the formulas in the right column are equivalent to the formulas in the left column. It is obvious that the length of the latter formulas is linear in n while the length of the former is exponential in n. Firstly, we would like to prove that even if we extend EL^m_n with formulas of the form $S_\Gamma \psi$ and thus, obtain the logic ELS^m_n, there is no sequence of ELS^m_n formulas δ_n such that $\phi_n \leftrightarrow \delta_n$ and at the same time the length of δ_n is subexponential in n. To this end, for every $n \geq 1$, we must define a suitable class of models $\mathbb{M}^n$ such that $\mathbb{M}^n \models \phi_n$. We need to find another class of models $\mathbb{N}^n$ such that $\mathbb{N}^n \models \neg\phi_n$. Finally, we must prove that Spoiler cannot win any *ELS*-game starting at $\langle M^n \circ N^n \rangle$ in less than 2^n moves. Intuitively, it is clear that the main problem in *ELS*-games is the "power" of the S_Γ-move. Therefore, we can try and define the models $\mathbb{M}^n$ and $\mathbb{N}^n$ in such a way as to force Spoiler *not to play any S_Γ-moves* during the game. Secondly, we would like to prove that there is no ELE^m_n formula γ_n of subexponential length such that $\theta_n \leftrightarrow \gamma_n$. Again, the main problem here is the power of the E_Γ-move. Guided by the same intuition, for every $n \geq 1$, we define sets of models $\mathbb{O}^n$ and $\mathbb{P}^n$ such that $\mathbb{O}^n \models \theta_n$, $\mathbb{P}^n \models \neg\theta_n$, Spoler cannot win the *ELE*-game in less than 2^n moves, and the models in $\mathbb{O}^n$ and $\mathbb{P}^n$ are such that he is forced not to play any E_Γ-moves.

Definition 11 and items (b) and (c) from Lemma 2 are the formalization of this idea.

Definition 11 (Tree models). *Figures 2 and 3 show the sets of models $\mathbb{M}^n, \mathbb{N}^n, \mathbb{O}^n$, and $\mathbb{P}^n$. We start with the tree-like models in $\mathbb{O}^1$ and $\mathbb{P}^1$ and define recursively the models in $\mathbb{O}^{n+1}$ and $\mathbb{P}^{n+1}$ by taking a model from $\mathbb{O}^1$, and using the leaves of the tree as roots for the respective models from $\mathbb{O}^n$ and $\mathbb{P}^n$[3] as shown. Similar strategy is employed in the construction of the models in $\mathbb{M}^n$ and $\mathbb{N}^n$.*

Lemma 2. *Let the sequences of formulas ϕ_n, θ_n, ψ_n, and χ_n be defined as in Table 1 and let $\mathbb{M}^n$, $\mathbb{N}^n$, $\mathbb{O}^n$, and $\mathbb{P}^n$ be as in Definition 11. Then, for every n*

[3] Intuitively, the subscript in the name of the model O^{n+1}_{ak} encodes a path (starting with an a-step) leading from the root of the tree to a leaf satisfying the proposition p. The same path in the model P^{n+1}_{ak} leads to a leaf that does not satisfy p. Appart from this difference, the models O^{n+1}_{ak} and P^{n+1}_{ak} look the same.

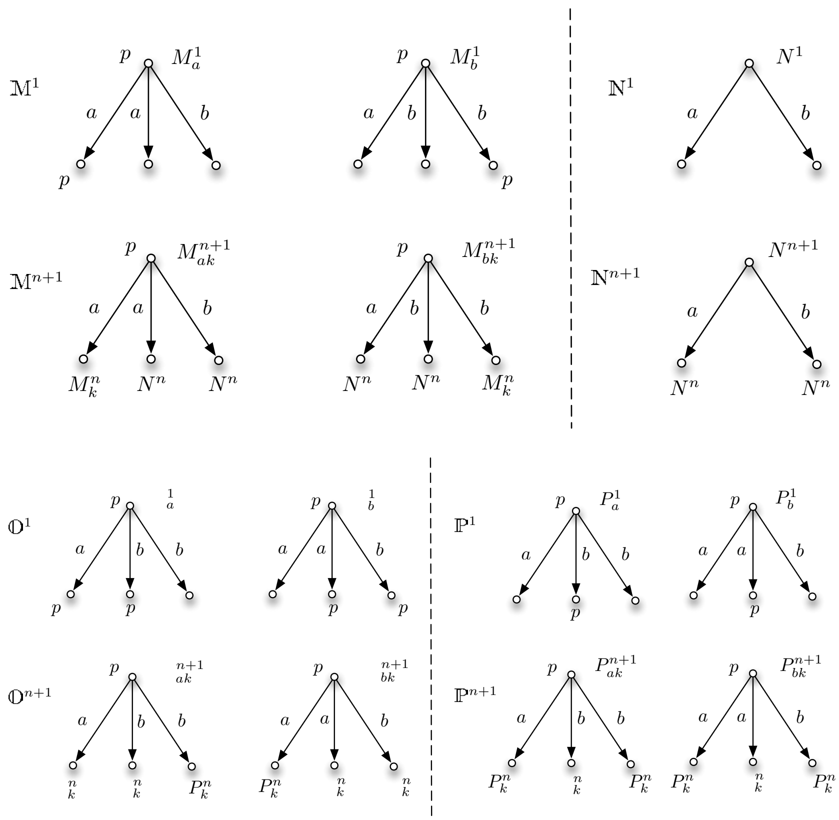


Fig. 2. The sets of models $\mathbb{M}^n$, $\mathbb{N}^n$, $\mathbb{O}^n$, and $\mathbb{P}^n$.

(a) $\mathbb{M}^n \models \phi_n$, $\mathbb{N}^n \models \neg\phi_n$, $\mathbb{O}^n \models \theta_n$, $\mathbb{P}^n \models \neg\theta_n$;

(b) *If Spoiler plays an $S_{\{a,b\}}$-move at a node of the form $\langle M_{ak}^{n+1} \circ N^{n+1}\rangle$ or $\langle N^{n+1} \circ M_{ak}^{n+1}\rangle$, he will lose the game. The same is true for nodes of the form $\langle M_{bk}^{n+1} \circ N^{n+1}\rangle$ or $\langle N^{n+1} \circ M_{bk}^{n+1}\rangle$, where $n \geq 0$ and k may be the empty word ϵ.*

(c) *If Spoiler plays an $E_{\{a,b\}}$-move at a node of the form $\langle O_{ak}^{n+1} \circ P_{ak}^{n+1}\rangle$ or $\langle P_{ak}^{n+1} \circ O_{ak}^{n+1}\rangle$, he will lose the game. The same is true for nodes of the form $\langle O_{bk}^{n+1} \circ P_{bk}^{n+1}\rangle$ or $\langle P_{bk}^{n+1} \circ O_{bk}^{n+1}\rangle$, where $n \geq 0$ and k may be the empty word ϵ.*

(d) *For $(n \geq 0)$, every closed game tree with root $\langle M_w^{n+1} \circ N^{n+1}\rangle$ has a branch B, where $Ag(B) = w$.*

(e) *For $(n \geq 0)$, every closed game tree with root $\langle O_w^{n+1} \circ P_w^{n+1}\rangle$ has a branch B, where $Ag(B) = w$.*

Proof.

(a) It is easily seen that $\mathbb{M}^n \models \psi_n$, $\mathbb{N}^n \models \neg\psi_n$, $\mathbb{O}^n \models \chi_n$, $\mathbb{P}^n \models \neg\chi_n$.

(b) Immediate from the definitions of the models. Playing an $S_{\{a,b\}}$ move at a node of this form will result in the occurrence of two bissimilar models, one on the left and the other on the right, in the successor node added to the game tree (see Figure 2).
(c) Similar to item (b) above.
(d) It is clear that there is at least one closed game tree with root $\langle M_w^{n+1} \circ N^{n+1} \rangle$. This follows from item (a) and Theorem 1. We prove the statement by induction on n. Consider the base case $n = 0$ and a game tree with root $\langle M_a^1 \circ N^1 \rangle$ (see Figure 2). The remaining cases can be proven in the same way. It is obvious that Spoiler cannot begin the game with an atomic, $\top$ or $\perp$ move. It follows from (b) that Spoiler cannot play an $S_{\{a,b\}}$-move, because he will lose the game. Playing an S_a or S_b-move is equivalent to playing a K_a or a K_b-move, respectively. Hence, w.l.o.g. we may assume that Spoiler does not play S_Γ-moves, where Γ is a singleton. Therefore, Spoiler can begin the game by playing either some *boolean*-moves, or an *agent*-move. Playing a *boolean*-move will result in adding at least one node of the form $\langle M_a^1 \circ N^1 \rangle$ or $\langle N^1 \circ M_a^1 \rangle$ to the game tree. The reasoning above shows that this node cannot be closed, hence Spoiler must play an *agent*-move in order to close the branch containing this node. It is obvious that this *agent*-move cannot be a K_b or M_b because this means that the successor node in the game tree will contain two bisimilar models - one on the left and the other on the right. Applying the first item from Lemma 1, we see that Spoiler will lose the game. Therefore, Spoiler must play a M_a or a K_a-move. Playing a K_a move at a node of the form $\langle M_a^1 \circ N^1 \rangle$, leads to loss for Spoiler. In the same way, playing an M_a-move at a node of the form $\langle N^1 \circ M_a^1 \rangle$ leads to loss. This means that in a node of the form $\langle M_a^1 \circ N^1 \rangle$, Spoiler must play an M_a-move and thus reaching the point that satisfies p in the model M_a^1 or a K_a-move at a node of the form $\langle N^1 \circ M_a^1 \rangle$ and again reaching the point that satisfies p in the model M_a^1. Hence, the statement is true for $n = 0$.
Assume now that the statement is true for n. We will prove it for $n+1$ when we have a closed game tree with root $\langle M_w^{n+1} \circ N^{n+1} \rangle$ such that $w = ak$. The remaining cases are similiar. Absolutely the same reasoning as above shows that there must be a branch in which the first *agent*-move is either an M_a-move leading to the point that is a root of a submodel M_k^n (see Figure 2) or a K_a-move again leading to the point that is a root of a submodel M_k^n. Such a move will result in reaching a submodel N^n in the model N^{n+1} and we can apply the inductive hypothesis.
(e) The proof is analogous to the proof of item d above.

Theorem 2. *Let $\mathbb{M}$ denote the union of all $\mathbb{M}_n$ and $\mathbb{N}_n$ models. Let $\mathbb{O}$ denote the union of all $\mathbb{O}_n$ and $\mathbb{P}_n$ models. Then*

(1) $PAL_n^m \not\leq_{\mathbb{M}}^{SUBEXP} ELS_n^m$.

(2) $PAL_n^m \not\leq_{\mathbb{O}}^{SUBEXP} ELE_n^m$.

Proof. We prove (2). The proof of (1) is analogous. We claim that for every ELE_n^m formula δ_n such that $\mathbb{O} \models \theta_n \leftrightarrow \delta_n$, $|\delta_n| \geq 2^n$. The general proof is best understood via an example. Let's consider the case $n = 2$. We know that $\mathbb{O}^2 = \{O_{aa}^2, O_{ab}^2, O_{ba}^2, O_{bb}^2\}$ and $\mathbb{P}^2 = \{P_{aa}^2, P_{ab}^2, P_{ba}^2, P_{bb}^2\}$. It follows from Lemma 2 (e) that every closed game tree with root $\langle O_{aa}^2 \circ P_{aa}^2 \rangle$ contains a branch B such that $Ag(B) = aa$; every closed game tree with root $\langle O^2_{ab} \circ P^2_{ab} \rangle$ contains a branch B_1 such that $Ag(B_1) = ab$, etc. Applying Lemma 1 items 7 and 8, we see that every closed game tree with root $\langle \mathbb{O}^2 \circ \mathbb{P}^2 \rangle$ must contain at least 2^2 different branches. This means that Spoiler must play at least $2^2 - 1$ splitting moves followed by at least 2^2 atomic moves since each branch must be closed. Had there been an ELE^m_n formula δ_2 such that $\mathbb{O} \models \langle \theta_1 \rangle (K_a p \vee K_b p) \leftrightarrow \delta_2$ and $|\delta_2| < 2^2$, Spoiler could have won the game starting at $\langle \mathbb{O}^2 \circ \mathbb{P}^2 \rangle$ in less than 2^2 moves[4] which is a contradiction.

4 Conclusion and Open Questions

As we pointed out in the introduction, one aspect of our result is unsatisfactory, namely the fact that the semantics of epistemic logics is usually given via models in which all relations are relations of equivalence, whereas the models we use do not have this property. Proving that PAL^m_n is exponentially more succinct than ELS^m_n or ELE^m_n on such a class of models presents technical difficulties which we intend to solve in a future paper. In addition, it would be nice to see whether at least one of ELE^m_n or ELS^m_n is exponentially more succinct than PAL^m_n. One possible way of attacking this problem is to define a suitable move in our formula-size games that corresponds to a formula of the form $[\phi]\psi$. This is not difficult to do but the resulting game is much more combinatorially involved than the games presented here. This suggests that proving such a result will be more difficult. Another seemingly difficult problem is whether PAL^m_n is exponentially more succinct than $ELES^m_n$.

In conclusion, we would like to stress two important points: Consider two logics L_1 and L_2

1. Even if L_1 and L_2 are not equally expressive they can be compared in terms of succinctness. As long as we have a set of properties P that are expressible in both L_1 and L_2, we can ask wether L_1 is more succinct than L_2 and by how much.
2. It is perfectly possible to have a situation where L_1 is exponentially more succinct than L_2 and vice versa even on the same class of models $\mathbb{C}$.

References

1. Adler, M., Immerman, N.: An n! lower bound on formula size. *ACM Transactions on Computational Logic*, (2003) vol 4(3):296–314.
2. Blackburn, P., de Rijke, M., Venema, Y. Modal logic. *Cambridge University Press*, (2001).
3. van Ditmarsch, H., van der Hoek, W., Kooi, B. Dynamic Epistemic Logic. *Springer*, (2007).
4. French, T., van der Hoek, W., Iliev, P., Kooi, B. Succinctness of epistemic languages. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, (2011): 881–886.
5. Gogic, G., Papadimitriou, C., Selman, B., Kautz, H. The comparative linguistics of knowledge representation. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, (1995): 862–869.

[4] This follows immediately from the fact that $\mathbb{O}^2 \models \langle \theta_1 \rangle (K_a p \vee K_b p) p, \mathbb{P}^2 \models \neg \langle \theta_1 \rangle (K_a p \vee K_b p)$ and Theorem 1.

6. Grohe, M., Schweikardt, N.: The succinctness of first-order logic on linear orders. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS)*, (2004) : 438–447.
7. Lutz, C.: Complexity and succinctness of public announcement logic. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, (2006) : 137 – 143.
8. Stockmeyer, L.: The Complexity of Decision Problems in Automata Theory. In *PhD thesis,* , (1999) : 110 – 121. Department of Electrical Engineering, MIT (1974)
9. Wilke, T.: *CTL*$^+$ is exponentially more succinct than *CTL* . In *Proceedings of Proceedings of the 19th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, (1999) : 110 – 121.

Ockhamist Propositional Dynamic Logic and its application to norm modelling: a preliminary study

Emiliano Lorini

IRIT-CNRS, Toulouse, France

Abstract. I present a new logic called Ockhamist Propositional Dynamic Logic OPDL and highlight its relationship with well-known logics in the field of multi-agent systems such as PDL and CTL^*. Furthermore, I show that OPDL provides a rich logical framework for representing different kinds of normative concepts such as conditional obligations and obligations with deadline.

1 Introduction

Different logical systems are traditionally used to model the properties of agents, multi-agent systems (MAS) and normative multi-agent systems (NorMAS). Among them we should mention Propositional Dynamic Logic PDL [8], Computational Tree Logic CTL [6], Full Computation Tree Logic CTL^* [13], Coalition Logic CL [12] and Alternating-time Temporal Logic ATL [1], STIT logic (the logic of "seeing to it that") by Belnap, Horty and coll. [10, 2], Dynamic Logic of Agency DLA [9]. Some relationships between these different logical systems have been studied. For instance, it is well-known that both CL and CTL are fragments of ATL [7]. Furthermore, it has been shown that the 'strategic' variant of STIT logic embeds ATL [4] and that DLA embeds both CL and STIT [9]. However at the current stage the general picture remains incomplete. For example, there are no clear relationships between the logic of programs PDL and Full Computational Tree Logic CTL^* (*e.g.*, it is not clear whether we can find a natural translation from PDL to CTL^*), or between PDL and logics of agency and capabilities such as ATL and STIT (*i.e.*, it is not clear whether PDL can be embedded in ATL or STIT or, viceversa, whether ATL or STIT can be embedded in PDL). Even more importantly, there is still no logical system that can be said to be more general than the others. For this reason a challenge arises of making the previous competing logical systems for modelling MAS converge into a single logical system. The aim of this work is to make a first step in this direction.

I propose an Ockhamist variant of Propositional Dynamic Logic PDL [8], called Ockhamist Propositional Dynamic Logic OPDL. Ockhamist semantics for temporal logic have been widely studied in the 80ies and in the 90ies (see, *e.g.*, [14, 15, 5]). The logic of agency STIT (the logic of "seeing to it that") by Belnap, Horty and coll. [2] is based on such semantics. According to the Ockhamist conception of time the truth of statements is evaluated with respect to a moment on a particular whole linear history through time. In this paper I present the general features of the OPDL syntax and semantics and discuss its relationship with PDL and CTL^* (Section 2). Moreover, I present an application of OPDL to norm modelling (Section 3). I show that OPDL provides a rich logical framework for representing different kinds of normative concepts such as obligations and permissions, conditional obligations and commitments, obligations and commitments with deadline.

There are plenty of interesting issues that are not investigated in the present paper and that are postponed to future work. My research agenda for the future includes: (1) a study of the mathematical properties of OPDL such as the decidability of its satisfiability problem as well as the search for a sound and complete axiomatization of OPDL, (2) a study of the relationship between OPDL and some well-known logics of agency and capabilities such as CL and the version of STIT logic with time [11] and, finally, (3) the proposal of an extension of OPDL which allows to capture the concept of strategic capability which is expressed in ATL and in 'strategic' STIT.

2 Ockhamist Propositional Dynamic Logic

The syntax and the semantics of OPDL are presented (Sections 2.1 and 2.2). As shown in Section 2.3, an interesting aspect of OPDL is its clear connection with PDL and CTL^*.

2.1 Syntax

Assume a countable set of atomic propositions $Prop = \{p, q, \ldots\}$ and a countable set of atomic programs (or atomic actions) $Atm = \{a, b, \ldots\}$. The language $\mathcal{L}$ of OPDL is defined by the following grammar in Backus-Naur Form (BNF):

$$\begin{array}{l} Prg: \ \pi ::= a \mid \equiv \mid \pi_1;\pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^* \mid ?\varphi \\ Fml: \varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid [\![\pi]\!]\varphi \end{array}$$

where p ranges over $Prop$ and a ranges over Atm. The other Boolean constructions $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined from p, $\neg$ and $\wedge$ in the standard way. The dual $\langle\!\langle\pi\rangle\!\rangle$ of the operator $[\![\pi]\!]$ is defined in the expected way as follows $\langle\!\langle\pi\rangle\!\rangle\varphi \stackrel{\mathtt{def}}{=} \neg[\![\pi]\!]\neg\varphi$.

Complex programs of sequential composition ($\pi_1;\pi_2$), non-deterministic choice ($\pi_1 \cup \pi_2$), iteration (π^*) and test ($?\varphi$) are built from atomic programs in Atm, from the special program $\equiv$ and from formulas in Fml. The special program $\equiv$ allows to move from a history to an alternative history passing through the same moment. The behavior of this program will become clearer in Section 2.2 when presenting the OPDL semantics.

The formula $[\![\pi]\!]\varphi$ has to be read "φ will be true at the end of all executions of program π" whereas $\langle\!\langle\pi\rangle\!\rangle\varphi$ has to be read "φ will be true at the end of some possible execution of program π". As it is assumed that atomic programs in Atm are linear (*i.e.*, all atomic programs in Atm occurring at a given state lead to the same successor state), $[\![a]\!]\varphi$ can also be read "if the atomic program a occurs, φ will be true afterwards". Indeed, from the assumption of linearity, it follows that atomic programs in Atm are deterministic (*i.e.*, there is at most one possible execution of an atomic program a at a given state). Finally, the formula $[\![\equiv]\!]\varphi$ has to be read "φ is true in all histories passing through the current moment" or, more shortly, "φ is true in the current moment".

2.2 Semantics

OPDL frames are structures with two dimensions: a vertical dimension corresponding to the concept of history, a horizontal dimension corresponding to the concept of moment.

Definition 1 (OPDL frame). *A OPDL frame is a tuple $F = \langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv} \rangle$ where:*

- *W is a set of states (or worlds),*
- *$\mathcal{S}$ is a successor state function $\mathcal{S} : W \longrightarrow W$*
- *$\mathcal{A}$ is a mapping $\mathcal{A} : T \longrightarrow 2^{Atm}$ from state transitions to sets of atomic programs,*
- *$\mathcal{R}_{\equiv}$ is an equivalence relation between states in W,*

and $T = \{(w, v) : w, v \in W$ and $\mathcal{S}(w) = v\}$ is the transition relation induced by the successor state function $\mathcal{S}$.

For every $w, v \in W$, $\mathcal{S}(w) = v$ means that v is the successor state of w.

If $\mathcal{A}(w, v) = \{a, b\}$, then the actions a and b are responsible for the transition from the state w to the state v. In other words, the function $\mathcal{A}$ labels every state transition with a set of atomic actions (*viz*. the actions that are responsible for the transition).

$\mathcal{R}_{\equiv}$-equivalence classes are called *moments*. If w and v belong to the same moment then they are called *alternatives*. A maximal sequence of states according to the transition relation T starting at a given state w is called *history starting in* w. If w and v belong to the same moment, then the history starting in w and the history starting in v are alternative histories (*viz*. histories starting at the same moment).

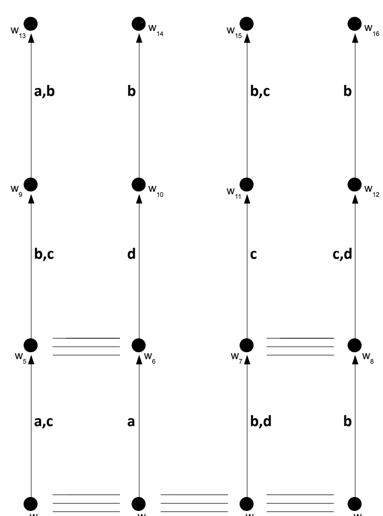


Fig. 1: A OPDL frame

Figure 1 is an example of OPDL frame. The $\mathcal{R}_{\equiv}$-equivalences classes $\{w_1, w_2, w_3, w_4\}$, $\{w_5, w_6\}$, $\{w_7, w_8\}$, $\{w_9\}$, $\{w_{10}\}$, $\{w_{11}\}$, $\{w_{12}\}$, $\{w_{13}\}$, $\{w_{14}\}$, $\{w_{15}\}$ and $\{w_{16}\}$ are the moments. The sequences of states (w_1, w_5, w_9, w_{13}), $(w_2, w_6, w_{10}, w_{14})$, $(w_3, w_7, w_{11}, w_{15})$ and $(w_4, w_8, w_{12}, w_{16})$ are the alternative histories starting at the same moment $\{w_1, w_2, w_3, w_4\}$. Actions a and c are responsible for the transition from the state w_1 to the state w_5, actions b and c are responsible for the transition from the state w_5 to the state w_9, and so on.

Definition 2 **(π-transitions).** *Given a OPDL frame $F = \langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv}\rangle$ and an atomic program $a \in Atm$, let*

$$\mathcal{R}_a = \{(w, v) : \mathcal{S}(w) = v \textit{ and } a \in \mathcal{A}(w, v)\}$$

be the set of a-transitions in the frame F. The binary relations for atomic programs are generalized to complex programs in Prg in the usual way as follows:

$$\begin{aligned}
\mathcal{R}_{\pi_1;\pi_2} &= \mathcal{R}_{\pi_1} \circ \mathcal{R}_{\pi_2} \\
\mathcal{R}_{\pi_1 \cup \pi_2} &= \mathcal{R}_{\pi_1} \cup \mathcal{R}_{\pi_2} \\
\mathcal{R}_{\pi^*} &= (\mathcal{R}_{\pi})^* \\
\mathcal{R}_{?\varphi} &= \{(w, w) : w \in W \textit{ and } M, w \models \varphi\}
\end{aligned}$$

A OPDL model is a OPDL frame supplemented with a valuation function mapping every state to the set of atomic propositions which are true in it. More precisely:

Definition 3 **(OPDL model).** *A OPDL model is a tuple $M = \langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv}, \mathcal{V}\rangle$ where:*

- *$\langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv}, \mathcal{V}\rangle$ is a OPDL frame and*
- *$\mathcal{V} : W \longrightarrow 2^{Prop}$.*

The truth of a OPDL formula is evaluated with respect to a world w in a OPDL model M.

Definition 4 **(Truth conditions).** *Given a OPDL model M, a world w and a formula φ, $M, w \models \varphi$ means that φ is true at world w in M. The rules defining the truth conditions of formulas are:*

- *$M, w \models p$ iff $p \in \mathcal{V}(w)$*
- *$M, w \models \neg\varphi$ iff not $M, w \models \varphi$*
- *$M, w \models \varphi \wedge \psi$ iff $M, w \models \varphi$ and $M, w \models \psi$*
- *$M, w \models [\![\pi]\!]\varphi$ iff $M, v \models \varphi$ for all v such that $(w, v) \in \mathcal{R}_\pi$*

2.3 Relationships between OPDL, PDL and CTL

Propositional Dynamic Logic PDL [8] is the well-known logic of programs. Again assume the countable set of atomic propositions $Prop = \{p, q, \ldots\}$ and the countable set of atomic programs $Atm = \{a, b, \ldots\}$. The language of PDL is defined by the following grammar in Backus-Naur Form (BNF):

$$\begin{aligned}
&Prg : \ \pi ::= a \mid \pi_1; \pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^* \mid ?\varphi \\
&Fml : \varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid [\pi]\varphi
\end{aligned}$$

where p ranges over $Prop$ and a ranges over Atm.

PDL models are tuples $M = \langle W, \{\mathcal{R}_a : a \in Atm\}, \mathcal{V}\rangle$ where:

- W is a nonempty set of possible worlds or states;
- $\{\mathcal{R}_a : a \in Atm\}$ is a set of binary relations on W;
- $\mathcal{V} : W \longrightarrow 2^{Prop}$ is a valuation function.

The accessibility relations for atomic programs are generalized to complex programs in the usual way.

PDL is completely axiomatized by the tautologies of propositional calculus and the following axioms and rules of inference:

(**K**$_{[\pi]}$)	$([\pi]\varphi \wedge [\pi](\varphi \to \psi)) \to [\pi]\psi$
(**Seq**)	$[\pi_1;\pi_2]\varphi \leftrightarrow [\pi_1][\pi_2]\varphi$
(**Choice**)	$[\pi_1 \cup \pi_2]\varphi \leftrightarrow [\pi_1]\varphi \wedge [\pi_2]\varphi$
(**Test**)	$[?\psi]\varphi \leftrightarrow (\psi \to \varphi)$
(**FixPoint**)	$[\pi^*]\varphi \leftrightarrow (\varphi \wedge [\pi][\pi^*]\varphi)$
(**Induction**)	$(\varphi \wedge [\pi^*](\varphi \to [\pi])) \to [\pi^*]\varphi$
(**Nec**$_{[\pi]}$)	$\dfrac{\varphi}{[\pi]\varphi}$

We can prove that OPDL is at least as expressive as PDL. Consider the following translation from the language of PDL to the OPDL language.

$$
\begin{aligned}
tr(p) &= p\\
tr(\neg\varphi) &= \neg tr(\varphi)\\
tr(\varphi_1 \wedge \varphi_2) &= tr(\varphi_1) \wedge tr(\varphi_2)\\
tr([\pi]\varphi) &= [\![tr(\pi)]\!]tr(\varphi)
\end{aligned}
$$

where

$$
\begin{aligned}
tr(a) &= \equiv; a\\
tr(\pi_1;\pi_2) &= tr(\pi_1); tr(\pi_2)\\
tr(\pi_1 \cup \pi_2) &= tr(\pi_1) \cup tr(\pi_2)\\
tr(\pi^*) &= (tr(\pi))^*\\
tr(?\varphi) &= ?tr(\varphi)
\end{aligned}
$$

As the following theorem shows, the preceding translation is a correct embedding.

Theorem 1. *Let φ be a PDL formula. φ is satisfiable in PDL iff $tr(\varphi)$ is OPDL satisfiable.*

Proof (Sketch). As to the right-to-left direction of the theorem, it is a routine task to prove that the translations of the axioms of PDL are OPDL valid and that the translated rules of inference of PDL preserve validity.

The proof of the left-to-right direction requires more work. Given a PDL model which satisfies φ, we have to build a OPDL model which satisfies $tr(\varphi)$. The construction goes as follows.

Suppose $M'' = \langle W'', \{\mathcal{R}''_a : a \in Atm\}, \mathcal{V}'' \rangle$ is a PDL model which satisfies φ. We can unravel M'' into a new PDL model $M' = \langle W', \{\mathcal{R}'_a : a \in Atm\}, \mathcal{V}' \rangle$ which satisfies the same formulas as M''. Therefore, there exists w_0 in M' such that $M', w_0 \models \varphi$.

Let $Z = \bigcup_{a \in Atm} \mathcal{R}'_a$ be the successor relation in M'. Z identifies the successors of a given world in M'. Let Z^* be the transitive closure of Z. Z^* is a strict partial order because M' satisfies the tree-like property. Let a *history* starting in w be a maximal Z^*-linearly ordered subset $\{w_1, \ldots, w_n, \ldots\} \subseteq W'$ with $w_1 = w$.[1] For any world $w \in W'$, let $H(w)$ be the set of all histories starting in w and let $H = \bigcup_{w \in W'} H(w)$ be the set of all histories. For notational convenience, elements of H are noted $h, h', \ldots$ We define the OPDL model $M = \langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv}, \mathcal{V} \rangle$ as follows:

- $W = \{w/h : w \in W' \text{ and } h \in H(w)\}$,
- for all $w/h, v/h' \in W$, $\mathcal{S}(w/h) = v/h'$ if and only if $(w, v) \in Z$ and $h = h' \cup \{w\}$,
- for all $w/h, v/h' \in W$ such that $\mathcal{S}(w/h) = v/h'$ and $a \in Atm$, $a \in \mathcal{A}(w/h, v/h')$ if and only if $(w, v) \in \mathcal{R}'_a$,
- for all $w/h \in W$ and $p \in Prop$, $p \in \mathcal{V}(w/h)$ if and only if $p \in \mathcal{V}'(w)$.

By induction on the structure of φ, it can be shown that $M, w_0/h \models tr(\varphi)$ for all $h \in H(w_0)$. □

In OPDL we can also reconstruct the basic operators of Full Computation Tree Logic CTL* [13]: the operators *next* and *until* of Linear Temporal Logic (LTL), and the modal operator quantifying over possible paths. In order to do this, it is necessary to add a specific constraint on OPDL frames, namely the assumption that the successor state function $\mathcal{S}$ in a OPDL frame is total. The two LTL operators can then be expressed as follows:

$$\mathsf{X}\varphi \stackrel{\text{def}}{=} [\![\bigcup_{a \in Atm} a]\!] \varphi$$

$$\varphi\, \mathcal{U}\, \psi \stackrel{\text{def}}{=} \langle\!\langle (?\varphi; \bigcup_{a \in Atm} a)^* \rangle\!\rangle \psi$$

where $\mathsf{X}\varphi$ and $\varphi\, \mathcal{U}\, \psi$ respectively mean that "φ will be true in the next state along the current history" and "ψ will be true at some point in the future along the current history and φ has to hold until ψ". The CTL* universal path-quantifier operator A is defined as follows:

$$\mathsf{A}\varphi \stackrel{\text{def}}{=} [\![\equiv]\!] \varphi$$

and its dual, the existential path-quantifier operator E, is defined as $\mathsf{E}\varphi \stackrel{\text{def}}{=} \neg \mathsf{A} \neg \varphi$. The preceding three operators can be used to express other kinds of temporal notions such as *eventually* $\mathsf{F}\psi \stackrel{\text{def}}{=} \top\, \mathcal{U}\, \psi$ (*i.e.*, ψ will be true at some point in the future along the

[1] Note that a history might be infinite.

current history), *henceforth* $\mathsf{G}\psi \stackrel{\mathtt{def}}{=} \neg\mathsf{F}\neg\psi$ (*i.e.*, ψ will be true in all future points along the current history), and *before* $\varphi\,\mathcal{B}\,\psi \stackrel{\mathtt{def}}{=} \neg(\neg\varphi\,\mathcal{U}\,\psi)$ (*i.e.*, φ will precede ψ along the current history).

3 Application to norm modeling

In order to model norms I need to enrich the definition of OPDL frame with the concept of illegal state transition.

Definition 5 (OPDL frame with illegal transitions). *A OPDL frame with illegal transitions is a tuple $F = \langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv}, I\rangle$ where:*

- *$\langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv}\rangle$ is a OPDL frame,*
- *$I \subseteq T$ is the set of illegal transitions,*

with $T = \{(w, v) : w, v \in W$ and $\mathcal{S}(w) = v\}$.

If $(w, v) \in I$, then the transition from the state w to the state v is illegal. Conversely, if $(w, v) \in T$ and $(w, v) \notin I$, then the transition from the state w to the state v is legal. It is assumed that the set I of illegal transitions satisfies the following constraint:

(IllHist) for every $w, v, u \in W$, if $(w, v) \in I$ and $(v, u) \in T$ then $(v, u) \in I$.

The preceding constraint (**IllHist**) ensures that the notion of *illegal state transition* can be generalized to the notion of *illegal history*. In fact, due to the constraint (**IllHist**), if the transition from the state w to its successor state v is an illegal transition, then the transition from the state v to its successor state v' is an illegal transition, the transition from the state v' to its successor state v'' is an illegal transition, and so on *ad infinitum*. In other words, saying that the transition from the state w to its successor state v is an illegal transition is the same as saying that, the history starting in w is an illegal history.

A OPDL model with illegal transitions is a OPDL frame with illegal transitions supplemented with a valuation function mapping every state to the set of atomic propositions which are true in it. More precisely:

Definition 6 (OPDL model with illegal transitions). *A OPDL model with illegal transitions is a tuple $M = \langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv}, I, \mathcal{V}\rangle$ where:*

- *$\langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv}, I\rangle$ is a OPDL frame with illegal transitions and*
- *$\mathcal{V} : W \longrightarrow 2^{Prop}$.*

Figures 2a and 2b are examples of OPDL models with illegal transitions and illegal histories. The illegal histories are the first and the third history (the thick red lines).

I extend the OPDL language with the special atomic formula Illeg whose reading is "the current transition is illegal" or alternatively, due to the constraint (**IllHist**), "the current history is illegal". This new construction is interpreted with respect to a given state w in a OPDL model $M = \langle W, \mathcal{S}, \mathcal{A}, \mathcal{R}_{\equiv}, I, \mathcal{V}\rangle$ with illegal transitions:

$$M, w \models \mathsf{Illeg} \text{ iff } (w, \mathcal{S}(w)) \in I$$

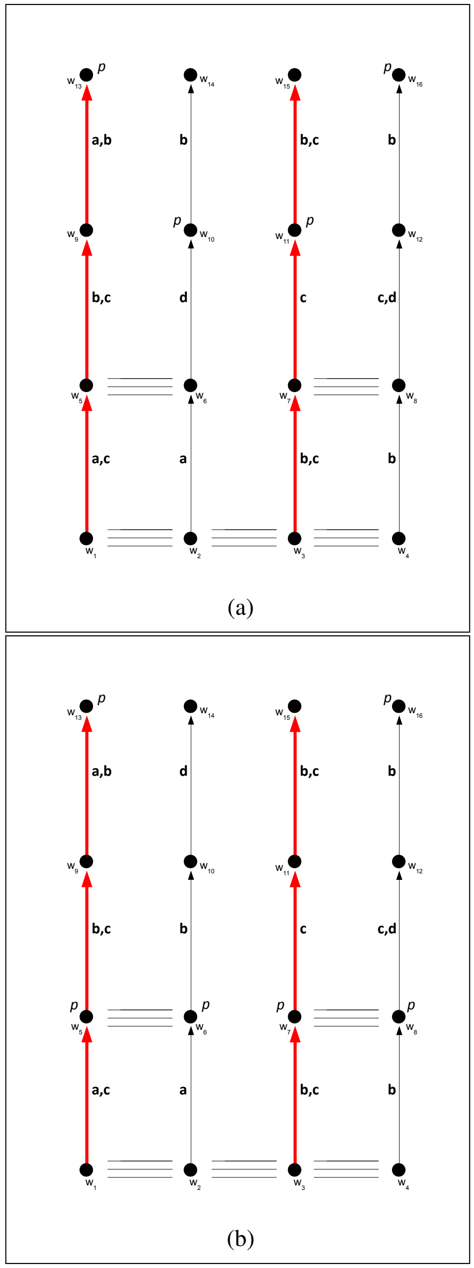


Fig. 2: Examples of OPDL models with illegal state transitions.

The extended OPDL language is rich enough to express interesting normative concepts. For instance, in OPDL we can define the concept of *obligation with deadline* $\mathsf{DObg}(a,\varphi)$, *i.e.*, the action a must be performed before the deadline φ:

$$\mathsf{DObg}(a,\varphi) \stackrel{\mathtt{def}}{=} [\![\equiv]\!](([\![a]\!]\bot\, \mathcal{U}\ \varphi) \rightarrow \mathsf{Illeg})$$

The formula $\mathsf{DObg}(a,\varphi)$ is true at a given state w of a OPDL model with illegal transitions if and only if, all histories which are alternatives to the history starting in w and in which the action a is not performed before φ becomes true are illegal histories. For example, in the model of Figure 2a the formula $\mathsf{DObg}(d,p)$ is true at the state w_1, because all histories which are alternatives to the history starting in w_1 and in which the action a is not performed before p are illegal histories.

OPDL also allows to define a concept of conditional obligation $\mathsf{CObg}(a,\varphi)$ which has to be read "it is obligatory that the action a is performed as soon as the condition φ becomes true":

$$\mathsf{CObg}(a,\varphi) \stackrel{\mathtt{def}}{=} [\![\equiv]\!]((\neg\varphi\, \mathcal{U}\ (\varphi \wedge [\![a]\!]\bot)) \rightarrow \mathsf{Illeg})$$

This kind of conditional obligation is typical of e-commerce scenarios. For example, in EBay it is obligatory that the seller sends the product as soon as he receives the payment. The formula $\mathsf{CObg}(a,\varphi)$ is true at a given state w of a OPDL model with illegal transitions if and only if, all histories which are alternatives to the history starting in w and in which the action a is not performed as soon as φ becomes true are illegal histories. For example, in the model of Figure 2b the formula $\mathsf{CObg}(d,p)$ is true at the state w_1 because at state w_1 it is the case that all histories which are alternatives to the history starting in w_1 and in which the action d is not performed as soon as p becomes true are illegal histories.

4 Conclusion

I have presented a new logic called Ockhamist Propositional Dynamic Logic OPDL and studied its relationship with PDL and CTL^*. I have presented an application of OPDL to norm modelling. As emphasized in the introduction, directions of future research are manifold. First of all, I plan to prove the decidability of the OPDL satisfiability problem using a technique similar to the one used by [3] for proving the decidability of Deterministic PDL. Secondly, I intend to refine the results given in Section 2.3 by providing a translation tr from CTL^* to OPDL and by proving that, for any CTL^* formula φ, φ is CTL^* satisfiable if and only if $tr(\varphi)$ is OPDL satisfiable. Indeed, the results of Section 2.3 only show that CTL^* operators can be reconstructed in OPDL. Finally, as emphasized in the introduction, the present paper is just a preliminary step towards a more comprehensive study of the relationship between OPDL and existing logics for multi-agent systems including Coalition Logic, STIT and ATL.

References

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

2. Belnap, N., Perloff, M., Xu, M.: Facing the future: agents and choices in our indeterminist world. Oxford University Press (2001)
3. Ben-Ari, M., Halpern, J.Y., Pnueli, A.: Deterministic propositional dynamic logic: finite models, complexity and completeness. Journal of Computer and System Sciences 25 (3), 402–417 (1982)
4. Broersen, J., Herzig, A., Troquard, N.: Embedding alternating-time temporal logic in strategic STIT logic of agency. Journal of Logic and Computation 16(5), 559–578 (2006)
5. Brown, M., Goranko, V.: An extended branching-time ockhamist temporal logic. Journal of Logic, Language and Information 8(2), 143–166 (1999)
6. Emerson, E.A.: Temporal and modal logic. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics. North-Holland Pub. Co./MIT Press (1990)
7. Goranko, V.: Coalition games and alternating temporal logics. In: Proceedings of the 8th conference on Theoretical aspects of rationality and knowledge (TARK'01). pp. 259–272. Morgan Kaufmann Publishers Inc. (2001)
8. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press (2000)
9. Herzig, A., Lorini, E.: A dynamic logic of agency I: STIT, abilities and powers. Journal of Logic, Language and Information 19(1), 89–121 (2010)
10. Horty, J.F.: Agency and Deontic Logic. Oxford Univ. Press, Oxford (2001)
11. Lorini, E.: A STIT logic analysis of commitment and its dynamics. Journal of Applied Logic To appear
12. Pauly, M.: A modal logic for coalitional power in games. Journal of Logic and Computation 12(1), 149–166 (2002)
13. Reynolds, M.: An axiomatization of full computation tree logic. Journal of Symbolic Logic 66(3), 1011–1057 (2001)
14. Thomason, R.: Combinations of tense and modality. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 2, p. 135165. Reidel, Dordrecht (1984), 2nd edition
15. Zanardo, A.: Branching-time logic with quantification over branches: The point of view of modal logic. Journal of Symbolic Logic 61(1), 143–166 (1996)

ATL with contexts: agency and explicit strategies

Nicolas Troquard[1] and Dirk Walther[2]

[1] Laboratory for Applied Ontology (ISTC-CNR), Trento, Italy
[2] Technical University of Madrid (UPM), Spain

Abstract. This paper reports on first results of our study of the recently proposed family of ATL_{sc} logics. We provide the intuitions for that it could serve as an (already well-studied) unifying framework for the verification of properties about time, ability, strategies and agency in societies of agents. We relate it with STIT logics of actual agency, and with ATL with explicit strategies. We establish that the problem of satisfiability checking for ATL_{sc} over general concurrent game structures (with possibly infinitely many moves and states) is undecidable.

1 Introduction

The first aim of this paper is to consider the recent extension of ATL with *strategy contexts* [6, 7, 11, 10] and reveal its relevance for the general discussion of strategies, ability and agency, and ground it in the topic of logics for multi-agent systems as well.[1] We explain how ATL_{sc} and ATL^*_{sc} can capture a variety of notions of *strategic actual agency* that lie beyond the mere ability of coalitions as captured by ATL.

The second goal is to contrast the use of strategy contexts with *explicit strategies*, to point out their similarities and differences in expressivity and flexibility. ATL_{sc} and ATLES capture both notions of commitment to and release of strategies. We relate the two logics and discuss how they capture these notions as well as notions like *irrevocable strategies*, *forgetting forever* and *recall* of strategies.

The third contribution is technical. The focus of ATL_{sc} has been on model checking so far, and not satisfiability. To make as few semantic assumptions as possible, we consider a generalisation of concurrent game structures with possibly infinitely many states and possibly infinitely many choices. We establish that, over such structures, the satisfiability problem for the logic ATL_{sc} is undecidable. In this more general setting, it is then not fit for *reasoning* about multi-agent systems. However, this is the price to pay as even apparently much simpler logics present the same drawback (e.g., Chellas' STIT logic of group agency [14]). Nevertheless, when interpreted over finite models, we identify a positive fragment

[1] For the moment, we leave aside the variants of ATL_{sc} designed to specify perfect-recall and bounded-memory strategies, although it is also a prominent aspect of societies of agents. We do not consider strategies in imperfect information either.

of ATL_{sc} that can be translated into ATLES, for which a decision procedure is known.

The paper is organized as follows. We present syntax and semantics of ATL_{sc} in Section 2. In Section 3, we explain how ATL_{sc} and ATL^*_{sc} are adequate for describing notions of actual agency and ATL-like ability. We try to illustrate the richness of the languages by proposing several variants and we point out a difficulty with how the ATL modality was defined in ATL_{sc}. We introduce ATLES on concurrent game structures in Section 4 and compare ATL_{sc} with ATLES. Moreover, we determine a fragment of ATL_{sc} that corresponds to ATLES. In Section 5, we show that ATL^*_{sc} is undecidable over general concurrent game structures (with possibly infinitely many states and choices).

2 ATL with strategy contexts

Any language in this paper is defined over a signature containing an infinite supply of ingredients. While it is typically accepted to have infinitely many propositions available to use in formulas, languages for multiple agents often assume the set of agents to be finite. Using a finite set of agents in the signature, instead of an infinite set, gives rise to a different language that, although of same cardinality, may lead to a different computational complexity for their respective reasoning problems. A finite bound on the number of agents limits the modelling capability of the language and, thus, restricts its generality. We fix $\mathbf{\Pi}$ and $\mathbf{\Sigma}$ to be countable infinite sets of, respectively, *atomic propositions* and *agents* (or *players*). All languages here are defined using $\mathbf{\Pi}$ and $\mathbf{\Sigma}$ as their signature.

The following grammar was given for ATL^*_{sc} in [11]. In this paper, however, we differ from the original definition in that we do not assume the set of agent symbols in the signature to be finite.

Definition 1 **(ATL^*_{sc} syntax).** *The following grammar defines state formulas φ and path formulas ψ, where p ranges over $\mathbf{\Pi}$ and A over finite subsets of $\mathbf{\Sigma}$. The language of ATL^*_{sc} consists of the state formulas.*

$$\begin{aligned}\varphi &::= \ p \ \mid\ \neg\varphi \ \mid\ \varphi\vee\varphi \ \mid\ \cdot\rangle A\langle\cdot\,\varphi \ \mid\ \langle\cdot A\cdot\rangle\psi \\ \psi &::= \ \varphi \ \mid\ \neg\psi \ \mid\ \psi\vee\psi \ \mid\ \bigcirc\varphi \ \mid\ \varphi\,\mathcal{U}\,\varphi\end{aligned}$$

The language is enumerable. To see this, verify that there are countably many coalitions, i.e. finite subsets of a countable set.

The remaining Boolean operators $\wedge$, $\rightarrow$ and $\leftrightarrow$ as well as the logical constants $\top$ and $\bot$ can be defined as usual in terms of the operators given. The linear temporal logic operators 'sometime' and 'forever' can be defined as path formulas $\Diamond\varphi = (\top\,\mathcal{U}\,\varphi)$ and $\Box\varphi = \neg(\top\,\mathcal{U}\,\neg\varphi)$. Informally, the formula $\langle\cdot A\cdot\rangle\psi$ states that A has a strategy to ensure the temporal property ψ. The modality $\langle\cdot A\cdot\rangle$ *commits* the members of A to their selected strategy, while the operator $\cdot\rangle A\langle\cdot$ *releases* this commitment.

The language of ATL_{sc} consists only of some formulas from ATL^*_{sc}. The syntax of the path formulas ψ is restricted as follows (where φ refers to the state formulas in Def. 1):

$$\psi ::= \neg\psi \ \mid\ \bigcirc\varphi \ \mid\ \varphi\,\mathcal{U}\,\varphi$$

Notice that $\Box\varphi$ is still definable in ATL_{sc} as this grammar allows for path formulas of the form $\neg(\varphi_1 \,\mathcal{U}\, \varphi_2)$. In contrast, the syntax of ATL [4] is restricted to not allow the application of negation to the next-time and until operator. It does not matter in the case of $\neg\bigcirc\varphi$, because the negation can be pushed inside the next-time operator yielding the equivalent ATL path formula $\bigcirc\neg\varphi$. But it does matter for path formulas of the form $\neg(\varphi_1 \,\mathcal{U}\, \varphi_2)$. Their absence in the ATL syntax is compensated by including the $\Box$ operator explicitly. However, the compensation is only partial, because the dual of until[2] cannot be expressed in ATL, cf. [17].

ATL has been defined using Alternating Transition Systems (ATSs) [2, 3] and Concurrent Game Structures (CGSs) [4]. It is readily seen that both types of structures can be used interchangeably for logics that do no address the names for moves in the object language. In the case of ATL, this has been shown in [13]. In terms of computational complexity of model checking, however, it makes a difference when we use ATSs or CGSs as was studied in [16, 17].

In this paper, we evaluate formulas on Concurrent Game Structures (CGSs), which are defined as follows.

Definition 2 (Concurrent Game Structure). *Let $\Sigma = \{1, \ldots, n\} \subset \mathbf{\Sigma}$, with $n \geq 1$, be a finite set of agents, and $\Pi \subset \mathbf{\Pi}$ be a finite set of atomic propositions. A* Concurrent Game Structure *(CGS) $\mathcal{C}$ for $\langle \Sigma, \Pi \rangle$ is a tuple $\mathcal{C} = \langle W, V, \Sigma, M, Mov, E \rangle$, where:*

- *W is a finite, non-empty set of worlds (or game positions);*
- *$V : W \to 2^{\Pi}$ is a valuation function;*
- *M is a finite, non-empty set of moves;*
- *$Mov : W \times \Sigma \to 2^M \setminus \emptyset$ specifies for every world w and agent a a set $Mov(w, a)$ of moves available to a at w;*
- *$E : W \times M^{\Sigma} \to W$ is a transition function mapping a world w and a move profile $\boldsymbol{m} = \langle m_1, \ldots, m_n \rangle$ (one move for each agent) to the world $E(w, \boldsymbol{m})$.*

Let $\mathcal{C}$ be a CGS. The component Mov determines which of the moves from M are available for an agent at a world w. Let $\mathsf{prof}(w)$ be the set of available move profiles at world w, i.e.,

$$\mathsf{prof}(w) = \{\langle m_1, \ldots, m_n \rangle \mid m_i \in Mov(w, i)\}.$$

A move profile is used to determine a successor of a world using the transition function E. Let $\mathsf{succ}(w)$ be the set of possible successors of w, formally

$$\mathsf{succ}(w) = \{E(w, \boldsymbol{m}) \mid \boldsymbol{m} \in \mathsf{prof}(w)\}.$$

An infinite sequence $\lambda = x_0 x_1 x_2 \cdots \in W^{\omega}$ of worlds is called a *play* or *computation* if $x_{i+1} \in \mathsf{succ}(x_i)$ for all positions $i \geq 0$. Denote with $\lambda[i]$ the i-th component x_i in λ, and with $\lambda[0, i]$ the initial sequence $x_0 \cdots x_i$ of λ.

[2] The dual of the temporal logic operator until $\mathcal{U}$ is called *release* $\mathcal{R}$, and it is defined as $(\varphi \mathcal{R} \psi) \stackrel{\text{def}}{=} (\neg\varphi \,\mathcal{U}\, \neg\psi)$. In LTL we have the equivalence $(\varphi \mathcal{R} \psi) \equiv \Box\psi \vee (\psi \,\mathcal{U}\, (\varphi \wedge \psi))$. In CTL, it is defined as $(\varphi \mathcal{R} \psi) \stackrel{\text{def}}{=} E\Box\psi \vee E(\psi \,\mathcal{U}\, (\varphi \wedge \psi))$.

A *strategy for an agent* $a \in \Sigma$ is a function f_a that maps a world w from W to a move profile $f_a(w) \in Mov(w, a)$ available to a at w. A *strategy for a coalition* $A \subseteq \Sigma$ is a set F_A of strategies with $F_A = \{\sigma_a \mid a \in A\}$ containing one strategy for every agent in A. We refer to a strategy also as *strategy context*. We denote with $\mathsf{strat}(A)$ the set of strategies available to coalition A. The strategies considered here are *memoryless* as they are functions from worlds to move profiles and, thus, do not take previously visited states into account.

We define two operations on strategies: upgrade and release of strategies. Let F_A and F be strategies for sets of agents, where F_A contains strategies for the agents in A. The *upgrade* of F with the strategies in F_A is the result of *overwriting* F with strategies for the agents in $A \cap \mathsf{dom}(F)$ and *supplementing* F with strategies for agents for which F does not already provide a strategy (i.e., for agents in $A \setminus \mathsf{dom}(F)$). We will use $\circ$ as a strategy upgrade operator. Formally,

$$F_A \circ F = F_A \cup \{f_a \in F \mid a \notin A\}.$$

The *release* of the strategies for the agents in B from F is the *restriction* of F to strategies for agents that do not occur in B (i.e., for agents in $\Sigma \setminus B$). Formally, for $C = \Sigma \setminus B$,

$$F|_C = \{f_a \in F \mid a \in C\}.$$

The set $\mathsf{out}(w, F_A)$ of *outcomes* of a strategy F_A for the agents in A starting at a world w is the set of all plays $\lambda = x_0 x_1 x_2 \cdots \in W^\omega$ such that $x_0 = w$ and, for every $i \geq 0$, there is a move profile $\boldsymbol{m} = \langle m_1, \ldots, m_n \rangle \in \mathsf{prof}(x_i)$ such that

(i) $m_a = f_a(x_i)$, for all $a \in A$, and
(ii) $x_{i+1} = E(x_i, \boldsymbol{m})$.

The semantics of ATL^*_{sc} over CGSs is given as follows, where state formulas are evaluated at worlds (or game positions) and path formulas over infinite paths in a CGS.

Definition 3 **(ATL^*_{sc} Semantics).** *Given a CGS $\mathcal{C} = \langle W, R, V, \Sigma, M, Mov, E \rangle$ for $\langle \Sigma, \Pi \rangle$ and a strategy context F, the consequence relation $\models$ is inductively defined as follows.*

- $\mathcal{C}, w \models_F p$ *iff* $p \in V(w)$*, for all atomic propositions* $p \in \Pi$*;*
- $\mathcal{C}, w \models_F \neg\varphi$ *iff* $\mathcal{C}, w \not\models_F \varphi$*;*
- $\mathcal{C}, w \models_F \varphi_1 \vee \varphi_2$ *iff* $\mathcal{C}, w \models_F \varphi_1$ *or* $\mathcal{C}, w \models_F \varphi_2$*;*
- $\mathcal{C}, w \models_F \,\rangle\!\cdot\! A \!\cdot\!\langle\, \varphi$ *iff* $\mathcal{C}, w \models_S \varphi$*, where* $S = F|_{\Sigma \setminus A}$*;*
- $\mathcal{C}, w \models_F \langle\!\cdot A \cdot\!\rangle \psi$ *iff there is* $F_A \in \mathsf{strat}(A)$ *such that for all plays* $\lambda \in \mathsf{out}(w, S)$*, it holds that* $\mathcal{C}, \lambda \models_S \psi$*, where* $S = F_A \circ F$*;*
- $\mathcal{C}, \lambda \models_F \varphi$ *iff* $\mathcal{C}, \lambda[0] \models_F \varphi$*, when* φ *is a state formula;*
- $\mathcal{C}, \lambda \models_F \neg\psi$ *iff* $\mathcal{C}, \lambda \not\models_F \psi$*;*
- $\mathcal{C}, \lambda \models_F \psi_1 \vee \psi_2$ *iff* $\mathcal{C}, \lambda \models_F \psi_1 \vee \psi_2$*;*
- $\mathcal{C}, \lambda \models_F \bigcirc\varphi$ *iff* $\mathcal{C}, \lambda[1] \models_F \varphi$*;*
- $\mathcal{C}, \lambda \models_F (\varphi_1 \,\mathcal{U}\, \varphi_2)$ *iff there is an* $i \geq 0$ *such that* $\mathcal{C}, \lambda[i] \models_F \varphi_2$ *and* $\mathcal{C}, \lambda[j] \models_F \varphi_1$ *for all* j *with* $0 \leq j < i$*.*

A formula φ *is* satisfiable *if* $\mathcal{C}, w \models_F \varphi$ *for some CGS* $\mathcal{C}$*, some world* w *in* $\mathcal{C}$ *and some strategy context* F *in* $\mathcal{C}$*; a formula is called* valid *if* $\mathcal{C}, w \models_F \varphi$ *for all* $\mathcal{C}$*, all* w *and all* F*.*

In this paper, we do not assume agents being capable of perfect recall. In fact, we use a semantics for ATL_{sc} and ATL^*_{sc} that is based on memoryless strategies. This means that agents use strategies that prescribe for every world which move to take. The history of previously visited worlds is not taken into account. This differs from the original definition in [11] that introduces the logics with a perfect recall semantics.

The language seems rather rich. Sometimes, different formulations of the same simple property will seem natural. We shall illustrate this in the next section by defining a few modalities that the community of logics in MAS has become familiar with.

3 Common modalities of agency

We now turn to the definition in the object language of ATL_{sc} and ATL^*_{sc} of a few modalities often discussed in the literature: $\mathsf{ATL}^{(*)}$ modality of ability (Section 3.2) and the modality of *strategic* actual agency (Section 3.3).[3] In order to express those modalities in the language of ATL^*_{sc}, it requires to write formulas referring explicilty to all agents. For this purpose we have to consider a fragment of the language defined in the previous section containing only a finite number of agents. We leave for future work a study of properties that can be expressed in the full language or variants of it.

3.1 Whatever A do

Brihaye *et al.* [6] define a modality that is going to be useful later:

$$[\cdot A\cdot]\psi \stackrel{\text{def}}{=} \neg\langle\cdot A\cdot\rangle\neg\langle\cdot\emptyset\cdot\rangle\psi.$$

They provide the reading: "for any strategy of coalition A, *every* run in the corresponding outcome satisfies a formula ψ". Notice that it is defined in the language of ATL^*_{sc} and not in the one of ATL_{sc}. Its semantics is:

– $\mathcal{C}, w \models_{F_B} [\cdot A\cdot]\psi$ iff $\forall S_A \in \mathsf{strat}(A), \forall\lambda \in \mathsf{out}(w, S_A \circ F_B)$. $\mathcal{C}, \lambda \models_{S_A \circ F_B} \psi$

The modality $[\cdot A\cdot]$ is *not* the dual of $\langle\cdot A\cdot\rangle$. It is also important to observe that the truth of the path formula ψ is in the context of $S_A \circ F_B$. A more precise reading of $[\cdot A\cdot]\psi$ is therefore: "for any strategy of coalition A, *every* run in the corresponding outcome satisfies a formula ψ *in the current strategy context updated by the new strategies of* A."

[3] 'Strategic' is not to be understood in the sense of game theory, where agents strategize to maximize their utility. It is to be opposed to actual agency that considers *only the current move*. Strategic actual agency is a property of agents or coalitions currently doing something by planning *more than one move ahead*.

3.2 Simulating the $\mathsf{ATL}^{(*)}$ path quantifier

Brihaye *et al.* [7, 11], propose to simulate the $\mathsf{ATL}^{(*)}$ formula $\langle\langle A\rangle\rangle\psi$ as follows:

$$\langle\langle A\rangle\rangle^1\psi \stackrel{\text{def}}{=} \cdot\rangle\Sigma\langle\cdot\langle\cdot A\cdot\rangle\psi.$$

That is, one first releases the current strategies of all agents, then we find a strategy for A that only yields runs that satisfy ψ. Its truth condition is:

- $\mathcal{C}, w \models_{F_B} \langle\langle A\rangle\rangle^1\psi$ iff $\exists S_A \in \mathsf{strat}(A), \forall\lambda \in \mathsf{out}(w, S_A).\ \mathcal{C}, \lambda \models_{S_A} \psi$

Notice that ψ must hold on each elected run, in the context of the current strategy of the members of A.

When the signature contains a finite set Σ of agents, $\langle\langle\Sigma\rangle\rangle$ and $\langle\langle\emptyset\rangle\rangle$ are dual: we have that $\langle\langle\Sigma\rangle\rangle\psi \leftrightarrow \neg\langle\langle\emptyset\rangle\rangle\neg\psi$ is a valid schema in ATL^*. Now, take the ATL^*_{sc} path formula $\Psi = \bigcirc[\cdot b\cdot]\bigcirc p$. We can see that $\langle\langle\Sigma\rangle\rangle^1\Psi \to \neg\langle\langle\emptyset\rangle\rangle^1\neg\Psi$ is not an ATL^*_{sc}-validity. It is falsified at the world w_0 of the model in Fig. 1.

We have that $\langle\langle\Sigma\rangle\rangle^1\bigcirc p \to \neg\langle\langle\emptyset\rangle\rangle^1\bigcirc\neg p$, with p a propositional variable from $\mathbf{\Pi}$ is indeed a valid formula in ATL^*_{sc}. But we have just seen that the uniform substitution of p with $[\cdot b\cdot]\bigcirc p$ yields a non-validity of ATL^*_{sc}. It means that ATL^*_{sc} does not obey the rule of uniform substitution.

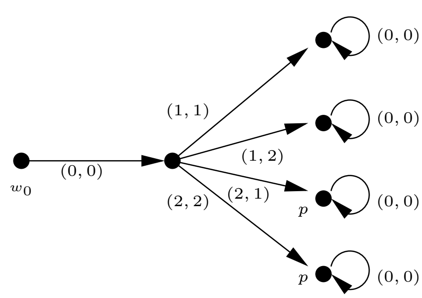


Fig. 1. A CGS for two agents.

A translation tr from the language of ATL^* into the language of ATL^*_{sc} such that $tr(\langle\langle A\rangle\rangle\psi) \stackrel{\text{def}}{=} \langle\langle A\rangle\rangle^1 tr(\psi)$ and homomorphic for the propositional connectives is indeed satisfiability preserving. But the definition does not interact well with the richer language of ATL^*_{sc}. A more fitting definition of the $\mathsf{ATL}^{(*)}$ modality would be:

$$\langle\langle A\rangle\rangle^2\psi \stackrel{\text{def}}{=} \cdot\rangle\Sigma\langle\cdot\langle\cdot A\cdot\rangle\cdot\rangle\Sigma\langle\cdot\psi.$$

That is, one first releases the current strategies of all agents, then we find a strategy for A, and one finally releases again all the current strategies to evaluate the path formula ψ. Its truth condition is:

- $\mathcal{C}, w \models_{F_B} \langle\langle A\rangle\rangle^2\psi$ iff $\exists S_A \in \mathsf{strat}(A), \forall\lambda \in \mathsf{out}(w, S_A).\ \mathcal{C}, \lambda \models_{\emptyset} \psi$

This seems more adequate with the notion of non-committed ability that we are familiar in $\mathsf{ATL}^{(*)}$. At least we regain duality in the sense that $\langle\langle\Sigma\rangle\rangle^2\bigcirc\varphi \leftrightarrow \neg\langle\langle\emptyset\rangle\rangle^2\bigcirc\neg\varphi$ is a valid axiom schema in ATL_{sc} and $\langle\langle\Sigma\rangle\rangle^2\psi \leftrightarrow \neg\langle\langle\emptyset\rangle\rangle^2\neg\psi$ is a valid axiom schema in ATL^*_{sc}.

In ATL^*_{sc}, there is at least one more way to capture the $\mathsf{ATL}^{(*)}$ path quantifier. It is sometimes interpreted as "coalition A has a strategy to enforce ψ whatever the choices of the other agents." This is actually the reading given in [7, p. 97].

It would then seem natural to express it as

$$\langle\langle A\rangle\rangle^{3}\psi \stackrel{\text{def}}{=} \langle\cdot A\cdot\rangle[\cdot\Sigma \setminus A\cdot]\psi.$$

(This definition does not fall into the language of ATL_{sc}, but of ATL^{*}_{sc}.) Its truth condition is:

- $\mathcal{C}, w \models_{F_B} \langle\langle A\rangle\rangle^{3}\psi$ iff $\exists S_A \in \mathsf{strat}(A), \forall S_{\overline{A}} \in \mathsf{strat}(\Sigma \setminus A)$, $\forall\lambda \in \mathsf{out}(w, S_{\overline{A}} \circ S_A)$. $\mathcal{C}, \lambda \models_{S_{\overline{A}}\circ S_A} \psi$

The path formula ψ is then evaluated with respect to a complete context of strategies, one for each member of the counter-coalition.

To conclude, we have now three sensible notions of ATL-like ability:

$\mathcal{C}, w \models_{F_B} \langle\cdot A\cdot\rangle[\cdot\Sigma \setminus A\cdot]\psi$	ψ eval. wrt. a Σ-commitment
$\mathcal{C}, w \models_{F_B} \rangle\Sigma\langle\cdot\langle\cdot A\cdot\rangle\psi$	ψ eval. wrt. an A-commitment
$\mathcal{C}, w \models_{F_B} \rangle\Sigma\langle\cdot\langle\cdot A\cdot\rangle\rangle\Sigma\langle\psi$	ψ eval. wrt. an $\emptyset$-commitment

The successive definitions involve an ever decreasing commitment for the evaluation of the path formula in its scope. Interestingly however, their sets of outcomes are identical, and correspond to these sets of runs that a coalition can enforce (in the sense of $\mathsf{ATL}^{(*)}$). They are distinct in ATL^{*}_{sc} because of the different commitments, but all are sufficient for an embedding of $\mathsf{ATL}^{(*)}$.

3.3 Strategic actual agency

The modality of actual agency has been widely studied, and is most prominently known for its treatment in the STIT theories (STIT for "seeing to it that"). It is a large family of logics with each their own semantics and modalities [5, 15]. Nonetheless, they all share an Ockhamist view of time [19]. Formulas are evaluated in tree models, with respect to a state and a play. The most basic modality is the Chellas' STIT operator (somewhat a misnomer) of actual (one-shot strategy) agency, proposed by Horty. Integrated in discrete time it allows to embed Coalition Logic ([9]).

A challenge in formal philosophy of action is to devise a modality similar to the Chellas' STIT but for long-term strategies. There is a truth-value gap of strategic statements, analogous to the truth-value gap for future-tense statements addressed, e.g., in [20] and [19]. In a nutshell, a state and a play are not enough to evaluate a statement reading that the coalition A strategically see to it that ψ is true. This is because in general, the context of only a state and a play does not determine a unique strategy of an arbitrary coalition. See [15, p. 149] for an illustration. (In a CGS, a play does determine a unique strategy for the coalition Σ, though.)

Horty observes that two lines of resolution are possible [15, Sec. 7.2]. The Peircean-like one is to consider all strategies that could determine the current play. The Ockhamist-like one, that we adopt here, is that a modality of strategic actual agency should additionally be evaluated wrt. to a strategy context. (This has been investigated by Müller [18] for the individual case and by Broersen *et al.* [8] for the case of coalitions.)

We can say here that a coalition A see to it that ψ in a context F_B iff the strategies of A in F_B are enough to make all the plays satisfy ψ. The truth value of such a modality would then be:

– $\mathcal{C}, w \models_{F_B} [A\ \mathsf{sstit}^1]\psi$ iff $\forall \lambda \in \mathsf{out}(w, F_B|_A)$. $\mathcal{C}, \lambda \models_{F_B|_A} \psi$

In fact, the modality $[A\ \mathsf{sstit}^1]$ can readily be captured in the language of ATL_{sc} as follows:

$$[A\ \mathsf{sstit}^1]\psi \overset{\text{def}}{=} \cdot\rangle \Sigma \setminus A\langle\cdot\langle\cdot\emptyset\cdot\rangle\psi.$$

The notion of strategic actual agency captured by $[A\ \mathsf{sstit}^1]$ is the one that appears the most immediate in the CGSs with contexts. It does capture perfectly that the current strategy of a coalition ensures that something happens, *independently of the commitment of the counter-coalition*, and *independently of the currently non-committed members of* A. We postpone for future research a thorough comparison, but this will turn out rather different from the solutions in the more traditional STIT literature, e.g., the proposal in [8]. A striking difference is that so far we did not feel compelled to explicitly put into the semantics of actual agency the fact that a coalition see to something *whatever the other agents do*. It might be a blunt conceptual error. But like in the simulation of the $\mathsf{ATL}^{(*)}$ path quantifier in Section 3.2, it might as well reveal interesting differences on the assumptions about agents' commitment to strategies between the two frameworks.

Trying to emulate *whatever the other agents do*, we can employ the modality $[\cdot A\cdot]$ defined in Section 3.1. A direct translation of "the coalition A see to it that ψ whatever the other agents do" would then be:

$$[A\ \mathsf{sstit}^2]\psi \overset{\text{def}}{=} [A\ \mathsf{sstit}^1][\cdot\Sigma \setminus A\cdot]\psi.$$

It is clearly equivalent to $[\cdot\Sigma \setminus A\cdot]\psi$. We would then have:

– $\mathcal{C}, w \models_{F_B} [A\ \mathsf{sstit}^2]\psi$ iff $\forall S_{\overline{A}} \in \mathsf{strat}(\overline{A}), \forall \lambda \in \mathsf{out}(w, S_{\overline{A}} \circ F_B)$. $\mathcal{C}, \lambda \models_{S_{\overline{A}} \circ F_B} \psi$

The modalities $[A\ \mathsf{sstit}^1]$ and $[A\ \mathsf{sstit}^2]$ are nevertheless very significantly different in that the evaluation of the path formula in the scope of $[A\ \mathsf{sstit}^2]$ is within the context of a commitment of the counter-coalition. (The evaluation of the path formula is still independent from the currently non-committed members of A.)

Variants of these modalities can be defined semantically, where instead of being *independent* of the strategies of the non-committed members of A, they reflect a type of actual agency that remains true *in whatever context of strategies for the non-committed members of* A.

– $\mathcal{C}, w \models_{F_B} [A\ \mathsf{sstit}^3]\psi$ iff $\mathcal{C}, w \models_{F_B} [A\ \mathsf{sstit}^1][\cdot A \setminus B\cdot]\psi$
– $\mathcal{C}, w \models_{F_B} [A\ \mathsf{sstit}^4]\psi$ iff $\mathcal{C}, w \models_{F_B} [A\ \mathsf{sstit}^2][\cdot A \setminus B\cdot]\psi$

Their truth condition is straightforward. However, note that there is no syntactic reference in $[A\ \mathsf{sstit}^3]$ and $[A\ \mathsf{sstit}^4]$ to the committed agents B. Hence, it is doubtful that they are expressible syntactically in the language of ATL_{sc} or ATL^*_{sc} as they require some sort of reification of who are the committed agents in the context.

To sum up, when giving an interpretation to a formula representing a statement about actual agency in the context of a long-term strategy, we are offered again more than one distinct possibility, depending on what commitments we wish to consider when evaluating the path formulas.

$\mathcal{C}, w \models_{F_B} \cdot\rangle \Sigma \setminus A\langle\cdot\langle\cdot\emptyset\cdot\rangle\psi$	ψ eval. wrt. a $(B \cap A)$-commitment
$\mathcal{C}, w \models_{F_B} [\cdot\Sigma \setminus A\cdot]\psi$	ψ eval. wrt. a $(B \cup (\Sigma \setminus A))$-commitment
$\mathcal{C}, w \models_{F_B} \cdot\rangle \Sigma \setminus A\langle\cdot\langle\cdot\emptyset\cdot\rangle[\cdot A \setminus B\cdot]\psi$	ψ eval. wrt. an A-commitment
$\mathcal{C}, w \models_{F_B} [\cdot\Sigma \setminus A\cdot][\cdot A \setminus B\cdot]\psi$	ψ eval. wrt. a Σ-commitment

Of course, we did not exhaust the seemingly sensible characterizations of a modality of strategic actual agency that can be directly expressed in the language of ATL^*_{sc}. One could also have the very simple variants where we release the commitment of all the agents when we evaluate the path formula. It is readily seen that for any $1 \leq i, j \leq 4$, we have that $[A\ \mathsf{sstit}^{\mathsf{i}}]\cdot\rangle\Sigma\langle\cdot\psi \leftrightarrow [A\ \mathsf{sstit}^{\mathsf{j}}]\cdot\rangle\Sigma\langle\cdot\psi$ is valid.

4 Strategy contexts and explicit strategies

We now turn to the second contribution of the paper. Here we contrast the notion of strategy contexts with explicit strategies. We first present ATLES, the extension of ATL with *explicit strategies* from [22] (Section 4.1), and then we translate a fragment of ATL_{sc} into ATLES (Section 4.2).

4.1 ATLES

The language of ATL is enriched with symbols for strategies and commitment functions that assign agents to strategies they are committed to play. Thus ATLES allows to reason explicitly about strategies. This is not possible with any of ATL and ATL_{sc} (and their respective LTL-extensions) as strategies are pure semantic constructs and they do not occur in the object language.

Formally, the signature of the language is extended by a set $\mathbf{\Upsilon}$ of *strategy terms*, where $\mathbf{\Upsilon} = \bigcup_{a\in\mathbf{\Sigma}} \mathbf{\Upsilon}_a$ and $\mathbf{\Upsilon}_a$ is a countable infinite set of *strategy terms* σ_a for agent a in $\mathbf{\Sigma}$. A *commitment function* is a partial function $\rho : \mathbf{\Sigma} \to \mathbf{\Upsilon}$ with a finite domain mapping an agent $a \in \mathbf{\Sigma}$ to a strategy term $\rho(a) \in \mathbf{\Upsilon}_a$ for a. Note that a commitment function ρ is a finite object and as such it is used to additionally parameterise path-quantifiers as $\langle\langle A\rangle\rangle_\rho$. The set $\mathsf{dom}(\rho)$ consists of the committed agents. If $\rho(a)$ is defined, then ρ contains a mapping of the form $a \mapsto \sigma_a$ which is called a *commitment* of agent a (or a *commits*) to play the strategy denoted by the strategy term σ_a. On the other hand, if $\rho(a)$ is undefined, then a does not commit to any strategy and, thus, a can quantify freely over the strategies available to a. The reading of an ATL-path quantifier $\langle\langle A\rangle\rangle$ with commitment function ρ is as follows:

> $\langle\langle A\rangle\rangle_\rho\varphi$ states that, given the commitment of any agent b in $\mathsf{dom}(\rho)$ to use the strategy denoted by $\rho(b)$, the agents in $A \setminus \mathsf{dom}(\rho)$ have a strategy to ensure the temporal property φ, no matter what the agents in $\Sigma \setminus (\mathsf{dom}(\rho) \cup A)$ do.

Notice that the committed agents in $\mathsf{dom}(\rho)$ do not take part in the quantification over strategies in $\langle\langle A\rangle\rangle_\rho$.

We remark that $\langle\langle A\rangle\rangle_\rho$ is not how the path quantifier really looks like when used in a formula. The symbol ρ is merely a meta-logical reference to an actual commitment function, which is a collection of mappings of the form $a \mapsto \sigma_a$, where σ_a is a strategy term for agent a. This should be considered when analysing the length of a formula. For instance, take $\rho = \{a \mapsto \sigma_a, b \mapsto \sigma_b\}$. Then we write $\langle\langle A\rangle\rangle_\rho$ for convenience in order to abbreviate $\langle\langle A\rangle\rangle_{\{a\mapsto\sigma_a, b\mapsto\sigma_b\}}$. For modelling purposes, one may modify the syntax and write $\langle\langle A : a \mapsto \sigma_a, b \mapsto \sigma_b\rangle\rangle$ instead.

The notion of commitment to strategies requires the same strategies to be played again later stage. This means, in formulas of the form $\langle\langle A\rangle\rangle_\rho \Psi$, the same commitment $a \mapsto \sigma_a$ from ρ occurs in a commitment function ξ of a nested path quantifier $\langle\langle B\rangle\rangle_\xi$ in Ψ. That is, both, ρ and ξ, prescribe the strategy term σ_a for agent a (or, in both cases, a commits to σ_a). We have that $\rho(a) = \xi(a)$. Release of commitment to σ_a is modelled as easy as committing to it in the first place. This is achieved by having a commitment function χ of a nested path quantifier not include the commitment $a \mapsto \sigma_a$, i.e., either $\chi(a) \neq \sigma_a$ or χ is undefined for a. In case release of commitment is not desired, the notion of irrevocable strategies is used. It can be modelled explicitly in ATLES by only allowing commitment functions ρ to extend conservatively the commitment functions ξ under whose range they occur, i.e., ρ and ξ agree for all agents in $\mathsf{dom}(\xi)$. Thus, IATL can be defined in ATLES while avoiding the update semantics employed in [1].

The language of ATLES is defined over the extended signature $\langle \mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{\Upsilon}\rangle$.

Definition 4 (ATLES Syntax). *The following grammar defines state formulas φ and path formulas ψ, where p ranges over $\mathbf{\Pi}$, A ranges over finite subsets of $\mathbf{\Sigma}$ and ρ over commitment functions. The language of ATLES consists of state formulas.*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle A\rangle\rangle_\rho \psi$$
$$\psi ::= \bigcirc\varphi \mid \Box\varphi \mid \varphi\,\mathcal{U}\,\varphi$$

The language of ATLES can easily be extended to allow for negation of the temporal operators next-time and until. But we refrain from extending the syntax in this paper as we use the established complexity result of the satisfiability problem for ATLES from [22] in order to use ATLES to determine a decidable fragment of ATL_{sc} whose satisfiability can be solved in ExpTime.

We define ATLES using concurrent game structures which differs from its original definition in [22]. The logic was introduced using alternating transition systems from [4] extended with strategy terms and a denotation function mapping the strategy terms to actual strategies. Another suitable extension of alternating transition systems was introduced in [21], so-called action-based alternating transition systems, which explicitly accounts for actions and action pre-conditions. It can readily be seen that these transition systems give rise to the same logics.

Strategy terms in $\mathbf{\Upsilon}$ are interpreted as strategies in a CGS via assignments. An *assignment* $\mathfrak{a}$ in $\mathcal{C}$ is a function mapping strategy terms σ_a in $\mathbf{\Upsilon}_a$ for any agent a in $\mathbf{\Sigma}$ to a strategy $\mathfrak{a}(\sigma_a)$ in $\mathsf{strat}(a)$ for a in $\mathcal{C}$. Note that the assignment $\mathfrak{a}$ in a CGS acts like an assignment in First-order Logic with the difference that in ATLES strategy terms are mapped to actual strategies in the CGS instead of domain elements as in FOL. In [22] an assignment is called denotation function, which comes as a component of an ATS.

To define the semantics of ATLES, we use the notions of a strategy and outcome as in Section 2. We lift the notion of assignment to commitment functions as follows. The application of an assignment $\mathfrak{a}$ to a commitment function ρ is the set $\mathfrak{a}(\rho)$ of strategies for the agents in $\mathsf{dom}(\rho)$. Formally,

$$\mathfrak{a}(\rho) = \{f_a \in \mathsf{strat}(a) \mid f_a = \mathfrak{a}(\rho(a)), a \in \mathsf{dom}(\rho)\}.$$

It is readily checked that $\mathfrak{a}(\rho)$ is indeed a set of strategies, one for each agent in $\mathsf{dom}(\rho)$. To see this, recall that ρ is functional, i.e., it yields exactly one strategy term $\rho(a)$ for every agent for which ρ is defined.

An assignment $\mathfrak{a}$ acts as an interpretation of the commitment function ρ (i.e. the strategy terms in ρ). We can view a strategy term $\sigma_a = \rho(a)$, for any a in $\mathsf{dom}(\rho)$, as a constant rather than a variable. As we will see below in the semantics of ATLES, the assignment $\mathfrak{a}$ does not change during the evaluation of a formula and, thus, the strategy $\mathfrak{a}(\sigma_a)$ is fixed. We can think of the strategy term σ_a as being existentially quantified in the sense that there exists a strategy for a that is referenced by σ_a and provided by $\mathfrak{a}$. ATLES does not provide references to universally quantified strategies.

Using the notion of assignments, we can now define how to interpret the formulas of ATLES over CGSs.

Definition 5 **(ATLES Semantics).** *Given a CGS $\mathcal{C} = \langle W, R, V, \Sigma, M, Mov, E\rangle$ for $\langle \Sigma, \Pi\rangle$ and an assignment $\mathfrak{a}$, the consequence relation $\models$ is inductively defined as follows. The notions of validity and satisfiability are defined as usual.*

- $\mathcal{C}, w \models^{\mathfrak{a}} p$ *iff* $w \in V(p)$, *for all atomic propositions* $p \in \Pi$*;*
- $\mathcal{C}, w \models^{\mathfrak{a}} \neg\varphi$ *iff* $\mathcal{C}, w \not\models^{\mathfrak{a}} \varphi$*;*
- $\mathcal{C}, w \models^{\mathfrak{a}} \varphi_1 \vee \varphi_2$ *iff* $\mathcal{C}, w \models^{\mathfrak{a}} \varphi_1$ *or* $\mathcal{C}, w \models^{\mathfrak{a}} \varphi_2$*;*
- $\mathcal{C}, w \models^{\mathfrak{a}} \langle\!\langle A\rangle\!\rangle_\rho \psi$ *iff there is a strategy* F_A *in* $\mathsf{strat}(A)$ *such that for all plays* $\lambda \in \mathsf{out}(w, S)$*, it holds that* $\mathcal{C}, \lambda \models^{\mathfrak{a}} \psi$*, where* $S = \mathfrak{a}(\rho) \circ F_A$*;*
- $\mathcal{C}, \lambda \models^{\mathfrak{a}} \bigcirc\varphi$ *iff* $\mathcal{C}, \lambda[1] \models^{\mathfrak{a}} \varphi$*;*
- $\mathcal{C}, \lambda \models^{\mathfrak{a}} \Box\varphi$ *iff* $\mathcal{C}, \lambda[i] \models^{\mathfrak{a}} \varphi$ *for all positions* $i \geq 0$*;*
- $\mathcal{C}, \lambda \models^{\mathfrak{a}} (\varphi_1 \mathcal{U} \varphi_2)$ *iff there is an* $i \geq 0$ *such that* $\mathcal{C}, a, \lambda[i] \models^{\mathfrak{a}} \varphi_2$ *and* $\mathcal{C}, \lambda[j] \models^{\mathfrak{a}} \varphi_1$ *for all positions* j *with* $0 \leq j < i$.

The ATLES semantics of $\langle\!\langle A\rangle\!\rangle_\rho$ is similar to the semantics of $\langle\!\cdot A\cdot\!\rangle$ in ATL_{sc}, which facilitates comparison. We recall that the operator $\circ$ from Section 2 yields $\mathfrak{a}(\rho) \circ F_A = \mathfrak{a}(\rho) \cup \{f_a \in F_A \mid a \notin \mathsf{dom}(\rho)\}$. Intuitively, $\mathfrak{a}(\rho) \circ F_A$ states that commitments of agents are respected as prescribed in ρ, all other agents in A play their just selected strategies.

4.2 Comparing ATL_{sc} and ATLES

Obvious differences between ATL_{sc} and ATLES are that, while the former includes a separate release modality $\cdot\rangle A\langle\cdot$ and strategy contexts in the semantics, the latter allows for commitments of the form $a \mapsto \sigma_a$ in the syntax which are interpreted using assignments. However, commitments and assignments play the role of strategy contexts in ATL_{sc}. A crucial difference between the logics is the semantics of the path quantifiers $\langle\cdot A\cdot\rangle$ and $\langle\langle A\rangle\rangle_\rho$; cf. Def. 3 and Def. 5, respectively. For $\langle\cdot A\cdot\rangle$, the strategies F_A selected by A upgrade (overwrite) the strategy context F_{context}, whereas, for $\langle\langle A\rangle\rangle_\rho$, the strategy commitments $\mathfrak{a}(\rho)$ are supplemented by F_A. Consequently, due to how context or commitments are respected in $\langle\cdot A\cdot\rangle$ and $\langle\langle A\rangle\rangle_\rho$, different agents end up quantifying over strategies in general. The following proposition states under which conditions $\langle\cdot A\cdot\rangle$ and $\langle\langle A\rangle\rangle_\rho$ determine the same set $\mathsf{out}(x, S)$ of plays, where S is defined as $S = F_A \circ F_{\mathsf{context}}$ in the former case, and $S = \mathfrak{a}(\rho) \circ F_A$ in the latter.

Proposition 1. *It holds that* $F_A \circ F_{\mathsf{context}} = \mathfrak{a}(\rho) \circ F_A$ *if one of the following conditions is satisfied:*

(i) $F_{\mathsf{context}} = \mathfrak{a}(\rho) = \emptyset$;
(ii) $F_A = \emptyset$ *and* $F_{\mathsf{context}} = \mathfrak{a}(\rho)$; *or*
(iii) $F_A = F_{\mathsf{context}} = \mathfrak{a}(\rho)$.

The proposition can be shown by using the fact that the strategy upgrade operator $\circ$ forms an idempotent semigroup on the set strat of strategies, and that $\circ$ is not commutative.[4]

Proposition 1 makes clear that a strategy context F_{context} in ATL_{sc} corresponds to the strategy commitment $\mathfrak{a}(\rho)$ in ATLES with the difference that F_{context} is a purely semantic object, whereas $\mathfrak{a}(\rho)$ consists of a syntactic component ρ and a semantic component $\mathfrak{a}$. This means we can explicitly describe strategy contexts in the language of ATLES, whereas in ATL_{sc} we have to make use of $\langle\cdot A\cdot\rangle$ and $\cdot\rangle A\langle\cdot$ that describe that strategies for A are either pushed into the context or released from it. Notice how using strategy commitments in the syntax is more flexible than the strategy context model as every path quantifier in ATLES can be parameterised with a different commitment function, which describes explicitly which agent is using what strategy. In particular, this does not require a dedicated release operator.

The notion of *irrevocable strategies* is captured in ATL_{sc} by carefully avoiding quantification over strategies of committed agents. In ATLES, irrevocability can be made explicit in the syntax.

Once a strategy in the strategy context is overwritten with a new strategy or released, it cannot be recovered in ATL_{sc}, because any reference to it is lost. This could be described with the notion of *forgetting forever*. Not so in ATLES, where 'forgetting forever' can be modelled explicitly in the language, but it is no restriction of the logic as in ATL_{sc}. In fact, an agent in ATLES may *resume* a commitment after releasing it, which also captures a notion of agents having a *strategy memory*.

A strength of ATL_{sc} is to push *any* strategy that is available to an agent into the context. This is achieved with formulas of the form $\neg\langle\cdot A\cdot\rangle\psi$, where the agents

[4] The operation $\circ$ is a binary function on strat, it is associative (i.e., $(F_A \circ F_B) \circ F_C = F_A \circ (F_B \circ F_C)$), the empty strategy $\emptyset$ forms the identity element (i.e., $F \circ \emptyset = \emptyset \circ F = F$), and $\circ$ is idempotent (i.e., $F \circ F = F$).

in A quantify universally over their strategies F_A. In the semantics, before we continue with the evaluation of the path formula ψ, the strategies F_A are used to upgrade the strategy context (cf. Def. 3). This is another crucial difference to ATLES, which is restricted to existential quantification over commitments. To make more precise the relationship between ATL_{sc} and ATLES, we present an equivalence preserving mapping from a fragment of ATL_{sc} into ATLES. We define a translation $tr(\cdot,\cdot)$ as a partial function that maps an ATL_{sc}-formula, in which every occurrence of a path quantifier $\langle\cdot A\cdot\rangle$ is under the scope of an even number of negations, and a commitment function to formulas of ATLES as follows:

$$
\begin{aligned}
tr(p,\xi) &\stackrel{\text{def}}{=} p;\\
tr(\neg\varphi,\xi) &\stackrel{\text{def}}{=} \neg tr(\varphi,\xi);\\
tr(\varphi_1 \vee \varphi_2,\xi) &\stackrel{\text{def}}{=} tr(\varphi_1,\xi) \vee tr(\varphi_2,\xi);\\
tr(\cdot\rangle A\langle\cdot\varphi,\xi) &\stackrel{\text{def}}{=} tr(\varphi,\chi),\ \text{where } \chi = \xi|_{\Sigma\setminus A};\\
tr(\neg\cdot\rangle A\langle\cdot\neg\varphi,\xi) &\stackrel{\text{def}}{=} tr(\varphi,\chi),\ \text{where } \chi = \xi|_{\Sigma\setminus A};\\
tr(\langle\cdot A\cdot\rangle\bigcirc\varphi,\xi) &\stackrel{\text{def}}{=} \langle\!\langle A\rangle\!\rangle_\rho \bigcirc tr(\varphi,\rho);\\
tr(\langle\cdot A\cdot\rangle\Box\varphi,\xi) &\stackrel{\text{def}}{=} \langle\!\langle A\rangle\!\rangle_\rho \Box\, tr(\varphi,\rho);\\
tr(\langle\cdot A\cdot\rangle(\varphi_1\,\mathcal{U}\,\varphi_2),\xi) &\stackrel{\text{def}}{=} \langle\!\langle A\rangle\!\rangle_\rho (tr(\varphi_1,\rho)\,\mathcal{U}\,tr(\varphi_2,\rho));
\end{aligned}
$$

where the commitment function ρ overwrites/updates ξ at A with fresh strategy terms. Formally,

$$\rho = \xi|_{\mathsf{dom}(\xi)\setminus A} \cup \{a \mapsto \sigma_a \mid a \in A, \sigma_a \text{ is fresh}\}.$$

The following proposition states that $tr(\cdot,\cdot)$ is indeed equivalence preserving. The proof works by induction on the structure of ATL_{sc}-formulas that are translated.

Proposition 2. *Let φ be an ATL_{sc}-formula, $\mathcal{C}$ a CGS, x a world in $\mathcal{C}$ and F a strategy in $\mathcal{C}$. The following are equivalent:*

(a) $\mathcal{C}, x \models_F \varphi$;
(b) $\mathcal{C}, x \models^{\mathfrak{a}} tr(\varphi, \rho_F)$, for some $\langle\rho_F, F\rangle$-compatible assignment $\mathfrak{a}$,

*where $\rho_F = \{a \mapsto \sigma_a \mid f_a \in F, \sigma_a$ is fresh$\}$ and an assignment $\mathfrak{a}$ is $\langle\rho_F, F\rangle$-*compatible *if $\mathfrak{a}(\rho_F(a)) = f_a$, for every $a \in \mathsf{dom}(\rho_F)$ and $f_a \in F$.*

The ATL_{sc}-fragment determined by $tr(\cdot,\cdot)$ is the fragment that does not allow for universal quantification over strategy commitments. The latter is expressed by formulas of the form $\neg\langle\cdot A\cdot\rangle\psi$ or, in general, by the modality $\langle\cdot A\cdot\rangle$ under the scope of an odd number of negations. The satisfiability checking problem for this fragment can be solved in ExpTime by Proposition 2 and the fact that ATLES is in ExpTime [22]. This is in contrast with the complexity of full ATL_{sc}, which we establish in the following section.

5 Complexity

This section is devoted to investigate the computational complexity of ATL_{sc} and ATL^*_{sc} over *general* CGSs: we relax CGSs from Def. 2 by allowing infinite number of states and infinite number of moves.

Generally, high expressiveness tends to come with the price of high computational complexity of reasoning problems. While the model checking problem was already considered in [11, 7] (and shown to be between 2ExpTime-hard and nonelementary for ATL_{sc}, while it is 2ExpTime-complete for ATL^* [4]), we focus here on the satisfiability problem. Clearly, the lower complexity bounds carry over to ATL_{sc} and ATL^*_{sc} from their respective fragments ATL and ATL^*. It turns out, however, that extending ATL with strategy contexts comes with a much higher price. In the following we show that ATL_{sc} is undecidable. To show this, we use a reduction of the satisfiability problem for the product logic $\mathsf{S5}^n$, which is known to be undecidable. In Section 3, we demonstrated that ATL_{sc} can capture some notion of actual group agency (cf. operator $[A\ \mathsf{sstit}^{\mathsf{1}}]$ in Section 3.3). Thus the undecidability of ATL_{sc} may not come as a surprise considering the undecidability of Chellas' STIT logic of group agency [14].

We obtain the following lower complexity bounds. It remains to be shown that Thm. 1 holds for finite CGSs (as defined in Def. 2), which amounts to showing that $\mathsf{S5}^n$ over finite frames is undecidable. We also leave the matching upper bounds as an open problem.

Theorem 1. *The satisfiability problem for* ATL_{sc} *(over general CGSs) is*

(i) NP-hard for formulas with $n = 1$ *agent;*
(ii) NExpTime-hard for formulas with $n = 2$ *agents; and*
(iii) undecidable for formuals with $n \geq 3$ *agents.*

The lower bounds in Theorem 1 can be shown by the following reduction of the satisfiability problem for $\mathsf{S5}^n$ to the problem for ATL_{sc}.[5] For a formal definition of $\mathsf{S5}^n$, we refer to, e.g., [12]. Define a translation $tr(\cdot)$ mapping $\mathsf{S5}^n$-formulas to formulas of ATL_{sc} as follows:

$$\begin{aligned} tr(p) &\stackrel{\text{def}}{=} \langle\!\cdot\emptyset\cdot\!\rangle \bigcirc p; \\ tr(\neg\varphi) &\stackrel{\text{def}}{=} \neg tr(\varphi); \\ tr(\varphi \vee \psi) &\stackrel{\text{def}}{=} tr(\varphi) \vee tr(\psi); \\ tr(\Diamond_i \varphi) &\stackrel{\text{def}}{=} \langle\!\cdot i\cdot\!\rangle (\bot\, \mathcal{U}\, tr(\varphi)). \end{aligned}$$

We can show the following lemma.

Lemma 1. *Let* φ *be an* $S5^n$*-formula and let* Σ_φ *be the set of agents that occur in* φ*. The following are equivalent:*

(i) φ *is satisfiable wrt.* $\models_{S5^n}$*;*
(ii) $\langle\!\cdot\Sigma_\varphi\cdot\!\rangle \Box tr(\varphi)$ *is satisfiable wrt.* $\models_{ATLsc}$.

Acknowledgements

We are indebted to the reviewers of LAMAS 2012 and the workshop "Modeling Strategic Reasoning", Lorentz Center, Leiden, The Netherlands, 20-24 February 2012. Nicolas is supported by a Marie Curie COFUND fellowship, and Dirk by a Juan de la Cierva fellowship of Spain and the project Agreement Technologies (Grant CONSOLIDER CSD2007-0022, INGENIO 2010), and the MICINN projects TIN2006-15455 and TIN2009-14562-CO5.

[5] Note that the lower bound of Theorem 1(i) already follows from propositional logic.

References

1. T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-time temporal logics with irrevocable strategies. In *Proceedings of TARK XI*, pages 15–24. Presses Universitaires de Louvain, 2007.
2. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of FOCS'97*, pages 100–109. IEEE Computer Society, 1997
3. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Lecture Notes in Computer Science*, 1536:23–60, 1998.
4. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.
5. N. Belnap, M. Perloff, and M. Xu. *Facing the future: agents and choices in our indeterminist world*. Oxford, 2001.
6. T. Brihaye, A. Da Costa, F. Laroussinie, and N. Markey. ATL with strategy contexts and bounded memory. Technical Report LSV-08-14, ENS Cachan, 2008.
7. T. Brihaye, A. Da Costa, F. Laroussinie, and N. Markey. ATL with strategy contexts and bounded memory. In *Proceedings of LFCS'09*, LNCS, vol. 5407, pages 92–106. Springer, 2009.
8. J. Broersen, A. Herzig, and N. Troquard. A STIT-extension of ATL. In *Proceedings of JELIA'06*, LNAI, vol. 4160, pages 69–81. Springer, 2006.
9. J. Broersen, A. Herzig, and N. Troquard. From Coalition Logic to STIT. In *Proceedings of LCMAS'05*, ENTCS, vol. 157:4, pages 23–35. Elsevier, 2006.
10. A. Da Costa. *Propriétés de jeux multi-agents*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, Sept. 2011.
11. A. Da Costa, F. Laroussinie, and N. Markey. ATL with strategy contexts: Expressiveness and model checking. In *Proceedings of FSTTCS'10*, LIPIsc, vol. 8, pages 120–132. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.
12. D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: theory and applications*. Studies in Logic, 148. Elsevier Science, 2003.
13. V. Goranko and W. Jamroga. Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2):241–280, 2004.
14. A. Herzig and F. Schwarzentruber. Properties of logics of individual and group agency. In *Proceedings of AiML*, pages 133–149. College Publications, 2008.
15. J. F. Horty. *Agency and Deontic Logic*. Oxford University Press, Oxford, 2001.
16. W. Jamroga and J. Dix. Do agents make model checking explode (computationally)? In *Proceedings of CEEMAS'05*, pages 398–407. Springer-Verlag, 2005.
17. F. Laroussinie, N. Markey, and G. Oreiby. On the expressiveness and complexity of ATL. *Logical Methods in Computer Science*, 4(2), 2008.
18. T. Müller. On the formal structure of continuous action. In *Proceedings of AiML 5*, pages 191–209. King's College Publications, 2005.
19. A. N. Prior. *Past, Present, and Future*. Clarendon Press, 1967.
20. R. H. Thomason. Indeterminist time and truth-value gaps. *Theoria*, 36:264–81, 1970.
21. W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. In *Proceedings of AAMAS'05*, pages 157–164. ACM Press, 2005.
22. D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time temporal logic with explicit strategies. In *Proceedings of TARK XI*, pages 269–278. Presses Universitaires de Louvain, 2007.