EPTCS 146

Proceedings of the

2nd International Workshop on

Strategic Reasoning

Grenoble, France, April 5-6, 2014

Edited by: Fabio Mogavero, Aniello Murano and Moshe Y. Vardi

Published: 1st April 2014
DOI: 10.4204/EPTCS.146
ISSN: 2075-2180
Open Publishing Association

Preface

This volume contains the proceedings of the Second International Workshop on Strategic Reasoning 2014 (SR 2014), held in Grenoble (France), April 5-6, 2014.

The SR workshop aims to bring together researchers, possibly with different backgrounds, working on various aspects of strategic reasoning in computer science, both from a theoretical and a practical point of view.

Strategic reasoning is one of the most active research area in multi-agent system domain. The literature in this field is extensive and provides a plethora of logics for modeling strategic ability. Theoretical results are now being used in many exciting domains, including software tools for information system security, robot teams with sophisticated adaptive strategies, efficient resource management for smart-city models, and automatic players capable of beating expert human adversary, just to cite a few. All these examples share the challenge of developing novel theories and tools for agent-based reasoning that takes into account the likely behavior of adversaries.

This year SR has hosted four invited talks:

- Simulation Games
  Thomas A. Henzinger (IST Austria)
- Two Themes in Modal Logic
  Wiebe van der Hoek (University of Liverpool)
- Model Checking Systems Against Epistemic Specifications
  Alessio R. Lomuscio (Imperial College London)
- What are "Good" Winning Strategies in Infinite Games?
  Wolfgang Thomas (RWTH Aachen)

Each submission to SR 2014 was evaluated by four reviewers for quality and relevance to the topics of the workshop. All submissions with positive scores were accepted, leading to 14 contributed talks at the workshop.

We would like to acknowledge the people and institutions, which contributed to the success of this edition of SR. We thank the organizers of the European Joint Conferences on Theory and Practice of Software (ETAPS 2014) for giving us the opportunity to host SR 2014. Many thanks go to all the Program Committee members and the additional reviewers for their excellent work, the fruitful discussions and the active participation during the reviewing process. We also thank Loredana Sorrentino for her work as member of the Organizing Committee. We would like to acknowledge the EasyChair organization for supporting all tasks related to the selection of contributions, and both EPTCS and arXiv for hosting the proceedings. We gratefully acknowledge the financial support to SR 2014 by ExCAPE - an NSF-funded Expeditions Project in Computer Augmented Program Engineering and by OR.C.HE.S.T.R.A. an Italian Ministry and EU research funded project on ORganization of Cultural HEritage for Smart Tourism and Real-time Accessibility. Finally, we acknowledge the patronage from the Department of Electrical Engineering and Information Technology of the Università degli Studi di Napoli Federico II.

Grenoble, April 2014
Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. i–ii, doi:10.4204/EPTCS.146.0

Table of Contents

Simulation Games

Invited Talk

Thomas A. Henzinger
IST Austria

Milner's simulation relation is a fundamental concept to compare the behaviors of two discrete dynamical systems. While originally defined for safety properties of state transition graphs, its game-theoretic formulation allows a natural generalization to liveness and quantitative properties. The resulting games are implication games on product graphs, i.e., the derived (simulation) objective takes the form of a logical implication between primary (system) objectives. We summarize the hardness of such implication games for important classes of system objectives: in some cases the implication game is no harder to solve than the corresponding primary game; in other cases the implication game is open even though we know how to solve the primary game.

This is joint work with Krishnendu Chatterjee and Jan Otop. It was supported in part by the European Research Council (ERC) and the Austrian Science Fund (FWF).

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. v–v, doi:10.4204/EPTCS.146.0.1

Two Themes in Modal Logic

Invited Talk

Wiebe van der Hoek
University of Liverpool

Modal logics form the basis for knowledge representation languages in AI, enabling us to reason about time, knowledge, beliefs, desires and obligations of agents. In my talk, I will address two contemporary research themes in this field.

A good old fashioned line of research in modal logic is that of "correspondence theory" which establishes a direct link between first order properties on Kripke models (basically, graphs) and modal sentences. Standard results have a typical global flavour: in terms of beliefs for instance, reflexive models guarantee that the agent's beliefs are correct, and inclusion of the doxastic relation of agent *a* in that of agent *b* guarantees that agent *a* believes whatever *b* believes. However, such results cannot cater for cases where we want to express that such properties only hold locally, as in "agent *a* believes his beliefs are correct, but this is not the case", or in "agent *a* believes anything agent *b* believes, but this will cease to hold as soon as *b* reads the letter". I will present a logic that can deal with such local cases.

The second theme concerns the question how we compare different logics. Standard ways to compare L_1 with L_2 address their expressivity, or the computational complexity of reasoning problems one can perform in each. In many cases, two logics are comparable on both measures. Only recently the field of knowledge representation has started to address the issue of succinctness: how economically can one express properties in each logic? I give a working definition of what it means that L_1 is exponentially more succinct than L_2, and then I present a tool which can be used to prove succinctness results, the so-called Formula Size Games. Such games are played on two sets of models, and it establishes a relation between the number of moves needed to win the game, and the length of a formula that discriminates between the sets. I will present some examples of succinctness results.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. vii–vii, doi:10.4204/EPTCS.146.0.2

Model Checking Systems Against Epistemic Specifications

Invited Talk *

Alessio R. Lomuscio
Imperial College London

Twenty years after the publication of the influential article ”Model checking vs theorem proving: a manifesto” by Halpern and Vardi, the area of model checking systems against agent-based specifications is flourishing.

In this talk I will present some of the approaches I have developed with collaborators. I will begin by discussing BDD-based model checking for epistemic logic combined with ATL operators and then move to abstraction techniques including symmetry reduction. I will then highlight how, in our experience, bounded model checking can also successfully be used in this context, particularly in combination with BDDs, and how synthesis problems can be formulated and solved in an epistemic setting.

The talk will include examples in the context of security protocols and a brief demo of MCMAS, an open-source model checker implementing some of these techniques.

*This talk was meant to feature in the SR 2013 programme but could not be given due to ill health of the speaker.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. ix–ix, doi:10.4204/EPTCS.146.0.3

What are "Good" Winning Strategies in Infinite Games?

Invited Talk

Wolfgang Thomas*
RWTH Aachen

Infinite games were invented in descriptive set theory, where the dominating question was determinacy - the mere existence of a winning strategy for one of the two players. In computer science the problem was put into an algorithmic setting: Can one decide who wins and can one effectively construct a winning strategy? In this talk we address quantitative refinements of the problem, reflecting a major current trend of research: How to construct winning strategies that are "good" or even "optimal" in some sense? The size of memory of finite-state machines executing winning strategies is a well-known criterion. Other criteria refer to the "efficient behavior" of strategies, as captured by the application of the solution of mean-payoff games. A third approach aims at novel formats of winning strategies, e.g. as "programs" (rather than state-machines). We survey old and recent work on these topics, spanning the literature from the beginnings (Büchi-Landweber 1969) to recent results obtained in the Aachen research group, among them the study of winning strategies as Boolean programs (Brütsch 2013) and the Turing machine based model of "strategy machine" (Gelderie 2014).

*Research supported by the project Cassting (Collective Adaptative Systems Synthesis With Non-Zero-Sum Games) funded as part of the FoCAS collaborative action by the European Commission under FP7.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. xi–xi, doi:10.4204/EPTCS.146.0.4

Expectations or Guarantees? I Want It All! A crossroad between games and MDPs*

Véronique Bruyère
Université de Mons
Belgium

Emmanuel Filiot
Université Libre de Bruxelles
Belgium

Mickael Randour
Université de Mons
Belgium

Jean-François Raskin
Université Libre de Bruxelles
Belgium

When reasoning about the strategic capabilities of an agent, it is important to consider the nature of its adversaries. In the particular context of controller synthesis for quantitative specifications, the usual problem is to devise a strategy for a reactive system which yields some desired performance, taking into account the possible impact of the environment of the system. There are at least two ways to look at this environment. In the classical analysis of two-player quantitative games, the environment is purely antagonistic and the problem is to provide strict performance guarantees. In Markov decision processes, the environment is seen as purely stochastic: the aim is then to optimize the expected payoff, with no guarantee on individual outcomes.

In this expository work, we report on recent results [10, 9] introducing the beyond worst-case synthesis problem, which is to construct strategies that guarantee some quantitative requirement in the worst-case while providing an higher expected value against a particular stochastic model of the environment given as input. This problem is relevant to produce system controllers that provide nice expected performance in the everyday situation while ensuring a strict (but relaxed) performance threshold even in the event of very bad (while unlikely) circumstances. It has been studied for both the mean-payoff and the shortest path quantitative measures.

1 Introduction

Classical models. Two-player zero-sum quantitative games [17, 31, 8] and Markov decision processes (MDPs) [27, 11] are two popular formalisms for modeling decision making in adversarial and uncertain environments respectively. In the former, two players compete with opposite goals (zero-sum), and we want strategies for player 1 (the system) that ensure a given *minimal performance against all possible strategies* of player 2 (its environment). In the latter, the system plays against a stochastic model of its environment, and we want strategies that ensure a *good expected overall performance*. Those two models are well studied and simple optimal memoryless strategies exist for classical objectives such as mean-payoff [25, 17, 18] or shortest path [4, 2]. But both models have clear weaknesses: strategies that are good for the worst-case may exhibit suboptimal behaviors in probable situations while strategies that are good for the expectation may be terrible in some unlikely but possible situations.

What if we want both? In practice, we want strategies that both ensure (a) some worst-case threshold no matter how the adversary behaves (i.e., against any arbitrary strategy) and (b) a good expectation against the expected behavior of the adversary (given as a stochastic model). We study how to construct such

*Work partially supported by European project CASSTING (FP7-ICT-601148). Filiot and Randour are respectively F.R.S.-FNRS research associate and research fellow. Raskin is supported by ERC Starting Grant inVEST (279499).

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 1–8, doi:10.4204/EPTCS.146.1

finite-memory strategies. We consider finite memory for player 1 as it can be implemented in practice (as opposed to infinite memory). Player 2 is not restricted in his choice of strategies, but we show that simple strategies suffice. Our problem, the **beyond worst-case synthesis problem**, makes sense for any quantitative measure. We focus on two classical ones: the *mean-payoff*, and the *shortest path*. Our results are summarized in Table 1.

		worst-case	expected value	**BWC**
mean-payoff	complexity	NP ∩ coNP	P	**NP ∩ coNP**
	memory	memoryless		**pseudo-poly.**
shortest path	complexity	P		**pseudo-poly. / NP-hard**
	memory	memoryless		**pseudo-poly.**

Table 1: Overview of decision problem complexities and memory requirements for winning strategies of the first player in games (worst-case), MDPs (expected value) and the BWC setting (combination).

Example. Consider the weighted game in Fig. 1 to illustrate the *shortest path* context. Circle states belong to player 1, square states to player 2, integer labels are durations in minutes, and fractions are probabilities that model the expected behavior of player 2. Player 1 wants a strategy to go from "home" to "work" such that "work" is *guaranteed* to be reached within 60 minutes (to avoid missing an important meeting), and player 1 would also like to minimize the expected time to reach "work".

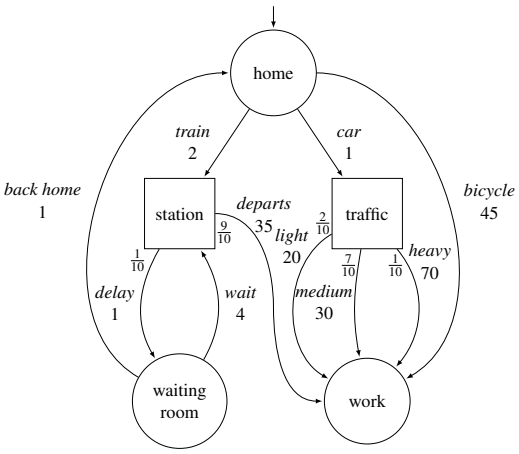


Figure 1: Player 1 wants to minimize its expected time to reach "work", but while ensuring it is less than an hour in all cases.

The strategy that minimizes the expectation is to take the car (expectation is 33 minutes) but it is excluded as there is a possibility to arrive after 60 minutes (in case of heavy traffic). Bicycle is safe but the expectation of this solution is 45 minutes. We can do better with the following strategy: try to take the train, if the train is delayed three time consecutively, then go back home and take the bicycle. This strategy is safe as it always reaches "work" within 59 minutes and its expectation is $\approx 37{,}56$ minutes (so better than taking directly the bicycle). Observe that this simple example already shows that, unlike the situation for classical games and MDPs, strategies using memory are strictly more powerful than memoryless ones. Our algorithms are able to decide the existence of (and synthesize) such finite-memory strategies.

Related work. This paper gives an expository presentation of results appeared in [10] (an extended version of the paper can be found in [9]).

Our problems generalize the corresponding problems for two-player zero-sum games and MDPs. In mean-payoff games, optimal memoryless worst-case strategies exist and the best known algorithm is in NP ∩ coNP [17, 31, 8]. For shortest path games, where we consider game graphs with strictly positive weights and try to minimize the cost to target, it can be shown that memoryless strategies also suffice, and the problem is in P. In MDPs, optimal expectation strategies are studied in [27, 18] for both measures:

memoryless strategies suffice and they can be computed in P.

Our strategies are *strongly risk averse*: they avoid at all cost outcomes below a given threshold (no matter their probability), and inside the set of those *safe* strategies, we maximize expectation. To the best of our knowledge, we are the first to consider such strategies.

Other notions of risk have been studied for MDPs: e.g., in [30], the authors want to find policies minimizing the probability (risk) that the total discounted rewards do not exceed a specified value; in [19], the authors want to achieve a specified value of the long-run limiting average reward at a given probability level (percentile). While those strategies limit risk, they only ensure *low probability* for bad behaviors but not their absence, furthermore, they do not ensure good expectation either.

Another body of related work is the study of strategies in MDPs that achieve a trade-off between the expectation and the variance over the outcomes (e.g., [6] for the mean-payoff, [26] for the cumulative reward), giving a statistical measure of the stability of the performance. In our setting, we strengthen this requirement by asking for *strict guarantees on individual outcomes*, while maintaining an appropriate expected payoff.

2 Beyond Worst-Case Synthesis

Preliminaries. We consider the classical models of *games* and *MDPs*. Both are based on underlying directed *graphs* with integer weights on edges.

In *games*, the set of vertices, called states, is partitioned between states of the first player, denoted by $\mathcal{P}_1$, and states of its adversary, denoted by $\mathcal{P}_2$. When the game is in a state belonging to $\mathcal{P}_i$, $i \in \{1,2\}$, then $\mathcal{P}_i$ chooses a successor state according to his *strategy*, which may in general use memory (i.e., depend on the history) and be randomized (i.e., prescribe a probability distribution over successor states). This process gives rise to a *play*, an infinite sequence of states corresponding to a path through the game graph. We assign real values to plays according to a *value function*.

In *MDPs*, the set of states is partitioned between states of $\mathcal{P}_1$ and stochastic states, where the successor state is chosen according to a given probability distribution. Basically, an MDP is a game where the strategy of $\mathcal{P}_2$ is fixed.

When we fix the strategy of $\mathcal{P}_1$ in an MDP, or the strategies of $\mathcal{P}_1$ and $\mathcal{P}_2$ in a game, we obtain a *Markov chain* (MC), a graph where all successor states are chosen according to a stochastic transition function. Given an MC, it is well-known that measurable sets of plays have uniquely defined probabilities [29], and if we have a measurable value function, we can also compute the *expected value* or *expectation* of this function when executing the MC from a given initial state.

Classical problems. In games, the *worst-case threshold problem* asks if $\mathcal{P}_1$ has a strategy such that any possible outcome, against any possible strategy of $\mathcal{P}_2$, gives a play with a value higher than a given threshold. In MDPs, the *expected value threshold problem* asks if $\mathcal{P}_1$ has a strategy such that the resulting MC yields an expectation higher than a given threshold.

Our model. The *beyond worst-case (BWC) problem* asks if $\mathcal{P}_1$ has a finite-memory strategy ensuring, *simultaneously*, a value greater than a threshold μ in the worst-case (i.e., against any strategy of the adversary), and an expected value greater than a threshold ν against a given finite-memory stochastic model of the adversary (e.g., representing commonly observed behavior of the environment). The *BWC synthesis problem* asks to synthesize such a strategy if one exists.

3 Mean-Payoff

What was known. Given a play, its mean-payoff is defined as the (inf or sup) limit of the mean encountered weights along its finite prefixes: essentially, it is the long-run average weight over the infinite play. For the worst-case threshold problem, pure memoryless optimal strategies exist for both players [25, 17] and deciding the winner is in NP $\cap$ coNP [31, 24, 21]. Whether the problem is in P is a long-standing open problem [8, 13]. Optimal expected values in MDPs can be achieved by memoryless strategies, and the corresponding decision problem can be solved in polynomial time through linear programming [18].

Our results. We prove that surprisingly, the BWC problem matches the decision complexity of the simpler worst-case problem, even collapsing to P if the latter were proved to be in P. Hence, we enrich the modeling and reasoning power over strategies without negative impact on the complexity class.

Theorem 1. *The beyond worst-case problem for the mean-payoff value function is in NP $\cap$ coNP and at least as hard as mean-payoff games.*

Furthermore, we establish that in contrast to the worst-case and expectation problems, some memory is now needed to win in general. Nevertheless, we show that elegantly implementable strategies suffice, constructed using clever alternation between memoryless strategies based on intuitive counters.

Theorem 2. *Memory of pseudo-polynomial size may be necessary and is always sufficient to satisfy the BWC problem for the mean-payoff: polynomial in the size of the game and the stochastic model, and polynomial in the weight and threshold values.*

Some key ideas. Our solving algorithm is too complex to be presented fully in this work. Nonetheless, we here give a few hints of its cornerstones, highlighting crucial aspects of the problem.

End-components. An important part of the algorithm relies on the analysis of *end-components* (ECs) in the MDP, i.e., strongly connected subgraphs in which $\mathcal{P}_1$ can ensure to stay when playing against the stochastic adversary. This is motivated by two facts. First, under any arbitrary strategy, the set of states that are seen infinitely often along an outcome corresponds with probability one to an EC [15, 1]. Second, the mean-payoff function is prefix-independent, therefore the value of any outcome only depends on the states that are seen infinitely often. Hence, the expected mean-payoff that $\mathcal{P}_1$ can achieve on the MDP depends *uniquely* on the value obtained in the ECs. Inside an EC, we can compute the maximal expected value that can be achieved by $\mathcal{P}_1$, and this value is the same in all states of the EC [18].

Classification of ECs. To be efficient w.r.t. the expectated value criterion, an acceptable strategy has to favor reaching ECs with a sufficient expectation, but under the constraint that it also guarantees satisfaction of the worst-case requirement: some ECs with high expected values may still need to be avoided because they do not permit to ensure this constraint. We establish a classification of ECs based on that observation, partitioning them between *winning ECs* (WECs) and *losing ECs* (LECs). Since the total number of ECs may be exponential, providing a representative subclass of polynomial size and computing it efficiently is a crucial point to maintain the overall NP $\cap$ coNP membership.

Within a WEC. We give a particularly interesting family of strategies for $\mathcal{P}_1$ that both guarantee safe outcomes for the worst-case, and prove to be efficient w.r.t. the expected value. Actually, we establish that the worst-case can be guaranteed *almost for free* in the sense that we can achieve expectations arbitrarily close (but not exactly equal) to what $\mathcal{P}_1$ could obtain without considering the worst-case requirement at all (i.e., in a classical MDP).

To obtain this result we use a finite-memory *combined strategy*. For two well-chosen parameters $K, L \in \mathbb{N}$, it is informally defined as follows: in phase *(a)*, play a memoryless expected value optimal

strategy for K steps and memorize $\mathsf{Sum} \in \mathbb{Z}$, the sum of weights along these steps; in phase *(b)*, if $\mathsf{Sum} > 0$, go to *(a)*, otherwise play a memoryless worst-case optimal strategy for L steps, then go to *(a)*. In phases *(a)*, $\mathcal{P}_1$ tries to increase its expectation and approach its optimal one, while in phase *(b)*, he compensates, if needed, losses that occurred in phase *(a)*.

The crux of the proof is to establish that adequate values of the parameters K and L exist. Essentially, K needs to be big enough so that the overall expectation is close to the optimal, but then L also needs to grow to be able to compensate sufficiently for the worst-case, hence lowering to some extent the overall expectation. Using results related to Chernoff bounds and Hoeffding's inequality in MCs [28, 22], we are able to show that the probability of having to compensate decreases exponentially when K increases, while L only needs to be polynomial in K. Overall, this implies the desired result that the parameters can be taken large enough for the strategy to be ε-optimal w.r.t. the expectation while worst-case safe.

4 Shortest Path

What was known. In this context, we consider game graphs where all weights are strictly positive, and a target set of states that $\mathcal{P}_1$ wants to reach while giving an upper bound on the cost to reach it. Hence the inequalities of the BWC problem are reversed. Given a play, the value function for the shortest path computes the sum of weights up to the first encounter of a state belonging to the target set, or assigning infinity if the play never reaches such a state. The worst-case threshold problem takes polynomial time, as a winning strategy of $\mathcal{P}_1$ should avoid all cycles (because they yield strictly positive costs), hence usage of attractors and comparison of the worst possible sum of costs with the threshold suffices. For the expected value threshold problem, memoryless strategies suffice and the problem is in P [4, 2].

Our results. In contrast to the mean-payoff case where we could maintain the complexity of the worst-case problem, we here provide an algorithm which operates in pseudo-polynomial time instead of truly-polynomial time. Nevertheless, we prove that the problem is actually NP-hard (reduction from the K^{th} *largest subset problem* [20]), hence establishing that a truly-polynomial-time algorithm is highly unlikely.

Theorem 3. *The beyond worst-case problem for the shortest path can be solved in pseudo-polynomial time: polynomial in the size of the underlying game graph, the stochastic model of the adversary and the encoding of the expected value threshold, and polynomial in the value of the worst-case threshold. The beyond worst-case problem for the shortest path is NP-hard.*

Once again, we show that pseudo-polynomial memory is both necessary and sufficient. Recall that the example of Fig. 1 already required memory to achieve some thresholds pair for the BWC problem.

Theorem 4. *Memory of pseudo-polynomial size may be necessary and is always sufficient to satisfy the BWC problem for the shortest path: polynomial in the size of the game and the stochastic model, and polynomial in the worst-case threshold value.*

Some key ideas. The shortest path setting has a useful property: the set of all winning strategies of $\mathcal{P}_1$ for the worst-case threshold problem can be represented through a finite game. Indeed, we construct, from the original game G and the worst-case threshold μ, a new game G_μ such that there is a bijection between the strategies of $\mathcal{P}_1$ in G_μ and the strategies of $\mathcal{P}_1$ in the original game G that are winning for the worst-case requirement: we unfold the original graph, tracking the current value of the sum of weights *up to the threshold* μ, and integrating this value in the states of an expanded graph. In the corresponding game G', we compute the set of states R from which $\mathcal{P}_1$ can reach the target set with cost lower than μ and we define the subgame $G_\mu = G' \restriction R$ such that any path in G_μ satisfies the worst-case requirement.

Assuming that G_μ is not empty, we can now combine it with the stochastic model of the adversary to construct an MDP in which we search for a $\mathcal{P}_1$ strategy that ensures reachability of the target set with an expected cost lower than the expectation threshold. If it exists, it is guaranteed that it will also satisfy the worst-case requirement against any strategy of $\mathcal{P}_2$ thanks to the bijection evoked earlier.

Hence, in the case of the shortest path, our approach is sequential, first solving the worst-case, then optimizing the expected value among the worst-case winning strategies. This sequential algorithm is depicted through Example 5. Observe that such an approach is not applicable to the mean-payoff, as in that case there exists no obvious finite representation of the worst-case winning strategies.

Example 5. Consider the game G depicted in Fig. 2. We want to synthesize a BWC strategy of $\mathcal{P}_1$ that minimizes the expected cost up to the target set $\{s_3\}$ under the (strict) worst-case threshold $\mu = 8$.

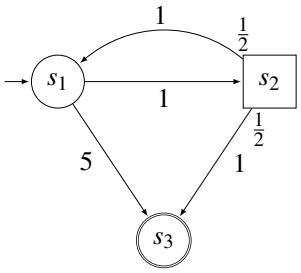


Figure 2: Simple BWC shortest path game with target set $\{s_3\}$ and worst-case threshold $\mu = 8$.

First, we unfold this game G up to the worst-case threshold (excluded), and obtain the game G' represented in Fig. 3. Observe that as soon as the worst-case threshold is reached, we stop the unfolding and associate symbol $\top$: the worst-case requirement is lost if such states are reached. This guarantees a finite (and at most pseudo-polynomial size) unfolding.

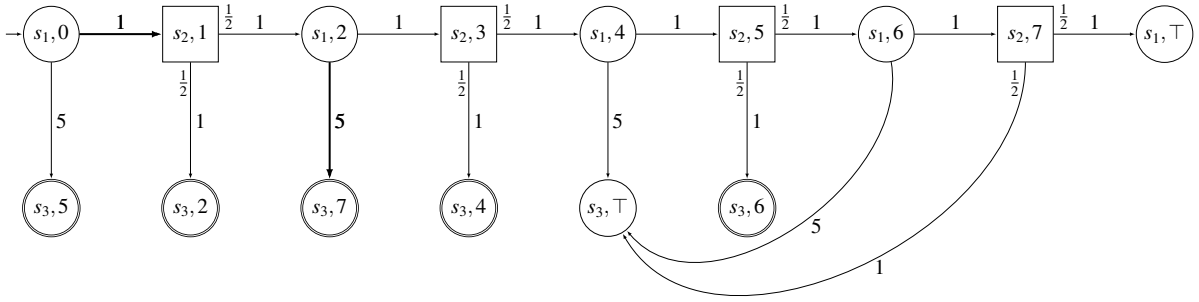


Figure 3: Unfolding of the game of Fig. 2: worst-case winning requires to reach a double state. Thick edges represent the strategy that minimizes the expected cost while ensuring this worst-case.

Therefore, it is clear that a BWC strategy of $\mathcal{P}_1$ must ensure reachability of states of G' that represent reaching the target state with a total cost strictly less than the worst-case threshold. Those states are depicted by double circles in the figure. Hence, $\mathcal{P}_1$ must stay within the attractor of those double states. It implies that state $(s_2,3)$ of the unfolding and subsequent states are off-limits.

Knowing that, it now suffices to minimize the expected value within the safe region, which is achieved by the memoryless (with regard to G') strategy that chooses to go in $(s_2,1)$ from $(s_1,0)$ and to $(s_3,7)$ from $(s_1,2)$. This strategy is depicted by the thick edges on the figure. Observe that this strategy is memoryless in G', hence requires at most pseudo-polynomial memory in G. $\triangleleft$

5 Future Work

We believe that the beyond worst-case framework is a powerful one, well-suited for specifications combining the quest of high expected performance with the need for strong worst-case guarantees. We want to build on the results presented here and consider several extensions of the initial setting.

The first line of work is applying the problem to other well-known quantitative measures and to more general classes of games (for example decidable classes of games with imperfect information [16, 23]).

A second interesting question is the extension of our results for mean-payoff and shortest path to multi-dimension games. It is already known that multi-dimension games are more complex than one-dimension ones for the worst-case threshold problem alone [12, 14]. Hence, a leap in complexity is also to be expected for the beyond worst-case problem.

Given the relevance of the framework for practical applications, it would certainly be worthwhile to develop tool suites supporting it. We could for example build on symblicit implementations recently developed for monotonic Markov decision processes by Bohy et al. [5].

Links outside computer science are also of interest. Economics is interested in strategies (i.e., investor profiles) that ensure both sufficient risk-avoidance and profitable expected return. Mathematical models powerful enough to tackle the previously discussed problems could be an advantage. A related approach to such questions is the concept of *solvency games* introduced by Berger et al. [3], and extended by Brázdil et al. [7]. Solvency games provide a framework for the analysis of risk-averse investors trying to avoid bankruptcy.

References

[1] L. de Alfaro (1997): *Formal verification of probabilistic systems*. Ph.D. thesis, Stanford University.

[2] L. de Alfaro (1999): *Computing Minimum and Maximum Reachability Times in Probabilistic Systems*. In: *Proc. of CONCUR*, LNCS 1664, Springer, pp. 66–81, doi:10.1007/3-540-48320-9_7.

[3] N. Berger, N. Kapur, L.J. Schulman & V.V. Vazirani (2008): *Solvency Games*. In: *Proc. of FSTTCS*, *LIPIcs* 2, Schloss Dagstuhl - LZI, pp. 61–72, doi:10.4230/LIPIcs.FSTTCS.2008.1741.

[4] D.P. Bertsekas & J.N. Tsitsiklis (1991): *An analysis of stochastic shortest path problems*. *Mathematics of Operations Research* 16, pp. 580–595, doi:10.1287/moor.16.3.580.

[5] A. Bohy, V. Bruyère & J.-F. Raskin (2014): *Symblicit algorithms for mean-payoff and shortest path in monotonic Markov decision processes*. *CoRR* abs/1402.1076. Available at `http://arxiv.org/abs/1402.1076`.

[6] T. Brázdil, K. Chatterjee, V. Forejt & A. Kucera (2013): *Trading Performance for Stability in Markov Decision Processes*. In: *Proc. of LICS*, IEEE Computer Society, pp. 331–340, doi:10.1109/LICS.2013.39.

[7] T. Brázdil, T. Chen, V. Forejt, P. Novotný & A. Simaitis (2013): *Solvency Markov Decision Processes with Interest*. In: *Proc. of FSTTCS*, *LIPIcs* 24, Schloss Dagstuhl - LZI, pp. 487–499, doi:10.4230/LIPIcs.FSTTCS.2013.487.

[8] L. Brim, J. Chaloupka, L. Doyen, R. Gentilini & J.-F. Raskin (2011): *Faster algorithms for mean-payoff games*. *Formal Methods in System Design* 38(2), pp. 97–118, doi:10.1007/s10703-010-0105-x.

[9] V. Bruyère, E. Filiot, M. Randour & J.-F. Raskin (2013): *Meet Your Expectations With Guarantees: Beyond Worst-Case Synthesis in Quantitative Games*. *CoRR* abs/1309.5439. Available at `http://arxiv.org/abs/1309.5439`.

[10] V. Bruyère, E. Filiot, M. Randour & J.-F. Raskin (2014): *Meet Your Expectations With Guarantees: Beyond Worst-Case Synthesis in Quantitative Games*. In: *Proc. of STACS*, *LIPIcs* 25, Schloss Dagstuhl - LZI, pp. 199–213.

[11] K. Chatterjee & L. Doyen (2011): *Games and Markov Decision Processes with Mean-payoff Parity and Energy Parity Objectives*. In: *Proc. of MEMICS*, LNCS, Springer, doi:10.1007/978-3-642-25929-6_3.

[12] K. Chatterjee, L. Doyen, T.A. Henzinger & J.-F. Raskin (2010): *Generalized Mean-payoff and Energy Games*. In: *Proc. of FSTTCS*, LIPIcs 8, Schloss Dagstuhl - LZI, pp. 505–516, doi:10.4230/LIPIcs.FSTTCS.2010.505.

[13] K. Chatterjee, L. Doyen, M. Randour & J.-F. Raskin (2013): *Looking at Mean-Payoff and Total-Payoff through Windows*. In: *Proc. of ATVA*, LNCS 8172, Springer, pp. 118–132, doi:10.1007/978-3-319-02444-8_10.

[14] K. Chatterjee, M. Randour & J.-F. Raskin (2012): *Strategy Synthesis for Multi-Dimensional Quantitative Objectives*. In: *Proc. of CONCUR*, LNCS 7454, Springer, pp. 115–131, doi:10.1007/978-3-642-32940-1_10.

[15] C. Courcoubetis & M. Yannakakis (1995): *The Complexity of Probabilistic Verification*. *J. ACM* 42(4), pp. 857–907, doi:10.1145/210332.210339.

[16] A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin & S. Torunczyk (2010): *Energy and Mean-Payoff Games with Imperfect Information*. In: *Proc. of CSL*, LNCS 6247, Springer, pp. 260–274, doi:10.1007/978-3-642-15205-4_22.

[17] A. Ehrenfeucht & J. Mycielski (1979): *Positional Strategies for Mean Payoff Games*. *Int. Journal of Game Theory* 8(2), pp. 109–113, doi:10.1007/BF01768705.

[18] J. Filar & K. Vrieze (1997): *Competitive Markov decision processes*. Springer, doi:10.1007/978-1-4612-4054-9.

[19] J.A. Filar, D. Krass & K.W. Ross (1995): *Percentile Performance Criteria For Limiting Average Markov Decision Processes*. *Transactions on Automatic Control*, pp. 2–10, doi:10.1109/9.362904.

[20] M.R. Garey & D.S. Johnson (1979): *Computers and intractability: a guide to the Theory of NP-Completeness*. Freeman New York.

[21] T. Gawlitza & H. Seidl (2009): *Games through Nested Fixpoints*. In: *Proc. of CAV*, LNCS 5643, Springer, pp. 291–305, doi:10.1007/978-3-642-02658-4_24.

[22] P.W. Glynn & D. Ormoneit (2002): *Hoeffding's inequality for uniformly ergodic Markov chains*. *Statistics & Probability Letters* 56(2), pp. 143–146, doi:10.1016/S0167-7152(01)00158-4.

[23] P. Hunter, G. A. Pérez & J.-F. Raskin (2013): *Mean-payoff Games with Incomplete Information*. *CoRR* abs/1309.5462. Available at `http://arxiv.org/abs/1309.5462`.

[24] M. Jurdziński (1998): *Deciding the Winner in Parity Games is in UP ∩ co-UP*. *Inf. Process. Lett.* 68(3), pp. 119–124, doi:10.1016/S0020-0190(98)00150-1.

[25] T.M. Liggett & S.A. Lippman (1969): *Stochastic games with perfect information and time average payoff*. *Siam Review* 11(4), pp. 604–607, doi:10.1137/1011093.

[26] S. Mannor & J.N. Tsitsiklis (2011): *Mean-Variance Optimization in Markov Decision Processes*. In: *Proc. of ICML*, Omnipress, pp. 177–184.

[27] M.L. Puterman (1994): *Markov decision processes: discrete stochastic dynamic programming*, 1st edition. John Wiley & Sons, Inc., New York, NY, USA, doi:10.1002/9780470316887.

[28] M. Tracol (2009): *Fast convergence to state-action frequency polytopes for MDPs*. *Oper. Res. Lett.* 37(2), pp. 123–126, doi:10.1016/j.orl.2008.12.003.

[29] M.Y. Vardi (1985): *Automatic Verification of Probabilistic Concurrent Finite-State Programs*. In: *Proc. of FOCS*, IEEE Computer Society, pp. 327–338, doi:10.1109/SFCS.1985.12.

[30] C. Wu & Y. Lin (1999): *Minimizing Risk Models in Markov Decision Processes with Policies Depending on Target Values*. *Journal of Mathematical Analysis and Applications* 231(1), pp. 47–67, doi:10.1006/jmaa.1998.6203.

[31] U. Zwick & M. Paterson (1996): *The complexity of mean payoff games on graphs*. *Theoretical Computer Science* 158, pp. 343–359, doi:10.1016/0304-3975(95)00188-3.

Games for the Strategic Influence of Expectations

Lluís Godo
Artificial Intelligence Research Institute, IIIA
Spanish National Research Council, CSIC
Campus UAB, 08193 Bellaterra, Spain
godo@iiia.csic.es

Enrico Marchioni
Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier
118 Route de Narbonne, 31062 Toulouse, France
enrico.marchioni@irit.fr

We introduce a new class of games where each player's aim is to randomise her strategic choices in order to affect the other players' expectations aside from her own. The way each player intends to exert this influence is expressed through a Boolean combination of polynomial equalities and inequalities with rational coefficients. We offer a logical representation of these games as well as a computational study of the existence of equilibria.[1]

1 Introduction

In the situations of strategic interactions modelled in Game Theory, the goal of each player is essentially the maximisation of her own expected payoff. Players, however, often care not only about maximising their own expectation, but also about influencing other players' expected outcomes. As an example, consider a number of competing investment banks selling and buying tradable assets so that the trading of financial products affects each other's profit. These banks might randomize their choices and obviously aim at maximizing their expected profit. Still, their strategy might go beyond the choice of a specific investment and they might be interested in influencing the market and the behavior of other banks possibly undermining the expected gain of their competitors.

In this work, we offer logical models to formalize these kinds of strategic interactions, called Expectation Games, where each player's aim is to randomise her strategic choices in order to affect the other players' expectations over an outcome as well as their own expectation. Expectation Games are an extension of Łukasiewicz games [9] and are based on the logics $\mathsf{E}(\mathfrak{G})$ that formalise reasoning about expected payoffs in a class of Łukasiewicz games [4]. Łukasiewicz games [9], a generalisation of Boolean games [7], involve a finite set of players P_i each controlling a finite set of propositional variables V_i, whose strategy corresponds to assigning values from the scale $L_k = \left\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\right\}$ to the variables in V_i. Strategies can be interpreted as efforts or costs, and each player's strategic choice can be seen as an assignment to each controlled variable carrying an intrinsic cost. Each player is given a finitely-valued Łukasiewicz logic formula φ_i, with variables from $\bigcup_i^n \mathsf{V}_i$, whose valuation is interpreted as the payoff function for P_i and corresponds to the restriction over L_k of a continuous piecewise linear polynomial function [2].

Expectation Games expand Lukasiewicz games by assigning to each player P_i a modal formula Φ_i of the logic $\mathsf{E}(\mathfrak{G})$, whose interpretation corresponds to a piecewise rational polynomial function whose variables are interpreted as the expected values of the payoff functions φ_i. Each formula Φ_i is then meant to represent a player's goal concerning the relation between her and other players' expectations.

[1] This extended abstract is based on the article [4] and an upcoming extended version of the same work.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 9–15, doi:10.4204/EPTCS.146.2

2 Logical Background

The language of Łukasiewicz logic Ł (see [2]) is built from a countable set of propositional variables $\{p_1, p_2, \ldots\}$, the binary connective $\to$ and the truth constant $\overline{0}$ (for falsity). Further connectives are defined as follows:

$$\begin{array}{rclrcl} \neg\varphi & \text{is} & \varphi \to \overline{0}, & \varphi \wedge \psi & \text{is} & \varphi\&(\varphi \to \psi), \\ \varphi\&\psi & \text{is} & \neg(\varphi \to \neg\psi), & \varphi \vee \psi & \text{is} & ((\varphi \to \psi) \to \psi), \\ \varphi \oplus \psi & \text{is} & \neg(\neg\varphi\&\neg\psi), & \varphi \leftrightarrow \psi & \text{is} & (\varphi \to \psi)\&(\psi \to \varphi), \\ \varphi \ominus \psi & \text{is} & \varphi\&\neg\psi, & d(\varphi,\psi) & \text{is} & \neg(\varphi \leftrightarrow \psi). \end{array}$$

Let *Form* denote the set of Łukasiewicz logic formulas. A valuation e from *Form* into $[0,1]$ is a mapping $e : \mathit{Form} \to [0,1]$ assigning to all propositional variables a value from the real unit interval (with $e(\overline{0}) = 0$) that can be extended to complex formulas as follows:

$$\begin{array}{rclrcl} e(\varphi \to \psi) & = & \min(1 - e(\varphi) + e(\psi), 1) & e(\neg\varphi) & = & 1 - e(\varphi) \\ e(\varphi\&\psi) & = & \max(0, e(\varphi) + e(\psi) - 1) & e(\varphi \oplus \psi) & = & \min(1, e(\varphi) + e(\psi)) \\ e(\varphi \ominus \psi) & = & \max(0, e(\varphi) - e(\psi)) & e(\varphi \wedge \psi) & = & \min(e(\varphi), e(\psi)) \\ e(\varphi \vee \psi) & = & \max(e(\varphi), e(\psi)) & e(d(\varphi,\psi)) & = & |e(\varphi) - e(\psi)| \\ e(\varphi \leftrightarrow \psi) & = & 1 - |e(\varphi) - e(\psi)| & & & \end{array}$$

A valuation *e satisfies* a formula φ if $e(\varphi) = 1$. As usual, a set of formulas is called a theory. A valuation e satisfies a theory T, if $e(\psi) = 1$, for every $\psi \in T$.

Infinite-valued Łukasiewicz logic has the following axiomatisation:

(Ł1) $\varphi \to (\psi \to \varphi)$, (Ł2) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$,
(Ł3) $(\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$, (Ł4) $((\varphi \to \psi) \to \psi) \to ((\psi \to \varphi) \to \varphi)$.

The only inference rule is *modus ponens*, i.e.: from $\varphi \to \psi$ and φ derive ψ.

A *proof* in Ł is a sequence $\varphi_1, \ldots, \varphi_n$ of formulas such that each φ_i either is an axiom of Ł or follows from some preceding φ_j, φ_k $(j,k < i)$ by modus ponens. We say that a formula φ can be derived from a theory T, denoted as $T \vdash \varphi$, if there is a proof of φ from a set $T' \subseteq T$. A theory T is said to be consistent if $T \nvdash \overline{0}$.

Łukasiewicz logic is complete with respect to deductions from finite theories for the given semantics, i.e.: for every finite theory T and every formula φ, $T \vdash \varphi$ iff every valuation e that satisfies T also satisfies φ.

For each $k \in \mathbb{N}$, the finite-valued Łukasiewicz logic Ł$_k$ is the schematic extension of Ł with the axiom schemas:

(Ł5) $(n-1)\varphi \leftrightarrow n\varphi$, (Ł6) $(k\varphi^{k-1})^n \leftrightarrow n\varphi^k$,

for each integer $k = 2, \ldots, n-2$ that does not divide $n-1$, and where $n\varphi$ is an abbreviation for $\varphi \oplus \cdots \oplus \varphi$ (n times) and φ^k is an abbreviation for $\varphi\&\ldots\&\varphi$, (k times). The notions of valuation and satisfiability for Ł$_k$ are defined as above just replacing $[0,1]$ by

$$L_k = \left\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\right\}$$

as set of truth values. Every Ł$_k$ is complete (in the above sense) with respect to deductions from finite theories for the given semantics.

It is sometimes useful to introduce constants in addition to $\overline{0}$ that will denote values in the domain L_k. Specifically, we will denote by Ł$_k^c$ the Łukasiewicz logic obtained by adding constants $\overline{c}$ for every value $c \in L_k$. We assume that valuation functions e interpret such constants in the natural way: $e(\overline{c}) = c$.

A McNaughton function [2] is a continuous piecewise linear polynomial functions with integer coefficients over the nth-cube $[0,1]^n$. To each Łukasiewicz formula $\varphi(p_1,\ldots,p_n)$ we can associate a McNaughton function f_φ so that, for every valuation e

$$f_\varphi(e(p_1),\ldots,e(p_n)) = e(\varphi(p_1,\ldots,p_n)).$$

Every Ł-formula is then said to define a McNaughton function. The converse is also true, i.e. every continuous piecewise linear polynomial function with integer coefficients over $[0,1]^n$ is definable by a formula in Łukasiewicz logic. In the case of finite-valued Łukasiewicz logics, the functions defined by formulas are just the restrictions of McNaughton functions over $(L_k)^n$. In this sense, we can associate to every formula $\varphi(p_1,\ldots,p_n)$ from Ł$_k$ a function $f_\varphi : (L_k)^n \to L_k$. As for each Ł$_k^c$, the functions defined by a formula are combinations of restrictions of McNaughton functions and, in addition, the constant functions for each $c \in L_k$. The class of functions definable by Ł$_k^c$-formulas exactly coincides with the class of all functions $f : (L_k)^n \to L_k$, for every $n \geq 0$.

The expressive power of infinite-valued Łukasiewicz logic lies in, and is limited to, the definability of piecewise linear polynomial functions. Expanding Ł with the connectives $\odot, \to_\Pi$ of Product logic [6], interpreted as the product of reals and as the truncated division, respectively, significantly augments the expressive power of the logic. The Ł$\Pi\frac{1}{2}$ logic [3] is the result of this expansion, obtained by adding the connectives $\odot, \to_\Pi, \overline{\frac{1}{2}}$, whose valuations e extend the valuations for Ł as follows:

$$e(\varphi \odot \psi) = e(\varphi)\cdot e(\psi), \qquad e(\varphi \to_\Pi \psi) = \begin{cases} 1 & e(\varphi) \leq e(\psi) \\ \frac{e(\psi)}{e\varphi} & \text{otherwise} \end{cases}, \qquad e\left(\overline{\tfrac{1}{2}}\right) = \tfrac{1}{2}.$$

Notice that the presence of the constant $\overline{\frac{1}{2}}$ makes it possible to define constants for all rationals in $[0,1]$ (see [3]). Ł$\Pi\frac{1}{2}$'s axioms include the axioms of Łukasiewicz and Product logics (see [6]) as well as the following additional axioms, where $\Delta\varphi$ is $\neg\varphi \to_\Pi \overline{0}$:

(ŁΠ1) $(\varphi \odot \psi) \ominus (\varphi \odot \chi) \leftrightarrow \varphi \odot (\psi \ominus \chi)$,
(ŁΠ2) $\Delta(\varphi \to \psi) \to (\varphi \to_\Pi \psi)$,
(ŁΠ3) $\Delta(\varphi \to_\Pi \psi) \to (\varphi \to \psi)$,
(ŁΠ4) $\overline{\frac{1}{2}} \leftrightarrow \neg\overline{\frac{1}{2}}$.

The deduction rules are modus ponens for & and $\to$, and the necessitation rule for Δ, i.e.: from φ derive $\Delta\varphi$. Ł$\Pi\frac{1}{2}$ is complete with respect to deductions from finite theories for the given semantics [3].

While Ł is the logic of McNaughton functions, Ł$\Pi\frac{1}{2}$ is the logic of piecewise rational functions over $[0,1]^n$, for all n (see [10]). In fact, the function defined by each Ł$\Pi\frac{1}{2}$-formula with n variables corresponds to a supremum of rational fractions

$$\frac{P(x_1,\ldots,x_n)}{Q(x_1,\ldots,x_n)}$$

over $[0,1]^n$, where $P(x_1,\ldots,x_n), Q(x_1,\ldots,x_n)$ are polynomials with rational coefficients. Conversely, every piecewise rational function with over the unit cube $[0,1]^n$ can be defined by an Ł$\Pi\frac{1}{2}$-formula.

3 Logics for Łukasiewicz Games with Expectations

In this section we briefly introduce Łukasiewicz games on Ł$_k^c$ along with the logics $\mathsf{E}(\mathfrak{G})$ to represent expected payoffs in classes of games. $\mathsf{E}(\mathfrak{G})$ will be the basis upon which Expectation Games are defined.

3.1 Łukasiewicz Games

Definition 3.1 ([9]) *A Łukasiewicz game $\mathscr{G}$ on $Ł_k^c$ is a tuple $\mathscr{G} = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}, \{\mathsf{S}_i\}, \{\varphi_i\}\rangle$ where:*

1. *$\mathsf{P} = \{P_1, \ldots, P_n\}$ is a set of* players*;*
2. *$\mathsf{V} = \{p_1, \ldots, p_m\}$ is a finite set of propositional variables;*
3. *For each $i \in \{1, \ldots, n\}$, $\mathsf{V}_i \subseteq \mathsf{V}$ is the set of propositional variables under control of player P_i, so that the sets V_i form a partition of V, with $|\mathsf{V}_i| = m_i$, and $\sum_{i=1}^{n} m_i = m$.*
4. *For each $i \in \{1, \ldots, n\}$, S_i is the strategy set for player P_i that consists of all valuations $s : \mathsf{V}_i \to L_k$ of the propositional variables in V_i, i.e. $\mathsf{S}_i = \{s \mid s : \mathsf{V}_i \to L_k\}$.*
5. *For each $i \in \{1, \ldots, n\}$, $\varphi_i(p_1, \ldots, p_t)$ is an $Ł_k^c$-formula, built from variables in V, whose associated function $f_{\varphi_i} : (L_k)^t \to L_k$ corresponds to the* payoff function *of P_i, and whose value is determined by the valuations in $\{\mathsf{S}_1, \ldots, \mathsf{S}_n\}$.*

We denote by $\mathsf{S} = \mathsf{S}_1 \times \cdots \times \mathsf{S}_n$ the product of the strategy spaces. A tuple $\vec{s} = (s_1, \ldots, s_n) \in \mathsf{S}$ of strategies is called a *strategy combination*. With an abuse of notation, we denote by $f_{\varphi_i}(\vec{s})$ the value of the payoff function f_{φ_i} under the valuation corresponding to the strategy combination $\vec{s}$.

Given a game $\mathscr{G}$, let $\delta : \mathsf{P} \to \{1, \ldots, m\}$ be a function assigning to each player P_i an integer from $\{1, \ldots, m\}$ that corresponds to the number of variables in V_i: i.e.: $\delta(P_i) = m_i$. δ is called a *variable distribution function*. Given a game $\mathscr{G}$, the *type* of $\mathscr{G}$ is the triple $\langle n, m, \delta\rangle$, where n is the number of players, m is the number of variables in V, and δ is the variable distribution function for $\mathscr{G}$.

Definition 3.2 (Class) *Let $\mathscr{G}$ and $\mathscr{G}'$ be two Łukasiewicz games $\mathscr{G}$ and $\mathscr{G}'$ on $Ł_k^c$ of type $\langle n, m, \delta\rangle$ and $\langle n, m, \delta'\rangle$, respectively. We say that $\mathscr{G}$ and $\mathscr{G}'$ belong to the same class $\mathfrak{G}$ if there exists a permutation $\mathfrak{j}$ of the indices $\{1, \ldots, n\}$ such that, for all P_i, $\delta(P_{\mathfrak{j}(i)}) = \delta'(P_i)$.*

Notice that what matters in the definition of a type is not which players are assigned certain variables, but rather their distribution.

Let $\mathscr{G}$ be a Łukasiewicz game on $Ł_k^c$. A *mixed strategy* π_i for player P_i is a probability distribution on the strategy space S_i. By π_{-i}, we denote the tuple of mixed strategies $(\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_n)$. P_{-i} denotes the tuple of players $(P_1, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n)$. Given the mixed strategies $(\pi_1, \ldots, \pi_n)$, the *expected payoff* for P_i of playing π_i, when P_{-i} play π_{-i}, is given by

$$exp_{\varphi_i}(\pi_i, \pi_{-i}) = \sum_{\vec{s}=(s_1,\ldots,s_n)\in\mathsf{S}} \left(\left(\prod_{j=1}^{n} \pi_j(s_j) \right) \cdot f_{\varphi_i}(\vec{s}) \right)$$

3.2 The Logics $\mathsf{E}(\mathfrak{G})$

Given a class of games $\mathfrak{G}$ on $Ł_k^c$, the language of $\mathsf{E}(\mathfrak{G})$ is defined as follows: (1) The set NModF of non-modal formulas corresponds to the set of $Ł_k^c$-formulas built from the propositional variables $p_1, \ldots, p_m$. (2) The set ModF of modal formulas is built from the atomic modal formulas $\mathsf{E}\varphi$, with $\varphi \in$ NModF, using the connectives of the $Ł\Pi\frac{1}{2}$ logic. $\mathsf{E}\varphi$ is meant to encode a player's expected payoff of playing a mixed strategy, given the payoff function associated to φ. Nested modalities are not allowed.

A model $\mathbf{M}$ for $\mathsf{E}(\mathfrak{G})$ is a tuple $\langle \mathsf{S}, e, \{\pi_i\}\rangle$, such that:

1. $\mathsf{S} = \mathsf{S}_1 \times \cdots \times \mathsf{S}_n$ is the set of all strategy combinations, i.e.

$$\{\vec{s} = (s_1, \ldots, s_n) \mid (s_1, \ldots, s_n) \in \mathsf{S}_1 \times \cdots \times \mathsf{S}_n\}.$$

2. $e : (\text{NModF} \times \mathsf{S}) \to L_k$ is a valuation of non-modal formulas, such that, for each $\varphi \in \text{NModF}$ $e(\varphi, \vec{s}) = f_\varphi(\vec{s})$, where f_φ is the function associated to φ and $\vec{s} = (s_1, \ldots, s_n)$.
3. $\pi_i : \mathsf{S}_i \to [0,1]$ is a probability distribution, for each P_i.

The truth value of a formula Φ in $\mathbf{M}$ at $\vec{s}$, denoted $\|\Phi\|_{\mathbf{M},\vec{s}}$, is inductively defined as follows:

1. If Φ is a non-modal formula $\varphi \in \text{NModF}$, then $\|\varphi\|_{\mathbf{M},\vec{s}} = e(\varphi, \vec{s})$,
2. If Φ is an atomic modal formula $\mathsf{E}\varphi$, then $\|\mathsf{E}\varphi\|_{\mathbf{M},\vec{s}} = exp_\varphi(\pi_1, \ldots, \pi_n)$.
3. If Φ is a non-atomic modal formula, its truth value is computed by evaluating its atomic modal subformulas and then by using the truth functions associated to the $\text{Ł}\Pi\frac{1}{2}$-connectives occurring in Φ.

Since the valuation of a modal formula Φ does not depend on a specific strategy combination but only on the model $\mathbf{M}$, we will often simply write $\|\Phi\|_{\mathbf{M}}$ to denote the valuation of Φ in $\mathbf{M}$.

Theorem 3.3 (Completeness) *Let Γ and Φ be a finite modal theory and a modal formula in $\mathsf{E}(\mathfrak{G})$. Then, $\Gamma \vdash_{\mathsf{E}(\mathfrak{G})} \Phi$ if and only if for every model $\mathbf{M}$ such that, for each $\Psi \in \Gamma$, $\|\Psi\|_{\mathbf{M}} = 1$, also $\|\Phi\|_{\mathbf{M}} = 1$.*

4 Expectation Games

In this section we introduce a class of games with polynomial constraints over expectations. These games expand Lukasiewicz games by assigning to each player a formula Φ_i of $\mathsf{E}(\mathfrak{G})$, whose interpretation corresponds to a piecewise rational polynomial function whose variables are expected values. The formula Φ_i is meant to represent a player's goal concerning the relation between her and other players' expectations.

Definition 4.1 *An Expectation Game $\mathscr{E}_\mathscr{G}$ on $\mathsf{E}(\mathfrak{G})$ is a tuple $\mathscr{E}_\mathscr{G} = \langle \mathscr{G}, \{\mathsf{M}_i\}, \{\Phi_i\} \rangle$, where:*

1. *$\mathscr{G}$ is a Łukasiewicz game on Ł_k^c, with $\mathscr{G} \in \mathfrak{G}$,*
2. *for each $i \in \{1, \ldots, n\}$, M_i is the set of all mixed strategies on S_i of player P_i,*
3. *for each $i \in \{1, \ldots, n\}$, Φ_i is an $\mathsf{E}(\mathfrak{G})$-formula such that every atomic modal formula occurring in Φ_i has the form $\mathsf{E}\psi$, with $\psi \in \{\varphi_1, \ldots, \varphi_n\}$, i.e. the payoff formulas in $\mathscr{G}$.*

A model $\mathbf{M} = \langle \mathsf{S}, e, \{\pi_i\} \rangle$ of $\mathsf{E}(\mathfrak{G})$ for a game $\mathscr{E}_\mathscr{G}$ is called a *best response model* for a player P_i whenever, for all models $\mathbf{M}' = \langle \mathsf{S}, e, \{\pi_i'\} \rangle$ with $\pi'_{-i} = \pi_{-i}$,

$$\|\Phi_i\|_{\mathbf{M}'} \leq \|\Phi_i\|_{\mathbf{M}}.$$

An expectation game $\mathscr{E}_\mathscr{G}$ on $\mathsf{E}(\mathfrak{G})$ is said to have a *Nash Equilibrium*, whenever there exists a model $\mathbf{M}^*$ that is a best response model for each player P_i. In that case $\mathbf{M}^*$ is called an *equilibrium model*.

Example 1. Let $\mathscr{E}_\mathscr{G}$ be any expectation game where each P_i is simply assigned the formula $\Phi_i := \mathsf{E}\varphi_i$. This game corresponds to the the situation where each player cares only about her own expectation and whose goal is its maximisation. Clearly, by Nash's Theorem [11], every $\mathscr{E}_\mathscr{G}$ of this form admits an Equilibrium, since it offers a formalisation of the classical case where equilibria are given by tuples of mixed strategies over valuations in a Łukasiewicz game.

Example 2. Not every expectation game has an equilibrium. In fact, consider the following game $\mathscr{E}_\mathscr{G} = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}, \{\mathsf{S}_i\}, \{\varphi_i\}, \{\mathsf{M}_i\}, \{\Phi_i\} \rangle$, with $i \in \{1, 2\}$, where:

(1) $\varphi_1 := p_1$ and $\varphi_2 := p_2$, and (2) $\Phi_1 := \neg d(\mathsf{E}(p_1), \mathsf{E}(p_2))$ and $\Phi_2 := d(\mathsf{E}(p_1), \mathsf{E}(p_2))$.[2]

The above game can be regarded as a particular version of Matching Pennies with expectations. In fact, while P_1 aims at matching P_2's expectation, P_2 wants their expectations to be as far as possible. It is easy to see that there is no model $\mathbf{M}$ that gives an equilibrium for $\mathscr{E}_\mathscr{G}$. Therefore:

Proposition 4.2 *There exist Expectation Games on* $\mathsf{E}(\mathfrak{G})$ *that do not admit a Nash Equilibrium.*

5 Complexity

Definition 5.1 *For a given game* $\mathscr{E}_\mathscr{G}$*, the* MEMBERSHIP *problem is the problem of determining whether there exists an equilibrium model* $\mathbf{M}$*. For a given game* $\mathscr{E}_\mathscr{G}$ *and model* $\mathbf{M}$ *with with rational mixed strategies* $(\pi_1, \ldots, \pi_n)$*, the* NON-EMPTINESS *problem is the problem of determining whether* $\mathbf{M}$ *belongs to the set of Nash Equilibria.*

Recall that the first-order theory $\mathsf{Th}(\mathbb{R})$ of real closed fields is the set of sentences in the language of ordered rings $\langle +, -, \cdot, 0, 1, < \rangle$ that are valid over the field of reals [8]. The existence of an equilibrium in a game $\mathscr{E}_\mathscr{G}$ can be expressed through a first-order sentence ξ of $\mathsf{Th}(\mathbb{R})$:

Proposition 5.2 *For each Expectation Game* $\mathscr{E}_\mathscr{G}$ *there exists a first-order sentence* ξ *of the theory* $\mathsf{Th}(\mathbb{R})$ *of real closed fields so that* $\mathscr{E}_\mathscr{G}$ *admits a Nash Equilibrium if and only if* ξ *holds in* $\mathsf{Th}(\mathbb{R})$*.*

As a consequence of the above, it is easy to see that a game $\mathscr{E}_\mathscr{G}$ admits an equilibrium if and only if there exists a quantifier-free formula in the language of ordered rings that defines a non-empty semialgebraic set over the reals [8].

We exploit the connection with $\mathsf{Th}(\mathbb{R})$ to determine the computational complexity of both the MEMBERSHIP and the NON-EMPTINESS problem. In fact, given a game $\mathscr{E}_\mathscr{G}$, it can be shown that the sentence ξ can be computed from $\mathscr{E}_\mathscr{G}$ but its length is exponential in the number of propositional variables of the payoff formulas ϕ_i. Deciding the validity of a sentence in $\mathsf{Th}(\mathbb{R})$ is singly exponential in the number of variables and doubly exponential in the number of alternations of quantifier blocks [5]. It can be shown that for every game the alternation of quantifiers in ξ is always fixed. As a consequence, we obtain:

Theorem 5.3 *Given an Expectation Game* $\mathscr{E}_\mathscr{G}$ *the* NON-EMPTINESS *problem can be decided in* 2-EXPTIME.

Deciding the validity of a sentence with only existential quantifiers in $\mathsf{Th}(\mathbb{R})$ can be solved in PSPACE [1]. We can show that, given a game $\mathscr{E}_\mathscr{G}$ and model $\mathbf{M}$ with rational mixed strategies $(\pi_1, \ldots, \pi_n)$, we can compute in polynomial time an existential sentence of $\mathsf{Th}(\mathbb{R})$ whose validity is equivalent to the fact that $\mathbf{M}$ is an equilibrium model.

Theorem 5.4 *Given an Expectation Game* $\mathscr{E}_\mathscr{G}$ *and a model* $\mathbf{M}$ *with rational mixed strategies* $(\pi_1, \ldots, \pi_n)$*, the* MEMBERSHIP *problem can be decided in* PSPACE*.*

[2] Where $\neg d(\mathsf{E}(p_1), \mathsf{E}(p_2))$ is interpreted as $1 - |exp_{p_1}(\pi_1, \pi_2) - exp_{p_2}(\pi_1, \pi_2)|$ and $d(\mathsf{E}(p_1), \mathsf{E}(p_2))$ as $|exp_{p_1}(\pi_1, \pi_2) - exp_{p_2}(\pi_1, \pi_2)|$ (see [4]).

6 Extensions and Future Work

This work lends itself to several extensions and generalizations. On the one hand we plan to study the notion of correlated equilibria for Expectation Games as well as to determine the complexity of checking their existence. In addition, we are interested in studying games where an external agent can exert influence on the game by imposing constraints on the payoffs and the expectations. This agent would then play the role of an enforcer by pushing the players to make choices that agree with her dispositions. Also, we plan to investigate games based on infinite-valued Łukasiewicz logic [2] where players have infinite strategy spaces. Finally, we intend to explore possible relations with stochastic games and whether our framework can be adapted to formalize those kinds of strategic interactions.

Acknowledgements

Godo acknowledges support from the Spanish projects EdeTRI (TIN2012-39348-C02-01) and AT (CONSOLIDER CSD 2007-0022). Marchioni acknowledges support from the Marie Curie Intra-European Fellowship NAAMSI (FP7-PEOPLE-2011-IEF).

References

[1] J.F. Canny. Some algebraic and geometric computations in PSPACE. In *Proc. of the 20th ACM Symposium on Theory of Computing*, 460–467, 1988, doi:10.1145/62212.62257.

[2] R. Cignoli, I. M. L. D'Ottaviano, D. Mundici. *Algebraic Foundations of Many-Valued Reasoning*, Volume 7 of *Trends in Logic*, Kluwer Academic Publishers, Dordrecht, 2000, doi:10.1007/978-94-015-9480-6.

[3] F. Esteva, L. Godo, F. Montagna. The ŁΠ and ŁΠ$\frac{1}{2}$ logics: two complete fuzzy systems joining Łukasiewicz and product logic. *Archive for Mathematical Logic*, 40: 39–67, 2001, doi:10.1007/s001530050173.

[4] L. Godo, E. Marchioni. Logics for Non-Cooperative Games with Expectations. In *Proc. of the 11th European Workshop on Multi-Agent Systems*, Toulouse, France, 2013, available at http://ceur-ws.org/Vol-1113/paper7.pdf.

[5] D. Y. Grigor'ev. Complexity of deciding Tarski algebra. *Journal of Symbolic Computation*, 5: 65–108, 1988, doi:10.1016/S0747-7171(88)80006-3.

[6] P. Hájek. *Metamathematics of Fuzzy Logic*. Volume 4 of *Trends in Logic*, Kluwer Academic Publishers, Dordrecht, 1998, doi:10.1007/978-94-011-5300-3.

[7] P. Harrenstein, W. van der Hoek, J.J.Ch. Meyer, C. Witteveen. Boolean games. In *Proc. of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, J. van Benthem (Ed.), Siena, Italy, 287–298, 2001, available at http://www.tark.org/proceedings/tark_jul8_01/p287-harrenstein.pdf.

[8] W. Hodges. *Model theory*. Cambridge University Press, Cambridge, 1993, doi:10.1017/CBO9780511551574.

[9] E. Marchioni, M. Wooldridge. Łukasiewicz Games. In *Proceedings of the 13th International Conferece on Autonomous Agents and Multiagent Systems*, Paris, France, 2014, to appear.

[10] F. Montagna, G. Panti. Adding structures to MV-algebras. *Journal of Pure and Applied Algebra*, 164: 365–387, 2001, doi:10.1016/S0022-4049(00)00169-9.

[11] J. Nash. Non-cooperative games. *The Annals of Mathematics*, Second Series, 54(2): 286–295, 1951, available at http://www.jstor.org/stable/1969529.

On Defendability of Security Properties

Wojciech Jamroga
Computer Science and Communication
& Interdisciplinary Centre for Security, Reliability, and Trust,
University of Luxembourg
wojtek.jamroga@uni.lu

Matthijs Melissen
Computer Science and Communication,
University of Luxembourg
& School of Computer Science,
University of Birmingham
m.melissen@cs.bham.ac.uk

Henning Schnoor
Arbeitsgruppe Theoretische Informatik,
University of Kiel
henning.schnoor@email.uni-kiel.de

We study the security of interaction protocols when incentives of participants are taken into account. We begin by formally defining correctness of a protocol, given a notion of rationality and utilities of participating agents. Based on that, we propose how to assess security when the precise incentives are unknown. Then, the security level can be defined in terms of *defender sets*, i.e., sets of participants who can effectively "defend" the security property as long as they are in favor of the property.

We present some theoretical characterizations of defendable protocols under Nash equilibrium, first for bijective games (a standard assumption in game theory), and then for games with non-injective outcomes that better correspond to interaction protocols. Finally, we apply our concepts to analyze fairness in the ASW contract-signing protocol.

1 Introduction

Interaction protocols are ubiquitous in multi-agent systems. Protocols can be modeled as games, since every participant in the protocol has several strategies that she can employ. From a game-theoretic perspective, protocols are an interesting class of games since they have a *goal*, i.e., a set of outcomes that are preferred by the designer of the protocol. *Security protocols* use cryptography to enforce their goals against any possible behavior of participants. Such a protocol is deemed correct with respect to its goal if the goal is achieved in all runs where a predefined subset of players follows the protocol.

We point out that this definition of correctness can be too strong, since violation of the goal may be achievable only by irrational responses from the other players. On the other hand, the definition may also prove too weak when the goal can be only achieved by an irrational strategy of agents supporting the goal, in other words: one that they should never choose to play. To describe and predict rational behavior of agents, game theory has proposed a number of *solution concepts* [13]. Each solution concept captures some notion of rationality which may be more or less applicable in different contexts. We do not fix a particular solution concept, but consider it to be a parameter of the problem.

Our main contributions are the following. First, in Section 3.1, we define a parametrized notion of *rational correctness* for security protocols, where the parameter is a suitable solution concept. Secondly, based on this notion, we define a concept of *defendability of security* in a protocol, where the security property is guaranteed under relatively weak assumptions (Section 3.3). Thirdly, in Section 4, we propose a *characterization* of defendable security properties when rationality of participants is based on Nash equilibrium. Finally, we consider the case of mixed strategies in Section 5, we generalize the results to

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 17–25, doi:10.4204/EPTCS.146.3

non-injective game models in Section 6, and apply our concepts to analyze fairness in the ASW contract-signing protocol in Section 7. Most of this paper (Sections 2–5) is a compressed version of the material already published in [9]. The novel contribution is presented in Sections 6 and 7.

We want to emphasize that our work does not focus on "classical" security protocols where most participants are assumed to be "honest", i.e., to follow a typically deterministic sequence of actions. More appropriately, we should say that we study *interaction protocols* in general, where actions of participants may or may not be "honest", and the actual set of available behaviors depends on the execution semantics of the protocol. We believe that the two kinds of assumptions (honesty vs. being in favor of the protocol objective) are largely orthogonal. A study of interplay between the two is left for future work.

1.1 Related Work

Researchers have considered protocol execution as a game with the very pessimistic assumption that the only goal of the other participants ("adversaries") is to break the intended security property of the protocol. In this case, a protocol is correct if the "honest" participants have a strategy such that, for all strategies of the other agents, the goal of the protocol is satisfied (cf. e.g. [10]). Recently, protocols have been analyzed with respect to some game theoretic notions of rationality [7, 2] where preferences of participants are taken into account. An overview of connections between cryptography and game theory is given in [6]. Another survey [12] presents arguments suggesting that study of incentives in security applications is crucial. Buttyán, Hubaux and Čapkun [4] model protocols in a way similar to ours, and also use incentives to model the behavior of agents. However, they restrict their analysis to strongly Pareto-optimal Nash equilibria which is not necessarily a good solution concept for security protocols: First, it is unclear why agents would *individually* converge to a strongly Pareto-optimal play. Moreover, in many protocols it is unclear why agents would play a Nash equilibrium in the first place. Our method is more general, as we use the solution concept as a parameter to our analysis. Asharov et al. (2011) [2] use game theory to study gradual-release fair exchange protocols. They consider a protocol to be game-theoretically fair if the strategy that never aborts the protocol is a computational Nash-equilibrium. They prove that their analysis allows for solutions that are not admitted by the traditional cryptographic definition. Groce and Katz [8] show that if agents have a strict incentive to achieve fair exchange, then gradual-release fair exchange without trusted third party (TTP) is possible under the assumption that the other agents play rationally. Syverson [14] presents a *rational exchange* protocol for which he shows that "enlightened, self-interested parties" have no reason to cheat. Finally, Chatterjee & Raman [5] use assume-guarantee synthesis for synthesis of contract signing protocols.

In summary, rationality-based correctness of protocols has been studied in a number of papers, but usually with a particular notion of rationality in mind. In contrast, we define a concept of correctness where a game-theoretic solution concept is a parameter of the problem. Even more importantly, our concept of *defendability* of a security property is completely novel. The same applies to our characterizations of defendable properties under Nash equilibrium.

2 Protocols and Games

A protocol is a specification of how agents should interact. Protocols can contain *choice points* where several actions are available to the agents. An agent is *honest* if he follows the protocol specification, and *dishonest* otherwise, i.e., when he behaves in a way that is not allowed by the protocol. In the latter case, the agent is only restricted by the physical and logical actions that are available in the environment.

For instance, in a cryptographic protocol, dishonest agents can do anything that satisfies properties of the cryptographic primitives, assuming perfect cryptography (as in [11]). The protocol, together with a model of the environment of action, a subset of agents who are assumed to be honest, and the operational semantics of action execution, defines a multi-agent transition system that we call the *model* of the protocol. In the rest of the paper, we focus on protocol models, and abstract away from how they arise. We also do not treat the usual "network adversary" that can intercept, delay and forge messages, but essentially assume the existence of secure channels. The issue of the "network adversary" is of course highly relevant for security protocols, but orthogonal to the aspects we discuss in this paper. In the full version of this paper [9], we present contract signing protocols as a running example. In such a protocol, Alice and Bob want to sign a contract. Among the most relevant game-theoretic security properties of such protocols are fairness, balancedness, and abuse-freeness.

We use *normal-form games* as abstract models of interaction in a protocol.

Definition 2.1 (Frames and games)**.** *A* game frame *is a tuple* $\Gamma = (N, \Sigma)$*, where* $N = \{A_1, \ldots, A_{|N|}\}$ *is a finite set of* agents*, and* $\Sigma = \Sigma_{A_1} \times \cdots \times \Sigma_{A_{|N|}}$ *is a set of strategy profiles.*

A normal-form (NF) game *is a game frame plus a* utility profile $u = \{u_1, \ldots, u_{|N|}\}$ *where* $u_i : \Sigma \to \mathbb{R}$ *is a utility function assigning utility values to strategy profiles.*

Game theory uses *solution concepts* to define which strategy profiles capture rational interactions. Let $\mathscr{G}$ be a class of games with the same strategy profiles Σ. Formally, a solution concept for $\mathscr{G}$ is a function $SC : \mathscr{G} \to \mathscr{P}(\Sigma)$ that, given a game, returns a set of *rational* strategy profiles. Well-known solution concepts include e.g. Nash equilibrium (NE), dominant and undominated strategies, Stackelberg equilibrium, Pareto optimality etc.

Protocols as Games. Let P be a model of a protocol. We will investigate properties of P through the game frame $\Gamma(P)$ in which strategies are *conditional plans* in P, i.e., functions that specify for each choice point which action to take. A set of strategies, one for each agent, uniquely determines a *run* of the protocol, i.e., a sequence of actions that the agents will take. $\Gamma(P)$ takes runs to be the outcomes in the game, and hence maps strategy profiles to runs.

Security protocols are designed to achieve one or more *security requirements* and/or *functionality requirements*. We only consider requirements that can be expressed in terms of single runs having a certain property. We model this by a subset of possible behaviors, called the *objective of the protocol*.

Definition 2.2. *Given a game frame* $\Gamma = (N, \Sigma)$*, an* objective *is a set* $\gamma \subseteq \Sigma$*. We call* γ nontrivial *in* Γ *iff* γ *is neither impossible nor guaranteed in* Γ*, i.e.,* $\emptyset \neq \gamma \neq \Sigma$*.*

3 Incentive-Based Security Analysis

In this section, we give a definition of correctness of security protocols that takes into account rational decisions of agents, based on their incentives.

3.1 Incentive-Based Correctness

As we have pointed out, the requirement that all strategy profiles satisfy the objective might be too strong. Instead, we will require that all *rational* runs satisfy the objective. In case there are no rational runs, all behaviors are equally rational; then, we require that all strategy profiles must satisfy γ.

Definition 3.1. *A protocol model represented as game frame* $\Gamma = (N, \Sigma)$ *with utility profile* u *is* correct with respect to objective γ under solution concept *SC, written* $(\Gamma, u) \models_{SC} \gamma$*, iff:*

$$\begin{cases} SC(\Gamma, u) \subseteq \gamma & \text{if } SC(\Gamma, u) \neq \emptyset \\ \gamma = \Sigma & \text{otherwise.} \end{cases}$$

3.2 Unknown Incentives

Definition 3.1 applies to a protocol when a utility profile is given. However, the exact utility profiles are often unknown. One way out is to require the protocol to be correct for *all possible* utility profiles.

Definition 3.2. *A protocol model represented by game frame* Γ *is* valid *with respect to objective* γ *under solution concept SC (written* $\Gamma \models_{SC} \gamma$*) iff* $(\Gamma, u) \models_{SC} \gamma$ *for all utility profiles* u*.*

It turns out that, under some reasonable assumptions, protocols are only valid for trivial objectives.

Definition 3.3. *Let* $G = (N, \Sigma, (u_1, \ldots, u_n))$*. Let* $\pi = (\pi_1, \ldots, \pi_n)$*, where for all* $i \in N$*,* $\pi_i : \Sigma_i \to \Sigma_i$ *is a permutation on* Σ_i*. We slightly abuse the notation by writing* $\pi((s_1, \ldots, s_n))$ *for* $(\pi_1(s_1), \ldots, \pi_n(s_n))$*. A solution concept is* closed under permutation *iff* $s \in SC((N, \Sigma, (u'_1, \ldots, u'_n)))$ *if and only if* $\pi(s) \in SC((N, \Sigma, (u'_1 \circ \pi_1^{-1}, \ldots, u'_n \circ \pi_n^{-1})))$*.*

Theorem 3.4. *If SC is closed under permutation, then* $\Gamma \models_{SC} \gamma$ *iff* $\gamma = \Sigma$*.*[1]

Thus, correctness for all distributions of incentives is equivalent to correctness in all possible runs.

3.3 Defendability of Protocols

Typical analysis of a protocol implicitly assumes some participants to be aligned with its purpose. E.g., one usually assumes that communicating parties are interested in exchanging a secret without the eavesdropper getting hold of it, that a bank wants to prevent web banking fraud etc. In this section, we formalize this idea by assuming a subset of agents, called the *defenders* of the protocol, to be in favor of its objective. Our new definition of correctness says that a protocol is correct with respect to some objective γ if and only if it is correct with respect to every utility profile in which the preferences of all defenders comply with γ.[2]

Definition 3.5. *A group of agents* $D \subseteq N$ supports *the objective* γ *in game* (N, Σ, u) *iff for all* $i \in D$*, if* $s \in \gamma$ *and* $s' \in \Sigma \setminus \gamma$ *then* $u_i(s) > u_i(s')$*.*

A protocol model represented as game frame Γ *is* defended by agents D*, written* $\Gamma \models_{SC} [D]\gamma$*, iff* $(\Gamma, u) \models_{SC} \gamma$ *for all utility profiles* u *such that* D *supports* γ *in game* (Γ, u)*.*

Clearly, if there are no defenders, then defendability is equivalent to ordinary protocol validity:

Proposition 3.6. *If* Γ *is a game frame and SC is a solution concept, we have that* $\Gamma \models_{SC} [\emptyset]\gamma$ *iff* $\Gamma \models_{SC} \gamma$*.*

If all agents are defenders, any protocol is correct, as long as the solution concept does not select *strongly Pareto-dominated* strategy profiles, and there always is some strategy profile which is rational according to the solution concept.

Definition 3.7. *A solution concept is* weakly Pareto *iff it never selects a strongly Pareto dominated outcome (i.e., such that there exists another outcome strictly preferred by all the players). It is* efficient *iff it never returns the empty set.*

[1] For proofs of all theorems and definitions of auxiliary concepts, we refer to the original paper [9].

[2] There is an analogy of the concept to [1] where "robust" goals are studied, i.e., goals that are achieved as long as a selected subset of agents behaves correctly.

Theorem 3.8. *If Γ is a game frame and SC is an efficient weakly Pareto solution concept then $\Gamma \models_{SC} [N]\gamma$.*

Many solution concepts are both efficient and weakly Pareto, for example: Stackelberg equilibrium, maximum-perfect cooperative equilibrium, backward induction and subgame-perfect Nash equilibrium in perfect information games. On the other hand, Nash equilibrium is neither weakly Pareto nor efficient, and equilibrium in dominant strategies is weakly Pareto but not necessarily efficient.

Clearly, defendability of a protocol is monotonic with respect to the set of defenders. This justifies the following definition.

Definition 3.9. *The* game-theoretic security level *of protocol P is the antichain of minimal sets of defenders that make the protocol correct.*

4 Characterizing Defendability under Nash Equilibrium

In this section, we turn to properties that can be defended if agents' rationality is based on Nash equilibrium or Optimal Nash Equilibrium.

4.1 Defendability under Nash Equilibrium

From Theorem 3.4, we know that no protocol is valid under Nash equilibrium (NE) for any nontrivial objective, since NE is closed under permutation. Do things get better if we assume some agents to be in favor of the security objective? We now look at the extreme variant of the question, i.e., defendability by the grand coalition N. Note that, by monotonicity of defendability wrt the set of defenders D, nondefendability by N implies that the objective is not defendable by any coalition at all.

Our first result in this respect is negative: we show that in every game frame there are nontrivial objectives that are not defendable under NE.

Theorem 4.1. *Let Γ be a game frame with at least two players and at least two strategies per player. Moreover, let γ be a singleton objective, i.e., $\gamma = \{\omega\}$ for some $\omega \in \Sigma$. Then, $\Gamma \not\models_{\mathrm{NE}} [N]\gamma$.*

In particular, the construction from the above proof shows that, as mentioned before, there are cases where the "defending" coalition has a strategy to achieve a goal γ, but there are still rational plays in which the goal is not achieved.

To present the general result that characterizes defendability of security objectives under Nash equilibrium, we need to introduce additional concepts. In what follows, we use $s[t_i/i]$ to denote $(s_1,\ldots,s_{i-1},t_i,s_{i+1},\ldots,s_N)$, i.e., the strategy profile that is obtained from s when player i changes her strategy to t_i.

Definition 4.2. *Let γ be a set of strategy profiles in Γ. The* deviation closure *of γ is defined as $Cl(\gamma) = \{s \in \Sigma \mid \exists i \in N, t_i \in \Sigma_i \,.\, s[t_i/i] \in \gamma\}$.*

$Cl(\gamma)$ extends γ with the strategy profiles that are reachable by unilateral deviations from γ. Thus, $Cl(\gamma)$ can be seen as the closure of γ with the behaviors that are relevant for Nash equilibrium. Moreover, the following notion captures strategy profiles that can be used to construct sequences of unilateral deviations ending up in a cycle.

Definition 4.3. *A* strategic knot *in γ is a subset of strategy profiles $S \subseteq \gamma$ such that there is a permutation $(s^1,\ldots,s^k)$ of S where: (a) for all $1 \le j < k$, $s^{j+1} = s^j[s_i^{j+1}/i]$ for some $i \in N$, and (b) $s^j = s^k[s_i^j/i]$ for some $i \in N$, $j < k$.*

Essentially, this means that every strategy s^{j+1} is obtained from s^j by a unilateral deviation of a single agent. If these deviations are rational (i.e., increase the utility of the deviating agent), then the knot represents a possible endless loop of rational, unilateral deviations which precludes a group of agents from reaching a stable joint strategy. We now state the main result of this section.

Theorem 4.4. *Let Γ be a finite game frame and γ a nontrivial objective in Γ. Then, $\Gamma \models_{\mathrm{NE}} [N]\gamma$ iff $Cl(\gamma) = \Sigma$ and there is a strategy profile in γ that belongs to no strategic knots in γ.*

4.2 Optimal Nash Equilibria

Nash equilibrium is a natural solution concept for a game played repeatedly until the behavior of all players converges to a stable point. For a one-shot game, NE possibly captures convergence of the process of deliberation. It can be argued that, among the available solutions, no player should contemplate those which are strictly worse for everybody when compared to another stable point. This gives rise to the following refinement of Nash equilibrium: $\mathrm{OptNE}(\Gamma, u)$ is the set of *optimal Nash equilibria* in game (Γ, u), defined as those equilibria *that are not strongly Pareto-dominated by another Nash equilibrium*. Defendability by the grand coalition under OptNE has the following simple characterization.

Theorem 4.5. *Let Γ be a finite game frame and γ a nontrivial objective in Γ. Then, $\Gamma \models_{\mathrm{OptNE}} [N]\gamma$ iff there is a strategy profile in γ that belongs to no strategic knots in γ.*

5 Defendability in Mixed Strategies

So far, we considered only deterministic (pure) strategies. It is well known that for many games and solution concepts, rational strategies exist only when taking mixed strategies into account. We now extend our definition of correctness to mixed strategies, i.e., randomized conditional plans represented by probability distributions over pure strategies from Σ_{A_i}. Let $dom(s)$ be the support (domain) of a mixed strategy profile s, i.e., the set of pure strategy profiles that have nonzero probability in s. We extend the notion to sets of mixed strategy profiles in the obvious way. By SC^m we denote the variant of SC in mixed strategy profiles. A protocol is correct in mixed strategies iff all the possible behaviors resulting from a rational (mixed) strategy profile satisfy the goal γ; formally: $\Gamma, u \models^m_{SC} \gamma$ iff $dom(SC^m(\Gamma, u)) \subseteq \gamma$ when $SC^m(\Gamma, u) \neq \emptyset$ and $\gamma = \Sigma_\Gamma$ otherwise. The definitions of protocol validity and defendability in mixed strategies ($\Gamma \models^m_{SC} \gamma$ and $\Gamma \models^m_{SC} [D]\gamma$) are analogous. For defendability in mixed strategies under Nash equilibrium, we have the following, rather pessimistic result.

Theorem 5.1. *Let Γ be a finite game frame, and γ an objective in it. Then, $\Gamma, u \models^m_{\mathrm{NE}} [N]\gamma$ iff $\gamma = \Sigma$.*

On the other hand, it turns out that *optimal Nash equilibrium* yields a simple and appealing characteristics of N-defendable properties. In the following, γ is closed under convex combination of strategies iff every combination of strategies that appear in some profile in γ again is an element of γ.

Theorem 5.2. *$\Gamma \models^m_{\mathrm{OptNE}} [N]\gamma$ iff $\gamma = Conv(\gamma)$, i.e., γ is closed under convex combination of strategies.*

Corollary 5.3. *$\Gamma \models^m_{\mathrm{OptNE}} [N]\gamma$ iff there exist subsets of individual strategies $\chi_1 \subseteq \Sigma_1, \ldots, \chi_{|N|} \subseteq \Sigma_{|N|}$ such that $\gamma = \chi_1 \times \cdots \times \chi_{|N|}$.*

That is, security property γ is defendable by the grand coalition in Γ iff γ can be *decomposed into constraints on individual behavior of particular agents*.

6 Defendability in Non-Injective Games

Normal game frames are usually defined in the literature as $\Gamma = (N, \Sigma, \Omega, o)$, where N, Σ are as before, Ω is the set of (abstract) *outcomes* of the game, and $o : \Sigma \to \Omega$ maps strategy profiles to outcomes. Our analysis so far has been based on the standard assumption that o is a bijection. In other words, there

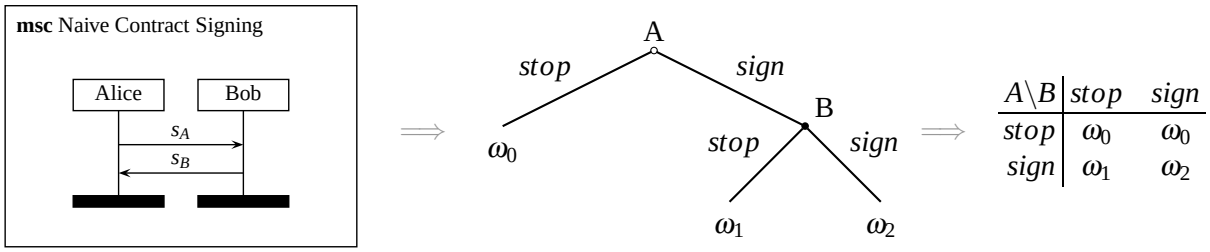


Figure 1: Naive contract-signing: from protocol to EF game to NF game

is a 1-1 relationship between joint behaviors of agents and the outcomes of those behaviors. Then, we can identify outcomes with strategy profiles, and omit the former from the game model. However, the standard construction of a game model from a protocol assumes the outcomes to be *runs* of the protocol. In that case, the assumption does *not* hold; in particular, the mapping is not injective.

Example 6.1. *Consider the naive contract signing protocol in Figure 1. Alice sends her signature to Bob, who responds with his signature. Alice and Bob can stop the protocol at any moment (thereby deviating from the protocol). If we assume runs of the protocol to be the outcomes, this gives rise to an Extensive Form game frame, which can be then transformed to an NF game frame by the canonical construction. Clearly, the mapping between strategy profiles and outcomes is not injective.*

In general NF games, utility functions assign utility values to *outcomes* rather than strategy profiles. That is, $u_i : \Omega \to \mathbb{R}$. Moreover, an objective is assumed to select a *subset of outcomes*. This follows from the methodological assumption that an outcome encapsulates every relevant aspect of the play that has occurred. We observe that the definitions in Section 3 can be lifted to the general case by changing the types of u_i and γ accordingly. However, the results in Sections 4–5 cannot be lifted that easily. Games with non-injective outcome functions require a more general treatment, which we present below.

Definition 6.2. *Given a game frame Γ, we define the* deviation graph of Γ ($Dev(\Gamma)$) *to be the undirected graph where outcomes from Γ are vertices, and edges connect outcomes that are obtained from strategy profiles which differ only in* 1 *individual strategy (thus corresponding to a potential unilateral deviation).*

Moreover, for an objective $\gamma \subseteq \Omega$, we will use $Dev_\gamma(\Gamma)$ to denote the subgraph of $Dev(\Gamma)$ consisting only of the vertices from γ and the edges between them.

It is easy to see that the construction of $Dev(\Gamma)$ and $Dev_\gamma(\Gamma)$ from Γ, γ is straightforward. Let V be a subset of nodes in a graph. We define the *neighborhood of* V, denoted $Neighb(V)$, as V together with all the nodes adjacent to V. We observe that $Neighb(V)$ "implements" the deviation closure of V in $Dev(\Gamma)$. Moreover, ω does not lie on a strategic knot iff its connected component does not include a cycle. This leads to the following, more general, characterizations of defendability (we omit the proofs due to lack of space). Again, we assume that γ is nontrivial, i.e., $\emptyset \neq \gamma \neq \Omega$.

Theorem 6.3. *γ is defended by the grand coalition in Γ under Nash equilibrium iff:*

1. *The neighborhood of γ in in $Dev(\Gamma)$ covers the whole graph ($Neighb(\gamma) = \Omega$), and*
2. *$Dev_\gamma(\Gamma)$ includes at least one connected component with no cycles.*

Theorem 6.4. *γ is defended by the grand coalition in Γ under optimal Nash equilibrium iff $Dev_\gamma(\Gamma)$ includes at least one connected component with no cycles.*

Theorem 6.5. *γ is defended in mixed strategies by the grand coalition in Γ under optimal Nash equilibrium iff γ is obtained by a convex combination of strategies.*

7 Example: The ASW contract-signing protocol

A contract-signing protocol is used by two participants, usually called Alice and Bob, to sign a contract over an asymmetric medium as the internet. The central security properties are *fairness* (Alice should get a signed copy of the contract if and only if Bob gets one), *balancedness* (there is no point in the protocol run where Bob alone can decide whether the contract will be signed or not, i.e., Alice cannot abort the signing anymore but Bob still can abort) and *abuse-freeness* (if balance cannot be achieved, then at least Bob should not be able to prove the fact that he has the above-mentioned strong position in the current state of the protocol to an outsider). The contract-signing protocol P_{ASW}, introduced in [3], uses *commitments*, which are legally binding "declarations of intent" by Alice and Bob to sign the contract. The protocol operates as follows: (1) Alice sends a commitment cm_A to Bob; (2) Bob sends his commitment cm_B to Alice; (3) Alice sends the contract sc_A, digitally signed with her signature, to Bob; (4) Bob sends the contract sc_B, signed with his signature, to Alice.

If one of these messages is not sent by the corresponding signer, the other party may contact the TTP:

- If Alice does not receive a commitment from Bob, she can contact the TTP with an *abort request*, which instructs the TTP to mark this session of the protocol as aborted;
- If Bob does not receive Alice's signature, but has her commitment, he can send a *resolve request* to the TTP, who then issues a *replacement contract* (a document that is legally equivalent to the contract signed by Alice), unless Alice has sent an abort request earlier,
- If Alice does not receive Bob's signature, but has his commitment, she can send a *resolve request* to the TTP as well, which allows her to receive a replacement contract.

It can be shown that the protocol is fair if the TTP is reliable (it will never stop the protocol on its own). It is also balanced if neither Alice nor Bob can drop or delay messages from the other signer to the TTP. Let us denote outcomes by sets of agents who have obtained the signature of the other player. Thus, $\emptyset$ represents the situation where nobody got a signed contract, $\{\mathsf{sign}_A\}$ the situation where Alice obtained Bob's signature but note vice versa, etc. Applying the definitions in Section 3.3, one can show the following. If *SC* is either *Nash equilibrium* or *undominated strategies*, we have:

1. $P_{\mathrm{ASW}} \models_{SC} [\{\mathrm{Bob}\}]\{\emptyset, \{\mathsf{sign}_B\}, \{\mathsf{sign}_A, \mathsf{sign}_B\}\}$,
2. $P_{\mathrm{ASW}} \models_{SC} [\{\mathrm{Alice}\}]\{\emptyset, \{\mathsf{sign}_A\}, \{\mathsf{sign}_A, \mathsf{sign}_B\}\}$.

We now consider the case where TTP is not necessarily reliable. If the TTP can stop the protocol at any time, then the protocol does not guarantee fairness anymore. On the other hand, if Bob wants the protocol to be fair, then he can ensure fairness by simply sending a signed contract to Alice as soon as he receives her signature. Clearly, Alice alone (without an honest TTP to assist her) cannot achieve fairness. Hence the game-theoretic security level of the ASW protocol without reliable TTP is the set $\{\{\mathrm{Bob}\}, \{\mathrm{TTP}\}\}$. This holds for both Nash equilibrium and undominated strategies.

8 Conclusions

We propose a framework for analyzing security protocols (and other interaction protocols), that takes into account the incentives of agents. In particular, we consider a novel notion of *defendability* that guarantees that all the runs of the protocol are correct as long as a given subset of the participants (the "defenders") is in favor of the security property. We have obtained some characterization results for defendability under Nash equilibria and optimal Nash equilibria. In the original paper [9], we also address the computational complexity of the corresponding decision problems, both in the generic case and in some special cases.

In the future, we plan to combine our framework with results for protocol verification using game logics (such as ATL), especially for those solution concepts that can be expressed in that kind of logics.

Acknowledgements. We thank the SR2014 reviewers for their extremely useful remarks. Addressing the fundamental ones was not possible in this extended abstract due to space and time constraints, but we will use them in the journal version of the paper (in preparation).

Wojciech Jamroga acknowledges support of the National Research Fund Luxembourg (FNR) under project GaLOT – INTER/DFG/12/06.

References

[1] T. Ågotnes, W. van der Hoek & M. Wooldridge (2010): *Robust normative systems and a logic of norm compliance*. *Logic Journal of the IGPL* 18(1), pp. 4–30, doi:10.1093/jigpal/jzp070.

[2] G. Asharov, R. Canetti & C. Hazay (2011): *Towards a Game Theoretic View of Secure Computation*. In: *Proceedings of EUROCRYPT*, pp. 426–445, doi:10.1007/978-3-642-20465-4_24.

[3] N. Asokan, V. Shoup & M. Waidner (1998): *Asynchronous protocols for optimistic fair exchange*. In: *Proceedings of the IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society Press, pp. 86–99, doi:10.1109/SECPRI.1998.674826.

[4] L. Buttyán, J. Hubaux & S. Čapkun (2004): *A formal model of rational exchange and its application to the analysis of Syverson's protocol*. *Journal of Computer Security* 12(3,4), pp. 551–587.

[5] K. Chatterjee & V. Raman (2010): *Assume-Guarantee Synthesis for Digital Contract Signing*. *CoRR* abs/1004.2697.

[6] Y. Dodis & T. Rabin (2007): *Cryptography and Game Theory*. In: *Algorithmic Game Theory*, chapter 8, pp. 181–208, doi:10.1017/CBO9780511800481.010.

[7] G. Fuchsbauer, J. Katz & D. Naccache (2010): *Efficient Rational Secret Sharing in Standard Communication Networks*. In: *Proceedings of TCC*, pp. 419–436, doi:10.1007/978-3-642-11799-2_25.

[8] A. Groce & J. Katz (2012): *Fair Computation with Rational Players*. In: *Proceedings of EUROCRYPT*, pp. 81–98, doi:10.1007/978-3-642-29011-4_7.

[9] W. Jamroga, M. Melissen & H. Schnoor (2013): *Defendable Security in Interaction Protocols*. In: *Proceedings of the 16th International Conference on Principles and Practice of Multi-Agent Systems PRIMA 2013*, *LNCS* 8291, Springer, pp. 132–148, doi:10.1007/978-3-642-44927-7_10.

[10] S. Kremer & J. Raskin (2002): *Game Analysis of Abuse-Free Contract Signing*. In: *Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW'02)*, IEEE Computer Society Press, pp. 206–220, doi:10.1109/CSFW.2002.1021817.

[11] S. Kremer & J.-F. Raskin (2003): *A game-based verification of non-repudiation and fair exchange protocols*. *Journal of Computer Security* 11(3), doi:10.1007/3-540-44685-0_37.

[12] T. Moore & R. Anderson (2011): *Economics and Internet Security: a Survey of Recent Analytical, Empirical and Behavioral Research*. Technical Report TR-03-11, Computer Science Group, Harvard University.

[13] M. Osborne & A. Rubinstein (1994): *A Course in Game Theory*. MIT Press.

[14] P. Syverson (1998): *Weakly Secret Bit Commitment: Applications to Lotteries and Fair Exchange*. In: *Proceedings of CSFW*, pp. 2–13, doi:10.1109/CSFW.1998.683149.

Reasoning about Knowledge and Strategies: Epistemic Strategy Logic

Francesco Belardinelli
Laboratoire IBISC – Unversité d'Evry
belardinelli@ibisc.fr

In this paper we introduce Epistemic Strategy Logic (ESL), an extension of Strategy Logic with modal operators for individual knowledge. This enhanced framework allows us to represent explicitly and to reason about the knowledge agents have of their own and other agents' strategies. We provide a semantics to ESL in terms of epistemic concurrent game models, and consider the corresponding model checking problem. We show that the complexity of model checking ESL is not worse than (non-epistemic) Strategy Logic.

1 Introduction

Formal languages to represent and reason about strategies and coalitions are a thriving area of research in Artificial Intelligence and multi-agent system [4, 8, 19]. Recently, a wealth of multi-modal logics have appeared, which allow to formalise complex strategic abilities and behaviours of individual agents and groups [2, 5]. In parallel to these developments, in knowledge representation there is a well-established tradition of extending logics for reactive systems with epistemic operators to reason about the knowledge agents have of systems evolution. These investigations began in the '80s with contributions on combinations of linear- and branching-time temporal logics with multi-agent epistemic languages [9, 10, 6]. Along this line of research, [11] introduced alternating-time temporal epistemic logic (ATEL), an extension of ATL with modalities for individual knowledge. The various flavours of logics of time and knowledge have been successfully applied to the specification of distributed and multi-agent systems in domains as diverse as security protocols, UAVs, web services, and e-commerce, as well as to verification by model checking [7, 16].

In this paper we take inspiration from the works above and pursue further this line of research by introducing Epistemic Strategy Logic, an extension of Strategy Logic (SL) [5, 17] that allows agents to reason about their strategic abilities. The extension here proposed is naive in the sense that it suffers many of the shortcomings of its relative ATEL [12]. Nonetheless, we reckon that it constitutes an excellent starting point to analyse the interaction of knowledge and strategic abilities in a language, such as SL, that explicitly allow for quantification on strategies.

Related Work. This paper builds on previous contributions on Strategy Logic. SL has been introduced in [5] for two-player concurrent game structures (CGS). In [17] the semantics has been extended to a multi-player setting. Also, [17] introduced bind operators for strategies in the syntax. In the present contribution we consider multi-agent CGS in line with [17]. However, we adopt an agent-based perspective and consider agents with possibly different actions and protocols [6]. Also, our language do not include bind operators to avoid the formal machinery associated with these operators. We leave such an extension for future and more comprehensive work. Finally, the model checking results in Section 4 are inspired by and use techniques from [17].

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 27–33, doi:10.4204/EPTCS.146.4

Even though to our knowledge no epistemic extension of SL has been proposed yet, the interaction between knowledge and strategic reasoning has been studied extensively, especially in the context of alternating-time temporal logic. An extension of ATL with knowledge operators, called ATEL, was put forward in [11], and immediately imperfect information variants of this logic were considered in [14], which introduces alternating-time temporal observational logic (ATOL) and ATEL-R*, as well as uniform strategies. Notice that [14] also analyses the distinction between *de re* and *de dicto* knowledge of strategies; this distinction will also be considered later on in the context of Epistemic Strategy Logic. Further, [13] enriches ATL with a constructive notion of knowledge. As regards (non-epistemic) ATL, more elaborate notions of strategy have been considered. In [1] commitment in strategies has been analysed; while [15] introduced a notion of "feasible" strategy. In future work it might be worth exploring to what extent the theoretical results available for the various flavours of ATEL transfer to ESL.

Scheme of the paper. In Section 2 we introduce the epistemic concurrent game models (ECGM), which are used in Section 3 to provide a semantics to Epistemic Strategy Logic (ESL). In Section 4 we consider the model checking problem for this setting and state the corresponding complexity results. Finally, in Section 5 we discuss the results and point to future research. For reasons of space, all proofs are omitted. An extended version of this paper with complete proofs is available [3].

2 Epistemic Concurrent Game Models

In this section we present the epistemic concurrent game models (ECGM), an extension of concurrent game structures [2, 11], starting with the notion of *agent*.

Definition 1 (Agent) *An* agent *is a tuple* $i = \langle L_i, Act_i, Pr_i \rangle$ *such that* (i) L_i *is the set of* local states $l_i, l_i', \ldots$; (ii) Act_i *is the finite set of* actions $\sigma_i, \sigma_i', \ldots$; *and* (iii) $Pr_i : L_i \mapsto 2^{Act_i}$ *is the* protocol function.

Intuitively, each agent i is situated in some local state $l_i \in L_i$, representing her local information, and performs the actions in Act_i according to the protocol function Pr_i [6]. Differently from [17], we assume that agents have possibly different actions and protocols. To formally describe the interactions between agents, we introduce their synchronous composition. Given a set AP of atomic propositions and a set $Ag = \{i_0, \ldots, i_n\}$ of agents, we define the set L of global states $s, s', \ldots$ (resp. the set Act of joint actions $\sigma, \sigma', \ldots$) as the cartesian product $L_0 \times \ldots \times L_n$ (resp. $Act_0 \times \ldots \times Act_n$). In what follows we denote the jth component of a tuple t as t_j or, equivalently, as $t(j)$.

Definition 2 (ECGM) *Given a set* $Ag = \{i_0, \ldots, i_n\}$ *of agents* $i = \langle L_i, Act_i, Pr_i \rangle$, *an* epistemic concurrent game model *is a tuple* $\mathscr{P} = \langle Ag, s_0, \tau, \pi \rangle$ *such that* (i) $s_0 \in L$ *is the* initial global state; (ii) $\tau : L \times Act \mapsto L$ *is the* global transition function, *where* $\tau(s, \sigma)$ *is defined iff* $\sigma_i \in Pr_i(l_i)$ *for every* $i \in Ag$; *and* (iii) $\pi : AP \mapsto 2^L$ *is the* interpretation function *for atomic propositions in AP.*

The transition function τ describes the evolution of the ECGM from the initial state s_0. We now introduce some notation that will be used in the rest of the paper. The *transition relation* $\rightarrow$ on global states is defined as $s \rightarrow s'$ iff there exists $\sigma \in Act$ s.t. $\tau(s, \sigma) = s'$. A *run* λ from a state s, or s-run, is an infinite sequence $s^0 \rightarrow s^1 \rightarrow \ldots$, where $s^0 = s$. For $n, m \in \mathbb{N}$, with $n \leq m$, we define $\lambda(n) = s^n$ and $\lambda[n, m] = s^n, s^{n+1}, \ldots, s^m$. A state s' is *reachable from* s if there exists an s-run λ s.t. $\lambda(i) = s'$ for some $i \geq 0$. We define S as the set of states reachable from the initial state s_0. Further, let $\sharp$ be a placeholder for arbitrary individual actions. Given a subset $A \subseteq Ag$ of agents, an *A-action* σ_A is an $|Ag|$-tuple s.t. *(i)* $\sigma_A(i) \in Act_i$ for $i \in A$, and *(ii)* $\sigma_A(j) = \sharp$ for $j \notin A$. Then, Act_A is the set of all A-actions and $D_A(s) = \{\sigma_A \in Act_A \mid \text{for every } i \in A, \sigma_i \in Pr_i(l_i)\}$ is the set of all A-actions enabled at $s = \langle l_0, \ldots, l_n \rangle$. A joint action σ *extends* an A-action σ_A, or $\sigma_A \sqsubseteq \sigma$, iff $\sigma_A(i) = \sigma(i)$ for all $i \in A$. The *outcome* $out(s, \sigma_A)$

of action σ_A *at state* s is the set of all states s' s.t. there exists a joint action $\sigma \sqsupseteq \sigma_A$ and $\tau(s,\sigma) = s'$. Finally, two global states $s = \langle l_0,\ldots,l_n\rangle$ and $s' = \langle l'_0,\ldots,l'_n\rangle$ are *indistinguishable* for agent i, or $s \sim_i s'$, iff $l_i = l'_i$ [6].

3 Epistemic Strategy Logic

We now introduce Epistemic Strategy Logic as a specification language for ECGM. Hereafter we consider a set Var_i of strategy variables $x_i, x'_i, \ldots$, for every agent $i \in Ag$.

Definition 3 (ESL) *For $p \in AP$, $i \in Ag$ and $x_i \in Var_i$, the ESL formulas ϕ are defined in BNF as follows:*

$$\phi \quad ::= \quad p \mid \neg\phi \mid \phi \rightarrow \phi \mid X\phi \mid \phi U \phi \mid K_i\phi \mid \exists x_i \phi$$

The language ESL is an extension of the Strategy Logic in [5] to a multi-agent setting, including an epistemic operator K_i for each $i \in Ag$. Alternatively, ESL can be seen as the epistemic extension of the Strategy Logic in [17], minus the bind operator. We do not consider bind operators in ESL for ease of presentation. The ESL formula $\exists x_i \phi$ is read as "agent i has some strategy to achieve ϕ". The interpretation of LTL operators X and U is standard. The epistemic formula $K_i\phi$ intuitively means that "agent i knows ϕ". The other propositional connectives and LTL operators, as well as the strategy operator $\forall$, can be defined as standard. Also, notice that we can introduce the *nested-goal* fragment ESL[NG], the *boolean-goal* fragment ESL[BG], and the *one-goal* fragment ESL[1G] in analogy to SL [17]. Further, the *free* variables $fr(\phi) \subseteq Ag$ of an ESL formula ϕ are inductively defined as follows:

$$\begin{array}{lcl} fr(p) & = & \emptyset \\ fr(\neg\phi) = fr(K_i\phi) & = & fr(\phi) \\ fr(\phi \rightarrow \phi') & = & fr(\phi) \cup fr(\phi') \\ fr(X\phi) = fr(\phi U \phi') & = & Ag \\ fr(\exists x_i \phi) & = & fr(\phi) \setminus \{i\} \end{array}$$

A *sentence* is a formula ϕ with $fr(\phi) = \emptyset$, and the set $bnd(\phi)$ of bound variables is defined as $Ag \setminus fr(\phi)$.

To provide a semantics to ESL formulas in terms of ECGM, we introduce the notion of strategy.

Definition 4 (Strategy) *Let γ be an ordinal s.t. $1 \leq \gamma \leq \omega$ and $A \subseteq Ag$ a set of agents. A γ-*recall *A-*strategy *is a function $F_A[\gamma] : \bigcup_{1\leq n<1+\gamma} S^n \mapsto \bigcup_{s\in S} D_A(s)$ s.t. $F_A[\gamma](\kappa) \in D_A(last(\kappa))$ for every $\kappa \in \bigcup_{1\leq n<1+\gamma} S^n$, where $1+\gamma = \gamma$ for $\gamma = \omega$ and $last(\kappa)$ is the last element of κ.*

Hence, a γ-recall A-strategy returns an enabled A-action for every sequence $\kappa \in \bigcup_{1\leq n<1+\gamma} S^n$ of states of length at most γ. Notice that for $A = \{i\}$, $F_A[\gamma]$ can be seen as a function from $\bigcup_{1\leq n<1+\gamma} S^n$ to Act_i s.t. $F_A[\gamma](\kappa) \in Pr_i(last(\kappa))$ for $\kappa \in \bigcup_{1\leq n<1+\gamma} S^n$. In what follows we write $F_i[\gamma]$ for $F_{\{i\}}[\gamma]$. Then, for $A = \{i_0,\ldots,i_m\} \subseteq Ag$, $F_A[\gamma]$ is equal to $F_{i_0}[\gamma] \times \ldots \times F_{i_m}[\gamma]$, where for every $\kappa \in \bigcup_{1\leq n<1+\gamma} S^n$, $(F_{i_0}[\gamma] \times \ldots \times F_{i_m}[\gamma])(\kappa)$ is defined as the set of actions $\sigma \in \bigcup_{s\in S} D_A(s)$ s.t. $\sigma_i = F_i[\gamma](\kappa)$ if $i \in A$, $\sigma_i = \sharp$ otherwise. Therefore, a group strategy is the composition of its members' strategies. Further, the *outcome* of strategy $F_A[\gamma]$ at state s, or $out(s, F_A[\gamma])$, is the set of all s-runs λ s.t. $\lambda(i+1) \in out(\lambda(i), F[\gamma](\lambda[j,i]))$ for all $i \geq 0$ and $j = max(i-\gamma+1, 0)$. Depending on γ we can define positional strategies, strategies with perfect recall, etc. [8]. However, these different choices do not affect the following results, so we assume that γ is fixed and omit it. Moreover, by Def. 4 it is apparent that agents have perfect information, as their strategies are determined by global states [4]; we leave contexts of imperfect information for future research.

Now let χ be an assignment that maps each agent $i \in Ag$ to an i-strategy F_i. For $Ag = \{i_0,\ldots,i_n\}$, we denote $\chi(i_0) \times \ldots \times \chi(i_n)$ as F^χ, that is, the Ag-strategy s.t. for every $\kappa \in \bigcup_{1\leq n<1+\gamma} S^n$, $F^\chi(\kappa) = \sigma \in$

$\bigcup_{s\in S} D_{Ag}(s)$ iff $\sigma_i = \chi(i)(\kappa)$ for every $i \in Ag$. Since $|out(s,F^\chi)| = 1$, we simply write $\lambda = out(s,F^\chi)$. Also, $\chi^i_{F_i}$ denotes the assignment s.t. *(i)* for all agents j different from i, $\chi^i_{F_i}(j) = \chi(j)$, and *(ii)* $\chi^i_{F_i}(i) = F_i$.

Definition 5 (Semantics of ESL) *We define whether an ECGM $\mathscr{P}$* satisfies *a formula φ at state s according to assignment χ, or $(\mathscr{P},s,\chi) \models \varphi$, as follows (clauses for propositional connectives are straightforward and thus omitted):*

$(\mathscr{P},s,\chi) \models p$ *iff* $s \in \pi(p)$
$(\mathscr{P},s,\chi) \models X\psi$ *iff* *for* $\lambda = out(s,F^\chi)$, $(\mathscr{P},\lambda(1),\chi) \models \psi$
$(\mathscr{P},s,\chi) \models \psi U \psi'$ *iff* *for* $\lambda = out(s,F^\chi)$ *there is* $k \geq 0$ *s.t.* $(\mathscr{P},\lambda(k),\chi) \models \psi'$ *and* $0 \leq j < k$ *implies* $(\mathscr{P},\lambda(j),\chi) \models \psi$
$(\mathscr{P},s,\chi) \models K_i\psi$ *iff* *for all* $s \in S$, $s \sim_i s'$ *implies* $(\mathscr{P},s',\chi) \models \psi$
$(\mathscr{P},s,\chi) \models \exists x_i \psi$ *iff* *there exists an i-strategy* F_i *s.t.* $(\mathscr{P},s,\chi^i_{F_i}) \models \psi$

An ESL formula φ is *satisfied* at state s, or $(\mathscr{P},s) \models \varphi$, if $(\mathscr{P},s,\chi) \models \varphi$ for all assignments χ; φ is *true* in $\mathscr{P}$, or $\mathscr{P} \models \varphi$, if $(\mathscr{P},s_0) \models \varphi$. The satisfaction of formulas is independent from bound variables, that is, $\chi(fr(\phi)) = \chi'(fr(\phi))$ implies that $(\mathscr{P},s,\chi) \models \phi$ iff $(\mathscr{P},s,\chi') \models \phi$. In particular, the satisfaction of sentences is independent from assignments.

We can now state the model checking problem for ESL.

Definition 6 (Model Checking Problem) *Given an ECGM $\mathscr{P}$ and an ESL formula ϕ, determine whether there exists an assignment χ s.t. $(\mathscr{P},s_0,\chi) \models \phi$.*

Notice that, if $y_1,\ldots,y_m$ is an enumeration of $fr(\phi)$, then the model checking problem amounts to check whether $\mathscr{P} \models \exists y_1,\ldots,\exists y_m\phi$, where $\exists y_1,\ldots,\exists y_m\phi$ is a sentence.

Hereafter we illustrate the formal machinery introduced thus far with a toy example.

Example. We introduce a turn-based ECGM with two agents, A and B. First, A secretly chooses between 0 and 1. Then, at the successive stage, B also chooses between 0 and 1. The game is won by agent A if the values provided by the two agents coincide, otherwise B wins. We formally describe this toy game starting with agents A and B. Specifically, A is the tuple $\langle L_A, Act_A, Pr_A\rangle$, where *(i)* $L_A = \{\varepsilon_A, 0, 1\}$; *(ii)* $Act_A = \{set(0), set(1), skip\}$; and *(iii)* $Pr_A(\varepsilon_A) = \{set(0), set(1)\}$ and $Pr_A(0) = Pr_A(1) = \{skip\}$. Further, agent B is defined as the tuple $\langle L_B, Act_B, Pr_B\rangle$, where $L_B = \{\varepsilon_B, \lambda, 0, 1\}$; $Act_B = \{wait, set(0), set(1), skip\}$; $Pr_B(\varepsilon_B) = \{wait\}$, $Pr_B(\lambda) = \{set(0), set(1)\}$ and $Pr_B(0) = Pr_B(1) = \{skip\}$. The intuitive meaning of local states, actions and protocol functions is clear. Also, we consider the set $AP = \{win_A, win_B\}$ of atomic propositions, which intuitively express that agent A (resp. B) has won the game. We now introduce the ECGM $\mathscr{Q}$, corresponding to our toy game, as the tuple $\langle Ag, s_0, \tau, \pi\rangle$, where *(i)* $s_0 = (\varepsilon_A, \varepsilon_B)$; *(ii)* the transition function τ is given as follows for $i, j \in \{0,1\}$:

- $\tau((\varepsilon_A,\varepsilon_B),(set(i),wait)) = (i,\lambda)$
- $\tau((i,\lambda),(skip,set(j))) = (i,j)$
- $\tau((i,j),(skip,skip)) = (\varepsilon_A,\varepsilon_B)$

and *(iii)* $\pi(win_A) = \{(0,0),(1,1)\}$, $\pi(win_B) = \{(1,0),(0,1)\}$. Notice that we suppose that our toy game, represented in Fig. 1, is non-terminating.

Now, we check whether the following ESL specifications hold in the ECGM $\mathscr{Q}$.

$$\mathscr{Q} \models \forall x_A\, X\, K_B\, \exists y_B\, X\, win_B \quad (1)$$
$$\mathscr{Q} \not\models \forall x_A\, X\, \exists y_B\, K_B\, X\, win_B \quad (2)$$
$$\mathscr{Q} \models \forall x_A\, X\, K_B\, K_A\, \exists y_B\, X\, win_A \quad (3)$$
$$\mathscr{Q} \models \forall x_A\, X\, K_B\, \exists y_B\, K_A\, X\, win_A \quad (4)$$

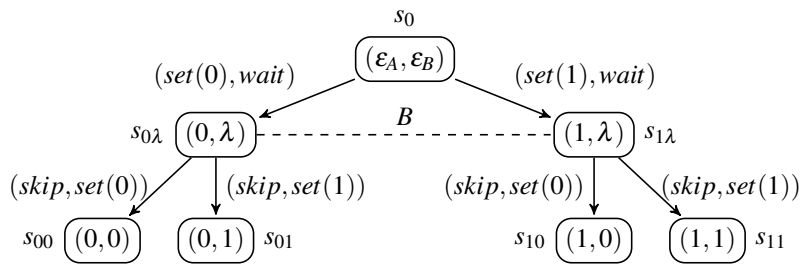


Figure 1: the ECGM $\mathscr{Q}$. Transitions from s_{00}, s_{01}, s_{10}, and s_{11} to s_0 are omitted.

Intuitively, (1) expresses the fact that at the beginning of the game, independently from agent A's move, at the next step agent B knows that there exists a move by which she can enforce her victory. That is, if agent A chose 0 (resp. 1), then B can choose 1 (resp. 0). However, B only knows that there exists a move, but she is not able to point it out. In fact, (2) does not hold, as B does not know which specific move A chose, so she is not capable of distinguishing states $s_{0\lambda}$ and $s_{1\lambda}$. Moreover, by (3) B knows that A knows that there exists a move by which B can let A win. Also, by (4) this move is known to A, as it is the B-move matching A's move.

Indeed, in ESL it is possible to express the difference between *de re* and *de dicto* knowledge of strategies. One of the first contributions to tackle this issue formally is [14]. Formula (1) expresses agent B's *de dicto* knowledge of strategy y_B; while (2) asserts *de re* knowledge of the same strategy. Similarly, in (3) agent A has *de re* knowledge of strategy y_B; while (4) states that agent A knows the same strategy *de dicto*. The *de re*/*de dicto* distinction is of utmost importance as, as shown above, having a *de dicto* knowledge of a strategy does not guarantee that an agent is actually capable of performing the associated sequence of actions. Ideally, in order to have an effective strategy, agents must know it *de re*.

4 Model Checking ESL

In this section we consider the complexity of the model checking problem for ESL. In Section 4.1 and 4.2 we provide the lower and upper bound respectively. For reasons of space, we do not provide full proofs, but only give the most important partial results. We refer to [3] for detailed definitions and complete proofs.

For an ESL formula ϕ we define $alt(\phi)$ as the maximum number of alternations of quantifiers $\exists$ and $\forall$ in ϕ. Then, ESL[k-alt] is the set of ESL formulas ϕ with $alt(\phi)$ equal to or less than k.

4.1 Lower Bound

In this section we prove that model checking ESL formulas is non-elementary-hard. Specifically, we show that for ESL formulas with maximum alternation k the model checking problem is k-EXPSPACE-hard. The proof strategy is similar to [17], namely, we reduce the satisfiability problem for quantified propositional temporal logic (QPTL) to ESL model checking. However, the reduction applied is different, as ESL does not contain the bind operator used in [17].

We first state that the satisfiability problem for QPTL sentences built on a finite set $AP = \{p_0, \ldots, p_n\}$ of atomic propositions can be reduced to model checking ESL sentences on a ECGM $\mathscr{Q}$ of fixed size on $|AP|$, albeit exponential.

Lemma 1 (QPTL Reduction) *Let $AP = \{p_0, \ldots, p_n\}$ be a finite set of atomic propositions. There exists an ECGM $\mathscr{Q}$ on AP s.t. for every QPTL[k-alt] sentence ϕ on AP, there exists an ESL[k-alt] sentence $\overline{\phi}$ s.t. ϕ is satisfiable iff $\mathscr{Q} \models \overline{\phi}$.*

By this result and the fact that the satisfiability problem for QPTL[k-alt] is k-EXPSPACE-hard [17], we can derive the lower bound for model checking ESL[k-alt].

Theorem 2 (Hardness) *The model checking problem for ESL[k-alt] is k-EXPSPACE-hard.*

In particular, it follows that ESL model checking is non-elementary-hard.

4.2 Upper Bound

In this section we extend to Epistemic Strategy Logic the model checking procedure for SL in [17], which is based on alternating tree automata (ATA) [18]. We state the following result, which extends Lemma 5.6 in [17].

Lemma 3 *Let $\mathscr{P}$ be an ECGM and ϕ an ESL formula. Then, there exists an alternating tree automaton $\mathscr{A}^{\phi}_{\mathscr{P}}$ s.t. for every state $s \in S$ and assignment χ, we have that $(\mathscr{P}, s, \chi) \models \phi$ iff the* assignment-state encoding *$\mathscr{T}^{\chi}_{s}$ belongs to the language $\mathscr{L}(\mathscr{A}^{\phi}_{\mathscr{P}})$.*

The following result corresponds to Theorem 5.4 in [17].

Theorem 4 (ATA Direction Projection) *Let $\mathscr{A}^{\phi}_{\mathscr{P}}$ be the ATA in Lemma 3, and $s \in S$ a distinguished state. Then, there exists a non-deterministic ATA $\mathscr{N}^{\phi}_{\mathscr{P},s}$ s.t. for all $Act_{fr(\phi)}$-labelled Δ-tree $\mathscr{T} = \langle T, V \rangle$, we have that $\mathscr{T} \in \mathscr{L}(\mathscr{N}^{\phi}_{\mathscr{P},s})$ iff $\mathscr{T}' \in \mathscr{L}(\mathscr{A}^{\phi}_{\mathscr{P}})$, where $\mathscr{T}'$ is the $(Act_{fr(\phi)} \times S)$-labelled Δ-tree $\langle T, V' \rangle$ s.t. $V'(x) = (V(x), last(\kappa_{s \cdot x}))$.*

Then, by using Lemma 3 and Theorem 4 we can state the following result.

Theorem 5 *Let $\mathscr{P}$ be an ECGM, s a state in $\mathscr{P}$, χ an assignment, and ϕ an ESL formula. The non-deterministic ATA $\mathscr{N}^{\phi}_{\mathscr{P},s}$ in Theorem 4 is such that $(\mathscr{P}, s, \chi) \models \phi$ iff $\mathscr{L}(\mathscr{N}^{\phi}_{\mathscr{P},s}) \neq \emptyset$.*

We can finally state the following extension to Theorem 5.8 in [17], which follows from the fact that the non-emptyness problem for alternating tree automata is non-elementary in the size of the formula.

Theorem 6 (Completeness) *The model checking problem for ESL is PTIME-complete w.r.t. the size of the model and NON-ELEMENTARYTIME w.r.t. the size of the formula.*

We remark that Theorem 6 can be used to show that the model checking problem for the nested-goal fragment ESL[NG] is PTIME-complete w.r.t. the size of the model and $(k+1)$-EXPTIME w.r.t. the maximum alternation k of a formula. We conclude that the complexity of model checking ESL is not worse than the corresponding problem for the Strategy Logic in [17].

5 Conclusions

In this paper we introduced Epistemic Strategy Logic, an extension of Strategy Logic [17] with modalities for individual knowledge. We provided this specification language with a semantics in terms of epistemic concurrent game models (ECGM), and analysed the corresponding model checking problem. A number of developments for the proposed framework are possible. Firstly, the model checking problem for the nested-goal, boolean-goal, and one-goal fragment of SL has lower complexity. It is likely that similar results hold also for the corresponding fragments of ESL. Secondly, we can extend ESL with modalities for group knowledge, such as common and distributed knowledge. Thirdly, we can consider various assumptions on ECGM, for instance perfect recall, no learning, and synchronicity. The latter two extensions, while enhancing the expressive power of the logic, are also likely to increase the complexity of the model checking and satisfiability problems.

References

[1] Thomas Agotnes, Valentin Goranko & Wojciech Jamroga (2007): *Alternating-time Temporal Logics with Irrevocable Strategies*. In: *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '07, ACM, New York, NY, USA, pp. 15–24, doi:10.1145/1324249.1324256.

[2] Rajeev Alur, Thomas A. Henzinger & Orna Kupferman (2002): *Alternating-time temporal logic*. *J. ACM* 49(5), pp. 672–713, doi:10.1145/585265.585270.

[3] Francesco Belardinelli (2014): *Reasoning about Knowledge and Strategies: Epistemic Strategy Logic*. Technical Report, Universit d'Evry, Laboratoire IBISC. Available at `https://www.ibisc.univ-evry.fr/~belardinelli/Documents/sr2014.pdf`.

[4] Nils Bulling, Jurgen Dix & Wojciech Jamroga (2010): *Model Checking Logics of Strategic Ability: Complexity**. In Mehdi Dastani, Koen V. Hindriks & John-Jules Charles Meyer, editors: *Specification and Verification of Multi-agent Systems*, Springer US, pp. 125–159, doi:10.1007/978-1-4419-6984-2.

[5] Krishnendu Chatterjee, Thomas A. Henzinger & Nir Piterman (2010): *Strategy logic*. *Inf. Comput.* 208(6), pp. 677–693, doi:10.1016/j.ic.2009.07.004.

[6] Ronald Fagin, Joseph Y. Halpern, Yoram Moses & Moshe Y. Vardi (1995): *Reasoning About Knowledge*. The MIT Press.

[7] Peter Gammie & Ron van der Meyden (2004): *MCK: Model Checking the Logic of Knowledge*. In Rajeev Alur & Doron Peled, editors: *CAV*, *Lecture Notes in Computer Science* 3114, Springer, pp. 479–483, doi:10.1007/978-3-540-27813-9_41.

[8] Valentin Goranko & Wojciech Jamroga (2004): *Comparing Semantics of Logics for Multi-Agent Systems*. *Synthese* 139(2), pp. 241–280, doi:10.1023/B:SYNT.0000024915.66183.d1.

[9] Joseph Y. Halpern & Moshe Y. Vardi (1986): *The Complexity of Reasoning about Knowledge and Time: Extended Abstract*. In Juris Hartmanis, editor: *STOC*, ACM, pp. 304–315, doi:10.1145/12130.12161.

[10] Joseph Y. Halpern & Moshe Y. Vardi (1989): *The Complexity of Reasoning about Knowledge and Time. I. Lower Bounds*. *J. Comput. Syst. Sci.* 38(1), pp. 195–237, doi:10.1016/0022-0000(89)90039-1.

[11] Wiebe van der Hoek & Michael Wooldridge (2003): *Cooperation, Knowledge, and Time: Alternating-time Temporal Epistemic Logic and its Applications*. *Studia Logica* 75(1), pp. 125–157, doi:10.1023/A:1026185103185.

[12] Wojciech Jamroga (2004): *Some Remarks on Alternating Temporal Epistemic Logic*. In: *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pp. 133–140.

[13] Wojciech Jamroga & Thomas Ågotnes (2007): *Constructive knowledge: what agents can achieve under imperfect information*. *Journal of Applied Non-Classical Logics* 17(4), pp. 423–475, doi:10.3166/jancl.17.423-475.

[14] Wojciech Jamroga & Wiebe van der Hoek (2004): *Agents that Know How to Play*. *Fundam. Inform.* 63(2-3), pp. 185–219. Available at `http://iospress.metapress.com/content/xh738axb47d8rchf/`.

[15] Geert Jonker (2003): *Feasible strategies in Alternating-time Temporal Epistemic Logic*. Master's thesis, University of Utrecht.

[16] Alessio Lomuscio, Hongyang Qu & Franco Raimondi (2009): *MCMAS: A Model Checker for the Verification of Multi-Agent Systems*. In A. Bouajjani & O. Maler, editors: *CAV*, *Lecture Notes in Computer Science* 5643, Springer, pp. 682–688, doi:10.1007/978-3-642-02658-4_55.

[17] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2011): *Reasoning About Strategies*. *CoRR* abs/1112.6275. Available at `http://arxiv.org/abs/1112.6275`.

[18] David E. Muller & Paul E. Schupp (1987): *Alternating Automata on Infinite Trees*. *Theor. Comput. Sci.* 54, pp. 267–276, doi:10.1016/0304-3975(87)90133-2.

[19] Marc Pauly (2002): *A Modal Logic for Coalitional Power in Games*. *J. Log. Comput.* 12(1), pp. 149–166, doi:10.1093/logcom/12.1.149.

An Epistemic Strategy Logic (Extended Abstract)

Xiaowei Huang
The University of New South Wales

Ron van der Meyden
The University of New South Wales

The paper presents an extension of temporal epistemic logic with operators that quantify over strategies. The language also provides a natural way to represent what agents would know were they to be aware of the strategies being used by other agents. Some examples are presented to motivate the framework, and relationships to several variants of alternating temporal epistemic logic are discussed. The computational complexity of model checking the logic is also characterized.

Introduction

There are many subtle issues concerning agent knowledge in settings where multiple agents act strategically. In the process of understanding these issues, there has been a proliferation of modal logics dealing with epistemic reasoning in strategic settings, e.g., [14, 12, 9]. The trend has been for these logics to contain large numbers of operators, each of which combines several different concerns, such as the existence of strategies, and knowledge that groups of agents may have about these strategies. We argued in a previous work [8] that epistemic temporal logic already has the expressiveness required for many applications of epistemic strategy logics, provided that one works in a semantic framework in which strategies are explicitly rather than (as in most alternating temporal epistemic logics) implicitly represented, and makes the minor innovation of including new agents whose local states correspond to the strategies being used by other agents. This gives a more compositional basis for epistemic strategic logic. In the case of imperfect recall strategies and knowledge operators, and a CTL* temporal basis, this leads to a temporal epistemic strategy logic with a PSPACE complete model checking problem.

However, some of our results in [8] required a restriction to cases not involving a common knowledge operator, because certain notions could not be expressed. In the present paper, we develop a remedy for this weakness. We propose an epistemic strategy logic which, like [3, 11], supports explicit naming and quantification over strategies. However we achieve this in a slightly more general way: we first generalize temporal epistemic logic to include operators for quantification over global states and reference to their components, and then apply this generalization to a system that includes strategies encoded in the global states and references these using the "strategic" agents of [8] . The resulting framework can express many of the subtly different notions that have been the subject of proposals for alternating temporal epistemic logics. In particular, it generalizes the expressiveness of the logic in [8] but is able to also deal with the common knowledge issues that restricted the scope of that work. The new logic retains the pleasant compositional capabilities of the prior proposal. There is, however, a computational cost to the generalization: the complexity of model checking for the extended language based on CTL* is EXPSPACE-complete, a jump over the previous PSPACE-completeness result. However, for the fragment based on CTL temporal operators, model checking remains PSPACE-complete.

An extended temporal epistemic logic

We extend temporal epistemic logic with a set of variables *Var*, an operator $\exists x.$ and a construct $\mathsf{e}_i(x)$, where x is a variable and $\exists x.\phi$ says, intuitively, that there exists in the system a global state x such that ϕ

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 35–41, doi:10.4204/EPTCS.146.5

holds at the current point, and $\mathsf{e}_i(x)$ says that agent i has the same local state at the current point and at the global state x. Let $Prop$ be a set of atomic propositions and let Ags be a set of agents. Formally, the language ETLK($Ags, Prop, Var$) has syntax given by the grammar:

$$\phi \equiv p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid A\phi \mid \bigcirc\phi \mid \phi_1 U \phi_2 \mid \exists x.\phi \mid \mathsf{e}_i(x) \mid D_G\phi \mid C_G\phi$$

where $p \in Prop$, $x \in Var$, $i \in Ags$, and $G \subseteq Ags$. The construct $D_G\phi$ expresses that agents in G have distributed knowledge of ϕ, i.e., could deduce ϕ if they pooled their information, and $C_G\phi$ says that ϕ is common knowledge to group G. The temporal formulas $\bigcirc\phi$, $\phi_1 U \phi_2$, $A\phi$ have the standard meanings from CTL^*, i.e., $\bigcirc\phi$ says that ϕ holds at the next moment of time, $\phi_1 U \phi_2$ says that ϕ_1 holds until ϕ_2 does, and $A\phi$ says that ϕ holds in all possible evolutions from the present situation. Other operators can be obtained in the usual way, e.g., $\phi_1 \wedge \phi_2 = \neg(\neg\phi_1 \vee \neg\phi_2)$, $\Diamond\phi = (trueU\phi)$, $\Box\phi = \neg\Diamond\neg\phi$, etc. The universal form $\forall x.\phi = \neg\exists x.\neg\phi$ expresses that ϕ holds for all global states x. For an agent $i \in Ags$, we write $K_i\phi$ for $D_{\{i\}}\phi$; this expresses that agent i knows the fact ϕ. The notion of everyone in group G knowing ϕ can then be expressed as $E_G\phi = \bigwedge_{i\in G} K_i\phi$. We write $\mathsf{e}_G(x)$ for $\bigwedge_{i\in G} \mathsf{e}_i(x)$. This says that at the current point, the agents in G have the same local state as they do at the global state named by variable x.

The semantics of ETLK($Ags, Prop, Var$) builds straightforwardly on the following definitions used in the standard semantics for temporal epistemic logic [4]. Consider a system for a set of agents Ags. A *global state* is an element of the set $\mathcal{G} = L_e \times \Pi_{i\in Ags} L_i$, where L_e is a set of states of the environment and each L_i is a set of *local states* for agent i. A *run* is a mapping $r : \mathbb{N} \to \mathcal{G}$ giving a global state at each moment of time. A *point* is a pair (r,m) consisting of a run r and a time m. An *interpreted system* is a pair $\mathcal{I} = (\mathcal{R}, \pi)$, where $\mathcal{R}$ is a set of runs and π is an *interpretation*, mapping each point (r,m) with $r \in \mathcal{R}$ to a subset of $Prop$. For $n \leq m$, write $r[n\ldots m]$ for the sequence $r(n)r(n+1)\ldots r(m)$. Elements of $\mathcal{R} \times \mathbb{N}$ are called the *points* of $\mathcal{I}$. For each agent $i \in Ags \cup \{e\}$, we write $r_i(m)$ for the component of $r(m)$ in L_i, and then define an equivalence relation on points by $(r,m) \sim_i (r',m')$ if $r_i(m) = r'_i(m')$. We also define $\sim_G^D \equiv \cap_{i\in G} \sim_i$, and $\sim_G^E \equiv \cup_{i\in G} \sim_i$, and $\sim_G^C \equiv (\cup_{i\in G} \sim_i)^*$ for $G \subseteq Ags$. We take $\sim_\emptyset^D$ to be the universal relation on points, and $\sim_\emptyset^E$ and $\sim_\emptyset^C$ to be the identity relation.

To extend this semantic basis for temporal epistemic logic to a semantics for ETLK($Ags, Prop, Var$), we just need need to add a construct that interprets variables as global states. A *context* for an interpreted system $\mathcal{I}$ is a mapping Γ from Var to global states occurring in $\mathcal{I}$. We write $\Gamma[g/x]$ for the context Γ' with $\Gamma'(x) = g$ and $\Gamma'(y) = \Gamma(y)$ for all variables $y \neq x$. The semantics of the language ETLK is given by a relation $\Gamma, \mathcal{I}, (r,m) \models \phi$, representing that formula ϕ holds at point (r,m) of the interpreted system $\mathcal{I}$, relative to context Γ. This is defined inductively on the structure of the formula ϕ, as follows:

- $\Gamma, \mathcal{I}, (r,m) \models p$ if $p \in \pi(r,m)$;
- $\Gamma, \mathcal{I}, (r,m) \models \neg\phi$ if not $\Gamma, \mathcal{I}, (r,m) \models \phi$;
- $\Gamma, \mathcal{I}, (r,m) \models \phi \wedge \psi$ if $\Gamma, \mathcal{I}, (r,m) \models \phi$ and $\Gamma, \mathcal{I}, (r,m) \models \psi$;
- $\Gamma, \mathcal{I}, (r,m) \models A\phi$ if $\Gamma, \mathcal{I}, (r',m) \models \phi$ for all $r' \in \mathcal{R}$ with $r[0\ldots m] = r'[0\ldots m]$;
- $\Gamma, \mathcal{I}, (r,m) \models \bigcirc\phi$ if $\Gamma, \mathcal{I}, (r,m+1) \models \phi$;
- $\Gamma, \mathcal{I}, (r,m) \models \phi U \psi$ if there exists $m' \geq m$ such that $\Gamma, \mathcal{I}, (r,m') \models \psi$ and $\Gamma, \mathcal{I}, (r,k) \models \phi$ for all k with $m \leq k < m'$;
- $\Gamma, \mathcal{I}, (r,m) \models \exists x.\phi$ if $\Gamma[r'(m')/x], \mathcal{I}, (r,m) \models \phi$ for some point (r',m') of $\mathcal{I}$;
- $\Gamma, \mathcal{I}, (r,m) \models \mathsf{e}_i(x)$ if $r_i(m) = \Gamma(x)_i$;
- $\Gamma, \mathcal{I}, (r,m) \models D_G\phi$ if $\Gamma, \mathcal{I}, (r',m') \models \phi$ for all (r',m') such that $(r',m') \sim_G^D (r,m)$;

- $\Gamma, \mathcal{I}, (r,m) \models C_G\phi$ if $\Gamma, \mathcal{I}, (r',m') \models \phi$ for all (r',m') such that $(r',m') \sim_G^C (r,m)$.

The definition is standard, except for the constructs $\exists x.\phi$ and $\mathsf{e}_i(x)$. The clause for the former says that $\exists x.\phi$ holds at a point (r,m) if there exists a global state $g = r'(m')$ such that ϕ holds at the point (r,m), provided, we interpret x as referring to g. Note that it is required that g is attained at some point (r',m'), so actually occurs in the system $\mathcal{I}$. The clause for $\mathsf{e}_i(x)$ says that this holds at a point (r,m) if the local state of agent i, i.e., $r_i(m)$, is the same as the local state $\Gamma(x)_i$ of agent i at the global state $\Gamma(x)$ that interprets the variable x according to Γ.

We remark that these novel constructs introduce some redundancy, in that the set of epistemic operators D_G could be reduced to the "universal" operator $D_\emptyset$, since $D_G\phi \equiv \exists x.(\mathsf{e}_G(x) \wedge D_\emptyset(\mathsf{e}_G(x) \Rightarrow \phi))$. Evidently, given the complexity of this formulation, D_G remains a useful notation.

Strategic Environments

In order to deal with agents that operate in an environment by strategically choosing their actions, we introduce a richer type of transition system that models the available actions and their effects on the state. An *environment* for agents Ags is a tuple $E = \langle S, I, Acts, \rightarrow, \{O_i\}_{i\in Ags}, \pi\rangle$, where S is a set of states, I is a subset of S, representing the initial states, $Acts = \Pi_{i\in Ags} Acts_i$ is a set of joint actions, where each $Acts_i$ is a nonempty set of actions that may be performed by agent i, component $\rightarrow \subseteq S \times Acts \times S$ is a transition relation, $O_i : S \rightarrow L_i$ is an observation function, and $\pi : S \rightarrow \mathcal{P}(Prop)$ is a propositional assignment. An environment is said to be finite if all its components, i.e., $S, Ags, Acts_i, L_i$ and $Prop$ are finite. Intuitively, a joint action $a \in Acts$ represents a choice of action $a_i \in Acts_i$ for each agent $i \in Ags$, performed simultaneously, and the transition relation resolves this into an effect on the state. We assume that $\rightarrow$ is serial in the sense that for all $s \in S$ and $a \in Acts$ there exists $t \in S$ such that $(s,a,t) \in \rightarrow$.

A *strategy* for agent $i \in Ags$ in such an environment E is a function $\alpha : S \rightarrow \mathcal{P}(Acts_i) \setminus \{\emptyset\}$, selecting a set of actions of the agent at each state.[1] We call these the actions *enabled* at the state. A *group strategy*, or *strategy profile*, for a group G is a tuple $\alpha_G = \langle \alpha_i \rangle_{i\in G}$ where each α_i is a strategy for agent i. A strategy α_i is *deterministic* if $\alpha_i(s)$ is a singleton for all s. A strategy α_i for agent i is *uniform* if for all states s,t, if $O_i(s) = O_i(t)$, then $\alpha_i(s) = \alpha_i(t)$. A strategy $\alpha_G = \langle \alpha_i \rangle_{i\in G}$ for a group G is *locally uniform (deterministic)* if α_i is uniform (respectively, deterministic) for each agent $i \in G$. Given an environment E, we write $\Sigma^{det}(E)$ for the set of deterministic strategies, $\Sigma^{unif}(E)$ for the set of all locally uniform joint strategies, and $\Sigma^{unif,det}(E)$ for the set of all deterministic locally uniform joint strategies.

We now define an interpreted system that contains all the possible runs generated when agents Ags behave by choosing a strategy from some set Σ of joint strategies in the context of an environment E. One innovation, introduced in [8], is that the construction introduces new agents $\sigma(i)$, for each $i \in Ags$. The observation of $\sigma(i)$ is the strategy currently being used by agent i. Agent $\sigma(i)$ is not associated with any actions, and is primarily for use in epistemic operators, to allow reference to what can be deduced were agents to reason using information about each other's strategies. For $G \subseteq Ags$, we write $\sigma(G)$ for the set $\{\sigma(i) \mid i \in G\}$. Additionally, we include an agent e for representing the state of the environment. (This agent, also, is not associated with any actions.)

Formally, given an environment $E = \langle S, I, Acts, \rightarrow, \{O_i\}_{i\in Ags}, \pi\rangle$ for agents Ags, where $O_i : S \rightarrow L_i$ for each $i \in Ags$, and a set $\Sigma \subseteq \Pi_{i\in Ags}\Sigma_i$ of joint strategies for the group Ags, we define the *strategy space* interpreted system $\mathcal{I}(E,\Sigma) = (\mathcal{R},\pi')$. The system $\mathcal{I}(E,\Sigma)$ has global states $\mathcal{G} = S \times \Pi_{i\in Ags} L_i \times \Pi_{i\in Ags}\Sigma_i$. Intuitively, each global state consists of a state of the environment E, a local state for each agent i in E,

[1] More generally, a strategy could be a function of the history, but we focus here on strategies that depend only on the final state.

and a strategy for each agent i. We index the components of this cartesian product by e, the elements of Ags and the elements of $\sigma(Ags)$, respectively. We take the set of runs $\mathcal{R}$ of $\mathcal{I}(E,\Sigma)$ to be the set of all runs $r : \mathbb{N} \to \mathcal{G}$ satisfying the following constraints, for all $m \in \mathbb{N}$ and $i \in Ags$

1. $r_e(0) \in I$ and $\langle r_{\sigma(i)}(0)\rangle_{i\in Ags} \in \Sigma$,
2. $r_i(m) = O_i(r_e(m))$,
3. $(r_e(m), a, r_e(m+1)) \in \rightarrow$ for some joint action $a \in Acts$ such that for all $j \in Ags$ we have $a_j \in \alpha_j(r_j(m))$, where $\alpha_j = r_{\sigma(j)}(m)$, and
4. $r_{\sigma(i)}(m+1) = r_{\sigma(i)}(m)$.

The first constraint, intuitively, says that runs start at an initial state of E, and the initial strategy profile at time 0 is one of the profiles in Σ. The second constraint states that the agent i's local state at time m is the observation that agent i makes of the state of the environment at time m. The third constraint says that evolution of the state of the environment is determined at each moment of time by agents choosing an action by applying their strategy at that time to their local state at that time. The joint action resulting from these individual choices is then resolved into a transition on the state of the environment using the transition relation from E. The final constraint says that agents' strategies are fixed during the course of a run. Intuitively, each agent picks a strategy, and then sticks to it. The interpretation π' of $\mathcal{I}(E,\Sigma)$ is determined from the interpretation π of E by taking $\pi'(r,m) = \pi(r_e(m))$ for all points (r,m).

Our epistemic strategy logic is now just an instantiation of the extended temporal epistemic logic in the strategy space generated by an environment. That is, we start with an environment E and an associated set of strategies Σ, and then work with the language ETLK($Ags \cup \sigma(Ags) \cup \{e\}, Prop, Var$) in the interpreted system $\mathcal{I}(E,\Sigma)$. We call this instance of the language ESL($Ags, Prop, Var$), or just ESL when the parameters are implicit.

Applications

In [8], we proposed a logic CTL*K($Ags \cup \sigma(Ags), Prop$) extending temporal epistemic logic with strategy agents to allow the reasoning about knowledge and strategy by standard epistemic operators. The language introduced above is a generalization of the definitions in [8], to which we have added the constructs $\exists x.\phi$ and $\mathsf{e}_i(x)$. For formulas without these constructs, the semantics of ESL ignores the context Γ, so this component of the triple $\Gamma, \mathcal{I}(E,\Sigma), (r,m)$ can be removed from the definition, and it collapses to the definitions for CTL*K($Ags \cup \sigma(Ags), Prop$) in [8].

In the system $\mathcal{I}(E,\Sigma)$ we may refer, using distributed knowledge operators D_G where G contains the new strategic agents $\sigma(i)$, to what agents would know, should they take into account not just their own observations, but also information about other agent's strategies. For example, the distributed knowledge operator $D_{\{i,\sigma(i),\sigma(j)\}}$ captures what agent i would know, taking into account its own strategy and the strategy being used by agent j. Various applications of the usefulness of these distributed knowledge operators containing strategic agents are given in [8]. For example, we describe an application to *erasure policies* in computer security in which we write formulas such as

$$\neg D_\emptyset \neg(\mathtt{done} \wedge \neg \mathtt{exploited} \wedge EF \bigvee_{x\in \mathtt{CCN}} D_{\{A,\sigma(A),\sigma(M)\}}(\mathtt{cc} \neq x)))$$

to state that it is possible for an attacker A on an e-commerce payment gateway to obtain information about a credit card number cc even after the transaction is done, provided that the attacker reasons using knowledge about their own observations, their own strategy, but also knowledge of the strategy being

used by the merchant M. Here $\texttt{done} \wedge \neg\texttt{exploited}$ captures a situation where the transaction is done but the attacker has not run any exploit, and $D_{\{A,\sigma(A),\sigma(M)\}}(\texttt{cc} \neq x)$ says that the attacker is able to rule out the specific credit card number x from the range of possible values $\texttt{CCN}$ for the actual credit card number $\texttt{cc}$ (so the attacker has at least one bit of information about the actual credit card number). The modality $\neg D_\emptyset \neg$ is used to state that there is a point of the system where the formula holds. In particular, since the system builds in all possible strategies for the players, this modality captures a quantification over strategies.

In further applications given in [8], we showed that $\mathrm{CTL}^*\mathrm{K}(Ags \cup \sigma(Ags), Prop)$ can be used to express game theoretic equilibria, to reason about knowledge-based programs [4], and that many variants of alternating temporal epistemic logics that have been proposed in the literature can be expressed using $\mathrm{CTL}^*\mathrm{K}(Ags \cup \sigma(Ags), Prop)$. We refer the reader to [8] for details.

However, we had to make a restriction for some of these expressiveness results to formulas that do not contain uses of a common knowledge operator. We now show how the extended language of the present paper can remove this restriction.

Jamroga and van der Hoek [10] formulate a construct $\langle\!\langle H \rangle\!\rangle^\bullet_{\mathcal{K}(G)}\phi$ that says, effectively, that there is a strategy for a group H that another group G knows (for notion of group knowledge $\mathcal{K}$, which could be E for everyone knows, D for distributed knowledge, or C for common knowledge) to achieve goal ϕ. The semantics of this construct is given with respect to an environment E and a state s, and (in outline) is given by $E, s \models \langle\!\langle H \rangle\!\rangle^\bullet_{\mathcal{K}(G)}\phi$ if there exists a uniform strategy α for group H such that for all states t with $s \sim^{\mathcal{K}}_G t$, we have that all paths ρ from t that are consistent with α satisfy ϕ. Here $\sim^{\mathcal{K}}_G$ is the appropriate epistemic indistinguishability relation on states of E. We show in [8] how $\langle\!\langle H \rangle\!\rangle^\bullet_{\mathcal{K}(G)}\phi$ can be expressed in $\mathrm{CTL}^*\mathrm{K}(Ags \cup \sigma(Ags), Prop)$ for the cases where $\mathcal{K}$ is either E or D.

In the case of the operator $\langle\!\langle H \rangle\!\rangle^\bullet_{C(G)}\phi$, the definition involves the common knowledge that a group G of agents would have if they were to reason taking into consideration the strategy being used by another group H. This does not appear to be expressible using $\mathrm{CTL}^*\mathrm{K}(Ags \cup \sigma(Ags), Prop)$. In particular, the formula $C_{G\cup\sigma(H)}\phi$ does not give the intended meaning. Instead, what needs to be expressed is the greatest fixpoint X of the equation $X \equiv \bigwedge_{i\in G} D_{\{i\}\cup\sigma(H)}(X \wedge \phi)$. The language $\mathrm{CTL}^*\mathrm{K}(Ags \cup \sigma(Ags), Prop)$ does not include fixpoint operators and it does not seem that the intended meaning is expressible. On the other hand, it can be expressed with $\mathrm{ESL}(Ags, Var, Prop)$ in a natural way by a formula $C_G(\mathsf{e}_{\sigma(H)}(x) \Rightarrow \phi)$, which says that it is common knowledge to the group G that ϕ holds if the group H is running the strategy profile capture by the variable x. Using this idea, the construct $\langle\!\langle H \rangle\!\rangle^\bullet_{C(G)}\phi$ can be represented with ESL as

$$\exists x. C_G(\mathsf{e}_{\sigma(H)}(x) \Rightarrow \phi) \ .$$

(We remark that a carefully stated equivalence result requires an appropriate treatment of initial states in the environment E. We refer to [8] for details.) Applying similar ideas, ESL can also be used to eliminate, from the results on reasoning about knowledge-based programs presented in [8], the restriction to knowledge-based programs not containing common knowledge operators.

Model Checking

Since interpreted systems are always infinite objects, we use environments to give a finite input for the model checking problem. For an environment E, a set of strategies Σ for E, and a context Γ for $\mathcal{I}(E,\Sigma)$, we write $\Gamma, E, \Sigma \models \phi$ if $\Gamma, \mathcal{I}(E,\Sigma), (r,0) \models \phi$ for all runs r of $\mathcal{I}$ of $\mathcal{I}(E,\Sigma)$. (Often, the formula ϕ will be a sentence, i.e., will have all variables x in the scope of an operator $\exists x$. In this case the statement $\Gamma, E, \Sigma \models \phi$ is independent of Γ and we may write simply $E, \Sigma \models \phi$) The model checking problem is to

determine whether $\Gamma, E, \Sigma \models \phi$ for a finite state environment E, a set Σ of strategies and a context Γ, where ϕ is an ESL($Ags, Var, Prop$) formula.

For generality, we abstract Σ to a paramaterized class such that for each environment E, the set $\Sigma(E)$ is a set of strategies for E. We say that the parameterized class $\Sigma(E)$ is *PTIME-presented*, if it is presented by means of an algorithm that runs in time polynomial in the size of E and verifies if a given strategy α is in $\Sigma(E)$. For example, the class $\Sigma(E)$ of all strategies for E can be PTIME-presented, as can $\Sigma^{unif}(E)$, $\Sigma^{det}(E)$ and $\Sigma^{unif,det}(E)$.

A naive model checking algorithm would construct a transition system over the set of states $S \times \Sigma(E)$, where S is the set of states of E, then apply model checking techniques on it. Note that a joint strategy for an environment E can be represented in space $|S| \times |Acts|$. Thus, the size of $S \times \Sigma(E)$ is exponential as a function of the size of E. This means that the naive procedure requires exponential space. This indeed turns out to be the complexity of model checking the logic. However, it is possible to do better than this provided we restrict to the CTL-based fragment of the language. This is the fragment in which the temporal operators occur only in the forms $A\bigcirc\phi$, $\neg A\neg\bigcirc\phi$, $A\phi_1 U\phi_2$, and $\neg A\neg\phi_1 U\phi_2$.

Theorem 1 *Let $\Sigma(E)$ be a PTIME presented class of strategies for environments E. The complexity of deciding, given an environment E, an ESL formula ϕ and a context Γ for the free variables in an ESL formula ϕ relative to E and $\Sigma(E)$, whether $\Gamma, E, \Sigma(E) \models \phi$, is EXPSPACE-complete. For the restriction of the problem to ϕ in the CTL-based fragment, the complexity is PSPACE-complete.*

Conclusions

Hybrid logic [1] is an approach to the extension of modal logics that uses "nominals", i.e., propositions p that hold at a single world. These can be used in combination with operators such as $\exists p$, which marks an arbitrary world as the unique world at which nominal p holds. Our construct $\exists x$ is closely related to the hybrid construct $\exists p$, but we work in a setting that is richer in both syntax and semantics than previous works. There have been a few works using hybrid logic ideas in the context of epistemic logic [7, 13] but none are concerned with temporal logic. Hybrid temporal logic has seen a larger amount of study [2, 6, 5, 15], with variances in the semantics used for the model checking problem.

We note that if we were to view the variable x in our logic as a propositional constant, it would be true at a set of points in the system $\mathcal{I}(E, \Sigma)$, hence not a nominal in that system. Results in [2], where a hybrid linear time temporal logic formula is checked in all paths in a given model, suggest that a variant of ESL in which x is treated as a nominal in $\mathcal{I}(E, \Sigma)$ would have a complexity of model checking at least non-elementary, compared to our EXPSPACE and PSPACE complexity results.

Our model checking result seems to be more closely related to the a result in [5] that model checking a logic HL($\exists, @, F, A$) is PSPACE-complete. Here F is essentially a branching time future operator and A is a universal operator (similar to our $D_\emptyset$), the construct $@_p\phi$ says that ϕ holds at the world marked by the nominal p, and $\exists p(\phi)$ says that ϕ holds after marking some world by p. The semantics in this case does not unfold the model into either a tree or a set of linear structures before checking the formula, so the semantics of the hybrid existential $\exists$ is close to our idea of quantifying over global states. Our language, however, has a richer set of operators, even in the temporal dimension, and introduces the strategic dimension in the semantics. It would be an interesting question for future work to consider fragments of our language to obtain more precise statement of the relationship with hybrid temporal logics.

Strategy Logic [3] is a (non-epistemic) generalization of ATL for perfect information strategies in which strategies may be explicitly named and quantified. Work on identification of more efficient variants

of quantified strategy logic includes [11], which formulates a variant with a 2-EXPTIME-complete model checking problem. In both cases, strategies are perfect recall strategies, rather than the imperfect recall strategies that form the basis for our PSPACE-completeness result for model checking. The exploration of our logic over such a richer space of strategies is an interesting topic for future research.

References

[1] P. Blackburn & J. Seligman (1998): *What are hybrid languages?* In M. de Rijke, H. Wansing & M. Zakharyaschev, editors: *Advances in Modal Logic*, 1, CSLI Publications, pp. 41–62.

[2] Laura Bozzelli & Ruggero Lanotte (2010): *Complexity and succinctness issues for linear-time hybrid logics*. *Theoretical Computer Science* 411(2), pp. 454–469, doi:10.1016/j.tcs.2009.08.009.

[3] Krishnendu Chatterjee, Thomas A. Henzinger & Nir Piterman (2010): *Strategy logic*. *Information and Computation* 208(6), pp. 677–693, doi:10.1016/j.ic.2009.07.004.

[4] R. Fagin, J.Y. Halpern, Y. Moses & M.Y. Vardi (1995): *Reasoning About Knowledge*. The MIT Press.

[5] Massimo Franceschet & Maarten de Rijke (2006): *Model checking hybrid logics (with an application to semistructured data)*. *Journal of Applied Logic* 4(3), pp. 279–304, doi:10.1016/j.jal.2005.06.010.

[6] Massimo Franceschet, Maarten de Rijke & Bernd-Holger Schlingloff (2003): *Hybrid Logics on Linear Structures: Expressivity and Complexity*. In: *10th International Symposium on Temporal Representation and Reasoning / 4th International Conference on Temporal Logic (TIME-ICTL 2003)*, pp. 166–173. Available at `http://doi.ieeecomputersociety.org/10.1109/TIME.2003.1214893`.

[7] Jens Ulrik Hansen (2011): *A Hybrid Public Announcement Logic with Distributed Knowledge*. *Electronic Notes in Theoretical Computer Science* 273, pp. 33–50, doi:10.1016/j.entcs.2011.06.011.

[8] Xiaowei Huang & Ron van der Meyden (2014): *A Temporal Logic of Strategic Knowledge*. To appear KR'14, extended version available at `http://www.cse.unsw.edu.au/~meyden/research/atl_obs.pdf`.

[9] Wojciech Jamroga & Thomas Ågotnes (2007): *Constructive knowledge: what agents can achieve under imperfect information*. *Journal of Applied Non-Classical Logics* 17(4), pp. 423–475, doi:10.3166/jancl.17.423-475.

[10] Wojciech Jamroga & Wiebe van der Hoek (2004): *Agents that Know How to Play* . *Fundamenta Informaticae* 62, pp. 1–35.

[11] Fabio Mogavero, Aniello Murano & Moshe Y. Vardi (2010): *Reasoning About Strategies*. In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*, pp. 133–144, doi:10.4230/LIPIcs.FSTTCS.2010.133.

[12] Sieuwert van Otterloo & Geert Jonker (2005): *On Epistemic Temporal Strategic Logic*. *Electronic Notes in Theoretical Computer Science* 126, pp. 77–92, doi:10.1016/j.entcs.2004.11.014.

[13] O. Roy (2009): *A dynamic-epistemic hybrid logic for intentions and information changes in strategic games*. *Synthese* 171(2), pp. 291–320, doi:10.1007/s11229-009-9644-3.

[14] Pierre-Yves Schobbens (2004): *Alternating-time logic with imperfect recall*. *Electronic Notes in Theoretical Computer Science* 85(2), pp. 82–93, doi:10.1016/S1571-0661(05)82604-0.

[15] Thomas Schwentick & Volker Weber (2007): *Bounded-Variable Fragments of Hybrid Logics*. In: *Proc. STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science*, *Springer LNCS* 4393, pp. 561–572, doi:10.1007/978-3-540-70918-3_48.

Doomsday Equilibria for Games on Graphs*

Krishnendu Chatterjee†
IST Austria

Laurent Doyen
LSV, ENS Cachan & CNRS

Emmanuel Filiot‡ Jean-François Raskin§
Université Libre de Bruxelles – U.L.B.

Two-player games on graphs provide the theoretical framework for many important problems such as reactive synthesis. While the traditional study of two-player zero-sum games has been extended to multi-player games with several notions of equilibria, they are decidable only for perfect-information games, whereas several applications require imperfect-information games.

In this paper we propose a new notion of equilibria, called doomsday equilibria, which is a strategy profile such that all players satisfy their own objective, and if any coalition of players deviates and violates even one of the players objective, then the objective of every player is violated.

We present algorithms and complexity results for deciding the existence of doomsday equilibria for various classes of ω-regular objectives, both for imperfect-information games, and for perfect-information games. We provide optimal complexity bounds for imperfect-information games, and in most cases for perfect-information games.

1 Introduction

Two-player games on finite-state graphs with ω-regular objectives provide the framework to study many important problems in computer science [22, 20, 9]. One key application area is synthesis of reactive systems [2, 21, 19]. Traditionally, the reactive synthesis problem is reduced to two-player zero-sum games, where vertices of the graph represent states of the system, edges represent transitions, one player represents a component of the system to synthesize, and the other player represents the purely adversarial coalition of all the other components. Since the coalition is adversarial, the game is zero-sum, i.e., the objectives of the two players are complementary. Two-player zero-sum games have been studied in great depth in literature [15, 9, 11].

Instead of considering all the other components as purely adversarial, a more realistic model is to consider them as individual players each with their own objective, as in protocol synthesis where the rational behavior of the agents is to first satisfy their own objective in the protocol before trying to be adversarial to the other agents. Hence, inspired by recent applications in protocol synthesis, the model of multi-player games on graphs has become an active area of research in graph games and reactive synthesis [1, 10, 23]. In a multi-player setting, the games are not necessarily zero-sum (i.e., objectives are not necessarily conflicting) and the classical notion of rational behavior is formalized as Nash equilibria [18]. Nash equilibria perfectly capture the notion of rational behavior in the absence of external criteria, i.e., the players are concerned only about their own payoff (internal criteria), and they are indifferent to the payoff of the other players. In the setting of synthesis, the more appropriate notion is the adversarial

*This work was published in the proceedings of the conference VMCAI'14 [7], and a long version is available online [6].

†Supported by Austrian Science Fund (FWF) Grant No P23499-N23, FWF NFN Grant No S11407-N23 (RiSE), ERC Start grant (279307: Graph Games), and Microsoft faculty fellows award.

‡Supported by the Belgian National Fund for Scientific Research.

§Supported by ERC Start grant (279499: inVEST).

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 43–48, doi:10.4204/EPTCS.146.6

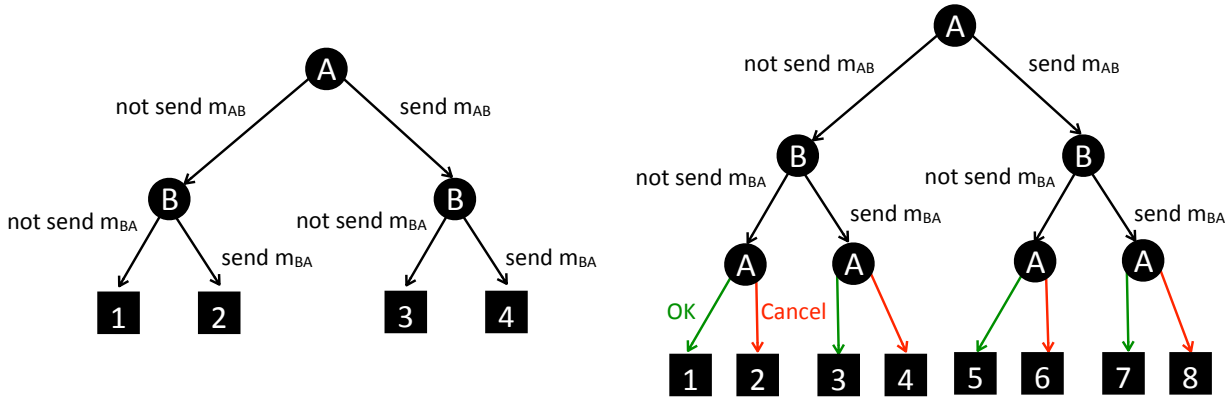


Figure 1: A simple example in the domain of Fair Exchange Protocols

external criteria, where the players are as harmful as possible to the other players without sabotaging with their own objectives. This has inspired the study of refinements of Nash equilibria, such as secure equilibria [4] (that captures the adversarial external criteria), rational synthesis [10], and led to several new logics where the non-zero-sum equilibria can be expressed [5, 8, 17, 24, 16]. The complexity of Nash equilibria [23], secure equilibria [4], rational synthesis [10], and of the new logics has been studied recently [5, 8, 17, 24].

Along with the theoretical study of refinements of equilibria, applications have also been developed in the synthesis of protocols. In particular, the notion of secure equilibria has been useful in the synthesis of mutual-exclusion protocol [4], and of fair-exchange protocols [13, 3] (a key protocol in the area of security for exchange of digital signatures). One major drawback that all the notions of equilibria suffer is that the basic decision questions related to them are decidable only in the setting of perfect-information games (in a perfect-information games the players perfectly know the state and history of the game, whereas in imperfect-information games each player has only a partial view of the state space of the game), and in the setting of multi-player imperfect-information games they are undecidable [19]. However, the model of imperfect-information games is very natural because every component of a system has private variables not accessible to other components, and recent works have demonstrated that imperfect-information games are required in synthesis of fair-exchange protocols [12]. In this paper, we provide the first decidable framework that can model them.

We propose a new notion of equilibria which we call *doomsday-threatening* equilibria (for short, doomsday equilibria). Given n objectives $\varphi_1, \ldots, \varphi_n$ and n strategies $\Lambda_1, \ldots, \Lambda_n$ for each of the n players respectively, the strategy profile $\Lambda = (\Lambda_1, \ldots, \Lambda_n)$ is a doomsday equilibrium if:

(a) all players satisfy their own objectives, that is $outcome(\Lambda) \in \varphi_i$ for all $1 \leq i \leq n$ (where $outcome(\Lambda)$ is the path obtained according to the strategies in the profile), and

(b) if any coalition of players deviates and violates even one of the players objective, then doomsday follows (every player objective is violated), that is for all $1 \leq i \leq n$, for all strategy profiles $\Lambda' = (\Lambda'_1, \ldots, \Lambda'_n)$ such that $\Lambda'_i = \Lambda_i$, if $outcome(\Lambda') \notin \varphi_i$, then $outcome(\Lambda') \notin \varphi_j$ for all $1 \leq j \leq n$.

Note that in contrast to other notions of equilibria, doomsday equilibria consider deviation by an arbitrary set of players, rather than individual players. Moreover, in case of two-player non-zero-sum games they coincide with the secure equilibria [4] where objectives of both players are satisfied.

Example 1. *Consider the two trees of Figure 1. They model the possible behaviors of two entities* Alice *and* Bob *that have the objective of exchanging messages:* m_{AB} *from* Alice *to* Bob*, and* m_{BA} *from* Bob

to Alice. *Assume for the sake of illustration that* m_{AB} *models the transfer of property of a house from* Alice *to* Bob*, while* m_{BA} *models the payment of the price of the house from* Bob *to* Alice.

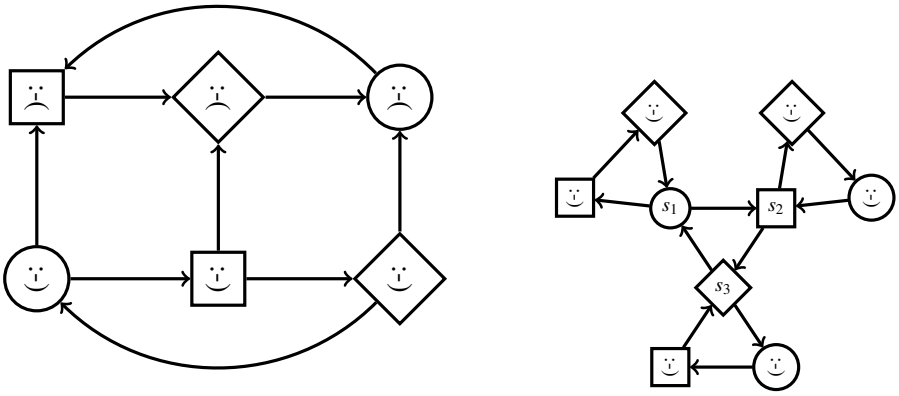
Having that interpretation in mind, let us consider the left tree. On the one hand, Alice *has as primary objective (internal criterion) to reach either state* 2 *or state* 4*, states in which she has obtained the money. She has a slight preference for* 2 *as in that case she received the money while not transferring the property of her house to* Bob*, this corresponds to her adversarial external criterion. On the other hand,* Bob *would like to reach either state* 3 *or* 4 *(similarly with a slight preference for* 3*). Also, it should be clear that* Alice *would hate to reach* 3 *because she would have transferred the property of her house to* Bob *but without being paid. Similarly,* Bob *would hate to reach* 2*. To summarize,* Alice *has the following preference order on the final states of the protocol:* $2 > 4 > 1 > 3$*, while for* Bob *the order is* $3 > 4 > 1 > 2$*. Is there a* doomsday-threatening equilibrium *in this game ? For such an equilibrium to exist, we must find a pair of strategies that please the two players for their primary objective (internal criterion): reach* $\{2,4\}$ *for* Alice *and reach* $\{3,4\}$ *for* Bob*. Clearly, this is only possible if at the root* Alice *plays "*send m_{AB}*", as otherwise we would not reach* $\{3,4\}$ *violating the primary objective of* Bob*. But playing that action is not safe for* Alice *as* Bob *would then choose "*not send m_{BA}*" because he slightly prefers* 3 *to* 4*. It can be shown that the only rational way of playing (taking into account both internal and external criteria) is for* Alice *to play "*not send m_{AB}*" and for* Bob *to play "*not send m_{BA}*". This profile is in fact the only secure equilibrium of the game but this is not what we hope from such a protocol.*

The difficulty in this exchange of messages comes from the fact that Alice *is starting the protocol by sending her part and this exposes her. To obtain a better behaving protocol, one solution is to add an extra stage after the exchanges of the two messages as shown in the right tree of Figure 1. In this new protocol,* Alice *has the possibility to cancel the exchange of messages (in practice this would be implemented by the intervention of a* TTP[1]*). For that new game, the preference orderings of the players are as follows: for* Alice *it is* $3 > 7 > 1 = 2 = 4 = 6 = 8 > 5$*, and for* Bod *it is* $5 > 7 > 1 = 2 = 4 = 6 = 8 > 3$*. Now let us show that there is a doomsday equilibrium in this new game. In the first round,* Alice *should play "*send m_{AB}*" as otherwise the internal objective of* Bob *would be violated, then* Bob *should play "*send m_{BA}*", and finally* Alice *should play "*OK*" to validate the exchange of messages. This profile of strategies satisfies the first property of a doomsday equilibrium: both players have reached their primary objective, and no player has an incentive to deviate. Indeed, if* Alice *deviates then* Bob *would play "*not send m_{BA}*", and we obtain a doomsday situation as both players have their primary objectives violated. If* Bob *deviates by playing "*not send m_{BA}*", then* Alice *would cancel the protocol exchange which again produces a doomsday. So, no player has an incentive to deviate from the equilibrium and the outcome of the protocol is the desired one: the two messages have been fairly exchanged. So, we see that the threat of a doomsday brought by the action "*Cancel*" has a beneficial influence on the behavior of the two players.* □

Example 2. *Figure 2 gives two examples of games with safety and Büchi objectives respectively.*

(Safety) Consider the 3-player game arena with perfect information of Figure 2(a) and safety objectives. Unsafe states for each player are given by the respective nodes of the upper part. Assume that the initial state is one of the safe states. This example models a situation where three countries are in peace until one of the countries, say country i*, decides to attack country* j*. This attack will then necessarily be followed by a doomsday situation: country* j *has a strategy to punish all other countries. The doomsday equilibrium in this example is to play safe for all players.*

[1] TTP stands for *Trusted Third Party*.



(a) Doomsday (Safety) (b) Büchi objectives

Figure 2: Examples of doomsday equilibria for Safety and Büchi objectives

(Büchi) Consider the 3-player game arena with perfect information of Figure 2(b) with Büchi objectives for each player: Player i wants to visit infinitely often one of its "happy" states. The position of the initial state does not matter. To make things more concrete, let us use this game to model a protocol where 3 players want to share in each round a piece of information made of three parts: for all $i \in \{1,2,3\}$*, Player i knows information* $(i+1)$ *mod* 3 *and* $(i+2)$ *mod* 3*. Player i can send or not these informations to the other players. This is modeled by the fact that Player i can decide to visit the happy states of the other players, or move directly to* $s_{(i \bmod 3)+1}$*. The objective of each player is to have an infinite number of successful rounds where they get all information.*

There are several doomsday equilibria. As a first one, let us consider the situation where for all $i \in \{1,2,3\}$*, if Player i is in state* s_i*, then he alternately moves to the happy states and to* $s_{(i \bmod 3)+1}$*. This defines an infinite play that visits all the states infinitely often. Whenever some player deviates from this play, the other players retaliate by always choosing in the future to go to the next s-state instead of visiting the happy states. Clearly, if all players follow their respective strategy, then all happy states are visited infinitely often. Now consider the strategy of Player i against two strategies of the other players that makes him lose. Clearly, the only way Player i loses is when the other two players eventually never visit their happy states anymore, but then all the players lose.*

As a second one, consider the strategies where Player 2 *and Player* 3 *always take their loop but Player* 1 *never takes his loop, and such that whenever the play deviates, Player* 2 *and* 3 *retialate by never taking their loops. For the same reasons as before this strategy profile is a doomsday equilibrium.*

Note that the first equilibrium requires one bit of memory for each player, to remember if they visit their s state for the first or second time. In the second equilibrium, only Player 2 *and* 3 *need a bit of memory. An exhaustive analysis shows that there is no memoryless doosmday equilibrium.* □

It should now be clear that multi-player games with doomsday equilibria provide a suitable framework to model various problems in protocol synthesis. In addition to the definition of doomsday equilibria, our main contributions are to present algorithms and complexity bounds for deciding the existence of such equilibria for various classes of ω-regular objectives both in the perfect-information and in the imperfect-information cases. Our technical contributions are summarized in Table 1. More specifically:

1. *(Perfect-information games).* We show that deciding the existence of doomsday equilibria in multi-player perfect-information games is (i) PTIME-complete for reachability, Büchi, and coBüchi ob-

objectives	safety	reachability	Büchi	co-Büchi	parity
perfect information	PSPACE-C	PTIME-C	PTIME-C	PTIME-C	PSPACE NP-HARD CONP-HARD
imperfect information	EXPTIME-C	EXPTIME-C	EXPTIME-C	EXPTIME-C	EXPTIME-C

Table 1: Summary of the results

jectives; (ii) PSPACE-complete for safety objectives; and (iii) in PSPACE and both NP-hard and coNP-hard for parity objectives.

2. *(Imperfect-information games).* We show that deciding the existence of doomsday equilibria in multi-player imperfect-information games is EXPTIME-complete for reachability, safety, Büchi, coBüchi, and parity objectives.

In a long version of this paper [6], we also prove that deciding the existence of a doomsday threatening equilibrium in a game whose objectives are given as LTL formula is 2EXPTIME-complete, but we devise a Safraless procedure [14] suitable to efficient implementation.

The area of multi-player games and various notions of equilibria is an active area of research, but notions that lead to decidability in the imperfect-information setting and has applications in synthesis has largely been an unexplored area. Our work is a step towards it.

References

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.

[2] J. R. Büchi and L. H. Landweber. Definability in the monadic second-order theory of successor. *J. Symb. Log.*, 34(2):166–170, 1969.

[3] R. Chadha, S. Kremer, and A. Scedrov. Formal analysis of multiparty contract signing. *J. Autom. Reasoning*, 36(1-2):39–83, 2006.

[4] K. Chatterjee, T. A. Henzinger, and M. Jurdzinski. Games with secure equilibria. *Theor. Comput. Sci.*, 365(1-2):67–82, 2006.

[5] K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. *Inf. Comput.*, 208(6):677–693, 2010.

[6] Krishnendu Chatterjee, Laurent Doyen, Emmanuel Filiot, and Jean-François Raskin. Doomsday equilibria for omega-regular games. *CoRR*, abs/1311.3238, 2013.

[7] Krishnendu Chatterjee, Laurent Doyen, Emmanuel Filiot, and Jean-François Raskin. Doomsday equilibria for omega-regular games. In *VMCAI*, pages 78–97, 2014.

[8] A. Da Costa Lopes, F. Laroussinie, and N. Markey. ATL with strategy contexts: Expressiveness and model checking. In *FSTTCS*, volume 8 of *LIPIcs*, pages 120–132, 2010.

[9] E. A. Emerson and C. Jutla. Tree automata, mu-calculus and determinacy. In *FOCS*, pages 368–377. IEEE Comp. Soc., 1991.

[10] D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In *Proc. of TACAS*, LNCS 6015, pages 190–204. Springer, 2010.

[11] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, LNCS 2500. Springer, 2002.

[12] W. Jamroga, S. Mauw, and M. Melissen. Fairness in non-repudiation protocols. In *Proc. of STM*, LNCS 7170, pages 122–139. Springer, 2012.

[13] S. Kremer and J.-F. Raskin. A game-based verification of non-repudiation and fair exchange protocols. *Journal of Computer Security*, 11(3):399–430, 2003.

[14] O. Kupferman and M. Y. Vardi. Safraless decision procedures. In *FOCS*, 2005.

[15] D. Martin. Borel determinacy. In *Annals of Mathematics*, volume 102, pages 363–371, 1975.

[16] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. What makes ATL* decidable? a decidable fragment of strategy logic. In *CONCUR*, pages 193–208, 2012.

[17] F. Mogavero, A. Murano, and M. Y. Vardi. Reasoning about strategies. In *Proc. of FSTTCS*, volume 8 of *LIPIcs*, pages 133–144. Schloss Dagstuhl - LZfI, 2010.

[18] J. F. Nash. Equilibrium points in *n*-person games. *PNAS*, 36:48–49, 1950.

[19] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *POPL*, pages 179–190. ACM Press, 1989.

[20] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. soc.*, 141:1–35, 1969.

[21] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987.

[22] L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39:1095–1100, 1953.

[23] M. Ummels and D. Wojtczak. The complexity of nash equilibria in stochastic multiplayer games. *Logical Methods in Computer Science*, 7(3), 2011.

[24] F. Wang, C. Huang, and F. Yu. A temporal logic for the interaction of strategies. In *Proc. of CONCUR*, LNCS 6901, pages 466–481. Springer, 2011.

Nash Equilibria in Symmetric Games with Partial Observation

Patricia Bouyer Nicolas Markey
LSV, CNRS & ENS Cachan, France
{bouyer,markey}@lsv.ens-cachan.fr

Steen Vester
DTU, Kgs. Lyngby, Denmark
stve@dtu.dk

We investigate a model for representing large multiplayer games, which satisfy strong symmetry properties. This model is made of multiple copies of an arena; each player plays in his own arena, and can partially observe what the other players do. Therefore, this game has partial information and symmetry constraints, which make the computation of Nash equilibria difficult. We show several undecidability results, and for bounded-memory strategies, we precisely characterize the complexity of computing pure Nash equilibria (for qualitative objectives) in this game model.

1 Introduction

Multiplayer games. Games played on graphs have been intensively used in computer science as a tool to reason about and automatically synthesize interacting reactive systems [10]. Consider a server granting access to a printer and connected to several clients. The clients may send requests to the server, and the server grants access to the printer depending on the requests it receives. The server could have various strategies: for instance, never grant access to any client, or always immediately grant access upon request. However, it may also have constraints to satisfy (which define its winning condition): for instance, that no two clients should access the printer at the same time, or that any request must eventually be granted. A strategy for the server is then a policy that it should apply in order to achieve these goals.

Until recently, more focus had been put on the study of purely antagonistic games (a.k.a. zero-sum games), which conveniently represent systems evolving in a (hostile) environment: the aim of one player is to prevent the other player from achieving his own objective.

Non-zero-sum games. Over the last ten years, computer scientists have started considering games with non-zero-sum objectives: they allow for conveniently modelling complex infrastructures where each individual system tries to fulfill its own objectives, while still being subject to uncontrollable actions of the surrounding systems. As an example, consider a wireless network in which several devices try to send data: each device can modulate its transmitting power, in order to maximize its bandwidth or reduce energy consumption as much as possible. In that setting, focusing only on optimal strategies for one single agent is too narrow. Game-theoreticians have defined and studied many other solution concepts for such settings, of which Nash equilibrium [11] is a prominent one. A Nash equilibrium is a strategy profile where no player can improve the outcome of the game by unilaterally changing his strategy. In other terms, in a Nash equilibrium, each individual player has a satisfactory strategy. Notice that Nash equilibria need not exist or be unique, and are not necessarily optimal: Nash equilibria where all players lose may coexist with more interesting Nash equilibria. Finding constrained Nash equilibria (*e.g.*, equilibria in which some players are required to win) is thus an interesting problem for our setting.

Networks of identical devices. Our aim in this paper is to handle the special case where all the interacting systems (but possibly a few of them) are identical. This encompasses many situations involving

Part of this work was sponsored by ERC Starting Grant EQualIS and by EU FP7 project Cassting.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 49–55, doi:10.4204/EPTCS.146.7

computerized systems over a network. We propose a convenient way of modelling such situations, and develop algorithms for synthesizing a single strategy that, when followed by all the players, leads to a global Nash equilibrium. To be meaningful, this requires symmetry assumptions on the arena of the game (the board should look the same to all the players). We also include *imperfect observation* of the other players, which we believe is relevant in such a setting.

Our contributions. We propose a convenient model for representing large interacting systems, which we call *game structure*. A game structure is made of multiple copies of a single arena (one copy per player); each player plays on his own copy of the arena. As mentioned earlier, the players have imperfect information about the global state of the game (they may have a perfect view on some of their "neighbours", but may be blind to some other players). In *symmetric* game structures, we additionally require that any two players are in similar situations: for every pair of players (A,B), we are able to map each player C to a corresponding player D with the informal meaning that 'player D is to B what player C is to A'. Of course, winning conditions and imperfect information should respect that symmetry. We present several examples illustrating the model, and argue why it is a relevant model for computing symmetric Nash equilibria.

We show several undecidability results, in particular that the parameterized synthesis problem (aiming to obtain one policy that forms a Nash equilibrium when applied to any number of participants) is undecidable. We then characterize the complexity of computing (constrained) pure symmetric Nash equilibria in symmetric game structures, when objectives are given as LTL formulas, and when restricting to memoryless and bounded-memory strategies. This problem with no memory bound is then proven undecidable.

Related work. Game theory has been a very active area since the 1940's, but its applications to computer science *via* graph games is quite recent. In that domain, until recently more focus had been put on zero-sum games [10]. Some recent works have considered multi-player non-zero-sum games, including the computation of (constrained) equilibria in turn-based and in concurrent games [5, 14, 2] or the development of temporal logics geared towards non-zero-sum objectives [4, 6].

None of those works distinguish symmetry constraints in strategy profiles nor in game description. Still, symmetry has been studied in the context of normal-form games [12, 7]: in such a game, each player has the same set of actions, and the utility function of a player only depends on his own action and on the number of players who played each action (it is independent on 'who played what'). Finally, let us mention that symmetry was also studied in the context of model checking, where different techniques have been developped to deal with several copies of the same system [9, 8, 1].

By lack of space, most of the technical developments could not be included in this extended abstract. They are available in the technical report [3].

2 Symmetric concurrent games

2.1 Definitions

For any $k \in \mathbb{N} \cup \{\infty\}$, we write $[k]$ for the set $\{i \in \mathbb{N} \mid 0 \leq i < k\}$ (in particular, $[\infty] = \mathbb{N}$). Let $s = (p_i)_{i\in[n]}$ be a sequence, with $n \in \mathbb{N} \cup \{\infty\}$ being the length $|s|$ of s. Let $j \in \mathbb{N}$ s.t. $j-1 < n$. The jth element of s, denoted s_{j-1}, is the element p_{j-1} (so that a sequence $(p_i)_{i\in[n]}$ may be named p when no ambiguity arises). The jth prefix $s_{<j}$ of s is the finite sequence $(p_i)_{i\in[j]}$. If s is finite, we write $\mathsf{last}(s)$ for its last element $s_{|s|-1}$.

Definition 1 *An* arena *is a tuple* $\langle \mathsf{States}, \mathsf{Agt}, \mathsf{Act}, \mathsf{Mov}, \mathsf{Tab}\rangle$ *where* States *is a finite set of states;* Agt *is a finite set of agents (also named players);* Act *is a finite set of actions;* $\mathsf{Mov}\colon \mathsf{States} \times \mathsf{Agt} \to 2^{\mathsf{Act}} \setminus \{\emptyset\}$ *is the set of actions available to a given player in a given state;* $\mathsf{Tab}\colon \mathsf{States} \times \mathsf{Act}^{\mathsf{Agt}} \to \mathsf{States}$ *is a transition function that specifies the next state, given a state and an action of each player.*

The evolution of such a game is as usual: at each step, the players propose a move, and the successor state is given by looking up this action vector in the transition table. A path is a sequence of states obtained this way; we write Hist for the set of finite paths (or *histories*).

Let $A \in \mathsf{Agt}$. A *strategy* for A is a mapping $\sigma_A : \mathsf{Hist} \to \mathsf{Act}$ such that for any $\rho \in \mathsf{Hist}$, $\sigma_A(\rho) \in \mathsf{Mov}(\mathsf{last}(\rho), A)$. Given a set of players $C \subseteq \mathsf{Agt}$, a strategy for C is a mapping σ assigning to each $A \in C$ a strategy for A (we write σ_A instead of $\sigma(A)$ to alleviate notations). As a special case, a strategy for Agt is called a *strategy profile*. A path π is *compatible* with a strategy σ of coalition C if, for any $i < |\pi|$, there exists a move $(m_A)_{A\in\mathsf{Agt}}$ such that $\mathsf{Tab}(\rho_{i-1}, (m_A)_{A\in\mathsf{Agt}}) = \rho_i$ and $m_A = \sigma_A(\rho_{<i})$ for all $A \in C$. The set of *outcomes* of σ from a state s, denoted $\mathsf{Out}(s,\sigma)$, is the set of plays from s that are compatible with σ.

Let $\mathscr{G}$ be a game. A *winning condition* for player A is a set Ω_A of plays of $\mathscr{G}$. We say that a play $\rho \in \Omega_A$ yields payoff 1 to A, and a play $\rho \notin \Omega_A$ yields payoff 0 to A. A strategy σ of a coalition C is *winning* for A from a state s if $\mathsf{Out}(s,\sigma) \subseteq \Omega_A$. A strategy profile σ is a *Nash equilibrium* if, for any $A \in \mathsf{Agt}$ and any strategy σ'_A, if σ is losing for A, then so is $\sigma[A \mapsto \sigma'_A]$. In other terms, no player can individually improve his payoff.

Remark 2 *In this paper, we only use purely boolean winning conditions, but our algorithms could easily be extended to the* semi-quantitative *setting of [2], where each player has several (pre)ordered boolean objectives. We omit such extensions in this paper, and keep focus on symmetry issues.*

The model we propose is made of a one-player arena, together with an observation relation. Intuitively, each player plays in his own copy of the one-player arena; the global system is the product of all the local copies, but each player observes the state of the global system only through the observation relation. This is in particular needed for representing large networks of systems, in which each player may only observe some of his neighbours.

Example 3 *Consider for instance a set of identical devices (e.g. cell phones) connected on a local area network. Each device can modulate its emitting power. In order to increase its bandwidth, a device tends to increase its emitting power; but besides consuming more energy, this also adds* noise *over the network, which decreases the other players' bandwidth and encourages them to in turn increase their power. We can model a device as an m-state arena (state i corresponding to some power p_i, with $p_0 = 0$ representing the device being off). Any device would not know the exact state of the other devices, but would be able to evaluate the surrounding noise; this can be modelled using our observation relation. Based on this information, the device can decide whether it should increase or decrease its emitting power (knowing that the other devices play the same strategy as it is playing), resulting in a good balance between bandwidth and energy consumption.*

Definition 4 *An n-*player game network *is a tuple $\mathscr{G} = \langle G, (\equiv_i)_{i\in[n]}, (\Omega_i)_{i\in[n]}\rangle$ s.t. $G = \langle \mathit{States}, \{A\}, \mathit{Act}, \mathit{Mov}, \mathit{Tab}\rangle$ is a one-player arena; for each $i \in [n]$, $\equiv_i$ is an equivalence relation on States^n (extended in a natural way to sequences of states of States^n). Two $\equiv_i$-equivalent configurations are indistinguishable to player i. This models* imperfect information *for player i; for each $i \in [n]$, $\Omega_i \subseteq (\mathit{States}^n)^\omega$ is the objective of player i. We require that for all $\rho, \rho' \in (\mathit{States}^n)^\omega$, if $\rho \equiv_i \rho'$ then ρ and ρ' are equivalently in Ω_i.*

The semantics of this game is defined as the "product game" $\mathscr{G}' = \langle \mathsf{States}', [n], \mathsf{Act}, \mathsf{Mov}', \mathsf{Tab}', (\Omega_i)_{i\in[n]}\rangle$ where $\mathsf{States}' = \mathsf{States}^n$, $\mathsf{Mov}'((s_0,\dots,s_{n-1}), i) = \mathsf{Mov}(s,i)$, and the transition table is defined as

$$\mathsf{Tab}'((s_0,\dots,s_{n-1}),(m_i)_{i\in[n]}) = (\mathsf{Tab}(s_0,m_0),\dots,\mathsf{Tab}(s_{n-1},m_{n-1})).$$

An element of States^n is called a *configuration* of $\mathscr{G}$. The equivalence relation $\equiv_i$ induces equivalence classes of configurations that player i cannot distinguish. We call these equivalence classes *information*

sets and denote $\mathscr{I}_i$ the set of information sets for player i. Strategies should respect these information sets: a strategy σ_i for player i is $\equiv_i$-*realisable* whenever for all $\rho, \rho' \in \mathsf{Hist}$, $\rho \equiv_i \rho'$ implies $\sigma_i(\rho) = \sigma_i(\rho')$. A strategy profile $\sigma = (\sigma_i)_{1 \leq i \leq n}$ is said *realisable in* $\mathscr{G}$ whenever σ_i is $\equiv_i$-realisable for every $i \in [n]$.

Remark 5 *We assume that each equivalence relation $\equiv_i$ is given* compactly *using templates whose size is independant of n. As an example, for $P \subseteq \mathsf{Agt}$, the relation $\mathsf{Id}(P)$ defined by $(t,t') \in \mathsf{Id}(P)$ iff $t[i] = t'[i]$ for all $i \in P$ encodes perfect observation of the players in P, and no information about the other players.*

Example 6 *Consider the cell-phone game again. It can be modelled as a game network where each player observes everything (i.e., the equivalence relations $\equiv_i$ are the identity). A more realistic model for the system can be obtained by assuming that each player only gets precise information about his close neighbours, and less precise information (only an estimation of the global noise in the network), or no information at all, about the devices that are far away.*

Despite the global arena being described as a product of identical arenas, not all games described this way are symmetric: the observation relation also has to be *symmetric*. We impose extra conditions on that relation in order to capture our expected notion of symmetry. Given a permutation π of $[n]$, for a configuration $t = (s_i)_{i \in [n]}$ we let $t(\pi) = (s_{\pi(i)})_{i \in [n]}$; for a path $\rho = (t_j)_{j \in \mathbb{N}}$, we let $\rho(\pi) = (t_j(\pi))_{j \in \mathbb{N}}$.

Definition 7 *A game network $\mathscr{G} = \langle G, (\equiv_i)_{i \in [n]}, (\Omega_i)_{i \in [n]} \rangle$ is* symmetric *whenever for any two players $i, j \in [n]$, there is a permutation $\pi_{i,j}$ of $[n]$ such that $\pi_{i,j}(i) = j$ and satisfying the following conditions for every $i, j, k \in [n]$:*

1. *$\pi_{i,i}$ is the identity, and $\pi_{k,j} \circ \pi_{i,k} = \pi_{i,j}$; hence $\pi_{i,j}^{-1} = \pi_{j,i}$.*
2. *the observation made by the players is compatible with the symmetry of the game: for any two configurations t and t', $t \equiv_i t'$ iff $t(\pi_{i,j}^{-1}) \equiv_j t'(\pi_{i,j}^{-1})$;*
3. *objectives are compatible with the symmetry of the game: for every play ρ, $\rho \in \Omega_i$ iff $\rho(\pi_{i,j}^{-1}) \in \Omega_j$.*

In that case, $\pi = (\pi_{i,j})_{i,j \in [n]}$ is called a symmetric representation *of $\mathscr{G}$.*

The mappings $\pi_{i,j}$ define the symmetry of the game: $\pi_{i,j}(k) = l$ means that player l plays vis-à-vis player j the role that player k plays vis-à-vis player i. We give the intuition why we apply $\pi_{i,j}^{-1}$ in the definition above, and not $\pi_{i,j}$. Assume configuration $t = (s_0, \ldots, s_{n-1})$ is observed by player i. The corresponding configuration for player j is $t' = (s'_0, \ldots, s'_{n-1})$ where player-$\pi_{i,j}(k)$ state should be that of player k in t. That is, $s'_{\pi_{i,j}(k)} = s_k$, so that $t' = t(\pi_{i,j}^{-1})$.

These mappings also define how symmetry must be used in strategies: let $\mathscr{G}$ be a symmetric n-player game network with symmetric representation π. We say that a strategy profile $\sigma = (\sigma_i)_{i \in [n]}$ is *symmetric* for the representation π if it is realisable (*i.e.*, each player only plays according to what he can observe) and if for all $i, j \in [n]$ and every history ρ, it holds $\sigma_i(\rho) = \sigma_j(\rho(\pi_{i,j}^{-1}))$.

The following lemma characterizes symmetric strategy profiles:

Lemma 8 *Fix a symmetric representation π for $\mathscr{G}$. If σ_0 is an $\equiv_0$-realisable strategy for player 0, then the strategy profile σ defined for all $i > 0$ by $\sigma_i(\rho) = \sigma_0(\rho(\pi_{i,0}^{-1}))$ is symmetric.*

Example 9 *Consider a card game tournament with six players, three on each table. Here each player has a left neighbour, a right neighbour, and three opponents at a different table. To model this, one could assume player 0 knows everything about himself, and has some informations about his right neighbour (player 1) and his left neighbour (player 2). But he knows nothing about players 3, 4 and 5.*

Now, the role of player 2 vis-à-vis player 1 is that of player 1 vis-à-vis player 0 (he is his right neighbour). Hence, we can define the symmetry as $\pi_{0,1}(0) = 1$, $\pi_{0,1}(1) = 2$, $\pi_{0,1}(2) = 0$, and $\pi_{0,1}(\{3,4,5\}) =$

$\{3,4,5\}$ *(any choice is fine here). As an example, the observation relation in this setting could be that player* 0 *has perfect knowledge of his set of cards, but only knows the number of cards of players* 1 *and* 2*, and has no information about the other three players. Notice that other observation relations would have been possible (for instance, giving more information about the right player).*

In this paper we are interested in computing (symmetric) Nash equilibria in symmetric game networks:

Problem 1 (Constrained existence of (symmetric) NE) *The* constrained existence problem *asks, given a symmetric game network $\mathcal{G}$, a symmetric representation π, a configuration t, a set $L \subseteq [n]$ of losing players, and a set $W \subseteq [n]$ of winning players, whether there is a (symmetric) Nash equilibrium σ in $\mathcal{G}$ from t for the representation π, such that all players in L lose and all players in W win. If L and W are empty, the problem is simply called the* existence problem*. If $W = [n]$, the problem is called the* positive existence problem*.*

We first realise that even though symmetric Nash equilibria are Nash equilibria with special properties, they are in some sense at least as hard to find as Nash equilibria. This can be proved by seeing the individual game structure as a product of n disconnected copies of the original individual structure. This way, the strategy played by one player on one copy imposes no constraints on the strategy played by another player on a different copy.

Proposition 10 *From a symmetric game network $\mathcal{G}$ we can construct in polynomial time a symmetric game network $\mathcal{H}$ such that there exists a symmetric Nash equilibrium in $\mathcal{H}$ if, and only if, there exists a Nash equilibrium in $\mathcal{G}$. Furthermore the construction only changes the arena, but does not change the number of players nor the objectives or the resulting payoffs.*

3 Our results

Undecidability with non-regular objectives. Our games allow for arbitrary boolean objectives, defined for each player as a set of winning plays. As can be expected, this is too general to get decidability of our problems even with perfect information, since it can be used to encode the runs of a two-counter machine:

Theorem 11 *The (constrained) existence of a symmetric Nash equilibrium for non-regular objectives in (two-player) perfect-information symmetric game networks is undecidable.*

Undecidability with a parameterized number of players. Parameterized synthesis of Nash equilibria (that is, synthesizing a single strategy that each player will apply, and that yields a Nash equilibrium for any number of players) was one of our targets in this work. We show that computing such equilibria is not possible, even in rather restricted settings.

Theorem 12 *The (positive) existence of a* parameterized *symmetric Nash equilibrium for LTL objectives in symmetric game networks is undecidable (even for memoryless strategies).*

This is proved by encoding a Turing machine as a game network with arbitrarily many players, each player controlling one cell of the tape. The machine halts if there exists a number n of players such that the play reaches the halting state. We use LTL formulas to enforce correct simulation of the Turing machine.

From positive existence to existence. Because of the previous result, we now fix the number n of players. Before turning to our decidability results, we begin with showing that positive existence of Nash equilibria is not harder than existence. Notice that this makes a difference with the setting of turn-based games, where Nash equilibria always exist.

Proposition 13 *Deciding the (symmetric) existence problem in (symmetric) game networks is always at least as hard as deciding the positive (symmetric) existence problem. The reduction doubles the number of players and uses LTL objectives, but does not change the nature of the strategies (memoryless, bounded-memory, or general).*

Bounded-memory strategies.

Theorem 14 *The (positive, constrained) existence of a bounded-memory symmetric Nash equilibrium for LTL objectives in symmetric game networks is* EXPSPACE*-complete.*

The EXPSPACE-hardness results are direct consequences of the proof of Theorem 12 (the only difference is that we restrict to a Turing machine using exponential space).

The algorithm for memoryless strategies is as follows: it first guesses a memoryless strategy for one player, from which it deduces the strategy to be played by the other players. It then looks for the players that are losing, and checks if they alone can improve their payoff. If they cannot improve the guessed strategy yields a Nash equilibrium, otherwise it does not yield an equilibrium.

The first step is to guess and store an $\equiv_0$-realisable memoryless strategy σ_0 for player 0, which we then prove witnesses the existence of a symmetric Nash equilibrium. Such a strategy is a mapping from States^n to Act. We intend player 0 to play according to σ_0, and any player i to play according to $\sigma_0(\pi_{i,0}^{-1}(s_0,\ldots,s_{n-1}))$ in state $(s_0,\ldots,s_{n-1})$. From Lemma 8 we know that all symmetric memoryless strategy profiles can be characterized by such an $\equiv_0$-realisable memoryless strategy for player 0.

The algorithm then guesses a set W of players (which satisfies the given constraint), and checks that under the strategy profile computed above, the players in W achieve their objectives while the players not in W do not. This is achieved by computing the non-deterministic Büchi automata for ϕ_i if $i \in W$ and for $\neg\phi_i$ if $i \notin W$, and checking that the outcome of the strategy profile above (which is a lasso-shaped path and can easily be computed from strategy σ_0) is accepted by all those automata.

It remains to check that the players not in W cannot win if they deviate from their assigned strategy. For each player i not in W, we build the one-player game where all players but player i play according to the selected strategy profile. The resulting automaton contains all the plays that can be obtained by a deviation of player i. It just remains to check that there is no path satisfying ϕ_i in that automaton. If this is true for all players not in W, then the selected strategy σ_0 gives rise to a memoryless symmetric Nash equilibrium.

Regarding (space) complexity, storing the guessed strategy requires space $O(|\mathsf{States}|^n)$. The Büchi automata have size exponential in the size of the formulas, but can be handled on-the-fly using classical constructions, so that the algorithm only requires polynomial space in the size of the formula. The lasso-shaped outcome, as well as the automata representing the deviations of the losing players, have size $O(|\mathsf{States}|^n)$, but can also be handled on-the-fly. In the end, the whole algorithm runs in exponential space in the number of players, and polynomial in the size of the game and in the size of the LTL formulas.

The above algorithm can be lifted to bounded-memory strategies: given a memory bound m, it guesses a strategy σ_0 using memory m, and does the same computations as above. Storing the strategy now requires space $O(m \cdot |\mathsf{States}|^n)$, which is still exponential, even if m is given in binary.

Remark 15 *Notice that the algorithms above could be adapted to handle non-symmetric equilibria in non-symmetric game networks: it would just guess all the strategies, the payoff, and check the satisfaction of the LTL objectives in the product automaton obtained by applying the strategies.*

The algorithm could also be adapted, still with the same complexity, to handle richer objectives, in particular in the semi-quantitative setting of [2], where the players have several (pre)ordered objectives. Instead of guessing the set of winners, the algorithm would guess, for each player, which objectives are satisfied, and check that no individual improvement is possible. The latter can be achieved by listing all possible improvements and checking that none of them can be reached.

General strategies. We already mentioned an undecidability result in Theorem 11 for two-player games and perfect information when general strategies are allowed. However, the objectives used for achieving the reduction are quite complex. On the other hand, imperfect information also leads to undecidability for

LTL objectives with only 3 players. To show this, we can slightly alter a proof from [13]. Here, synthesis of distributed reactive systems (corresponding to finding sure-winning strategies) with LTL objectives is shown undecidable in the presence of imperfect information. The situation used in the proof can be modelled in our framework and by adding a matching-penny module in the beginning and slightly changing the LTL objectives, we can obtain undecidability of Nash equilibria instead of sure-winning strategies.

Theorem 16 *The existence of a (symmetric) Nash equilibrium for LTL objectives in symmetric game networks is undecidable for $n \geq 3$ players.*

References

[1] Parosh Aziz Abdulla & Bengt Jonsson (1999): *On the Existence of Network Invariants for Verifying Parameterized Systems*. In: *Correct System Design, Recent Insight and Advances*, *Lecture Notes in Computer Science* 1710, Springer-Verlag, pp. 180–197, doi:http://dx.doi.org/10.1007/3-540-48092-7_9.

[2] Patricia Bouyer, Romain Brenguier, Nicolas Markey & Michael Ummels (2012): *Concurrent games with ordered objectives*. In: *Proc. 15th International Conference on Foundations of Software Science and Computation Structure (FoSSaCS'12)*, *Lecture Notes in Computer Science* 7213, Springer-Verlag, pp. 301–315, doi:http://dx.doi.org/10.1007/978-3-642-28729-9_20.

[3] Patricia Bouyer, Nicolas Markey & Steen Vester (2014): *Nash Equilibria in Symmetric Games with Partial Observation*. Technical Report LSV-14-01, Lab. Spécification & Vérification, ENS Cachan, France.

[4] Krishnendu Chatterjee, Thomas A. Henzinger & Nir Piterman (2007): *Strategy Logic*. In: *Proc. 18th International Conference on Concurrency Theory (CONCUR'07)*, *Lecture Notes in Computer Science* 4703, Springer-Verlag, pp. 59–73, doi:http://dx.doi.org/10.1007/978-3-540-74407-8_5.

[5] Krishnendu Chatterjee, Rupak Majumdar & Marcin Jurdziński (2004): *On Nash equilibria in stochastic games*. In: *Proc. 18th International Workshop on Computer Science Logic (CSL'04)*, *Lecture Notes in Computer Science* 3210, Springer-Verlag, pp. 26–40, doi:http://dx.doi.org/10.1007/978-3-540-30124-0_6.

[6] Arnaud Da Costa, François Laroussinie & Nicolas Markey (2010): *ATL with strategy contexts: Expressiveness and Model Checking*. In: *Proc. 30th Conferentce on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'10)*, *Leibniz International Proceedings in Informatics* 8, Leibniz-Zentrum für Informatik, pp. 120–132, doi:http://dx.doi.org/10.4230/LIPIcs.FSTTCS.2010.120.

[7] Partha Dasgupta & Eric Maskin (1986): *The Existence of Equilibrium in Discontinuous Economic Games, 1: Theory*. *The Review of Economic Studies* 53(1), pp. 1–26, doi:http://dx.doi.org/10.2307/2297589.

[8] E. Allen Emerson & A. Prasad Sistla (1996): *Symmetry and model checking*. *Formal Methods in System Design* 9(1-2), pp. 105–131, doi:http://dx.doi.org/10.1007/BF00625970.

[9] Steven M. German & A. Prasad Sistla (1992): *Reasoning about Systems with Many Processes*. *Journal of the ACM* 39(3), pp. 675–735, doi:http://dx.doi.org/10.1145/146637.146681.

[10] Thomas A. Henzinger (2005): *Games in system design and verification*. In: *Proc. 10th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'05)*, pp. 1–4, doi:http://dx.doi.org/10.1145/1089933.1089935.

[11] John F. Nash, Jr. (1950): *Equilibrium Points in n-Person Games*. *Proc. National Academy of Sciences* 36(1), pp. 48–49, doi:http://dx.doi.org/10.1073/pnas.36.1.48.

[12] John F. Nash, Jr. (1951): *Non-cooperative Games*. *Annals of Mathematics* 54(2), pp. 286–295, doi:http://dx.doi.org/10.2307/1969529.

[13] Amir Pnueli & Roni Rosner (1990): *Distributed Reactive Systems Are Hard to Synthesize*. In: *Proc. 31st Annual Symposium on Foundations of Computer Science (FOCS'90)*, IEEE Computer Society Press, pp. 746–757, doi:http://dx.doi.org/10.1109/FSCS.1990.89597.

[14] Michael Ummels & Dominik Wojtczak (2011): *The Complexity of Nash Equilibria in Stochastic Multiplayer Games*. *Logical Methods in Computer Science* 7(3:20), doi:http://dx.doi.org/10.2168/LMCS-7(3:20)2011.

Refining and Delegating Strategic Ability in ATL

Dimitar P. Guelev
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences, Sofia, Bulgaria
gelevdp@math.bas.bg

We propose extending Alternating-time Temporal Logic (ATL) by an operator $\langle i \sqsubseteq \Gamma\rangle\varphi$ to express that i can distribute its powers to a set of sub-agents Γ in a way which satisfies ATL condition φ on the strategic ability of the coalitions they may form, possibly together with others agents. We prove the decidability of model-checking of formulas whose $\langle . \sqsubseteq .\rangle$-subformulas have the form $\langle i_1 \sqsubseteq \Gamma_1\rangle \ldots \langle i_m \sqsubseteq \Gamma_m\rangle\varphi$, with no further occurrences of $\langle . \sqsubseteq .\rangle$ in φ.

Introduction

The basic co-operation modality of Alternating-time Temporal Logics (ATL, [AHK97, AHK02]) invites perceiving agent coalitions as single agents who enjoy the combined powers of the coalition members. We investigate an operator to reverse this, by addressing the possibility to partition the strategic ability of a single agent among several sub-agents. We write $\langle i \sqsubseteq \Gamma\rangle\varphi$ to denote that agent i can partition its strategic ability among the members of a set of fresh sub-agents Γ in a way which satisfies φ, a formula written in terms of the new agents Γ who assume i's powers, and the other original agents, except i. For example, a purchase scenario with the vendor represented by salesperson *SP* and delivery team *DT* can be described as

$$\langle vendor \sqsubseteq SP, DT\rangle \left(\begin{array}{l} \langle\langle customer, SP\rangle\rangle \Diamond purchase\ agreement \wedge \\ [\![SP]\!]\Box(purchase\ agreement \Rightarrow \langle\langle DT, customer\rangle\rangle \circ delivery) \end{array} \right).$$

The combined powers of all of i's sub-agents are always equal to i's:

$$\langle\langle \Delta \cup \{i\}\rangle\rangle\varphi \Leftrightarrow [i \sqsubseteq \Gamma]\langle\langle(\Delta \setminus \{i\}) \cup \Gamma\rangle\rangle\varphi$$

where $[i \sqsubseteq \Gamma]$ stands for $\neg\langle i \sqsubseteq \Gamma\rangle\neg$. Coalitions $\Delta \not\supseteq \Gamma$ may be weaker than i, but also have abilities contributed by agents from $\Delta \setminus \Gamma$. The realizability of schemes such as the example one generally depends on the basic composition of agents' actions. For instance, simple mechanisms make it always possible to deny the *proper* subsets of Γ all substantial strategic ability or make Γ use simple majority vote as indicated by the validity of the formula:

$$\neg\langle\langle\emptyset\rangle\rangle\varphi \wedge \langle\langle i\rangle\rangle\varphi \Rightarrow \langle i \sqsubseteq \Gamma\rangle \bigwedge_{\Delta \subsetneq \Gamma} \neg\langle\langle\Delta\rangle\rangle\varphi \wedge \langle i \sqsubseteq \Gamma\rangle \bigwedge_{\Delta \subset \Gamma, |\Delta| \leq |\Gamma\setminus\Delta|} \neg\langle\langle\Delta\rangle\rangle\varphi \wedge \bigwedge_{\Delta \subseteq \Gamma, |\Delta| > |\Gamma\setminus\Delta|} \langle\langle\Delta\rangle\rangle\varphi.$$

Subtracting strategic ability from one agent and transfering it in the form of a virtual sub-agent to another is a way of implementing *delegation*. Refinement can be instrumental in expressing the *alienability* of the ability in question. E.g.,

$$\langle\langle i\rangle\rangle \circ unlock \wedge \neg\langle\langle j\rangle\rangle \circ unlock \wedge \langle i \sqsubseteq i', key\rangle(\neg\langle\langle i'\rangle\rangle \circ unlock \wedge \langle\langle \underbrace{j, key}_{j'}\rangle\rangle \circ unlock)$$

states the possibility of giving i's *unlock*ing ability separate identity *key* which enables its passage to j. The relevant vocabulary introduced consists of *key* itself, $\{j, key\}$ for j *key*-in-hand and i' for i without *key*, respectively.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 57–63, doi:10.4204/EPTCS.146.8

Notably we investigate refining and delegating powers and not responsibilities as in, e.g., [NR02]. Sub-agents can pursue their own goals. As it becomes clear below, they do so by influencing the choice of actions on behalf of their super-agent with the share of the super-agents' power given to them. Unlike proper delegation as in, e.g., [vdHWW10] and [BFD02], where givers and receivers of control co-exist, just $\langle i \sqsubseteq \Gamma \rangle$ is about *replacing* i by its sub-agents Γ.

Our main result about ATL with $\langle . \sqsubseteq . \rangle$ in this paper is a model-checking procedure for the subset in which $\langle . \sqsubseteq . \rangle$ is restricted to occur only in subformulas of the form $\langle i_1 \sqsubseteq \Gamma_1 \rangle \dots \langle i_m \sqsubseteq \Gamma_m \rangle \varphi$, with no further occurrences of $\langle . \sqsubseteq . \rangle$ in φ. This is sufficient for the handling of scenarios like the example one above, but with refinements affecting more than one primary agent.

Structure of the paper After brief formal preliminaries on ATL on GCMs, we introduce our proposed operator and model-checking algorithm. We conclude by briefly commenting on some more related work, assessing our result and mentioning some work in progress.

1 Preliminaries

Definition 1 (concurrent game structures and models) A *concurrent game structure* (CGS) for some given set of agents $\Sigma = \{1, \dots, N\}$ is a tuple of the form $\langle W, \langle Act_i : i \in \Sigma \rangle, o \rangle$ where

W is a non-empty set of *states*;

Act_i is a non-empty set of *actions*, $i \in \Sigma$; given a $\Gamma \subseteq \Sigma$, Act_Γ stands for $\prod_{i \in \Gamma} Act_i$;

$o : W \times Act_\Sigma \to W$ is a *transition* function.

A *concurrent game model* (CGM) for Σ and atomic propositions AP is a tuple of the form $\langle W, \langle Act_i : i \in \Sigma \rangle, o, V \rangle$ where $\langle W, \langle Act_i : i \in \Sigma \rangle, o \rangle$ is a CGS for Σ and $V \subseteq W \times AP$ is a valuation relation.

In the sequel we always assume Act_i, $i \in \Sigma$ to be pairwise disjoint.

Below we write a_Γ to indicate that $a \in Act_\Gamma$ where $\Gamma \subseteq \Sigma$. If $a \in Act_\Delta$ and $\Gamma \subseteq \Delta$, then a_Γ also stands for the subvector of a consisting of the actions for the members of Γ. Given disjoint $\Gamma, \Delta \subseteq \Sigma$, we write $a_\Gamma \cdot b_\Delta$ for $c \in Act_{\Gamma \cup \Delta}$ which is defined by putting $c_i = a_i$ for $i \in \Gamma$ and $c_i = b_i$ for $i \in \Delta$.

Definition 2 (ATL **on CGMs)** The syntax of ATL formulas φ is given by the BNF

$$\varphi, \psi ::= \bot \mid p \mid (\varphi \Rightarrow \psi) \mid \langle\langle \Gamma \rangle\rangle \circ \varphi \mid \langle\langle \Gamma \rangle\rangle (\varphi \mathsf{U} \psi) \mid [\![\Gamma]\!] (\varphi \mathsf{U} \psi)$$

where p ranges over atomic propositions and Γ ranges over finite sets of agents. Satisfaction of ATL formulas are defined in terms of strategies. A *strategy* for $i \in \Sigma$ in CGM $M = \langle W, \langle Act_i : i \in \Sigma \rangle, o, V \rangle$ is a function from W^+ to Act_i. Given a vector of strategies $s_\Gamma = \langle s_i : i \in \Gamma \rangle$ for the members of $\Gamma \subseteq \Sigma$, the possible outcomes of Γ starting from state w and following s_Γ is the set of infinite runs

$$\mathrm{out}(w, s_\Gamma) = \{ w^0 w^1 \dots \in W^\omega : w^0 = w, w^{k+1} = o(w^k, a^k), a^0 a^1 \dots \in Act_\Sigma^\omega,\ a_\Gamma^k = s_\Gamma(w^0 \dots w^k), k < \omega \}.$$

Assuming a fixed M, we write S_Γ for the set of all vectors of strategies for Γ in M. Satisfaction is defined on CGMs M, states $w \in W$ and formulas φ:

$M,w \not\models \bot$

$M,w \models p$ iff $V(w,p)$

$M,w \models \varphi \Rightarrow \psi$ iff either $M,w \models \psi$ or $M,w \not\models \varphi$

$M,w \models \langle\langle\Gamma\rangle\rangle \circ \varphi$ iff there exists an $s_\Gamma \in S_\Gamma$ s. t. $w^0w^1\ldots \in \mathrm{out}(w,s_\Gamma)$ implies $M,w^1 \models \varphi$

$M,w \models \langle\langle\Gamma\rangle\rangle(\varphi \mathsf{U} \psi)$ iff there exists an $s_\Gamma \in S_\Gamma$ s. t. for any $w^0w^1\ldots \in \mathrm{out}(w,s_\Gamma)$ there exists a $k < \omega$ s. t. $M,w^0 \models \varphi,\ldots,M,w^{k-1} \models \varphi$ and $M,w^k \models \psi$

$M,w \models [\![\Gamma]\!](\varphi \mathsf{U} \psi)$ iff for every $s_\Gamma \in S_\Gamma$ there exists a $w^0w^1\ldots \in \mathrm{out}(w,s_\Gamma)$ and a $k < \omega$ s. t. $M,w^0 \models \varphi,\ldots,M,w^{k-1} \models \varphi$ and $M,w^k \models \psi$

$\top$, $\neg$, $\vee$, $\wedge$ and $\Leftrightarrow$ and the remaining combinations of $\langle\langle.\rangle\rangle$ and $[\![.]\!]$ with the temporal connectives $\circ$, $\Diamond$ and $\Box$ are regarded as derived constructs. See, e.g., [AHK02] for the definitions.

2 Refining Strategic Ability in ATL: $\mathrm{ATL}_{\sqsubseteq}$

Definition 3 (Γ-to-i homomorphisms of CGMs) Given Σ and AP, an $i \in \Sigma$ and some non-empty set of agent names Γ which is disjoint with Σ, consider CGMs $M = \langle W, \langle Act_j : j \in \Sigma\rangle, o, V\rangle$ and $M' = \langle W', \langle Act'_j : j \in \Sigma'\rangle, o', V'\rangle$ for AP, and Σ and $\Sigma' = (\Sigma \setminus \{i\}) \cup \Gamma$, respectively. A mapping $h : \prod_{j\in\Gamma} Act'_j \to Act_i$ is a Γ-*to*-i *homomorphism from* M' *to* M, if

$W' = W$, $V' = V$ and $Act_j = Act'_j$ for $j \in \Sigma \setminus \{i\}$;

range $h = Act_i$ and $o'(w,a) = o(w, a_{\Sigma\setminus\{i\}} \cdot h(a_\Gamma))$ for all $w \in W$ and all $a \in Act'_{\Sigma'}$.

Informally, if M is a Γ-to-i homomorphism of M, then the strategic ability of i in M is distributed among the new agents $j \in \Gamma$ in M'. For each action a_i of i in M there exists a vector of actions a_Γ for the members of Γ in M' such that $h(a_\Gamma) = a_i$. Together with the correspondence between the outcome functions o and o' of the two models, this means that the combined powers of the members of Γ in M' are equal to those of i in M, but proper sub-coalitions of Γ may be less powerful. Next we introduce the operator which is central to this work. Let M, i and Γ be as above.

Definition 4 (refinement operator) Let φ be written in terms of $(\Sigma \setminus \{i\}) \cup \Gamma$. Then

$$M,w \models \langle i \sqsubseteq \Gamma\rangle\varphi$$

iff there exist an M' for Σ' and AP such that $M',w \models \varphi$, and a Γ-to-i homomorphism from M' to M.

The occurrences of $j \in \Gamma$ in $\langle i \sqsubseteq \Gamma\rangle\varphi$ are *bound* in the usual sense. Informally, $\langle i \sqsubseteq \Gamma\rangle\varphi$ means that i can distribute its powers among the members of Γ so that φ holds in about the new set of agents. Its dual $[i \sqsubseteq \Gamma]\varphi$ means that φ holds regardless of how the powers of i are distributed among the agents from Γ.

3 Model-checking $\langle . \sqsubseteq . \rangle^*$-Flat $\mathrm{ATL}_{\sqsubseteq}$

$\langle . \sqsubseteq . \rangle^*$-flat $\mathrm{ATL}_{\sqsubseteq}$ is the subset of $\mathrm{ATL}_{\sqsubseteq}$ in which $\langle . \sqsubseteq . \rangle$-subformulas have the form

$$\langle i_1 \sqsubseteq \Gamma_1\rangle \ldots \langle i_m \sqsubseteq \Gamma_m\rangle\varphi \tag{1}$$

where φ has no further occurrences of $\langle . \sqsubseteq . \rangle$. Note that only occurrences of $\langle . \sqsubseteq . \rangle$ of the same polarity can be chained. E.g., if φ and ψ are $\langle . \sqsubseteq . \rangle$-free, then $\langle\langle i\rangle\rangle\Diamond(\langle i \sqsubseteq \Gamma\rangle\langle j \sqsubseteq \Delta\rangle\varphi \wedge [k \sqsubseteq \Upsilon][l \sqsubseteq \Xi]\psi)$ is $\langle . \sqsubseteq . \rangle^*$-flat, but $[i \sqsubseteq \Gamma]\langle j \sqsubseteq \Delta\rangle\varphi$ and $\langle i \sqsubseteq \Gamma\rangle\langle\langle k\rangle\rangle\Diamond\langle j \sqsubseteq \Delta\rangle\varphi$ are not. Our algorithm reduces the model-checking problem to satisfiability in the $\langle\langle.\rangle\rangle\circ$-subset of ATL, or, equivalently, in Coalition Logic [Pau02], which is known to be decidable. We first do the case of $m = 1$ and φ being a boolean combination of

$\langle\langle . \rangle\rangle\circ$-formulas with boolean combinations of atomic propositions as the arguments of $\langle\langle . \rangle\rangle\circ$, in full detail. Then we explain how the technique extends to arbitrary m, and, finally, however inefficiently, to formulas of the form (1) with an $\langle . \sqsubseteq . \rangle$-free φ in which the use of the ATL connectives is unrestricted.

The case of $m = 1$ Consider some formula $\langle i \sqsubseteq \Gamma\rangle\varphi$ with φ restricted as above. Let CGM M be as above and consider a CGM $M' = \langle W, \langle Act'_i : i \in \Sigma'\rangle, o', V\rangle$, $\Sigma' = \Sigma \setminus \{i\} \cup \Gamma$, and a Γ-to-i homomorphism h from M' to M. Let $\langle\langle\Delta\rangle\rangle \circ \chi$ be a subformula of φ. For $M', w \models \langle\langle\Delta\rangle\rangle \circ \chi$ to hold, there should be a vector of actions a_Δ such that, for any $b_{\Gamma\setminus\Delta}$, $a_{\Delta\setminus\Gamma} \cdot h(a_{\Delta\cap\Gamma} \cdot b_{\Gamma\setminus\Delta})$ gives $\Delta \setminus \Gamma \cup \{i\}$ a strategy to achieve $\circ\chi$ in M. For a fixed $a_{\Delta\setminus\Gamma}$ this means

$$h(a_{\Delta\cap\Gamma} \cdot b_{\Gamma\setminus\Delta}) \in \{a_i \in Act_i : \forall c_{\Sigma\setminus(\Delta\cup\{i\})} M, o(w, a_{\Delta\setminus\Gamma} \cdot a_i \cdot c_{\Sigma\setminus(\Delta\cup\{i\})}) \models \chi\} \tag{2}$$

Henceforth we write $A_{i,a_{\Delta\setminus\Gamma},w,\chi}$ for the subset of Act_i in (2).

Now consider a CGM $\overline{M} = \langle \overline{W}, \langle \overline{Act}_j : j \in \Gamma\rangle, \overline{o}, \overline{V}\rangle$ for Γ as the set of agents, $\overline{AP} = Act_i$ as the set of atomic propositions and $\overline{W} = Act_i \cup \{w^0\}$ as the set of states. Let $\overline{V}(w,a)$ be equivalent to $w = a$ for $a \in Act_i$, thus enabling reference to each individual action of i. The intended meaning of the states of $\overline{M}$ from Act_i is to represent the possible choices of i's actions by the members of Γ; w^0 is a distinguished reference state. Let $\overline{Act}_j = Act'_j$ for $j \in \Gamma$, and let $\overline{o}(w^0, a) = h(a)$ for all $a \in \overline{Act}_\Gamma$. Then

$$\overline{M}, w^0 \models \langle\langle\emptyset\rangle\rangle \circ \bigvee_{a\in Act_i} a \wedge \bigwedge_{a,b\in Act_i, a\neq b} \langle\langle\emptyset\rangle\rangle \circ \neg(a \wedge b) \wedge \bigwedge_{a\in Act_i} \langle\langle\Gamma\rangle\rangle \circ a, \tag{3}$$

since, due to the surjectivity of h, each of i's actions can be enforced by Γ, which is the grand coalition in $\overline{M}$.

Let the translation $\mathtt{t}$ replace subformulas of φ of the form $\langle\langle\Delta\rangle\rangle \circ \chi$ by their corresponding

$$\bigvee_{a_{\Delta\setminus\Gamma}\in Act_{\Delta\setminus\Gamma}} \langle\langle\Delta\cap\Gamma\rangle\rangle \circ \bigvee_{a_i\in A_{i,a_{\Delta\setminus\Gamma},w,\chi}} a_i.$$

Then $M, w \models \langle i \sqsubseteq \Gamma\rangle\varphi$ is equivalent to $\overline{M}, w^0 \models \mathtt{t}(\varphi)$.

Conversely, let a model $\overline{M} = \langle \overline{W}, \langle \overline{Act}_j : j \in \Gamma\rangle, \overline{o}, \overline{V}\rangle$ exist such that $\overline{M}, w^0 \models \mathtt{t}(\varphi)$ and (3) hold. Then we can define an M' and a Γ-to-i homomorphism h to witness $M, w \models \langle i \sqsubseteq \Gamma\rangle\varphi$ as follows. We put $Act'_j = \overline{Act}_j$, $j \in \Gamma$. For every $a_\Gamma \in \overline{Act}_\Gamma$, we define $h(a_\Gamma)$ as the unique $a_i \in Act_i$ such that $\overline{M}, o(w^0, a_\Gamma) \models a_i$. The identity $o'(w,a) = o(w^0, h(a))$ determines o'. Now a direct check shows that $M, w \models \langle i \sqsubseteq \Gamma\rangle\varphi$.

Hence, the existence of a model $\overline{M}$ which satisfies $\mathtt{t}(\varphi)$ and (3) at some state is equivalent to the satisfaction of φ at the given state w of the given M. Since satisfiability of formulas such as $\mathtt{t}(\varphi)$ and (3) is solvable, this entails the solvability of model-checking $\langle . \sqsubseteq . \rangle$-formulas.

The case of $m > 1$ To keep notation simple, let $m = 2$, i.e., consider formulas of the form $\langle 1 \sqsubseteq \Gamma_1\rangle\langle 2 \sqsubseteq \Gamma_2\rangle\varphi$. Bigger m are handled analogously. We first revise condition (2), with respect to formulas $\langle\langle\Delta\rangle\rangle \circ \chi \in \mathrm{Subf}(\varphi)$ in which $\Delta \subseteq \Sigma'$, $\Sigma' = \Sigma \setminus \{1,2\} \cup \Gamma_1 \cup \Gamma_2$. The $m = 2$-form of (2) is about sets of *pairs* of actions, for 1 and 2, respectively. Given a fixed $a_{\Delta\setminus(\Gamma_1\cup\Gamma_2)}$, (2) assumes the form

$$\langle h_1(a_{\Delta\cap\Gamma_1} \cdot b_{\Gamma_1\setminus\Delta}), h_2(a_{\Delta\cap\Gamma_2} \cdot b_{\Gamma_2\setminus\Delta})\rangle \in \\ \{\langle a_1, a_2\rangle \in Act_1 \times Act_2 : \forall c_{\Sigma\setminus(\Delta\cup\{1,2\})} M, o(w, a_1 \cdot a_2 \cdot a_{\Delta\setminus(\Gamma_1\cup\Gamma_2)} \cdot c_{\Sigma\setminus(\Delta\cup\{1,2\})}) \models \chi\}$$

We denote the subset of $Act_1 \times Act_2$ above by $A_{1,2,a_{\Delta\setminus(\Gamma_1\cup\Gamma_2)},w,\chi}$. The ability of Δ to achieve χ in one step from w is equivalent to the ability of each of $\Delta\cap\Gamma_1$ and $\Delta\cap\Gamma_2$ to enforce actions a_1 and a_2 on behalf of 1 and 2, respectively, so that $\langle a_1, a_2\rangle \in A_{1,2,a_{\Delta\setminus(\Gamma_1\cup\Gamma_2)},w,\chi}$ for some appropriate $a_{\Delta\setminus(\Gamma_1\cup\Gamma_2)}$. Therefore we define $\mathtt{t}(\langle\langle\Delta\rangle\rangle \circ \chi)$ as

$$\bigvee_{a_{\Delta\setminus(\Gamma_1\cup\Gamma_2)}\in Act_{\Delta\setminus(\Gamma_1\cup\Gamma_2)}} \bigvee_{A_1\times A_2\subseteq A_{1,2,a_{\Delta\setminus(\Gamma_1\cup\Gamma_2)},w,\chi}} \langle\langle\Delta\cap\Gamma_1\rangle\rangle \circ \bigvee_{a_1\in A_1} a_1 \wedge \langle\langle\Delta\cap\Gamma_2\rangle\rangle \circ \bigvee_{a_2\in A_2} a_2.$$

Formulas obtained by the $\langle 1 \sqsubseteq \Gamma_1\rangle\langle 2 \sqsubseteq \Gamma_2\rangle$-form of $\mathtt{t}$ are boolean combinations of formulas of the form $\langle\langle\Delta\rangle\rangle \circ \chi$ where $\Delta \subseteq \Gamma_k$ and χ is a disjunction of members of Act_k, for k being either 1 or 2. In the single $\langle . \sqsubseteq . \rangle$ case we are interested in the existence of a satisfying model $\overline{M}$ for $\mathtt{t}(\varphi)$ as the transitin function $\overline{o}$ of such a model can be used to determine the homomorphism h we need. For the case of $m = 2$, the part of $\overline{M}$ is played by a pair of models $\overline{M}_k = \langle \underbrace{Act_k \cup \{w_{0,k}\}}_{=\overline{W}_k}, \langle \overline{Act}_{k,j} : j \in \Gamma_k\rangle, \overline{o}_k, \overline{V}_k\rangle$ to represent the ability of coalitions withing Γ_k to enforce actions with some desired effect on behalf of agent k, $k = 1,2$. We are interested in the satisfiability of $\mathtt{t}$-translations at pairs of such models in the following sense. Consider a $\langle\langle\Delta\rangle\rangle \circ \chi \in \mathrm{Subf}(\mathtt{t}(\varphi))$ with either $\Delta \subseteq \Gamma_1$ and χ a boolean combination of atomic propositions from $\overline{AP}_1 = Act_1$, or $\Delta \subseteq \Gamma_2$ and χ a boolean combination of atomic propositions from $\overline{AP}_2 = Act_2$. We define $\overline{M}_1, \overline{M}_2, w_{0,1}, w_{0,2} \models \langle\langle\Delta\rangle\rangle \circ \chi$ as $\overline{M}_k, w_{0,k} \models \langle\langle\Delta\rangle\rangle \circ \chi$ for ψ being $\langle\langle\Delta\rangle\rangle \circ \chi$ with $\Delta \subseteq \Gamma_k$ and χ written in terms of Act_k, $k = 1,2$. The clauses for $\bot$ and for formulas built using $\Rightarrow$ are as usual.

Satisfiability at pair of models of the special type of formulas above straightforwardly reduces to the usual satisfiability at single models once $\mathtt{t}(\varphi)$ is given a disjunctive normal form: a $\mathtt{t}(\varphi)$ of this form is satisfiable iff some of its disjunctive members is, and each disjunctive member can be viewed as a conjunction of two formulas ψ_k, ψ_k being a conjunction of formulas of the form $\langle\langle\Delta\rangle\rangle \circ \chi$ with $\Delta \subseteq \Gamma_k$ and χ written in terms of $\overline{AP}_k$, $k = 1,2$. The satisfiability of $\psi_1 \wedge \psi_2$ is obviously equivalent to the satisfiability of both ψ_1 and ψ_2 in the usual sense, at a model of the type of $\overline{M}_k$.

Formulas (1) with arbitrary $\langle . \sqsubseteq . \rangle$-free φ Removing the restriction on φs to be in the flat $\langle\langle . \rangle\rangle\circ$-subset of ATL makes it necessary to synthesise an M' and the respective h with conditions such as (the many-dimensional form of) (2) associated with not just one but all the states w of M. To enable this, we first elimitate the use of $(.\mathsf{U}.)$ in φ using that $|W|$ is known.[1] Assuming that φ is $(.\mathsf{U}.)$-free, and that $m = 1$ again, for the sake of simplicity, we consider assignments $\|.\| : \mathrm{Subf}(\varphi) \to 2^W$. We are interested in the existence of an assignment $\|.\|$ such that an M' that admits a Γ-to-i homomorphism h to M exists in which φ holds at the given state w and $\{w' : M', w' \models \psi\} = \|\psi\|$ for all $\psi \in \mathrm{Subf}(\varphi)$. For ψ being some $p \in AP$ the latter condition holds iff $\|p\|$ is as detemined from the valuation V of M. For ψ being either $\bot$, or with $\Rightarrow$ as the main connective, or of the form $\langle\langle\Delta\rangle\rangle \circ \psi'$ where $\Delta \cap \Gamma = \emptyset$, $\|\psi\|$ is similarly unambiguously determined by the identities $\|\bot\| = \emptyset$, $\|\psi' \Rightarrow \psi''\| = \|\psi'\| \Rightarrow \|\psi''\|$ and $\|\langle\langle\Delta\rangle\rangle \circ \psi'\| = \{w' \in W : M, w' \models \langle\langle\Delta\rangle\rangle \circ \psi'\}$. The latter set can be computed using just ATL model-checking. Similarly, $\|\langle\langle\Delta\rangle\rangle \circ \psi'\| = \{w' \in W : M, w' \models \langle\langle(\Delta \setminus \Gamma) \cup \{i\}\rangle\rangle \circ \psi'\}$ in case $\Delta \supseteq \Gamma$. Therefore every acceptable assignment is determined unambiguously as soon as its values $\|\langle\langle\Delta\rangle\rangle \circ \psi\|$ for $\langle\langle\Delta\rangle\rangle \circ \psi \in \mathrm{Subf}(\varphi)$ such that $\emptyset \neq \Delta \cap \Gamma \neq \Gamma$ are specified, and the latter values satisfy the inclusions

$$\{w' \in W : M, w' \models \langle\langle(\Delta \setminus \Gamma)\rangle\rangle \circ \psi'\} \subseteq \|\langle\langle\Delta\rangle\rangle \circ \psi\| \subseteq \{w' \in W : M, w' \models \langle\langle(\Delta \setminus \Gamma) \cup \{i\}\rangle\rangle \circ \psi'\}.$$

Assuming an assignment $\|.\|$ of the above form, the existence of the required o' and h which link M' to M depends on the satisfiability of the conjunction

$$\bigwedge_{\substack{\langle\langle\Delta\rangle\rangle \circ \psi \in \mathrm{Subf}(\varphi) \\ \emptyset \neq \Delta \cap \Gamma \neq \Gamma}} \bigwedge_{w' \in \|\langle\langle\Delta\rangle\rangle \circ \psi\|} \bigvee_{a_{\Delta\setminus\Gamma} \in Act_{\Delta\setminus\Gamma}} \langle\langle\Delta \cap \Gamma\rangle\rangle \circ \bigvee_{a_i \in A_{i,a_{\Delta\setminus\Gamma},w,\|\psi\|}} a_i$$

at a model of the type of $\overline{M}$ already introduced above. As expected, here $A_{i,a_{\Delta\setminus\Gamma},w,\|\psi\|} = \{a_i \in Act_i : \forall c_{\Sigma\setminus(\Delta\cup\{i\})}(o(w, a_{\Delta\setminus\Gamma} \cdot a_i \cdot c_{\Sigma\setminus(\Delta\cup\{i\})}) \in X)\}$.

Obviously the algorithm implied by the above argument is only good to conclude decidability in principle because of the forbidding number of $\|.\|$s to be considered.

[1] This can cause an $O(|W|)$-blowup in the number of the subformulas of the given φ, making it clear that we are after nothing more than decidability in principle.

4 Concluding Remarks

Related Work There is an analogy between our $\langle . \sqsubseteq . \rangle$ and the refinement quantifier of *Refinement Modal Logic* [BvDF$^+$12] and its extensions to special classes of multimodal frames [HFD12]. Formal studies focusing on controlling the decisions of self-interested delegates can be found in [KW12, EPW13]. A notion of *refinement* of alternating transition systems, ATL's original type of models from [AHK97], allowing, unlike [AHKV98], the powers of different *sets* of agents to be related, was studied in [RS01]. The approach of [RS01] suggests considering a refinement modality of the form $\langle \Delta \sqsubseteq \Gamma \rangle$ with $|\Delta| \geq 1$. The authors of [RS01] stopped short of extending ATL *syntax* by such an operator. Our model-checking algorithm extends to the case of non-singleton coalition-to-coalition refinement as in our CGM-based setting in a straightforward way. Abstraction techniques with the agents being just *knowers* were studied in [ED07, CDLR09]. Abstraction involving over- and under-approximation of coalitions to contain model size was proposed in [KL11]. A formalization of teaming sub-agents under a scheduler as turn-based simulation was proposed in [GF10, GPS13]. Modelling varying the considered set of agents is addressed in *modular interpreted systems* [JÅ07, JMS13]. Distinctively, our setting is about varying the set of agents in a system by just redistributing strategic ability, with the overall activities which the system can accommodate unchanged. In CGMs, the effect of actions is defined by means of the transition function. Considering actions which are complete with a description of their effect and an additional parameter to the co-operation modality meant to specify the availability of actions to agents as in [HLW13, Her14] enables specifying delegation too, by varying availability of actions to express their changing hands with their effect on system state being transferred too. This form of delegation is, broadly speaking, complementary to our work as we propose reasoning about migrating the ability to enforce temporal conditions, and *synthesizing* implementations in terms of actions through satisfiability checking.

Some Work in Progress $\langle . \sqsubseteq . \rangle$ admits a definition with no reference to Γ-to-i homomorphisms, which enables translating the $\langle\langle . \rangle\rangle\circ$-subset of ATL$_\sqsubseteq$ into a promising looking subset of many-sorted predicate logic or, similarly, into $\langle\langle . \rangle\rangle\circ$-subsets of explicit strategy languages such as strategy logics [CHP07, MMV10]. Exploring the tractability of the translated formulas is one way of addressing satisfiability in ATL$_\sqsubseteq$, which is yet to be done. The translation gives rise to a companion operator, which holds some promise as the means for indirect axiomatization. Regarding direct axiomatization, for any fixed i and Γ, $\langle i \sqsubseteq \Gamma \rangle$ is a **KD**- and, with some adjustment to compensate for switching to the local agent vocabulary $\Sigma \setminus \{i\} \cup \Gamma$, also a **T**-modality. We have also established some non-trivial specific basic equivalences leading to a normal form, and a conventional-looking rule for introducing negative occurrences of $\langle . \sqsubseteq . \rangle$, but still lack sufficiently strong axioms for the positive occurrences.

Acknowledgements The research in this paper was partially supported through Bulgarian National Science Fund Grant DID02/32/2009. The author is thankful to the anonymous referees for their careful proof-reading, and to Valentin Goranko, Mark Ryan, Pierre Yves Schobbens and Andreas Herzig for their comments and suggestions.

References

[AHK97] Rajeev Alur, Tom Henzinger, and Orna Kupferman. Alternating-time Temporal Logic. In *Proceedings of FCS'97*, pages 100–109, 1997, doi:10.1007/3-540-49213-5_2.

[AHK02] Rajeev Alur, Tom Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):1–42, 2002, doi:10.1145/585265.585270.

[AHKV98] Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, and Moshe Y. Vardi. Alternating refinement relations. In *CONCUR*, pages 163–178, 1998, doi:10.1007/BFb0055622.

[BFD02] Olav L. Bandmann, Babak Sadighi Firozabadi, and Mads Dam. Constrained delegation. In *IEEE Symposium on Security and Privacy*, pages 131–140. IEEE Computer Society, 2002, doi:10.1109/SECPRI.2002.1004367.

[BvDF+12] Laura Bozzelli, Hans P. van Ditmarsch, Tim French, James Hales, and Sophie Pinchinat. Refinement modal logic. *CoRR*, abs/1202.3538, 2012.

[CDLR09] Mika Cohen, Mads Dam, Alessio Lomuscio, and Francesco Russo. Abstraction in model checking multi-agent systems. In *AAMAS (2)*, pages 945–952, 2009, doi:10.1145/1558109.1558144.

[CHP07] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy Logic. In *CONCUR*, pages 59–73, 2007, doi:10.1007/978-3-540-74407-8_5.

[ED07] Constantin Enea and Catalin Dima. Abstractions of multi-agent systems. In *CEEMAS*, pages 11–21, 2007, doi:10.1007/978-3-540-75254-7_2.

[EPW13] Edith Elkind, Dmitrii V. Pasechnik, and Michael Wooldridge. Strategic considerations in the design of committees. In *AAMAS*, pages 439–446, 2013.

[GF10] Giuseppe De Giacomo and Paolo Felli. Agent composition synthesis based on ATL. In *AAMAS*, pages 499–506, 2010.

[GPS13] Giuseppe De Giacomo, Fabio Patrizi, and Sebastian Sardiña. Automatic behavior composition synthesis. *Artif. Intell.*, 196:106–142, 2013, doi:10.1016/j.artint.2012.12.001.

[Her14] Andreas Herzig. Private communication, 2014.

[HFD12] James Hales, Tim French, and Rowan Davies. Refinement quantified logics of knowledge and belief for multiple agents. In *Advances in Modal Logic*, pages 317–338, 2012.

[HLW13] Andreas Herzig, Emiliano Lorini, and Dirk Walther. Reasoning about actions meets strategic logics. In *LORI*, pages 162–175, 2013, doi:10.1007/978-3-642-40948-6_13.

[JÅ07] Wojciech Jamroga and Thomas Ågotnes. Modular interpreted systems. In *AAMAS*, page 131, 2007.

[JMS13] Wojciech Jamroga, Artur Meski, and Maciej Szreter. Modularity and openness in modeling multi-agent systems. In *GandALF*, pages 224–239, 2013, doi:10.4204/EPTCS.119.19.

[KL11] Michael Köster and Peter Lohmann. Abstraction for Model Checking Modular Interpreted Systems over ATL. In *ProMAS*, pages 95–113, 2011, doi:10.1007/978-3-642-31915-0_6.

[KW12] Sarit Kraus and Michael Wooldridge. Delegating decisions in strategic settings. In *ECAI*, pages 468–473, 2012, doi:10.3233/978-1-61499-098-7-468.

[MMV10] Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi. Reasoning About Strategies. In *FSTTCS*, pages 133–144, 2010, doi:10.4230/LIPIcs.FSTTCS.2010.133.

[NR02] Timothy J. Norman and Chris Reed. Group Delegation and Responsibility. In *Proceedings of AAMAS 2002: Part 1*, AAMAS '02, pages 491–498. ACM, 2002.

[Pau02] Marc Pauly. A Modal Logic for Coalitional Power in Games. *Journal of Logic and Computation*, 12(1):149–166, 2002, doi:10.1093/logcom/12.1.149.

[RS01] Mark Ryan and Pierre-Yves Schobbens. Agents and roles: Refinement in alternating-time temporal logic. In *ATAL*, pages 100–114, 2001, doi:10.1007/3-540-45448-9_8.

[vdHWW10] Wiebe van der Hoek, Dirk Walther, and Michael Wooldridge. Reasoning about the transfer of control. *J. Artif. Intell. Res. (JAIR)*, 37:437–477, 2010, doi:10.1613/jair.2901.

A Resolution Prover for Coalition Logic

Cláudia Nalon
Departament of Computer Science
University of Brasília (Brazil)
nalon@unb.br

Lan Zhang
Information School
Capital University of Economics and Business (China)
lan@cueb.edu.cn

Clare Dixon Ullrich Hustadt
Department of Computer Science
University of Liverpool (UK)
{CLDixon,U.Hustadt}@liverpool.ac.uk

We present a prototype tool for automated reasoning for Coalition Logic, a non-normal modal logic that can be used for reasoning about cooperative agency. The theorem prover **CLProver** is based on recent work on a resolution-based calculus for Coalition Logic that operates on coalition problems, a normal form for Coalition Logic. We provide an overview of coalition problems and of the resolution-based calculus for Coalition Logic. We then give details of the implementation of **CLProver** and present the results for a comparison with an existing tableau-based solver.

1 Introduction

Coalition Logic CL is a formalism intended to describe the ability of groups of agents to achieve an outcome in a strategic game [14]. CL is a multi-modal logic with modal operators of the form $[\mathscr{A}]$, where $\mathscr{A}$ is a set of agents. The formula $[\mathscr{A}]\varphi$ reads as *the coalition $\mathscr{A}$ has a strategy to achieve φ*, where φ is a formula. We note that CL is a non-normal modal logic, as the schema that represents *additivity*, $[\mathscr{A}]\varphi \wedge [\mathscr{A}]\psi \Rightarrow [\mathscr{A}](\varphi \wedge \psi)$, is not valid. However, *monotonicity*, $[\mathscr{A}](\varphi \wedge \psi) \Rightarrow [\mathscr{A}]\varphi \wedge [\mathscr{A}]\psi$, holds.

Coalition Logic is equivalent to the next-time fragment of *Alternating-Time Temporal Logic* (ATL) [1, 5], where $[\mathscr{A}]\varphi$ translates into $\langle\langle\mathscr{A}\rangle\rangle\bigcirc\varphi$ (read as *the coalition $\mathscr{A}$ can ensure φ at the next moment in time*). The satisfiability problems for ATL and CL are EXPTIME-complete [16] and PSPACE-complete [14], respectively. Proof methods for these logics include, for instance, tableau-based methods for ATL [16, 6] and a tableau-based method for CL [8].

In order to make the paper self-contained, we present here the resolution-based calculus for CL, $\mathsf{RES}_{\mathsf{CL}}$ [12]. As to the best of our knowledge, there are no other resolution-based methods for either ATL or CL. Providing such a method for CL gives the user a choice of proof methods. Several comparisons of tableau algorithms and resolution methods [10, 7] indicate that there is no overall best approach: for some classes of formulae tableau algorithms perform better whilst on others resolution performs better. So, with a choice of different provers, for the best result, the user could run several in parallel or the one most likely to succeed depending on the type of the input formulae. $\mathsf{RES}_{\mathsf{CL}}$ is sound, complete, and terminating as shown in [12].

The paper is organised as follows. In the next section, we present the syntax, axiomatisation, and semantics of CL. In Section 3, we introduce the resolution-based method for CL, the main results, and provide a small example. In Section 4, we introduce the theorem-prover for CL. We give details of the implementation and discuss the results for a comparison with an existing tool. Conclusions and future work are given in Section 5.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 65–73, doi:10.4204/EPTCS.146.9

2 Coalition Logic

As in [6], we define $\Sigma \subset \mathbb{N}$ to be a finite, non-empty set of agents. A **coalition** $\mathscr{A}$ is a subset of Σ. Formulae in CL are constructed from propositional symbols ($\Pi = \{p, q, r, \ldots, p_1, q_1, r_1, \ldots\}$) and constants (**true**, **false**), together with Boolean operators ($\neg$, for negation, and $\wedge$, for conjunction) and coalition modalities. Formulae whose main operator is classical are built in the usual way. A **coalition modality** is either of the form $[\mathscr{A}]\varphi$ or $\langle\mathscr{A}\rangle\varphi$, where φ is a well-formed CL formula. The coalition operator $\langle\mathscr{A}\rangle$ is the dual of $[\mathscr{A}]$, that is, $\langle\mathscr{A}\rangle\varphi$ is an abbreviation for $\neg[\mathscr{A}]\neg\varphi$, for every coalition $\mathscr{A}$ and formula φ. We denote by $\mathsf{WFF}_{\mathsf{CL}}$ the set of CL well-formed formulae. Parentheses will be omitted if the reading is not ambiguous. We also omit the curly brackets within modalities. For instance, we write $[1,2]\varphi$ instead of $[\{1,2\}]\varphi$. Formulae of the form $\bigvee \varphi_i$ (resp. $\bigwedge \varphi_i$), $1 \leq i \leq n$, $n \in \mathbb{N}$, $\varphi_i \in \mathsf{WFF}_{\mathsf{CL}}$, represent arbitrary disjunctions (resp. conjunctions) of formulae. If $n = 0$, $\bigvee \varphi_i$ (resp. $\bigwedge \varphi_i$) is called the **empty disjunction** (resp. **empty conjunction**), denoted by **false** (resp. **true**).

A **literal** is either p or $\neg p$, for $p \in \Pi$. For a literal l of the form $\neg p$, where p is a propositional symbol, $\neg l$ denotes p; for a literal l of the form p, $\neg l$ denotes $\neg p$. The literals l and $\neg l$ are called **complementary literals**. We assume that literals are in simplified form, that is, $\neg\neg l$ is assumed to be l. A **positive coalition formula** (resp. **negative coalition formula**) is a formula of the form $[\mathscr{A}]\varphi$ (resp. $\langle\mathscr{A}\rangle\varphi$), where $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$. A **coalition formula** is either a positive or a negative coalition formula.

Coalition logic can be axiomatised by the following schemata (where $\mathscr{A}, \mathscr{A}'$ are coalitions and φ, φ_1, φ_2 are well-formed formulae) [14]:

$$
\begin{array}{lcl}
\bot & : & \neg[\mathscr{A}]\textbf{false} \\
\top & : & [\mathscr{A}]\textbf{true} \\
\Sigma & : & \neg[\emptyset]\neg\varphi \Rightarrow [\Sigma]\varphi \\
\mathbf{M} & : & [\mathscr{A}](\varphi_1 \wedge \varphi_2) \Rightarrow [\mathscr{A}]\varphi_1 \\
\mathbf{S} & : & [\mathscr{A}]\varphi_1 \wedge [\mathscr{A}']\varphi_2 \Rightarrow [\mathscr{A} \cup \mathscr{A}'](\varphi_1 \wedge \varphi_2), \text{ if } \mathscr{A} \cap \mathscr{A}' = \emptyset
\end{array}
$$

together with propositional tautologies and the following inference rules: **modus ponens** (from φ_1 and $\varphi_1 \Rightarrow \varphi_2$ infer φ_2) and **equivalence** (from $\varphi_1 \Leftrightarrow \varphi_2$ infer $[\mathscr{A}]\varphi_1 \Leftrightarrow [\mathscr{A}]\varphi_2$). It can be shown that the inference rule **monotonicity** (from $\varphi_1 \Rightarrow \varphi_2$ infer $[\mathscr{A}]\varphi_1 \Rightarrow [\mathscr{A}]\varphi_2$) is a derivable rule in this system. The next result will be used later.

Lemma 1 *The formula* $[\mathscr{A}]\psi_1 \wedge \langle\mathscr{B}\rangle\psi_2 \Rightarrow \langle\mathscr{B} \setminus \mathscr{A}\rangle(\psi_1 \wedge \psi_2)$ *where* $\mathscr{A}$ *and* $\mathscr{B}$ *are coalitions,* $\mathscr{A} \subseteq \mathscr{B}$, *and* $\psi_1, \psi_2 \in \mathsf{WFF}_{\mathsf{CL}}$, *is valid.*

Proof.

1.	$[\mathscr{A}]\psi_1 \wedge [\mathscr{B} \setminus \mathscr{A}](\psi_1 \Rightarrow \neg\psi_2) \Rightarrow [\mathscr{B}](\psi_1 \wedge (\psi_1 \Rightarrow \neg\psi_2))$	$\mathbf{S}, \mathscr{A}' = \mathscr{B} \setminus \mathscr{A}$ $\varphi_1 = \psi_1, \varphi_2 = \psi_1 \Rightarrow \neg\psi_2$
2.	$\psi_1 \wedge (\psi_1 \Rightarrow \neg\psi_2) \Rightarrow \neg\psi_2$	*propositional tautology*
3.	$[\mathscr{B}](\psi_1 \wedge (\psi_1 \Rightarrow \neg\psi_2)) \Rightarrow [\mathscr{B}]\neg\psi_2$	2, *monotonicity*
4.	$[\mathscr{A}]\psi_1 \wedge [\mathscr{B} \setminus \mathscr{A}](\psi_1 \Rightarrow \neg\psi_2) \Rightarrow [\mathscr{B}]\neg\psi_2$	1,3, *chaining*
5.	$[\mathscr{A}]\psi_1 \wedge \neg[\mathscr{B}]\neg\psi_2 \Rightarrow \neg[\mathscr{B} \setminus \mathscr{A}](\neg\psi_1 \vee \neg\psi_2)$	4, *rewriting*
6.	$[\mathscr{A}]\psi_1 \wedge \langle\mathscr{B}\rangle\neg\neg\psi_2 \Rightarrow \langle\mathscr{B} \setminus \mathscr{A}\rangle\neg(\neg\psi_1 \vee \neg\psi_2)$	5, *def. dual*
7.	$[\mathscr{A}]\psi_1 \wedge \langle\mathscr{B}\rangle\psi_2 \Rightarrow \langle\mathscr{B} \setminus \mathscr{A}\rangle(\psi_1 \wedge \psi_2)$	6, *rewriting* $\square$

The semantics of CL is given in terms of *Concurrent Game Structures* (CGS) [2] and it is *positional*, that is, agents have no memory of their past decisions and, thus, those decisions are made by taking into account only the current state. We note that the semantics of CL is often presented in terms of

Multiplayer Game Models (MGMs) [13]. Note also that MGMs yield the same set of validities as CGSs [5]. As we intend to extend the proof method given here to full ATL, the correctness proofs are based on the tableau procedure for full ATL [6] and we follow the semantics presentation given there.

Def. 1 A **Concurrent Game Frame** (CGF) is a tuple $\mathscr{F} = (\Sigma, \mathscr{S}, s_0, d, \delta)$, where

- Σ is a finite non-empty set of **agents**;
- $\mathscr{S}$ is a non-empty set of **states**, with a distinguished state s_0, termed *initial state*;
- $d : \Sigma \times \mathscr{S} \longrightarrow \mathbb{N}^+$, where the natural number $d(a,s) \geq 1$ represents the **number of moves** that the agent a has at the state s. Every **move** for agent a at the state s is identified by a number between 0 and $d(a,s)-1$. Let $D(a,s) = \{0,\ldots,d(a,s)-1\}$ be the set of all moves available to agent a at s. For a state s, a **move vector** is a k-tuple $(\sigma_1,\ldots,\sigma_k)$, where $k = |\Sigma|$, such that $0 \leq \sigma_a \leq d(a,s)-1$, for all $a \in \Sigma$. Intuitively, σ_a represents an arbitrary move of agent a in s. Let $D(s) = \Pi_{a\in\Sigma} D(a,s)$ be the set of all move vectors at s. We denote by σ an arbitrary member of $D(s)$.
- δ is a **transition function** that assigns to every $s \in \mathscr{S}$ and every $\sigma \in D(s)$ a state $\delta(s,\sigma) \in \mathscr{S}$ that results from s if every agent $a \in \Sigma$ plays move σ_a.

In the following, let $\mathscr{F} = (\Sigma, \mathscr{S}, s_0, d, \delta)$ be a CGF with $s, s' \in \mathscr{S}$. We say that s' is a **successor** of s (an s-successor) if $s' = \delta(s,\sigma)$, for some $\sigma \in D(s)$. If κ is a tuple, then κ_n (or $\kappa(n)$) denotes the n-th element of κ. Let $|\Sigma| = k$ and let $\mathscr{A} \subseteq \Sigma$ be a coalition. An $\mathscr{A}$**-move** $\sigma_\mathscr{A}$ at $s \in \mathscr{S}$ is a k-tuple such that $\sigma_\mathscr{A}(a) \in D(a,s)$ for every $a \in \mathscr{A}$ and $\sigma_\mathscr{A}(a') = *$ (i.e. an arbitrary move) for every $a' \notin \mathscr{A}$. We denote by $D(\mathscr{A},s)$ the set of all $\mathscr{A}$-moves at state s. A move vector σ **extends** an $\mathscr{A}$-move vector $\sigma_\mathscr{A}$, denoted by $\sigma_\mathscr{A} \sqsubseteq \sigma$ or $\sigma \sqsupseteq \sigma_\mathscr{A}$, if $\sigma(a) = \sigma_\mathscr{A}(a)$ for every $a \in \mathscr{A}$. Let $\sigma_\mathscr{A} \in D(\mathscr{A},s)$ be an $\mathscr{A}$-move. The **outcome** of $\sigma_\mathscr{A}$ at s, denoted by $out(s,\sigma_\mathscr{A})$, is the set of all states $s' \in \mathscr{S}$ for which there exists a move vector $\sigma \in D(s)$ such that $\sigma_\mathscr{A} \sqsubseteq \sigma$ and $\delta(s,\sigma) = s'$.

Def. 2 A **Concurrent Game Model** (CGM) is a tuple $\mathscr{M} = (\mathscr{F}, \Pi, \pi)$, where $\mathscr{F} = (\Sigma, \mathscr{S}, s_0, d, \delta)$ is a CGF; Π is the set of propositional symbols; and $\pi : \mathscr{S} \longrightarrow 2^\Pi$ is a valuation function.

Def. 3 Let $\mathscr{M} = (\Sigma, \mathscr{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM with $s \in \mathscr{S}$. The satisfaction relation, denoted by $\models$, is inductively defined as follows.

- $\langle \mathscr{M}, s\rangle \models$ **true**;
- $\langle \mathscr{M}, s\rangle \models p$ iff $p \in \pi(s)$, for all $p \in \Pi$;
- $\langle \mathscr{M}, s\rangle \models \neg\varphi$ iff $\langle \mathscr{M}, s\rangle \not\models \varphi$;
- $\langle \mathscr{M}, s\rangle \models \varphi \wedge \psi$ iff $\langle \mathscr{M}, s\rangle \models \varphi$ and $\langle \mathscr{M}, s\rangle \models \psi$;
- $\langle \mathscr{M}, s\rangle \models [\mathscr{A}]\varphi$ iff there exists a $\mathscr{A}$-move $\sigma_\mathscr{A} \in D(\mathscr{A},s)$ s.t. $\langle \mathscr{M}, s'\rangle \models \varphi$ for all $s' \in out(s,\sigma_\mathscr{A})$;
- $\langle \mathscr{M}, s\rangle \models \langle\mathscr{A}\rangle\varphi$ iff for all $\mathscr{A}$-moves $\sigma_\mathscr{A} \in D(\mathscr{A},s)$ exists $s' \in out(s,\sigma_\mathscr{A})$ s.t. $\langle \mathscr{M}, s'\rangle \models \varphi$.

Semantics of **false**, disjunctions, and implications are given in the usual way. Given a model $\mathscr{M}$, a state s in $\mathscr{M}$, and a formula φ, if $\langle \mathscr{M}, s\rangle \models \varphi$, $s \in \mathscr{S}$, we say that φ is **satisfied at the state** s **in** $\mathscr{M}$.

In this work, we consider *tight satisfiability*, i.e. the evaluation of a formula φ depends only on the agents occurring in φ [16]. We denote by Σ_φ, where $\Sigma_\varphi \subseteq \Sigma$, the set of agents occurring in a well-formed formula φ. If Φ is a set of well-formed formulae, $\Sigma_\Phi \subseteq \Sigma$ denotes $\bigcup_{\varphi\in\Phi}\Sigma_\varphi$. Let $\varphi \in \mathsf{WFF}_{\mathsf{CL}}$ and $\mathscr{M} = (\Sigma_\varphi, \mathscr{S}, s_0, d, \delta, \Pi, \pi)$ be a CGM. Formulae are interpreted with respect to the distinguished world s_0. Thus, a formula φ is said to be **satisfiable in** $\mathscr{M}$, denoted by $\mathscr{M} \models \varphi$, if $\langle \mathscr{M}, s_0\rangle \models \varphi$; it is said to be **satisfiable** if there is a model $\mathscr{M}$ such that $\langle \mathscr{M}, s_0\rangle \models \varphi$; and it is said to be **valid** if for all models $\mathscr{M}$ we have $\langle \mathscr{M}, s_0\rangle \models \varphi$. A finite set $\Gamma \subset \mathsf{WFF}_{\mathsf{CL}}$ is **satisfiable in a state** s **in** $\mathscr{M}$, denoted by $\langle \mathscr{M}, s\rangle \models \Gamma$, if for all $\gamma_i \in \Gamma$, $0 \leq i \leq n$, $n \in \mathbb{N}$, $\langle \mathscr{M}, s\rangle \models \gamma_i$; Γ is **satisfiable in a model** $\mathscr{M}$, $\mathscr{M} \models \Gamma$, if $\langle \mathscr{M}, s_0\rangle \models \Gamma$; and Γ is **satisfiable**, if there is a model $\mathscr{M}$ such that $\mathscr{M} \models \Gamma$.

3 Resolution Calculus

The resolution calculus for CL, $\mathsf{RES}_{\mathsf{CL}}$, operates on sets of clauses. A formula in CL is firstly converted into a coalition problem, which is then transformed into a coalition problem in *Divided Separated Normal Form for Coalition Logic*, $\mathsf{DSNF}_{\mathsf{CL}}$.

Def. 4 A **coalition problem** is a tuple $(\mathscr{I},\mathscr{U},\mathscr{N})$, where $\mathscr{I}$, the set of initial formulae, is a finite set of propositional formulae; $\mathscr{U}$, the set of global formulae, is a finite set of formulae in $\mathsf{WFF}_{\mathsf{CL}}$; and $\mathscr{N}$, the set of coalition formulae, is a finite set of coalition formulae, i.e. those formulae in which a coalition modality occurs.

The semantics of coalition problems assumes that initial formulae hold at the initial state; and that global and coalition formulae hold at every state of a model.

Def. 5 Given a coalition problem $\mathscr{C} = (\mathscr{I},\mathscr{U},\mathscr{N})$, we denote by $\Sigma_{\mathscr{C}}$ the set of agents $\Sigma_{\mathscr{U}\cup\mathscr{N}}$. If $\mathscr{C} = (\mathscr{I},\mathscr{U},\mathscr{N})$ is a coalition problem and $\mathscr{M} = (\Sigma_{\mathscr{C}},\mathscr{S},s_0,d,\delta,\Pi,\pi)$ is a CGM, then $\mathscr{M} \models \mathscr{C}$ if, and only if, $\langle\mathscr{M},s_0\rangle \models \mathscr{I}$ and $\langle\mathscr{M},s\rangle \models \mathscr{U}\cup\mathscr{N}$, for all $s \in \mathscr{S}$. We say that $\mathscr{C} = (\mathscr{I},\mathscr{U},\mathscr{N})$ is **satisfiable**, if there is a model $\mathscr{M}$ such that $\mathscr{M} \models \mathscr{C}$.

In order to apply the resolution method, we further require that formulae within each of those sets are in *clausal form*: **initial clauses** and **global clauses** are of the form $\bigvee_{j=1}^{n} l_j$; **positive coalition clauses** are of the form $\bigwedge_{i=1}^{m} l'_i \Rightarrow [\mathscr{A}]\bigvee_{j=1}^{n} l_j$; and **negative coalition clauses** are of the form $\bigwedge_{i=1}^{m} l'_i \Rightarrow \langle\mathscr{A}\rangle\bigvee_{j=1}^{n} l_j$; where $m,n \geq 0$ and l'_i, l_j, for all $1 \leq i \leq m$, $1 \leq j \leq n$, are literals or constants. We assume that clauses are kept in the simplest form by means of usual Boolean simplification rules. Tautologies are removed from the set of clauses as they cannot contribute to finding a contradiction. A **coalition problem in** $\mathsf{DSNF}_{\mathsf{CL}}$ is a coalition problem $(\mathscr{I},\mathscr{U},\mathscr{N})$ such that $\mathscr{I}$ is a set of initial clauses, $\mathscr{U}$ is a set of global clauses, and $\mathscr{N}$ is a set of positive and negative coalition clauses.

The transformation of a coalition logic formula into a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ is analogous to the approach taken in [4]. The transformation of a formula into a **coalition problem in** $\mathsf{DSNF}_{\mathsf{CL}}$, which is given in [11, 12], reduces the number of operators and separates the contexts to which the resolution inference rules are applied, but may add new propositional symbols.

The set of inference rules for $\mathsf{RES}_{\mathsf{CL}}$ are given as follows. Let $(\mathscr{I},\mathscr{U},\mathscr{N})$ be a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$; C,C' be conjunctions of literals; D,D' be disjunctions of literals; l,l_i be literals; and $\mathscr{A},\mathscr{B} \subseteq \Sigma$ be coalitions (where Σ is the set of all agents). The first rule, **IRES1**, is classical resolution applied to clauses which are true at the initial state. The next inference rule, **GRES1**, performs resolution on clauses which are true in all states.

$$\textbf{IRES1}\quad \frac{D\vee l \in \mathscr{I} \qquad D'\vee\neg l \in \mathscr{I}\cup\mathscr{U}}{D\vee D' \in \mathscr{I}} \qquad\qquad \textbf{GRES1}\quad \frac{D\vee l \in \mathscr{U} \qquad D'\vee\neg l \in \mathscr{U}}{D\vee D' \in \mathscr{U}}$$

Soundness of **IRES1** and **GRES1** follow from the semantics of coalition problems and the soundness result for classical propositional resolution [15]. The following rules perform resolution on positive and negative coalition clauses.

$$\textbf{CRES1}\quad \mathscr{A}\cap\mathscr{B}=\emptyset\quad \frac{C \Rightarrow [\mathscr{A}](D\vee l) \in \mathscr{N} \qquad C' \Rightarrow [\mathscr{B}](D'\vee\neg l) \in \mathscr{N}}{C\wedge C' \Rightarrow [\mathscr{A}\cup\mathscr{B}](D\vee D') \in \mathscr{N}} \qquad \textbf{CRES2}\quad \frac{D\vee l \in \mathscr{U} \qquad C \Rightarrow [\mathscr{A}](D'\vee\neg l) \in \mathscr{N}}{C \Rightarrow [\mathscr{A}](D\vee D') \in \mathscr{N}}$$

$$\textbf{CRES3}\quad \mathscr{A}\subseteq\mathscr{B}\quad \frac{C \Rightarrow [\mathscr{A}](D\vee l) \in \mathscr{N} \qquad C' \Rightarrow \langle\mathscr{B}\rangle(D'\vee\neg l) \in \mathscr{N}}{C\wedge C' \Rightarrow \langle\mathscr{B}\setminus\mathscr{A}\rangle(D\vee D') \in \mathscr{N}} \qquad \textbf{CRES4}\quad \frac{D\vee l \in \mathscr{U} \qquad C \Rightarrow \langle\mathscr{A}\rangle(D'\vee\neg l) \in \mathscr{N}}{C \Rightarrow \langle\mathscr{A}\rangle(D\vee D') \in \mathscr{N}}$$

Soundness of the inference rules **CRES1-4** follow from the axiomatisation of CL, given in Section 2. We give sketches of the proofs here. Let $\mathscr{M}$ be a CGM and $s \in \mathscr{M}$ a state. Recall that coalition clauses are satisfied at any state in $\mathscr{M}$. For **CRES1**, if $\langle \mathscr{M}, s\rangle \models C \wedge C'$, by the semantics of conjunction and implication, we have that $\langle \mathscr{M}, s\rangle \models C \wedge C' \Rightarrow [\mathscr{A}](D \vee l) \wedge [\mathscr{B}](D' \vee \neg l)$. By axiom **S**, we have that $[\mathscr{A}](D \vee l) \wedge [\mathscr{B}](D' \vee \neg l)$ implies $[\mathscr{A} \cup \mathscr{B}]((D \vee l) \wedge (D' \vee \neg l))$. Therefore, $\langle \mathscr{M}, s\rangle \models C \wedge C' \Rightarrow [\mathscr{A} \cup \mathscr{B}]((D \vee l) \wedge (D' \vee \neg l))$. By classical resolution applied within the successor states, we obtain that $\langle \mathscr{M}, s\rangle \models C \wedge C' \Rightarrow [\mathscr{A} \cup \mathscr{B}](D \vee D')$. For **CRES3**, by Lemma 1, we have that $[\mathscr{A}](D \vee l) \wedge \langle \mathscr{B}\rangle(D' \vee \neg l) \Rightarrow \langle \mathscr{B} \setminus \mathscr{A}\rangle((D \vee l) \wedge (D' \vee \neg l))$, with $\mathscr{A} \subseteq \mathscr{B}$, is valid. If $\langle \mathscr{M}, s\rangle \models C \wedge C'$, by the semantics of implication, we have that $\langle \mathscr{M}, s\rangle \models C \wedge C' \Rightarrow \langle \mathscr{B} \setminus \mathscr{A}\rangle((D \vee l) \wedge (D' \vee \neg l))$. Applying classical resolution within the successor states, we obtain that $\langle \mathscr{M}, s\rangle \models C \wedge C' \Rightarrow \langle \mathscr{B} \setminus \mathscr{A}\rangle(D \vee D')$. Soundness of the inference rules **CRES2** and **CRES4** follow from the above and the semantics of coalition problems: as $D \vee l$ in $\mathscr{U}$ is satisfied at all states, we have that $\mathbf{true} \Rightarrow [\emptyset](D \vee l)$ is also satisfied at all states.

The next two inference rules are justified by the axioms $\bot$ and $\top$, given by $\neg[\mathscr{A}]\mathbf{false}$ and $[\mathscr{A}]\mathbf{true}$, respectively, which imply that the consequent in both rewriting rules cannot be satisfied.

$$\textbf{RW1}\ \frac{\bigwedge_{i=1}^{n} l_i \Rightarrow [\mathscr{A}]\mathbf{false} \quad \in \mathscr{N}}{\bigvee_{i=1}^{n} \neg l_i \quad \in \mathscr{U}} \qquad \textbf{RW2}\ \frac{\bigwedge_{i=1}^{n} l_i \Rightarrow \langle \mathscr{A}\rangle\mathbf{false} \quad \in \mathscr{N}}{\bigvee_{i=1}^{n} \neg l_i \quad \in \mathscr{U}}$$

As sketched above, the resolution-based calculus for Coalition Logic is sound.

Theorem 1 (Soundness) *Let $\mathscr{C}$ be a coalition problem in* $\mathsf{DSNF}_{\mathsf{CL}}$. *Let $\mathscr{C}'$ be the coalition problem in* $\mathsf{DSNF}_{\mathsf{CL}}$ *obtained from $\mathscr{C}$ by applying any of the inference rules* **IRES1**, **GRES1**, **CRES1-4** *or* **RW1-2** *to $\mathscr{C}$. If $\mathscr{C}$ is satisfiable, then $\mathscr{C}'$ is satisfiable.*

A **derivation** from a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $\mathscr{C} = (\mathscr{I}, \mathscr{U}, \mathscr{N})$ by $\mathsf{RES}_{\mathsf{CL}}$ is a sequence $\mathscr{C}_0, \mathscr{C}_1, \mathscr{C}_2, \ldots$ of problems such that $\mathscr{C}_0 = \mathscr{C}$, $\mathscr{C}_i = (\mathscr{I}_i, \mathscr{U}_i, \mathscr{N}_i)$, and $\mathscr{C}_{i+1}$ is either $(\mathscr{I}_i \cup \{D\}, \mathscr{U}_i, \mathscr{N}_i)$, where D is the conclusion of **IRES1**; $(\mathscr{I}_i, \mathscr{U}_i \cup \{D\}, \mathscr{N}_i)$, where D is the conclusion of **GRES1**, **RW1**, or **RW2**; or $(\mathscr{I}_i, \mathscr{U}_i, \mathscr{N}_i \cup \{D\})$, where D is the conclusion of **CRES1**, **CRES2**, **CRES3**, or **CRES4**; and D is not a tautology.

A **refutation** for a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $\mathscr{C} = (\mathscr{I}, \mathscr{U}, \mathscr{N})$ (by $\mathsf{RES}_{\mathsf{CL}}$) is a derivation from $\mathscr{C}$ such that for some $i \geq 0$, $\mathscr{C}_i = (\mathscr{I}_i, \mathscr{U}_i, \mathscr{N}_i)$ contains a contradiction, where a contradiction is given by either $\mathbf{false} \in \mathscr{I}_i$ or $\mathbf{false} \in \mathscr{U}_i$. A derivation *terminates* if, and only if, either a contradiction is derived or no new clauses can be derived by further application of resolution rules of $\mathsf{RES}_{\mathsf{CL}}$.

The completeness proof for $\mathsf{RES}_{\mathsf{CL}}$ is based on the tableau construction given in [6]. Given an unsatisfiable coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$ $\mathscr{C}$, an initial tableau is obtained by this construction which is then reduced to an empty tableau via a sequence of deletion steps. We show that each deletion step corresponds to an application of the resolution inference rules to (sub)sets of clauses in $\mathscr{C}$ or clauses previously derived from $\mathscr{C}$. The derivation constructed in this way is shown to be a refutation of $\mathscr{C}$.

Theorem 2 (Completeness) *Let $\mathscr{C} = (\mathscr{I}, \mathscr{U}, \mathscr{N})$ be an unsatisfiable coalition problem in* $\mathsf{DSNF}_{\mathsf{CL}}$. *Then there is a refutation for $\mathscr{C}$ using the inference rules* **IRES1**, **GRES1**, **CRES1-4**, *and* **RW1-2**.

The proof that every derivation terminates is trivial and based on the fact that we have a finite number of clauses that can be expressed. As the number of propositional symbols after translation into the normal form is finite and the inference rules do not introduce new propositional symbols, we have that the number of possible literals occurring in clauses is finite and the number of conjunctions (resp. disjunctions) on the left-hand side (resp. right-hand side) of clauses is finite (modulo simplification). As the number of agents is finite, the number of coalition modalities that can be introduced by inference rules is also finite. Thus, only a finite number of clauses can be expressed (modulo simplification), so at some point either we derive a contradiction or no new clauses can be generated.

Theorem 3 *Let $\mathscr{C} = (\mathscr{I}, \mathscr{U}, \mathscr{N})$ be a coalition problem in* $\mathsf{DSNF}_{\mathsf{CL}}$. *Then any derivation from $\mathscr{C}$ by* $\mathsf{RES}_{\mathsf{CL}}$ *terminates.*

Full proofs for soundness, completeness, termination, and complexity of the resolution-based method for CL are given in [11, 12].

Example 1 *We show a simple example, adapted from [9], of the application of* $\mathsf{RES}_{\mathsf{CL}}$ *to a problem involving the cooperation of agents. There are two agents (*1 *and* 2*) and two toggle switches. For each agent $a = 1,2$, there are two possible actions: $[a]tog_a \wedge [a]\neg tog_a$, where tog_a denotes that the agent a can toggle the switch a (clauses 3, 9–13). The light is initially off, i.e. we have that $t_0 \Rightarrow \neg l$ (clauses 1 and 2). If the light is off and the switch is toggled, then at the next moment the light is on: $tog_a \wedge \neg l \Rightarrow [a]l$ (clauses 5 and 6). Similarly, if the light is on and the agent toggles the switch, then at the next moment the light is off: $tog_a \wedge l \Rightarrow [a]\neg l$ (clauses 7 and 8). We prove that the agents can cooperate to turn on the light, that is, we introduce the clauses 4 and 14, which corresponds to the negation of $[1,2]l$.*

1.			t_0	$[\mathscr{I}]$	*14.*	t_4	$\Rightarrow$	$[\emptyset]\neg l$	$[\mathscr{N}]$
2.			$\neg t_0 \vee \neg l$	$[\mathscr{U}]$	*15.*			$\neg t_0 \vee t_4$	$[\mathscr{U}$,**GRES1**,*3, 4*$]$
3.			$\neg t_0 \vee t_1$	$[\mathscr{U}]$	*16.*	$t_4 \wedge tog_1 \wedge \neg l$	$\Rightarrow$	$[1]$**false**	$[\mathscr{N}$,**CRES1**,*5, 14*$]$
4.			$\neg t_1 \vee t_4$	$[\mathscr{U}]$	*17.*	t_1	$\Rightarrow$	$[\emptyset]t_4$	$[\mathscr{N}$,**CRES2**,*13, 4*$]$
5.	$tog_1 \wedge \neg l$	$\Rightarrow$	$[1]l$	$[\mathscr{N}]$	*18.*			$l \vee \neg t_4 \vee \neg tog_1$	$[\mathscr{U}$,**RW1**,*16*$]$
6.	$tog_2 \wedge \neg l$	$\Rightarrow$	$[2]l$	$[\mathscr{N}]$	*19.*	t_1	$\Rightarrow$	$[\emptyset]l \vee \neg tog_1$	$[\mathscr{N}$,**CRES2**,*17, 18*$]$
7.	$tog_1 \wedge l$	$\Rightarrow$	$[1]\neg l$	$[\mathscr{N}]$	*20.*	t_1	$\Rightarrow$	$[1]l$	$[\mathscr{N}$,**CRES1**,*19, 9*$]$
8.	$tog_2 \wedge l$	$\Rightarrow$	$[2]\neg l$	$[\mathscr{N}]$	*21.*	$t_1 \wedge t_4$	$\Rightarrow$	$[1]$**false**	$[\mathscr{N}$,**CRES1**,*20, 14*$]$
9.	t_1	$\Rightarrow$	$[1]tog_1$	$[\mathscr{N}]$	*22.*			$\neg t_1 \vee \neg t_4$	$[\mathscr{U}$,**RW1**,*21*$]$
10.	t_1	$\Rightarrow$	$[2]tog_2$	$[\mathscr{N}]$	*23.*			$\neg t_0 \vee \neg t_1$	$[\mathscr{U}$,**GRES1**,*22, 15*$]$
11.	t_1	$\Rightarrow$	$[1]\neg tog_1$	$[\mathscr{N}]$	*24.*			$\neg t_0$	$[\mathscr{U}$,**GRES1**,*23, 3*$]$
12.	t_1	$\Rightarrow$	$[2]\neg tog_2$	$[\mathscr{N}]$	*25.*			**false**	$[\mathscr{I}$,**IRES1**,*1, 24*$]$
13.	t_1	$\Rightarrow$	$[\emptyset]t_1$	$[\mathscr{N}]$					

4 CLProver

CLProver is a prototype implementation of the resolution-based method given in [12]. The prover is written in SWI-Prolog (Multi-threaded, 64 bits, Version 6.0.2) and the compiled binaries for Linux x86_64 together with instructions for usage and example files are available at `http://www.cic.unb.br/docentes/nalon/#software`.

The prover recurs over the set of clauses using breadth-first search for a proof. The resolution inference rules for CL are in fact variations of the propositional resolution rule. Before presenting the general form of the inference rules, we explain the data structures that are employed by the prover. A *clause core* is implemented as a list with three elements, all of which are lists: a list of literals on the left-hand side of a clause, a list of agents, and a list of literals on the right-hand side of a clause. The only operator allowed within the lists of literals is the negation operator, `neg`. Clauses are then given as Prolog lists, with four elements. The first element is the clause number, the second is the clause core, the third is the justification ('given', if the clause is an input clause; or a list containing the numbers of the clauses from which it was derived, together with the literal being resolved, and the inference rule applied), and the fourth is an indication to which set within a coalition problem the clause belongs ('i' for initial, 'u' for global, 'p' for positive, and 'n' for negative). Thus, for instance, the clauses 1, 3, and 20 from Example 1

are represented as `[1,[[],[],[t0]],[given],i]`, `[3,[[],[],[neg t0, t1]],[given],u]`, and `[20,[[t1],[1],[l]],[9,19,tog1,cres1],n]`, respectively.

Given this representation, the propositional resolution inference rule is modified in such a way that a clause `[_,[LHS1],[AG1],[RHS1],_,S1]` is resolved with `[_,[LHS2],[AG2],[RHS2],_,S2]`, if such clauses meet the side conditions given by the inference rules presented in Section 3. For instance, the rule **CRES1** is applied if both `S1` and `S2` are equal to 'p' and if the intersection between `LHS1` and `LHS2` is empty. The prover then recurs over the set of initial, global and coalition clauses using the following procedure (where S is a saturated set of clauses and N is a non-saturated set of clauses):

procedure resolution(S, N)
while (N $\neq \emptyset$ and **false** $\notin$ N)
do Given $\leftarrow$ choose(N);
N $\leftarrow$ N $\setminus$ {Given};
S $\leftarrow$ S $\cup$ {Given};
New $\leftarrow$ rewrite(res(Given,S));
`/* Forward Subsumption */`
N $\leftarrow$ sub(sub(New,S),N);
end-while
if false $\in$ N **then** S $\leftarrow$ S $\cup$ {**false**};
return S;

where choose(N) randomly picks a clause in N; res(C, N) is the set of all non-tautological resolvents in simplified form derivable between a clause C and a set of clauses N by one of the inference rules; rewrite(N) is the union of N and the set of clauses derived by the rewriting rules; and sub(M,N) is the set of clauses in M not subsumed by a clause in N. Forward subsumption is implemented for both the propositional and modal portions of the language. For the propositional part, a clause D in $\mathscr{I}$ (resp. $\mathscr{U}$) is subsumed by a clause D' in $\mathscr{I} \cup \mathscr{U}$ (resp. $\mathscr{U}$) if $D' \subseteq D$. A positive coalition clause $C \Rightarrow [\mathscr{A}]D$ is subsumed by another positive coalition clause $C' \Rightarrow [\mathscr{A}']D'$, if $C' \subseteq C$, $\mathscr{A}' \subseteq \mathscr{A}$, and $D' \subseteq D$. A negative coalition clause $C \Rightarrow \langle\mathscr{A}\rangle D$ is subsumed by another negative coalition clause $C' \Rightarrow \langle\mathscr{A}'\rangle D'$, if $C' \subseteq C$, $\mathscr{A} \subseteq \mathscr{A}'$, and $D' \subseteq D$. Some other forms of subsumption have not been implemented in the current version of the prover, as, for instance, coalition clauses which are subsumed by global clauses.

The current version of **CLProver** is a prototype. The prover implements subsumption, but it does not implement any of the usual performance improving techniques for resolution-based methods. For example, the function choose(N) does not use any heuristic to determine what given clause to pick. Further refinements of the resolution calculus, for example, ordered resolution or the the use of a set of support strategy would also greatly improve the performance of the prover. **CLProver**, however, performs well when compared with both versions of another tool, namely, TATL, a tableau-based prover for ATL [3]. In the following, TATL-A refers to the April version of the TATL prover, available at at `https://www.ibisc.univ-evry.fr/~adavid/bin/tatl.tar.gz`; and TATL-N refers to the November version, available at `http://atila.ibisc.univ-evry.fr/tableau_ATL/bin/tatl.tar.gz`.

A benchmark, consisting of six sets of randomly generated CL formulae, was designed to compare the performance of both provers. The formulae in the benchmark are characterised by five parameters: (1) the number of propositional symbols N; (2) the number of agents A; (3) the number of conjuncts L; (4) the modal degree D; and (5) the probability P. Based on a given choice of parameters random formulae in conjunctive normal form (CNF) are defined inductively as follows. A *random (coalition) atom* of degree 0 is a propositional variable randomly chosen from the set of N propositional symbols. A *random coalition atom* of degree D, $D > 0$, is with probability P: (a) an expression of the form $[\mathscr{A}]\varphi$,

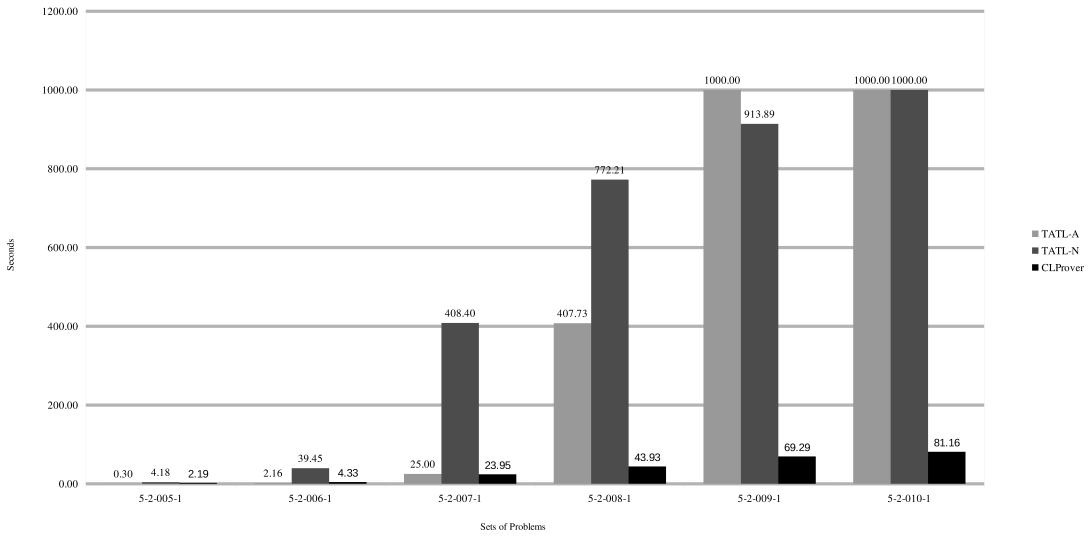


Figure 1: Performance Comparison between **CLProver**, TATL-A, and TATL-N.

where $[\mathscr{A}]$ is a coalition modality with a set of agents $\mathscr{A}$ randomly chosen from $\mathscr{P}^{\{1,\ldots,A\}}$ and φ is a random coalition CNF clause of modal degree $D-1$ (defined below), or (b) a random atom of degree 0, otherwise. A *random coalition literal* (of degree D) is with probability 0.5 a random coalition atom (of degree D) or its negation, otherwise. A *random coalition CNF clause* (of degree D) is a disjunction of three random coalition literals (of degree D). A *random coalition CNF formula* (of degree D) is a conjunction of L random coalition CNF clauses (of degree D).

The six sets of problems used to compare **CLProver** and TATL were generated using $N = 5$, $A = 2$, $L \in \{5,\ldots,10\}$, $D = 1$, and $P = 1$. The experiment was run on an i7-3537U CPU at 3.00GHz, 8GB RAM, under Linux kernel 3.11.10-100. The average run-time for each set of problems is shown in Figure 1. The provers were given a timeout of 1000 seconds. **CLProver** has solved all problems in all sets within the given time. TATL-A has failed to solve any problems in the sets 5-2-009-1 and 5-2-010-1. TATL-N has solved all problems in the sets 5-2-005-1 and 5-2-006-1; nine problems in 5-2-007-1; three in 5-2-008-1; four in 5-2-009-1; and none in 5-2-010-1. For the calculation of the average times, whenever the prover has timed out, we have set the corresponding time to 1000 seconds.

5 Conclusion

The resolution-based calculus for the Coalition Logic CL is applied to a coalition problem in $\mathsf{DSNF}_{\mathsf{CL}}$, which separates the dimensions to which the resolution rules are applied. The transformation into the normal form is satisfiability preserving and polynomially bounded by the size of the original formula. Soundness of the method follows from the axiomatisation of CL. Completeness is proved with respect to the tableau procedure given in [6]: if a tableau for a coalition problem is closed, there is a refutation based on the calculus given here. Termination is ensured by the fact that number of propositional symbols and agents is finite, so there are only a finite number of clauses that can be generated.

The decision procedure based on $\mathsf{RES}_{\mathsf{CL}}$ is in EXPTIME, as shown in [12]. This is optimal, as the satisfiability problem for coalition problems in $\mathsf{DSNF}_{\mathsf{CL}}$ is EXPTIME-hard, thus more expressive than

the language of CL. This result follows from [16, Lemma 4.10, page 785] and the fact that an extension of CL with positive occurrences of ATL's $\langle\langle\emptyset\rangle\rangle\,\Box$ operator can be translated into $\mathsf{DSNF}_{\mathsf{CL}}$. It also follows that $\mathsf{DSNF}_{\mathsf{CL}}$ is more expressive than CL.

CLProver is the first (prototype) implementation of $\mathsf{RES}_{\mathsf{CL}}$. The experiments we have performed suggest that it is a viable tool for reasoning about Coalition Logic. Future work includes further improvements to **CLProver**. We also intend to extend our calculus to the full language of ATL.

References

[1] R. Alur, T. A. Henziger & O. Kupferman (1997): *Alternating-Time Temporal Logic*. In: *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pp. 100–109, doi:10.1109/SFCS.1997.646098.

[2] R. Alur, T. A. Henziger & O. Kupferman (2002): *Alternating-Time Temporal Logic*. *Journal of the ACM* 49(5), pp. 672–713, doi:10.1145/585265.585270.

[3] A. David (2013): *TATL: Implementation of ATL Tableau-Based Decision Procedure*. *LNCS* 8123, pp. 97–103, doi:10.1007/978-3-642-40537-2_10.

[4] A. Degtyarev, M. Fisher & B. Konev (2006): *Monodic temporal resolution*. *ACM Trans. Comput. Log* 7(1), pp. 108–150, doi:10.1145/1119439.1119443.

[5] V. Goranko (2001): *Coalition games and alternating temporal logics*. In: *TARK '01*, Morgan Kaufmann, San Francisco, CA, USA, pp. 259–272. Available at `http://dl.acm.org/citation.cfm?id=1028128.1028157`.

[6] V. Goranko & D. Shkatov (2009): *Tableau-Based Decision Procedures for Logics of Strategic Ability in Multiagent Systems*. *ACM Transactions on Computational Logic* 11(1), pp. 3:1–3:51, doi:10.1145/1614431.1614434.

[7] R. Goré, J. Thomson & F. Widmann (2011): *An Experimental Comparison of Theorem Provers for CTL*. In C. Combi, M. Leucker & F.. Wolter, editors: *TIME 2011,Germany, September 12-14*, IEEE, pp. 49–56, doi:10.1109/TIME.2011.16.

[8] H. Hansen (2004): *Tableau Games for Coalition Logic and Alternating-Time Temporal Logic – theory and implementation*. Master's thesis, University of Amsterdam.

[9] A. Herzig (2007): *Logics for Agency and Multi-Agent Systems*. Slides. ESSLLI. Available at `http://www.staff.science.uu.nl/~broer110/ESSLLI07/`.

[10] U. Hustadt & R. A. Schmidt (2002): *Scientific Benchmarking with Temporal Logic Decision Procedures*. In D. Fensel, F. Giunchiglia, D. McGuinness & M-A. Williams, editors: *KR'2002*, M. Kaufmann, pp. 533–544.

[11] C. Nalon, L. Zhang, C. Dixon & U. Hustadt (2013): *A resolution-based calculus for Coalition Logic (Extended Version)*. Technical Report ULCS-13-004, University of Liverpool, Liverpool, UK. Available at `http://intranet.csc.liv.ac.uk/research/techreports/?id=ULCS-13-004`.

[12] C. Nalon, L. Zhang, C. Dixon & U. Hustadt (2014): *A resolution-based calculus for Coalition Logic*. *Journal of Logic and Computation*, doi:10.1093/logcom/ext074. To appear.

[13] M. Pauly (2001): *Logic for Social Software*. Ph.D. thesis, University of Amsterdam. Dissertation Series 2001-10.

[14] M. Pauly (2002): *A Modal Logic for Coalitional Power in Games*. *Journal of Logic and Computation* 12(1), pp. 149–166, doi:10.1093/logcom/12.1.149.

[15] J.A. Robinson (1965): *A Machine–Oriented Logic Based on the Resolution Principle*. *Journal of the ACM* 12(1), pp. 23–41, doi:10.1145/321250.321253.

[16] D. Walther, C. Lutz, F. Wolter & M. Wooldridge (2006): *ATL Satisfiability is Indeed ExpTime-complete*. *Journal of Logic and Computation* 16(6), pp. 765–787, doi:10.1093/logcom/exl009.

Efficient Decomposition of Bimatrix Games (Extended Abstract)*

Xiang Jiang & Arno Pauly
Computer Laboratory
University of Cambridge, United Kingdom
Arno.Pauly@cl.cam.ac.uk

Exploiting the algebraic structure of the set of bimatrix games, a divide-and-conquer algorithm for finding Nash equilibria is proposed. The algorithm is fixed-parameter tractable with the size of the largest irreducible component of a game as parameter. An implementation of the algorithm is shown to yield a significant performance increase on inputs with small parameters.

1 Introduction

A bimatrix game is given by two matrices (A,B) of identical dimensions. The first player picks a row i, the second player independently picks a column j. As a consequence, the first player receives the payoff A_{ij}, the second player B_{ij}. Both player are allowed to randomize over their choices, and will strive to maximize their expected payoff. A Nash equilibrium is a pair of strategies, such that no player can improve her expected payoff by deviating unilaterally.

If the payoff matrices are given by natural numbers, then there always is a Nash equilibrium using only rational probabilities. The computational task to find a Nash equilibrium of a bimatrix game is complete for the complexity class PPAD [21, 7, 6]. PPAD is contained in FNP, and commonly believed to exceed FP. In particular, it is deemed unlikely that a polynomial-time algorithm for finding Nash equilibria exists.

The next-best algorithmic result to hope for could be a fixed-parameter tractable (fpt) algorithm [8], that is an algorithm running in time $f(k)p(n)$ where n is the size of the game, p a polynomial and k a parameter. For such an algorithm to be useful, the assumption the parameter were usually small needs to be sustainable. The existence of fpt algorithms for finding Nash equilibria with various choices of parameters has been studied in [9, 13, 10].

In the present paper we demonstrate how *products* and *sums* of games – and their inverse operations – can be used to obtain a divide-and-conquer algorithm to find Nash equilibria. This algorithm is fpt, if the size of the largest component not further dividable is chosen as a parameter. *Products* of games were introduced in [23] as a means to classify the Weihrauch-degree [4, 3, 5, 14] of finding Nash equilibria for real-valued payoff matrices. *Sums* appear originally in the PhD thesis [24] of the second author; the algorithm we discuss was implemented in the Bachelor's thesis [15] of the first author.

2 Products and Sums of Games

Both products and sums admit an intuitive explanation: The product of two games corresponds to playing both games at the same time, while the sum involves playing *matching pennies* to determine which game to play, with one player being rewarded and the other one punished in the case of a failure to agree.

*A full version including proofs omitted here is available as [16].

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 75–81, doi:10.4204/EPTCS.146.10

2.1 Products

In our definition of products, we let $[\ ,\ ]:\{1,\ldots,n\}\times\{1,\ldots,m\}\to\{1,\ldots,nm\}$ denote the usual bijection $[i,j]=(i-1)n+j$. The relevant values of n,m will be clear from the context. We point out that $[\ ,\ ]$ is polynomial-time computable and polynomial-time invertible.

Definition 1. Given an $n_1\times m_1$ bimatrix game (A^1,B^1) and an $n_2\times m_2$ bimatrix game (A^2,B^2), we define the $(n_1n_2)\times(m_1m_2)$ product game $(A^1,B^1)\times(A^2,B^2)$ as (A,B) with $A_{[i_1,i_2][j_1,j_2]}=A^1_{i_1j_1}+A^2_{i_2j_2}$ and $B_{[i_1,i_2][j_1,j_2]}=B^1_{i_1j_1}+B^2_{i_2j_2}$.

Theorem 2. If (x^k,y^k) is a Nash equilibrium of (A^k,B^k) for both $k\in\{1,2\}$, then (x,y) is a Nash equilibrium of (A,B), where $x_{[i_1i_2]}=x^1_{i_1}x^2_{i_2}$ and $y_{[m_1m_2]}=y^1_{m_1}y^2_{m_2}$.

Theorem 3. If (x,y) is a Nash equilibrium of (A,B), then (x^1,y^1) given by $x^1_i=\sum_{l=1}^{n_2}x_{[i,l]}$ and $y^1_j=\sum_{l=1}^{m_2}y_{[j,l]}$ is a Nash equilibrium of (A^1,B^1).

2.2 Sums

The sum of games involves another parameter besides the two component games, which just is a number exceeding the absolute value of all the payoffs.

Definition 4. Given an $n_1\times m_1$ bimatrix game (A^1,B^1) and an $n_2\times m_2$ bimatrix game (A^2,B^2), we define the $(n_1+n_2)\times(m_1+m_2)$ sum game $(A^1,B^1)+(A^2,B^2)$ via the constant $K>\max_{i,j}\{|A_{i,j}|,B_{i,j}|\}$ as (A,B) with:

$$A_{i,j}=\begin{cases}A^1_{ij} & \text{if } i\le n_1, j\le m_1\\ A^2_{(i-n_1),(j-m_1)} & \text{if } i>n_1, j>m_1\\ K & \text{otherwise}\end{cases}$$

$$B_{i,j}=\begin{cases}B^1_{ij} & \text{if } i\le n_1, j\le m_1\\ B^2_{(i-n_1),(j-m_1)} & \text{if } i>n_1, j>m_1\\ -K & \text{otherwise}\end{cases}$$

Lemma 5. Let (x,y) be a Nash equilibrium of $(A^1,B^1)+(A^2,B^2)$. Then $0<\left(\sum_{i=1}^{n_1}x_i\right)<1$ and $0<\left(\sum_{j=1}^{m_1}y_j\right)<1$.

Theorem 6. If (x,y) is a Nash equilibrium of $(A^1,B^1)+(A^2,B^2)$, then a Nash equilibrium (x^1,y^1) of (A^1,B^1) can be obtained as $x^1_i=\frac{x_i}{\sum_{l=1}^{n_1}x_l}$ and $y^1_j=\frac{y_i}{\sum_{l=1}^{m_1}y_l}$.

Theorem 7. Let (x^k,y^k) be a Nash equilibrium of (A^k,B^k) resulting in payoffs (P^k,Q^k) for both $k\in\{1,2\}$. Then (x,y) is a Nash equilibrium of $(A^1,B^1)+(A^2,B^2)$, where $x_i=x^1_i\frac{K-Q^2}{2K-Q^1-Q^2}$ for $i\le n_1$, $x_i=x^2_{i-n_1}\frac{K-Q^1}{2K-Q^1-Q^2}$ for $i>n_1$, $y_j=y^1_j\frac{K-P^2}{2K-P^1-P^2}$ for $j\le m_1$, $y_j=y^2_{j-m_1}\frac{K-P^1}{2K-P^1-P^2}$ for $j>m_1$.

If a game is iteratively decomposed into sums, the resulting structure corresponds to a Blackwell game [1, 19] of finite length. The reasoning underlying the theorems above then provides a means of backwards-induction to show that such games always admit Nash equilibria without a direct appeal to their normal form version. The latter observation is the foundation for [18, Corollary 8].

3 Examples

In order to illuminate both how the operations work, and how the component games can be recovered from the compound game, we shall briefly consider some examples. As all relevant features already appear for zero-sum games, we shall restrict the examples to this case, and suppress explicit reference to the second player's payoffs.

$$A := \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 1 \\ 2 & 2 & 2 & 2 \\ 4 & 1 & 2 & 3 \end{pmatrix} \quad B := \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 2 & 3 \end{pmatrix}$$

$$A \times B = \begin{pmatrix} 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 & 1 & 2 & 3 & 4 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 3 & 4 & 5 & 6 & 4 & 5 & 6 & 7 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 2 & 1 & 2 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 2 & 2 & 3 & 2 & 3 & 3 & 4 & 3 & 4 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 5 & 5 & 5 & 5 \\ 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 \\ 5 & 2 & 3 & 4 & 4 & 1 & 2 & 3 & 5 & 2 & 3 & 4 \\ 5 & 2 & 3 & 4 & 6 & 3 & 4 & 5 & 7 & 4 & 5 & 6 \end{pmatrix} \qquad \begin{aligned} A_{ij} + B_{11} &= (A \times B)_{3i-2,j} \\ B_{ij} + A_{21} &= (A \times B)_{3+i,4j-3} \end{aligned}$$

As demonstrated by the colour-markings in the product game, if a game is indeed a product game, then the payoffs of the components can be read off from the payoff matrix of the composed game (plus some constant). Indeed, there are many different positions where the component games are found. Verifying that a game indeed is a product requires to ensure the consistency of these answers.

$$A + B = \begin{pmatrix} 1 & 2 & 3 & 4 & K & K & K \\ 0 & 1 & 0 & 1 & K & K & K \\ 2 & 2 & 2 & 2 & K & K & K \\ 4 & 1 & 2 & 3 & K & K & K \\ K & K & K & K & 0 & 0 & 0 \\ K & K & K & K & 1 & 0 & 1 \\ K & K & K & K & 1 & 2 & 3 \end{pmatrix}$$

Recovering the component games from a sum is even simpler: The payoff matrices are found in the left-upper and right-lower corner, while the remaining two cornes are covered by a suitable constant K. The latter allows us to determine the precise size of the corners.

4 The algorithm

Our basic algorithm proceeds as follows: To solve a game (A,B)

1. test whether (A,B) is the sum of (A^1,B^1) and (A^2,B^2) via some constant K. If yes, solve (A^1,B^1) and (A^2,B^2) and combine their Nash equilibria to an equilibrium of (A,B) via Theorem 7. If no,

2. test whether (A,B) is the product of (A^1,B^1) and (A^2,B^2). If yes, solve (A^1,B^1) and (A^2,B^2) and combine their Nash equilibria to an equilibrium of (A,B) via Theorem 2. If no,

3. find a Nash equilibrium of (A,B) by some other means.

For some $n \times m$ game (A,B) let let $S(A,B)$ denote its size, i.e. $S(A,B) = nm$, and let $\lambda(A,B)$ be the size of the largest game for which 3. in our algorithm is called. Let $f(k)$ be the time needed for the external algorithm called in 3. on a game of size k. Then the runtime of our algorithm is bounded by $O(S^2 f(\lambda))$, in particular, it is an fpt-algorithm:

Testing whether a game is a sum, and computing the components, if applicable, can be done in linear time: As the value K has to appear in the corner, one look up whether two suitable rectangular regions have payoffs K and $-K$. If this is the case, the remaining two rectangular regions contain the two subgames, provided they do not contain payoffs p with $|p| \geq K$. The sum of the sizes of the components is less than the size of the original game. Finally, combining Nash equilibria can be done in linear time, too. Only this case for yield quadratic runtime.

Whether a game is a product of factors of a fixed size can also be tested in linear time. Essentially, the payoffs of the putative component games are found as differences between corresponding payoffs in the original game. Then the product of the two component games can be computed, and finally compared to the original game to verify the decomposition. Testing the different possible factors adds an additional factor $\sqrt{S}$ for this part. The product of the sizes of the factors is equal to size of the original game. Again, combining the Nash equilibria takes linear time. The underlying recurrence relation on its own would yield a time bound of $O(S^{1.5} \log S)$, hence the quadratic runtime bound from the first relation is dominating.

Note that a game cannot simultaneously be the result of a sum and a product of smaller games. Thus, a full decomposition (and the size of the largest component) of a game is independent of the order in which decomposability is tested.

As a slight modification of our algorithm, one can eliminate (iteratively) strictly dominated strategies at each stage of the algorithm. We recall that a strategy i of some player is called strictly dominated by some other strategy j, if against any strategy chosen by the opponent, i provides its player with a strictly better payoff than j. A strictly dominated strategy can never be used in a Nash equilibrium. It is easy to verify that a game decomposable as a sum never has any strictly dominated strategies, but may occur as the result of the elimination of such strategies. Hence, including an elimination step for each stage increases the potential for decomposability.

Proposition 8. Elimination of strictly dominated strategies commutes with decomposition of products, i.e. the reduced form of the product is the product of the reduced forms of the factors.

The algorithm remains fpt if such a step is included, using e.g. the algorithm presented in [17]. The exponent would presumably increase to $O(S^4 f(\lambda))$ though. Discussion of complexity issues regarding removal of dominated strategies can be found in e.g. [2, 22].

5 Empirical evaluation

Only a small fraction of the bimatrix games of a given size and bounded integer payoffs will be decomposable by our techniques, this limiting the applicability of the algorithm in Section 4. In particular, if sample games were drawn from a uniform distribution, one should not expect any speedup using decomposability-tests. However, to some extent we can expect patterns in the definitions of real-world

game situation to increase the decomposability of the derived bimatrix games. For example, the structure of Poker-style games implies decomposability, as can be concluded from the considerations[1] in [11]. Note that an explicit understanding of such patterns is not required to benefit from our algorithm – a reasonable expectation that suitable patterns could occur is enough to justify the use of our algorithm, which then identifies the actual patterns.

To obtain a first impression whether using the decomposition algorithm is indeed beneficial for computing Nash equilibria, a collection of 100 random decomposable games was created. Each game has 95-105 strategies per player, and payoff values range from 0 to 50. The decomposability was ensure by creating a random tree representing the relevant decomposition structure first, using probabilities of 0.4 each for sum and product decomposition, and of 0.2 for an elimination of strictly dominated strategies step. The height of the trees was limited to 80, additionally vertices corresponding to games of size up to 6 were turned into leaves. At the leaves, the payoffs were chosen uniformly subject to the constraints derived from the structure and the overall constraint of payoff values being between 0 and 50. Finally, the corresponding bimatrix games were computed. As the payoffs for both players were chosen independently, the expected fraction of zero-sum games in the sample set is negligible.

Both as a benchmark, and in order to compute Nash equilibria of the irreducible component games, the tool GAMBIT [20] was used. GAMBIT offers a variety of algorithm for computing Nash equilibria of bimatrix games, we used:

1. gambit-enummixed: using extreme point enumeration
2. gambit-gnm: using a global Newton method approach
3. gambit-lcp: using linear complementarity
4. gambit-simpdiv: using simplicial subdivision

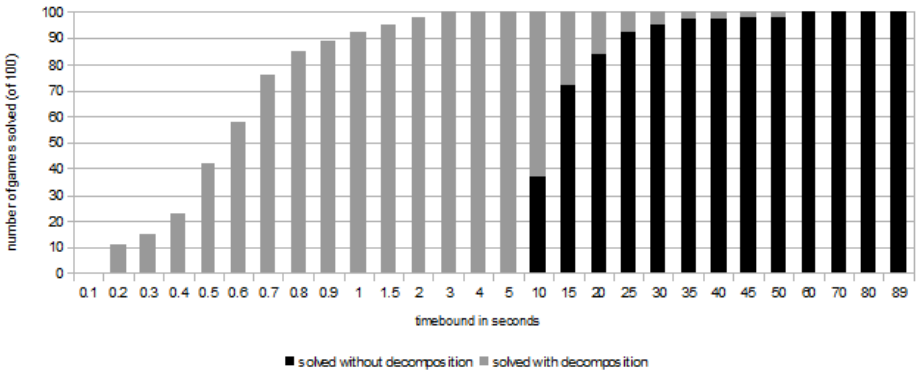


Figure 1: gambit-gnm

Figures 1. & 2. show for gambit-gmn and gambit-lcp how many of our decomposable example games could be solved in some given time bound (per game, not total) using only the GAMBIT algorithm directly, or exploiting decomposition implemented in C++ first. Despite the fact that our decomposition algorithm was not optimized, it turned out that using decomposition almost all games could be solved

[1] Making decisions on whether to *fold*, *call* or *raise* corresponds to choosing the type of the remaining game, i.e. a sum decomposition. Similarly, the cards chosen by chance induce a product decomposition of the expected values of the game.

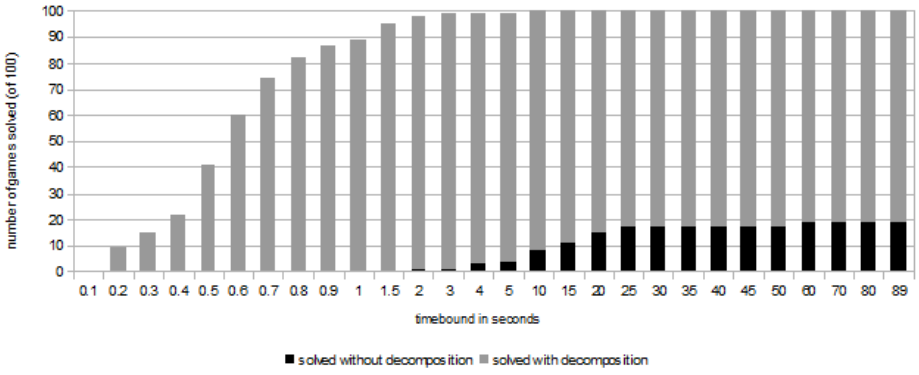


Figure 2: gambit-lcp

in under 3 seconds, whereas even gambit-gnm[2] as the fastest GAMBIT algorithm on the sample took 30 seconds for a similar feat. Neither gambit-enummixed nor gambit-subdiv were able to solve any of the example games in less than 90 seconds without decomposition. Note that the GAMBIT algorithms generally include elimination of strictly dominated strategies as well, so this alone cannot account for the differences. Thus, there is clear indication that on suitable data, exploiting the algebraic structure underlying the decomposition algorithm yields a significant increase in performance.

References

[1] D. Blackwell (1969): *Infinite G_δ games with imperfect information*. Zastowania Matematyki Applicationes Mathematicae.

[2] Felix Brandt, Markus Brill, Felix Fischer & Paul Harrenstein (2011): *On the Complexity of Iterated Weak Dominance in Constant-sum Games*. Theory of Computing Systems 49(1), pp. 162–181. Available at `http://dx.doi.org/10.1007/s00224-010-9282-7`.

[3] Vasco Brattka, Matthew de Brecht & Arno Pauly (2012): *Closed Choice and a Uniform Low Basis Theorem*. Annals of Pure and Applied Logic 163(8), pp. 968–1008, doi:10.1016/j.apal.2011.12.020.

[4] Vasco Brattka & Guido Gherardi (2011): *Effective Choice and Boundedness Principles in Computable Analysis*. Bulletin of Symbolic Logic 1, pp. 73 – 117, doi:10.2178/bsl/1294186663. ArXiv:0905.4685.

[5] Vasco Brattka, Stéphane Le Roux & Arno Pauly (2012): *On the Computational Content of the Brouwer Fixed Point Theorem*. In S.Barry Cooper, Anuj Dawar & Benedikt Löwe, editors: How the World Computes, Lecture Notes in Computer Science 7318, Springer Berlin Heidelberg, pp. 56–67, doi:10.1007/978-3-642-30870-3_7.

[6] Xi Chen, Xiaotie Deng & Shang-Hua Teng (2009): *Settling the Complexity of Computing Two-player Nash Equilibria*. J. ACM 56(3), pp. 14:1–14:57, doi:10.1145/1516512.1516516.

[7] Constantinos Daskalakis, Paul Goldberg & Christos Papadimitriou (2006): *The Complexity of Computing a Nash Equilibrium*. In: 38th ACM Symposium on Theory of Computing, pp. 71–78, doi:10.1145/1132516.1132527.

[8] Rod Downey & Michael Fellows (1999): *Parameterized Complexity*. Springer, doi:10.1007/978-1-4612-0515-9.

[2] Which is based on [12].

[9] Vladimir Estivill-Castro & Mahdi Parsa (2009): *Computing Nash equilibria Gets Harder – New Results Show Hardness Even for Parameterized Complexity*. In Rod Downey & Prabhu Manyem, editors: *CATS 2009*, *CRPIT* 94.

[10] Vladimir Estivill-Castro & Mahdi Parsa (2011): *Single Parameter FPT-Algorithms for Non-trivial Games*. In Costas Iliopoulos & William Smyth, editors: *Combinatorial Algorithms*, *Lecture Notes in Computer Science* 6460, Springer, pp. 121–124. Available at `http://dx.doi.org/10.1007/978-3-642-19222-7_13`.

[11] Andrew Gilpin, Javier Pena, Samid Hoda & Tuomas Sandholm (2007): *Gradient-based algorithms for finding Nash equilibria in extensive form games*. In: *Proceedings of the 18th Int Conf on Game Theory*, doi:10.1007/978-3-540-77105-0_9.

[12] Srihari Govindan & Robert Wilson (2003): *A Global Newton Method to Compute Nash Equilibria*. *Journal of Economic Theory* 110(1), pp. 65–86, doi:10.1016/S0022-0531(03)00005-X.

[13] Danny Hermelin, Chien-Chung Huang, Stefan Kratsch & Magnus Wahlström (2010): *Parameterized Two-Player Nash Equilibrium*. *CoRR* abs/1006.2063. Available at `http://arxiv.org/abs/1006.2063`.

[14] Kojiro Higuchi & Arno Pauly (2013): *The degree-structure of Weihrauch-reducibility*. *Logical Methods in Computer Science* 9(2), doi:10.2168/LMCS-9(2:2)2013.

[15] Xiang Jiang (2011): *Efficient Decomposition of Games*. Bachelor's thesis, University of Cambridge.

[16] Xiang Jiang & Arno Pauly (2012): *Efficient Decomposition of Bimatrix Games*. http://arxiv.org/abs/1212.6355.

[17] Donald Knuth, Christos Papadimitriou & John Tsitsiklis (1988): *A note on strategy elimination in bimatrix games*. *Operations Research Letters* 7(3), pp. 103–107, doi:10.1016/0167-6377(88)90075-2.

[18] Stéphane Le Roux & Arno Pauly (2014): *Infinite sequential games with real-valued payoffs*. arXiv:1401.3325.

[19] Donald A. Martin (1998): *The Determinacy of Blackwell Games*. *Journal of Symbolic Logic* 63(4), pp. 1565–1581, doi:10.2307/2586667.

[20] Richard McKelvey, Andrew McLennan & Theodore Turocy (2010): *Gambit: Software Tools for Game Theory*. http://www.gambit-project.org. Version 0.2010.09.01.

[21] Christos H. Papadimitriou (1994): *On the complexity of the parity argument and other inefficient proofs of existence*. *Journal of Computer and Systems Science* 48(3), pp. 498–532, doi:10.1016/S0022-0000(05)80063-7.

[22] Arno Pauly (2009): *The Complexity of Iterated Strategy Elimination*. arXiv:0910.5107.

[23] Arno Pauly (2010): *How Incomputable is Finding Nash Equilibria?* *Journal of Universal Computer Science* 16(18), pp. 2686–2710, doi:10.3217/jucs-016-18-2686.

[24] Arno Pauly (2012): *Computable Metamathematics and its Application to Game Theory*. Ph.D. thesis, University of Cambridge.

Acknowledgements

We are grateful for various helpful comments from referees. One comment in particular was instrumental in improving the runtime analysis of our algorithm to a quadratic exponent.

First Cycle Games*

Benjamin Aminof
IST Austria
Vienna, Austria
benj@ist.ac.at

Sasha Rubin
IST Austria and TU Wien
Vienna, Austria
srubin@ist.ac.at

First cycle games (FCG) are played on a finite graph by two players who push a token along the edges until a vertex is repeated, and a simple cycle is formed. The winner is determined by some fixed property Y of the sequence of labels of the edges (or nodes) forming this cycle. These games are traditionally of interest because of their connection with infinite-duration games such as parity and mean-payoff games.

We study the memory requirements for winning strategies of FCGs and certain associated infinite duration games. We exhibit a simple FCG that is not memoryless determined (this corrects a mistake in *Memoryless determinacy of parity and mean payoff games: a simple proof* by Björklund, Sandberg, Vorobyov (2004) that claims that FCGs for which Y is closed under cyclic permutations are memoryless determined). We show that $\Theta(n)!$ memory (where n is the number of nodes in the graph), which is always sufficient, may be necessary to win some FCGs. On the other hand, we identify easy to check conditions on Y (i.e., Y is closed under cyclic permutations, and both Y and its complement are closed under concatenation) that are sufficient to ensure that the corresponding FCGs and their associated infinite duration games are memoryless determined. We demonstrate that many games considered in the literature, such as mean-payoff, parity, energy, etc., satisfy these conditions. On the complexity side, we show (for efficiently computable Y) that while solving FCGs is in PSPACE, solving some families of FCGs is PSPACE-hard.

1 Introduction

First cycle games (FCGs) are played on a finite graph by two players who push a token along the edges of the graph until a simple cycle is formed. Player 0 wins the play if the sequence of labels of the edges (or nodes) of the cycle satisfies some fixed cycle property Y, and otherwise Player 1 wins. For instance, if every vertex has an integer priority, the cycle property $Y = \mathsf{cyc\text{-}Parity}$ states that the largest priority occurring on the cycle should be even. For a fixed cycle property Y, we write $\mathrm{FCG}(Y)$ for the family of games over all possible arenas with this winning condition. We are motivated by two questions: Under what conditions on Y is every game in $\mathrm{FCG}(Y)$ memoryless determined? What is the connection between FCGs and infinite-duration games?

First cycle games. First, we give a simple example showing that first cycle games (FCGs) are not necessarily memoryless determined, even if Y is closed under cyclic permutations (i.e., even if winning depends on the cycle but not on how it was traversed), contrary to the claim in [2][Page 370]. We then show that, for a graph with n nodes, whereas no winning strategy needs more than $(n-1)!$ memory (since this is enough to remember the whole history of the game), some FCGs require at least $\Omega(n!)$ memory. To complete the picture, we analyse the complexity of solving FCGs and show that it is PSPACE-complete. More specifically, we show that if one can decide in PSPACE whether a given cycle satisfies the property

*This work is supported by the Austrian Science Fund through grant P23499-N23, and through the RiSE network (S11403-N23, S11407-N23); ERC Start grant (279307: Graph Games); and Vienna Science and Technology Fund (WWTF) grant PROSEED Nr. ICT 10-050.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 83–90, doi:10.4204/EPTCS.146.11

Y, then solving the games in $\mathrm{FCG}(Y)$ is in PSPACE; and that even for a trivially computable cycle property Y (namely, that the cycle ends with the label 0), solving the games in $\mathrm{FCG}(Y)$ is PSPACE-hard.

First Cycle Games and Infinite-Duration Games. The main object used to connect FCGs and infinite-duration games (such as parity games) is the *cycles-decomposition* of a path. Informally, a path is decomposed by pushing the edges of the path onto a stack; as soon as a cycle is detected in the stack it is popped and output, and the algorithm continues. We then say that a winning condition W (such as the parity or energy winning condition) is *Y-greedy on $\mathscr{A}$* if in the game on arena $\mathscr{A}$ and winning condition W, Player 0 is guaranteed to win if he ensures that every cycle in the cycles-decomposition of the play satisfies Y, and Player 1 is guaranteed to win if she ensures that every cycle in the cycles-decomposition does not satisfy Y. We prove a *Transfer Theorem:* if W is Y-greedy on $\mathscr{A}$, then the winning regions in the following two games on arena $\mathscr{A}$ coincide, and memoryless winning strategies transfer between them: the infinite duration game with winning condition W, and the FCG with winning condition Y.

To illustrate the usefulness of the concept of being Y-greedy, we instantiate the definition to well-studied infinite-duration games: i) the parity winning condition (the largest priority occurring infinitely often is even) is Y-greedy on every arena $\mathscr{A}$ where $Y = \mathsf{cyc\text{-}Parity}$, ii) the mean-payoff condition (the mean payoff is at least v) is $\mathsf{cyc\text{-}MeanPayoff}_v$-greedy on every arena $\mathscr{A}$ (where $\mathsf{cyc\text{-}MeanPayoff}_v =$ average payoff is at least v), and iii) for every arena $\mathscr{A}$ with vertex set V, and largest weight W, the energy condition stating that the energy level is always non-negative starting with initial credit $W(|V|-1)$ is $\mathsf{cyc\text{-}Energy}$-greedy on $\mathscr{A}$ (where $\mathsf{cyc\text{-}Energy}$ = the energy level is non-negative).

In order to prove memoryless determinacy of certain FCGs (and related infinite-duration games) we generalise techniques used to prove that mean-payoff games are memoryless determined (Ehrenfeucht and Mycielski [4]). Given a cycle property Y, we first consider the infinite duration games $\mathrm{ACG}(Y)$ (all cycles), and $\mathrm{SCG}(Y)$ (suffix all-cycles). A game in the family $\mathrm{ACG}(Y)$ requires Player 0 to ensure that every cycle in the cycles-decomposition of the play (starting from the beginning) satisfies Y. A game in the family $\mathrm{SCG}(Y)$ requires Player 0 to ensure that every cycle in the cycles-decomposition of *some suffix* of the play satisfies Y. As was done in [4], reasoning about infinite and finite duration games is intertwined – in our case, we simultaneously reason about games in $\mathrm{FCG}(Y)$ and $\mathrm{SCG}(Y)$. We define a property of arenas, which we call *Y-unambiguous*, and prove a *Memoryless Determinacy Theorem*: a game from $\mathrm{FCG}(Y)$ whose arena $\mathscr{A}$ is Y-unambiguous is memoryless determined. Combining this with the Transfer Theorem above, we also get that if $\mathscr{A}$ is Y-unambiguous, then any game with a winning condition W that is Y-greedy on $\mathscr{A}$, is memoryless determined[1].

Although checking if an arena is Y-unambiguous may not be hard, it has two disadvantages: it involves reasoning about infinite paths and it involves reasoning about the arena whereas, in many cases, memoryless determinacy is guaranteed by the cycle property Y regardless of the arena (this is the case for example with $Y = \mathsf{cyc\text{-}Parity}$). Therefore, we also provide easy to check 'finitary' sufficient conditions on Y (namely that Y is closed under cyclic permutations, and both Y and its complement are closed under concatenation) that ensure Y-unambiguity of every arena, and thus memoryless determinacy for all games in $\mathrm{FCG}(Y)$. We demonstrate the usefulness of these conditions by observing that typical cycle properties are easily seen to satisfy them, e.g., $\mathsf{cyc\text{-}Parity}, \mathsf{cyc\text{-}MeanPayoff}_v, \mathsf{cyc\text{-}Energy}$.

We conclude by noting that, in particular, if Y is closed under cyclic permutations, and both Y and its complement are closed under concatenation, then games with winning condition W are memoryless determined on every arena $\mathscr{A}$ for which W is Y-greedy on $\mathscr{A}$. As noted above, for many winning

[1] Taking Y to be $\mathsf{cyc\text{-}GoodForEnergy}$ (defined to be that either the energy level is positive, or it is zero and the largest priority occurring is even) and noting that for every arena $\mathscr{A}$ we have: i) $\mathscr{A}$ is Y-unambiguous and, ii) the game in $\mathrm{ACG}(Y)$ over $\mathscr{A}$ is Y-greedy on $\mathscr{A}$; we obtain a proof of [3][Lemma 4] that no longer relies on the incorrect result from [2].

conditions W (such as mean-payoff, parity, and energy winning conditions) it is easy to find a cycle property Y satisfying the mentioned closure conditions, and for which W is Y-greedy on the arena of interest. This provides an easy way to deduce memoryless determinacy of these classic games.

Related work. As just discussed, this work extends [4], finds a counter-example to a claim in [2], and supplies a proof of a lemma in [3]. Conditions that ensure (or characterise) which games have memoryless strategies appear for example in [1, 5, 6]. However, all of these deal with infinite duration games and do not exploit the connection to finite duration games.

Due to space limitations, proofs appear in the full version of the article.

2 Definitions

In this paper all games are two-player turn-based games of perfect information played on finite graphs. The players are called Player 0 and Player 1.

Arena An *arena* is a labeled directed graph $\mathscr{A} = (V_0, V_1, E, \mathbb{U}, \lambda)$ where

1. V_0 and V_1 are disjoint sets of vertices of Player 0 and Player 1, respectively; the set of vertices of the arena $V := V_0 \cup V_1$ is non-empty.
2. $E \subseteq V \times V$ is a set of edges with no dead-ends (i.e., for every $v \in V$ there is some edge $(v,w) \in E$);
3. $\mathbb{U}$ is a set of possible labels.
4. $\lambda : E \to \mathbb{U}$ is a *labeling* function, used by the winning condition.

Typical choices for $\mathbb{U}$ are $\mathbb{R}$ and $\mathbb{N}$. Games in which vertices are labeled instead of edges can be modeled by ensuring $\lambda(v,w) = \lambda(v,w')$ for all $v,w,w' \in V$. Similarly, games in which vertices are labeled by elements of $\mathbb{U}'$ and edges are labeled by elements of $\mathbb{U}''$ can be modeled by labeling edges by elements of $\mathbb{U}' \times \mathbb{U}''$. As usual, if $u = e_1 e_2 \cdots$ is a (finite or infinite) sequence of edges in the arena, we write $\lambda(u)$ for the string of labels $\lambda(e_1)\lambda(e_2)\cdots$.

Plays and strategies A *play* $\pi = \pi_0, \pi_1, \ldots$ in an arena is an infinite[2] sequence over V such that $(\pi_j, \pi_{j+1}) \in E$ for all $j \in \mathbb{N}$. The node π_0 is called the *starting* node of the play. We denote the set of all plays in the arena $\mathscr{A}$ by $plays(\mathscr{A})$. A *strategy* for Player i is a function $S : V^* V_i \to V$ such that if $u \in V^*$ and $v \in V_i$ then $(v, S(uv)) \in E$. A strategy S for Player i is *memoryless* if $S(uv) = S(u'v)$ for all $u,u' \in V^*, v \in V_i$. A play π is *consistent* with S, where S is a strategy for Player i, if for every $j \in \mathbb{N}$ such that $\pi_j \in V_i$, it is the case that $\pi_{j+1} = S(\pi_0 \cdots \pi_j)$. A strategy S for Player i is *generated by a Moore machine* if there exists a finite set M of *memory states*, an *initial state* $m_I \in M$, a *memory update* function $\delta : V \times M \to M$, and a *next-move function* $\rho : V \times M \to V$ such that if $u = u_0 u_1 \cdots u_l$ is a prefix of a play with $u_l \in V_i$ then $S(u) = \rho(u_l, m_l)$ where m_l is defined inductively by $m_0 = m_I$ and $m_{i+1} = \delta(u_i, m_i)$. A strategy S is *finite-memory* if it is generated by some Moore machine. A strategy S *uses memory at most* k if it is generated by some Moore machine with $|M| \leq k$. A strategy S *uses memory at least* k if every Moore machine generating S has $|M| \geq k$.

Games, Winning Conditions, and Memoryless Determinacy A *game* is a pair $(\mathscr{A}, O)$ where $\mathscr{A} = (V_0, V_1, E, \mathbb{U}, \lambda)$ is an arena and $O \subseteq plays(\mathscr{A})$ is an *objective* (usually induced by the labeling). If either V_0 or V_1 is empty, then the game (A, O) is called a *solitaire game*. A play π in a game $(\mathscr{A}, O)$ is *won by Player* 0 if $\pi \in O$, and *won by Player* 1 otherwise. A strategy S for Player i is *winning starting from a node* $v \in V$ if every play π that starts from v and is consistent with S is won by Player i.

[2] For simplicity, we consider plays of both finite and infinite duration games to be infinite. However, in a finite duration game (and thus in any FCG) the winner is determined by a finite prefix of the play, and the moves after this prefix are immaterial.

A *winning condition* is a set $W \subseteq \mathbb{U}^\omega$. If W is a winning condition and $\mathscr{A}$ is an arena, the objective $O_W(\mathscr{A})$ induced by W is defined as follows: $O_W(\mathscr{A}) = \{v_0v_1v_2\cdots \in \mathit{plays}(\mathscr{A}) \mid \lambda(v_0,v_1)\lambda(v_1,v_2)\cdots \in W\}$. Here are some standard winning conditions:

- The *parity condition* PARITY consists of those infinite sequences $c_1c_2\cdots \in \mathbb{N}^\omega$ such that the largest label occurring infinitely often is even.
- For $\nu \in \mathbb{R}$, the *ν-mean-payoff condition* consists of those infinite sequences $c_1c_2\cdots \in \mathbb{R}$ such that $\limsup_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} c_i$ is at least ν.
- The *energy condition* for a given *initial credit* $r \in \mathbb{N}$, written $\textsc{Energy}(r)$, consists of those infinite sequences $c_1c_2\cdots \in \mathbb{Z}^\omega$ such that $r + c_1 + \cdots + c_k \geq 0$ for all $k \geq 1$.
- The *energy-parity condition* $\textsc{Energy-Parity}(r)$ is defined as consisting of $(c_1,d_1)(c_2,d_2)\cdots \in \mathbb{N}\times\mathbb{Z}$ such that $c_1c_2\cdots$ is in PARITY and $d_1d_2\cdots$ is in $\textsc{Energy}(r)$.

The *(memoryless) winning region* of Player i is the set of vertices $v \in V$ such that Player i has a (memoryless) winning strategy starting from v. A game is *pointwise memoryless for Player i* if the memoryless winning region for Player i coincides with the winning region for Player i. A game is *uniform memoryless for Player i* if there is a memoryless strategy for Player i that is winning starting from every vertex in that player's winning region.

A game is *determined* if the winning regions partition V. A game is *pointwise memoryless determined* if it is determined and it is pointwise memoryless for both players. A game is *uniform memoryless determined* if it is determined and uniform memoryless for both players.

Cycles-decomposition A *cycle* in an arena $\mathscr{A}$ is a sequence of edges $(v_1,v_2)(v_2,v_3)\cdots(v_{k-1},v_k)(v_k,v_1)$.

Define an algorithm that processes a play $\pi \in \mathit{plays}(\mathscr{A})$ and outputs a sequence of cycles: at step 0 start with empty stack; at step j push the edge (π_j,π_{j+1}), and if for some k, the top k edges on the stack form a cycle, this cycle is popped and output, and the algorithm continues to step $j+1$. The sequence of cycles output by this algorithm is called the *cycles-decomposition of π*, and is denoted by $\mathit{cycles}(\pi)$. The *first cycle of π* is the first cycle in $\mathit{cycles}(\pi)$. For example, if $\pi = vwxwvs(xyz)^\omega$, then $\mathit{cycles}(\pi) = (w,x)(x,w),(v,w)(w,v),(x,y)(y,z)(z,x),(x,y)(y,z)(z,x),\ldots$, and the first cycle of π is $(w,x)(x,w)$.Note that $\mathit{cycles}(\pi)$ is such that at most $|V|-1$ edges of π do not appear in it (i.e, they are pushed but never popped – like the edge (v,s) in the example above). As we show in the full version, this allows one to reason, for instance, about the initial credit problem for energy games (cf. [3]).

Cycle properties A *cycle property* is a set $Y \subseteq \mathbb{U}^*$, used later on to define winning conditions for games. Here are some cycle properties that we refer to in the rest of the article:

1. Let cyc-EvenLen be those sequences $c_1c_2\cdots c_k \in \mathbb{U}^*$ such that k is even.
2. Let cyc-Parity be those sequences $c_1\cdots c_k \in \mathbb{N}^*$ such that $\max_{1\leq i\leq k} c_i$ is even.
3. Let cyc-Energy be those sequences $c_1\cdots c_k \in \mathbb{Z}^*$ such that $\sum_{i=1}^{k} c_i \geq 0$.
4. Let cyc-GoodForEnergy be those sequences $(c_1,d_1)\cdots(c_k,d_k) \in (\mathbb{N}\times\mathbb{Z})^*$ such that either $\sum_{i=1}^{k} d_i > 0$, or both $\sum_{i=1}^{k} d_i = 0$ and $c_1\cdots c_k \in$ cyc-Parity.
5. Let $\text{cyc-MeanPayoff}_\nu$ be those sequences $c_1\cdots c_k \in \mathbb{R}^*$ such that $\frac{1}{k}\sum_{i=1}^{k} c_i \leq \nu$, for some $\nu \in \mathbb{R}$.
6. Let cyc-MaxFirst be those sequences $c_1\cdots c_k \in \mathbb{N}^*$ such that $c_1 \geq c_i$ for all $1 \leq i \leq k$.
7. Let cyc-EndsZero be those sequences $c_1\cdots c_k \in \mathbb{N}^*$ such that $c_k = 0$.

If $Y \subseteq \mathbb{U}^*$ is a cycle property, write $\neg Y$ for the cycle property $\mathbb{U}^* \setminus Y$. We isolate two important classes of cycle properties (the first is inspired by [2]):

1. Say that Y is *closed under cyclic permutations* if $ab \in Y$ implies $ba \in Y$, for all $a \in \mathbb{U}, b \in \mathbb{U}^*$.
2. Say that Y is *closed under concatenation* if $a \in Y$ and $b \in Y$ imply that $ab \in Y$, for all $a, b \in \mathbb{U}^*$.

Note that the cycle properties 1-5 above are closed under cyclic permutations and concatenation; and that $\neg$cyc-EvenLen is closed under cyclic permutations but not under concatenation.

First Cycle Games (FCGs) Given a cycle property $Y \subseteq \mathbb{U}^*$, and an arena $\mathscr{A} = (V_0, V_1, E, \mathbb{U}, \lambda)$, let the objective $O_{\mathrm{FCG}(Y)}(\mathscr{A}) \subseteq \mathit{plays}(\mathscr{A})$ be such that $\pi \in O_{\mathrm{FCG}(Y)}(\mathscr{A})$ iff $\lambda(u) \in Y$ where u is the *first cycle* in the cycles-decomposition of π. The family $\mathrm{FCG}(Y)$ of *first cycle games of* Y consists of all games of the form $(A, O_{\mathrm{FCG}(Y)}(\mathscr{A}))$ where $\mathscr{A}$ is an arena with labels in $\mathbb{U}$. For instance, $\mathrm{FCG}(\textsf{cyc-Parity})$ consists of those games such that Player 0 wins iff the largest label occurring on the first cycle is even.[3]

3 Finite Duration Cycle Games (on being first)

In this section we analyse the memory required for winning strategies in first cycle games, and the complexity of solving these games. We begin by correcting a mistake in [2].

Proposition 1. *There exists a cycle property Y closed under cyclic permutations and a game in $\mathrm{FCG}(Y)$ that is not pointwise memoryless determined.*

To see this, consider a game where Player 1 chooses from $\{a, b\}$ and Player 0 must match the choice. This clearly requires Player 0 to have memory. The claim follows by simply encoding this game as a FCG. For example, let the cycle-property Y be cyc-EvenLen, let the vertex set be $\{v_1, v_2, v_3, v_4\}$, let $V_0 = \{v_1\}$, and let the edges be $\{(v_1, v_2), (v_2, v_1), (v_1, v_3), (v_3, v_2), (v_2, v_4), (v_4, v_1)\}$.

We now consider the difference between pointwise and uniform memoryless determinacy of FCGs.

Theorem 1.
1. *Solitaire FCGs are pointwise memoryless determined.*
2. *There is a solitaire FCG that is not uniform memoryless determined.*
3. *If cycle property Y is closed under cyclic permutations, and a game from $\mathrm{FCG}(Y)$ is pointwise memoryless for Player i, then that game is uniform memoryless for Player i.*

Proposition 2.
1. *For a FCG on an arena with n vertices, if Player i wins from v, then every winning strategy for Player i starting from v uses memory at most $(n-1)!$.*
2. *For every n there exists a FCG on an arena with $3n+1$ vertices, and a vertex v, such that every winning strategy for Player 0 starting from v uses memory at least $n!$.*

The first item is immediate since $(n-1)!$ is enough to remember the whole history of the game up to the point a cycle is formed. The proof of the second item is by showing a game where Player 1 can "weave" any possible permutation of n nodes, whereas in order to win Player 0 must remember this permutation. The construction is in the full version of the paper.

Finally, we analyse the complexity of solving FCGs with efficiently computable cycle properties.

Theorem 2.
1. *If Y is a cycle property for which solving membership is in PSPACE, then the problem of solving games in $\mathrm{FCG}(Y)$ is in PSPACE.*
2. *The problem of solving games in $\mathrm{FCG}(\textsf{cyc-EndsZero})$ is PSPACE-complete.*

[3] Formally, then, first cycle games are of infinite duration, although the winner is determined after the first cycle appears on the play.

Sketch. For the first item, observe that solving the game amounts to evaluating the finite AND-OR tree obtained by unwinding the arena into all possible plays, up to the point on each play where a cycle is formed; nodes belonging to Player 0 are 'or' nodes, nodes belonging to Player 1 are 'and' nodes, and a leaf is marked by 'true' iff the cycle formed on the way to it is in Y. Since this tree has depth at most n (the size of the arena), and since we assumed membership in Y is in PSPACE, marking the leaves can be done in PSPACE. So evaluating the tree can be done in PSPACE.

For the second item, note that Generalised Geography can be thought of as a first cycle game in which Player i nodes are labeled by i, and $Y = \text{cyc-EndsZero}$. Note that computing Y is computationally trivial, but solving Generalised Geography is PSPACE-hard (see for instance [7][Theorem 8.11]). □

4 Infinite Duration Cycle Games

4.1 On being greedy

We start by defining two types of infinite duration games called the *All-Cycles* and the *Suffix All-Cycles* games, whose winning condition is derived from Y. Informally, All-Cycles games are games in which Player 0 wins iff all cycles in the cycles-decomposition of the play are in Y, and Suffix All-Cycles Games are games in which Player 0 wins iff all cycles in the cycles-decomposition of *some suffix* of the play are in Y. Formally, for arena $\mathscr{A} = (V_0, V_1, E, \mathbb{U}, \lambda)$ and cycle property $Y \subseteq \mathbb{U}^*$, we define two objectives $O \subseteq \mathit{plays}(\mathscr{A})$ and corresponding families of games as follows:

1. $\pi \in O_{\mathrm{ACG}(Y)}(\mathscr{A})$:if $\lambda(u) \in Y$ for *all cycles* u in $\mathit{cycles}(\pi)$.
2. $\pi \in O_{\mathrm{SCG}(Y)}(\mathscr{A})$:if some *suffix* π' of π satisfies that $\lambda(u) \in Y$ for all cycles u in $\mathit{cycles}(\pi')$. [4]

Define the corresponding families of games:

1. The family $\mathrm{ACG}(Y)$ of *all-cycles games of* Y consists of all games of the form $(\mathscr{A}, O_{\mathrm{ACG}(Y)}(\mathscr{A}))$.
2. The family $\mathrm{SCG}(Y)$ of *suffix all-cycles games of* Y consists of all games of the form $(\mathscr{A}, O_{\mathrm{SCG}(Y)}(\mathscr{A}))$.

Definition 1. *Say that a* game $(\mathscr{A}, O)$ is Y-greedy *if* $O_{\mathrm{ACG}(Y)}(\mathscr{A}) \subseteq O$ *and* $O_{\mathrm{ACG}(\neg Y)}(\mathscr{A}) \subseteq V^\omega \setminus O$. *Say that a* winning condition W is Y-greedy on arena $\mathscr{A}$ *if the game* $(\mathscr{A}, O_W)$ *is* Y*-greedy.*

Intuitively, W being Y-greedy on $\mathscr{A}$ means that Player 0 can win the game on arena $\mathscr{A}$ with winning condition W if he ensures that every cycle in the cycles-decomposition of the play is in Y, and Player 1 can win if she ensures that every cycle in the cycles-decomposition of the play is not in Y.

For instance, the winning condition PARITY (the largest priority occurring infinitely often is even) is cyc-Parity-greedy on every arena $\mathscr{A}$, the ν-mean-payoff condition (the lim sup average is at least ν) is $\text{cyc-MeanPayoff}_\nu$-greedy on every arena $\mathscr{A}$, and the energy condition (stating that the energy level is always non-negative starting with initial credit $W(|V|-1)$, where W is the largest weight and V are the vertices of the arena $\mathscr{A}$) is cyc-Energy-greedy on $\mathscr{A}$.

Theorem 3 (Transfer)**.** *Let* $(\mathscr{A}, O)$ *be a* Y*-greedy game, and let* $i \in \{0,1\}$.

1. *The winning regions for Player* i *in the games* $(\mathscr{A}, O)$ *and* $(\mathscr{A}, O_{\mathrm{FCG}(Y)}(\mathscr{A}))$ *coincide.*
2. *For every memoryless strategy* S *for Player* i *starting from* v *in arena* $\mathscr{A}$*:* S *is winning in the game* $(\mathscr{A}, O)$ *if and only if* S *is winning in the game* $(\mathscr{A}, O_{\mathrm{FCG}(Y)}(\mathscr{A}))$.

[4] Note that this is *not* the same as saying that $\lambda(u) \in Y$ for all but finitely many cycles u in $\mathit{cycles}(\pi)$. For instance, let Y be the property that the cycle has odd length, and take $\pi := (v_1 v_2 v_1 v_3 v_2 v_4)^\omega$. Note that i) decomposing the suffix π' starting with the second vertex results in all cycles having odd length, and ii) it is not the case that almost all cycles in the cycles-decomposition of π have odd length (in fact, they all have even length).

Corollary 1. *Let W be Y-greedy on arena $\mathscr{A}$. Then the game $(\mathscr{A}, O_W)$ is determined, and is pointwise (uniform) memoryless determined if and only if the game $(\mathscr{A}, O_{\mathrm{FCG}(Y)}(\mathscr{A}))$ is pointwise (uniform) memoryless determined.*

4.2 On being unambiguous

Definition 2. *An arena $\mathscr{A}$ is Y-*unambiguous *if $O_{\mathrm{SCG}(Y)}(\mathscr{A}) \cap O_{\mathrm{SCG}(\neg Y)}(\mathscr{A}) = \emptyset$.*

Lemma 1. *If $\mathscr{A}$ is Y-unambiguous then the game $(\mathscr{A}, O_{\mathrm{SCG}(Y)}(\mathscr{A}))$ is Y-greedy.*

Theorem 4 (Memoryless Determinacy)**.** *If arena $\mathscr{A}$ is Y-unambiguous, then the game $(\mathscr{A}, O_{\mathrm{FCG}(Y)}(\mathscr{A}))$ is pointwise memoryless determined. If Y is also closed under cyclic permutations, then this game is uniform memoryless determined.*

It is of interest to note that the proof of this theorem is a generalisation of the proof used in [4] for showing memoryless determinacy of mean-payoff games. As in [4], our proof reasons about infinite plays. More specifically, we obtain from Theorem 3 and Lemma 1 that the winning regions of each player in the games $(\mathscr{A}, O_{\mathrm{SCG}(Y)}(\mathscr{A}))$ and $(\mathscr{A}, O_{\mathrm{FCG}(Y)}(\mathscr{A}))$ coincide, and then go on and use this fact to derive memoryless strategies for the game $(\mathscr{A}, O_{\mathrm{FCG}(Y)}(\mathscr{A}))$.

Corollary 2. *Suppose arena $\mathscr{A}$ is Y-unambiguous.*

1. *If $(\mathscr{A}, O)$ is Y-greedy, then the game $(\mathscr{A}, O)$ is pointwise memoryless determined.*
2. *The games $(\mathscr{A}, O_{\mathrm{SCG}(Y)}(\mathscr{A}))$ and $(\mathscr{A}, O_{\mathrm{ACG}(Y)}(\mathscr{A}))$ are pointwise memoryless determined.*

If in addition Y is closed under cyclic permutations, then these game are uniform memoryless determined.

Proof. For the first item combine Theorems 3 and 4. For the second, use Lemma 1 and the fact that $(\mathscr{A}, O_{\mathrm{ACG}(Y)}(\mathscr{A}))$ is always Y-greedy. For the final statement apply Theorem 1 item 3. □

We now provide a simple sufficient condition on Y — that does not involve reasoning about cycles-decompositions of infinite paths — that ensures that every arena $\mathscr{A}$ is Y-unambiguous:

Theorem 5. *Let $Y \subseteq \mathbb{U}^*$ be a cycle property. If Y is closed under cyclic permutations*[5]*, and both Y and $\neg Y$ are closed under concatenation, then every arena $\mathscr{A}$ is Y-unambiguous.*

It is easy to check that the following cycle properties satisfy the hypothesis of Theorem 5: cyc-Parity, cyc-Energy, cyc-MeanPayoff$_v$, and cyc-GoodForEnergy. On the other hand, ¬cyc-EvenLen is not closed under concatenation, whereas cyc-MaxFirst is not closed under cyclic permutations.

We conclude with the main result of this section:

Corollary 3. *Suppose Y is closed under cyclic permutations, and both Y and its complement are closed under concatenation. Then the following games are uniform memoryless determined for every arena $\mathscr{A}$: $(\mathscr{A}, O_W)$ if W is Y-greedy on $\mathscr{A}$, $(\mathscr{A}, O_{\mathrm{SCG}(Y)}(\mathscr{A}))$, and $(\mathscr{A}, O_{\mathrm{ACG}(Y)}(\mathscr{A}))$.*

We believe that Corollary 3 provides a practical and easy way of deducing that many infinite duration games are uniform memoryless determined, as follows: exhibit a cycle property Y that is closed under cyclic permutations and both Y and $\neg Y$ are closed under concatenation, such that the winning condition W is Y-greedy on the arena A of interest. Finding such a Y is usually easy since it is simply a 'finitary' version of the winning condition W. For example, uniform memoryless determinacy of parity games, mean-payoff games, and energy-games, can easily be deduced by considering the cycle properties cyc-Parity, cyc-MeanPayoff$_v$, and cyc-Energy.

[5] It may be worth noting that Y is closed under cyclic permutations iff so is $\neg Y$.

References

[1] Alessandro Bianco, Marco Faella, Fabio Mogavero & Aniello Murano (2011): *Exploring the boundary of half-positionality*. *Ann. Math. Artif. Intell.* 62(1-2), pp. 55–77. Available at http://dx.doi.org/10.1007/s10472-011-9250-1.

[2] Henrik Björklund, Sven Sandberg & Sergei G. Vorobyov (2004): *Memoryless determinacy of parity and mean payoff games: a simple proof*. *Theor. Comput. Sci.* 310(1-3), pp. 365–378. Available at http://dx.doi.org/10.1016/S0304-3975(03)00427-4.

[3] Krishnendu Chatterjee & Laurent Doyen (2012): *Energy parity games*. *Theoretical Computer Science* 458(0), pp. 49 – 60, doi:10.1016/j.tcs.2012.07.038. Available at http://www.sciencedirect.com/science/article/pii/S0304397512007475.

[4] A. Ehrenfeucht & J. Mycielski (1979): *Positional strategies for mean payoff games*. *International Journal of Game Theory* 8(2), pp. 109–113. Available at http://dx.doi.org/10.1007/BF01768705.

[5] Hugo Gimbert & Wieslaw Zielonka (2005): *Games Where You Can Play Optimally Without Any Memory*. In: *CONCUR*, pp. 428–442. Available at http://dx.doi.org/10.1007/11539452_33.

[6] Eryk Kopczynski (2006): *Half-Positional Determinacy of Infinite Games*. In: *ICALP (2)*, pp. 336–347. Available at http://dx.doi.org/10.1007/11787006_29.

[7] Michael Sipser (1997): *Introduction to the theory of computation*. PWS Publishing Company.

Games with recurring certainty*

Dietmar Berwanger
Laboratoire Spécification et Vérification
CNRS & ENS Cachan, France
dwb@lsv.ens-cachan.fr

Anup Basil Mathew
Institute of Mathematical Sciences
Chennai, India
anupbasil@imsc.res.in

Infinite games where several players seek to coordinate under imperfect information are known to be intractable, unless the information flow is severely restricted. Examples of undecidable cases typically feature a situation where players become uncertain about the current state of the game, and this uncertainty lasts forever.

Here we consider games where the players attain certainty about the current state over and over again along any play. For finite-state games, we note that this kind of *recurring* certainty implies a stronger condition of *periodic* certainty, that is, the events of state certainty ultimately occur at uniform, regular intervals. We show that it is decidable whether a given game presents recurring certainty, and that, if so, the problem of synthesising coordination strategies under ω-regular winning conditions is solvable.

1 Introduction

Automated synthesis of systems that are correct by construction is a persistent ambition of computational engineering. One major challenge consists in controlling components that have only partial information about the global system state. Building on automata and game-theoretic foundations, significant progress has been made towards synthesising finite-state components that interact with an uncontrollable environment either individually, or in coordination with other controllable components — provided the information they have about the global system is distributed hierarchically [10, 9, 8]. Absent such restrictions, however, the problem of coordinating two or more components of a distributed system with non-terminating executions is generally undecidable [11, 2].

The distributed synthesis problem can be formulated alternatively in terms of games between n players (the components) that move along the edges of a finite graph (the state transitions of the global system) with imperfect information about the the current position and the moves of the other players. The outcome of a play is an infinite path (system execution) determined by the joint actions of the players and moves of Nature (the uncontrollable environment). The players have a common winning condition: that the play corresponds to a correct execution with respect to the system specification, no matter how Nature moves. Thus, distributed synthesis under partial information corresponds to the problem of constructing a winning profile of finite-state strategies in a coordination game with imperfect information, which was shown to be undecidable already in [12], for the basic setting of two players with a reachability condition, and in [7], for more complex winning conditions.

The cited undecidability arguments share a basic scenario: two players – he and she – become uncertain about the current state of the game, due to moves of Nature. As her (partial) knowledge of the state differs from his, and their actions need to respect the uncertainty of both, she needs to keep track not only of what she or he knows about the game state, but also, e.g., of what he knows about what she

*This work was partly supported by European project Cassting (FP7-ICT-601148).

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 91–96, doi:10.4204/EPTCS.146.12

knows that he knows, and so on. The scenario, set up so that the uncertainty never vanishes, leads to undecidability as the knowledge hierarchies grow unboundedly while the play proceeds [3].

The *information fork* criterion of [5] identifies distributed system architectures that may allow the knowledge of players to develop differently, for an unbounded number of rounds. Nevertheless, information forks may not cause undecidability in every context, for instance, if the "forked knowledge" is irrelevant for enforcing the winning condition, or if the effect of forking can be undone within a few rounds every time it occurs.

In this paper, we consider n-player games with imperfect information where the uncertainty of players about the game state cannot last forever. Our intuition of *recurring certainty* is that, whenever players are uncertain about the state of the game during a play, it takes only finitely many rounds until they can deduce the current state with certainty, and it becomes common knowledge among them. A faithful formalisation of this common knowledge property would most likely be undecidable. Thus, we resort to a weakening which intuitively states that the current state is evident to all players.

We show that the following two questions are decidable:

- Given an n-player game structure with imperfect information, does it satisfy the condition of recurring certainty?
- Given a game with recurring certainty and an ω-regular winning condition, does the grand coalition have a winning strategy?

Towards this, we first prove that, under recurring certainty, the intervals where the current state of the game is not common knowledge are bounded uniformly. We call this periodic certainty. Then, we show that the perfect-information *tracking* [4] of a game with periodic certainty is finite. This allows to solve the synthesis problem.

Acknowledgement. The authors thank Marie Van den Bogaard for useful discussions on related topics and for proof-reading this paper.

2 Coordination games with imperfect information

Our game model is close to that of concurrent games [1]. There are n players $1, \dots, n$ and a distinguished agent called nature. The *grand coalition* is the set $N = \{1, \dots, n\}$ of all players. We refer to a list of elements $x = (x^i)_{i \in N}$, one for each player, as a *profile*.

For each player i we fix a set A^i of *actions* and a set B^i of observations, finite unless stated otherwise. The *action space* A consists of all action profiles. A *game structure* $G = (V, \Delta, (\beta^i)_{i \in N})$ consists of a finite set V of *states*, a relation $\Delta \in V \times A \times V$ of simultaneous *moves* labelled by action profiles, and a profile of *observation* functions $\beta^i : V \to B^i$. We assume that each state has at least one outgoing move for every action profile, i.e., $\Delta(v,a) \neq \emptyset$, for all $v \in V$ and all $a \in A$.

Plays start at an initial state $v_0 \in V$ known to all players, and proceed in rounds. In a round, all players i choose an action $a^i \in A^i$ simultaneously, then nature chooses a successor state $v' \in \Delta(v,a)$ and each player i receives the observation $\beta^i(v')$. Notice that the players are not directly informed about the action chosen by other players nor the state chosen by nature. However, we assume that the player's own action is part of his observation at the target state. Formally, a *play* is an infinite sequence $\pi = v_0, a_0, v_1, a_1, \dots$ alternating between positions and action profiles with $(v_\ell, a, v_{\ell+1}) \in \Delta$, for all $\ell \geq 0$. A *history* is a prefix $v_0, a_0, \dots, a_{\ell-1}, v_\ell$ of a play. The observation function extends from states to histories and plays $\pi = v_0, a_0, v_1, a_1, \dots$ as $\beta^i(\pi) = \beta^i(v_0), \beta^i(v_1), \dots$. We say that two histories π, π' are *indistinguishable* to Player i, and write $\pi \sim^i \pi'$, if $\beta^i(\pi) = \beta^i(\pi')$. This is an equivalence relation, and its classes are called the *information sets* of Player i.

A *strategy* for Player i is a mapping $s^i : (VA)^*V \to A^i$ from histories to actions such that $s^i(\pi) = s^i(\pi')$, for any pair $\pi \sim^i \pi'$ of indistinguishable histories. We denote the set of all strategies of Player i with S^i and the set of all strategy profiles by S. A history or play $\pi = v_0, a_0, v_1, a_1, \ldots$ *follows* the strategy $s^i \in S^i$, if $a^i_\ell = s^i(v_0, a_0, v_1, \ldots, a_{\ell-1}, v_\ell)$ for every $\ell > 0$. For the grand coalition, the play π follows a strategy profile s, if it follows all strategies s^i. The set of possible *outcomes* of a strategy profile s is the set of plays that follow s.

A *winning condition* over a game structure G is a set $W \subseteq (VA)^\omega$ of plays. A *game* $\mathscr{G} = (G, W)$ consists of a game structure and a winning condition. We say that a play π on G is winning in $\mathscr{G}$ if $\pi \in W$; a strategy profile s is winning in $\mathscr{G}$, if all its possible outcomes are so. To describe winning conditions, we use a colouring function $\gamma : V \to C$ with a finite range C of colours, and refer to the set $W \subseteq C^\omega$ of all plays $v_0, a_0, v_1, a_1, \ldots$ with $\gamma(v_0), \gamma(v_1), \cdots \in W$. In this paper, we assume that the colouring is *observable* to each player i, that is, $\beta^i(v) \neq \beta^i(v')$ whenever $\gamma(v) \neq \gamma(v')$.

We consider coordination games over finite game structures where the winning condition is given by finite-state automata. (See [6], for a comprehensive background.) Given such a game $\mathscr{G}$, we are interested in the following questions: (1) Does the grand coalition have a winning strategy profile in $\mathscr{G}$? and (2) How to synthesise (distributed) winning strategies, if they exists?

3 Recurring certainty

We consider a class of games where the uncertainty of players about the current state is temporary and vanishes after a finite number of rounds.

To explain our notion of certainty, we introduce a fictitious player, let us call him Player 0, who is less informed than any actual player. He does not contribute to joint actions (i.e., his action set A^0 is a singleton), and his observation function is a coarsening of all observations of other players: for any pair v, v' of game states, $\beta^0(v) = \beta^0(v')$ whenever $\beta^i(v) = \beta^i(v')$ for some player i. Thus, for histories π, π', we have $\pi \sim^0 \pi'$, whenever $\pi \sim^i \pi'$ for some player i (the converse does not hold, in general).

For a given game structure G, we say that the grand coalition *attains certainty* at history $\pi = v_0, a_0, \ldots, a_{\ell-1}, v_\ell$, if any indistinguishable history $\pi' \sim^0 \pi$ ends at the same state v_ℓ. An infinite play π has *recurring certainty*, if the grand coalition attains certainty at infinitely many of its histories. Finally, we say that the game structure G has recurring certainty, if this is the case for every play in G.

As a simple example of a game with recurring certainty, consider the infinite repetition of a finite extensive game with imperfect information where the root is a perfect-information node, i.e., it is distinguishable from any other node, for every player. Likewise, games on graphs with the property that every cycle passes through a perfect-information state have recurring certainty.

We will also encounter the following stronger property. A game structure G has *periodic certainty* if there exists a uniform bound $t \in \mathbb{N}$ such that for every play π in G, every history ρ of π has a continuation ρ' by at most t rounds in π, such that the grand coalition attains certainty at ρ'.

3.1 Recognising games with recurring certainty

Our first result states that recurring certainty is a regular property of plays in finite game structures.

Lemma 3.1. *For any finite game structure, the set of plays where the grand coalition has recurring certainty is recognisable by a finite-state automaton.*

Proof. Let us fix a finite game structure G. First, we construct a word automaton $\mathscr{A}$ over the alphabet AV that recognises histories ρ at which the grand coalition does not attain certainty. To witness this, the

automaton guesses a second history ρ' (of the same length) that is $\sim^0$-indistinguishable from ρ and ends at a different state.

The state space of $\mathscr{A}$ consists of pairs of game states in V, plus a sink. The first component of the automaton state keeps track of the input history and the second one of the uncertainty witness that is guessed nondeterministically. The transition function ensures that both components evolve according to the moves available in the game structure and yield the same observation to all players; otherwise, they lead to the sink. Accepting states are those where the first and the second component differ.

By complementing the automaton $\mathscr{A}$, we obtain an automaton $\overline{\mathscr{A}}$ that accepts the set of histories at which the grand coalition attains certainty (plus sequences that do not correspond to histories, which can be excluded easily by intersection with the unravelling of G). Next, we determinise $\overline{\mathscr{A}}$ and view the outcome as a deterministic Büchi automaton $\mathscr{B}$ which accepts the input word, if it hits the set of final states infinitely often. Thus, $\mathscr{B}$ accepts all plays where the grand coalition has recurring certainty. □

The synchronous product of the deterministic Büchi automaton $\mathscr{B}$ constructed above with the game structure G is universal, i.e accepts every play of G, if and only if, G has recurring certainty.

Theorem 1. *The question whether a given game structure has recurring certainty is decidable.*

A further consequence of the automaton construction is that we obtain a uniform bound on the distance between two rounds at which the grand coalition attains certainty.

Theorem 2. *Every game with recurring certainty also has periodic certainty.*

Proof. Let G be a game structure with recurring certainty, $\mathscr{B}$ the deterministic Büchi automaton constructed for G as above, and let t be the number of states in $\mathscr{B}$ plus one. Towards a contradiction, suppose there exists a play π in G with a collection of $t > |\mathscr{B}|$ many consecutive histories $\rho_0, \rho_1, \dots \rho_t$ at which the grand coalition does not attain certainty. Accordingly, the uniquely determined run of $\mathscr{B}$ on input π hits no accepting state of the automaton $\mathscr{B}$ while reading the continuation of ρ_0 up to ρ_t. On the other hand, as $t > |\mathscr{B}|$, there exists a state in $\mathscr{B}$ that is reached by two different histories, say ρ_k and ρ_ℓ, with $0 \leq k \leq \ell \leq t$. Now we consider the play π' on G that begins with ρ_ℓ, and then repeats the continuation of ρ_k up to ρ_ℓ forever. Thus, the run of $\mathscr{C}$ on π' will finally not hit any accepting state and be rejected, in contradiction to our assumption that G has recurring certainty. □

3.2 Winner determination and strategy synthesis

Theorem 3. *Let $\mathscr{G}$ be a coordination game with an ω-regular winning condition. If $\mathscr{G}$ has recurring certainty, then the question whether the grand coalition has a winning strategy profile is decidable and the strategy synthesis problem is effectively solvable.*

Our argument relies on the tracking construction proposed in [4] that eliminates imperfect information in n-player games by an unravelling process that generates epistemic models of the player's information along the stages of a play. An *epistemic model* for a game structure $\mathscr{G}$ is a Kripke structure $\mathscr{K} = (K, (Q_v)_{v \in V}, (\sim^i)_{i \in N})$ over a set K of histories in $\mathscr{G}$, equipped with predicates Q_v designating the histories that end in state $v \in V$ and the players' indistinguishability relations $\sim^i$. The construction keeps track of how the knowledge of players is updated by generating, for each epistemic model $\mathscr{K}$, a set of successor models along tuples $(a_k)_{k \in K}$ of action profiles $a_k \in A$ compatible with the player's current knowledge, i.e. for every $i \in N$ and for all $k, k' \in K$ with $k \sim^i k'$, we have $a^i_k = a^i_{k'}$. This leads to a possibly disconnected epistemic model with universe $K' = \{ka_k v \mid k \in K, k \in Q_w \text{ and } (w, a_k, v) \in \Delta\}$ with $Q_v = \{ka_k v \mid ka_k v \in K'\}$ and $ka_k v \sim_i k' a_k v' \iff k \sim_i^{\mathscr{K}} k'$ and $v \sim_i^{\mathscr{G}} v'$. By taking the connected

components of this model under the coarsening $\sim^{\cup} := \bigcup_{i=0}^{n-1} \sim_i$, we obtain the set of epistemic successor models. When starting from the trivial model that consists only of the initial node of the game, and successively applying the update, one unravels a tree labelled with epistemic models, which corresponds to a two-player game of perfect information where the strategies of one player translate to coordination strategies of the grand coalition in the original game, and vice versa. This tree structure, which in general may contain infinitely many distinct labels for its nodes (the undecidable game in [3], for example), is called the *tracking* of the game structure.

The main result of [4] shows that, whenever two nodes of the unravelling tree carry homomorphically equivalent labels, they can be identified without changing the (winning or losing) status of the game. This holds for all imperfect-information games with ω-regular winning conditions that are *observable*. Consequently, the strategy synthesis problem is decidable for a subclass of such games, whenever the unravelling process is guaranteed to generate only finitely many epistemic models, up to homomorphic equivalence.

Let us now consider the tracking of a game $\mathscr{G}$ with an observable ω-regular winning condition. We claim that every history where the grand coalition attains certainty leads to an epistemic model that is homomorphically equivalent to the trivial structure consisting of a singleton labelled with the (certain) state at which the history ends. This is because every $\sim^{\cup}$-connected component is also $\sim^{0}$-connected, and all histories in such a component end at the same state. On the other hand, when updating an epistemic model, the successor models can be at most exponentially larger (for fixed action space). The property of periodic certainty implied by recurring certainty, allows us to conclude that the number of updating rounds in which the models can grow is bounded by the certainty period of G. Therefore, games with recurring certainty have finite tracking. By [4], this implies that the winner determination problem is decidable for such games, and finite-state winning strategies can be effectively synthesised whenever the grand coalition has a winning strategy.

References

[1] Rajeev Alur, Thomas A. Henzinger & Orna Kupferman (2002): *Alternating-time temporal logic*. *J. ACM* 49(5), pp. 672–713, doi:10.1145/585265.585270.

[2] André Arnold & Igor Walukiewicz (2007): *Nondeterministic controllers of nondeterministic processes*. In: *Logic and Automata*, 2, Amsterdam University Press.

[3] Dietmar Berwanger & Łukasz Kaiser (2010): *Information Tracking in Games on Graphs*. *Journal of Logic, Language and Information* 19(4), pp. 395–412, doi:10.1007/s10849-009-9115-8.

[4] Dietmar Berwanger, Łukasz Kaiser & Bernd Puchala (2011): *Perfect-Information Construction for Coordination in Games*. In: *Proceedings of the 31st Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'11)*, *Leibniz International Proceedings in Informatics* 13, Leibniz-Zentrum für Informatik, Mumbai, India, pp. 387–398, doi:10.4230/LIPIcs.FSTTCS.2011.387.

[5] B. Finkbeiner & S. Schewe (2005): *Uniform Distributed Synthesis*. In: *Proc. of LICS '05*, IEEE, pp. 321–330, doi:10.1109/LICS.2005.53.

[6] Erich Grädel, Wolfgang Thomas & Thomas Wilke, editors (2002): *Automata, Logics, and Infinite Games*. *LNCS* 2500, Springer-Verlag, doi:10.1007/3-540-36387-4.

[7] David Janin (2007): *On the (High) Undecidability of Distributed Synthesis Problems*. In: *Proc. of Theory and Practice of Computer Science (SOFSEM 2007)*, *Lecture Notes in Computer Science* 4362, Springer, pp. 320–329, doi:10.1007/978-3-540-69507-3_26.

[8] Łukasz Kaiser (2006): *Game Quantification on Automatic Structures and Hierarchical Model Checking Games*. In: *Proc. of CSL '06*, *LNCS* 4207, Springer, pp. 411–425, doi:10.1007/11874683_27.

[9] Orna Kupferman & Moshe Y. Vardi (2001): *Synthesizing Distributed Systems*. In: *Proc. of LICS '01*, IEEE Computer Society Press, pp. 389–398, doi:10.1109/LICS.2001.932514.

[10] Amir Pnueli & Roni Rosner (1989): *On the synthesis of a reactive module*. In: *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, POPL '89*, ACM Press, pp. 179–190, doi:10.1145/75277.75293.

[11] Amir Pnueli & Roni Rosner (1990): *Distributed Reactive Systems Are Hard to Synthesize*. In: *Proceedings of the 31st Annual Symposium on Foundations of Computer Science, FoCS '90*, IEEE Computer Society Press, pp. 746–757, doi:10.1109/FSCS.1990.89597.

[12] John H. Reif (1984): *The Complexity of Two-Player Games of Incomplete Information*. *Journal of Computer and Systems Sciences* 29(2), pp. 274–301, doi:10.1016/0022-0000(84)90034-5.

Automata Techniques for Epistemic Protocol Synthesis

Guillaume Aucher
Université de Rennes 1 - INRIA
Rennes, France
guillaume.aucher@irisa.fr

Bastien Maubert
Université de Rennes 1
Rennes, France
bastien.maubert@irisa.fr

Sophie Pinchinat
Université de Rennes 1
Rennes, France
sophie.pinchinat@irisa.fr

In this work we aim at applying automata techniques to problems studied in Dynamic Epistemic Logic, such as epistemic planning. To do so, we first remark that repeatedly executing *ad infinitum* a propositional event model from an initial epistemic model yields a relational structure that can be finitely represented with automata. This correspondence, together with recent results on *uniform strategies*, allows us to give an alternative decidability proof of the epistemic planning problem for propositional events, with as by-products accurate upper-bounds on its time complexity, and the possibility to synthesize a finite word automaton that describes the set of all solution plans. In fact, using automata techniques enables us to solve a much more general problem, that we introduce and call *epistemic protocol synthesis*.

1 Introduction

Automated planning, as defined and studied in [9], consists in computing a finite sequence of actions that takes some given system from its initial state to one of its designated "goal" states. The Dynamic Epistemic Logic (DEL) community has recently investigated a particular case of automated planning, called *epistemic planning* [7, 11, 1]. In DEL, epistemic models and event models can describe accurately how agents perceive the occurrence of events, and how their knowledge or beliefs evolve. Given initial epistemic states of the agents, a finite set of available events, and an epistemic objective, the epistemic planning problem consists in computing (if any) a finite sequence of available events whose occurrence results in a situation satisfying the objective property. While this problem is undecidable in general [7, 1], restricting to *propositional events* (those whose pre and postconditions are propositional) yields decidability [19].

In this paper, preliminary to our main results we bring a new piece to the merging of various frameworks for knowledge and time. Some connections between DEL and Epistemic Temporal Logics (ETL) are already known [10, 4, 2, 18]. We establish that structures generated by iterated execution of an event model from an epistemic model are regular structures, *i.e.* they can be finitely represented with automata, in case the event model is propositional. This allows us to reduce the epistemic planning problem for propositional events to the *uniform strategy problem*, as studied in [13, 14, 12]. The automata techniques developed for uniform strategies then provide an alternative proof of [19], with the additional advantage of bringing accurate upper-bounds on the time complexity of the problem, as well as an effective synthesis procedure to generate the recognizer of all solution plans. In fact, our approach allows us to solve a generalized problem in DEL, that we call *epistemic protocol synthesis problem*, and which is essentially the problem of synthesizing a protocol from an epistemic temporal specification; its semantics relies on the interplay between DEL and ETL. We then make use of the connections with regular structures and uniform strategies to solve this latter general problem.

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 97–103, doi:10.4204/EPTCS.146.13

2 DEL models

For this paper we fix Ag, a finite set of *agents*, and AP always denotes a finite set of atomic propositions (which is not fixed). The epistemic language $\mathscr{L}^{EL}$ is simply the language of propositional logic extended with "knowledge" modalities, one for each agent. Intuitively, $K_i\varphi$ reads as "agent i knows φ". The syntax of $\mathscr{L}^{EL}$ is given by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid K_i\varphi, \qquad (\text{where } p \in AP \text{ and } i \in Ag)$$

The semantics of $\mathscr{L}^{EL}$ is given in terms of epistemic models. Intuitively, a (pointed) epistemic model $(\mathscr{M},w)$ represents how the agents perceive the actual world w.

Definition 1 *An* epistemic model *is a tuple $\mathscr{M} = (W,\{R_i\}_{i\in Ag},V)$ where W is a finite set of possible* worlds, *$R_i \subseteq W \times W$ is an* accessibility relation *on W for agent $i \in Ag$, and $V : AP \to 2^W$ is a* valuation function.

We write $w \in \mathscr{M}$ for $w \in W$, and we call $(\mathscr{M},w)$ a *pointed epistemic model*. Formally, given a pointed epistemic model $(\mathscr{M},w)$, we define the semantics of $\mathscr{L}^{EL}$ by induction on its formulas: $\mathscr{M},w \models p$ if $w \in V(p)$, $\mathscr{M},w \models \neg\varphi$ if it is not the case that $\mathscr{M},w \models \varphi$, $\mathscr{M},w \models \varphi \vee \psi$ if $\mathscr{M},w \models \varphi$ or $\mathscr{M},w \models \psi$, and $\mathscr{M},w \models K_i\varphi$ if for all w' such that wR_iw', $\mathscr{M},w' \models \varphi$.

Definition 2 *An* event model *is a tuple $\mathscr{E} = (\mathsf{E},\{\mathsf{R}_i\}_{i\in Ag},pre,post)$ where E is finite set of* events, *for each $i \in Ag$, $\mathsf{R}_i \subseteq \mathsf{E} \times \mathsf{E}$ is an* accessibility relation *on E for agent i, $pre : \mathsf{E} \to \mathscr{L}^{EL}$ is a* precondition function *and $post : \mathsf{E} \to AP \to \mathscr{L}^{EL}$ is a* postcondition function.

We write $e \in \mathscr{E}$ for $e \in \mathsf{E}$, and call $(\mathscr{E},e)$ a *pointed event model*. For an event $e \in \mathscr{E}$, the precondition $\mathrm{pre}(e)$ and the postconditions $\mathrm{post}(e)(p)$ ($p \in AP$) are epistemic formulas. They respectively describe the set of worlds where event e may take place and the set of worlds where proposition p will hold after event e has occurred.

Definition 3 *A* proposition event model *is an event model whose preconditions and postconditions all lie in the propositional fragment of $\mathscr{L}^{EL}$.*

We now define the *update product* which, given an epistemic model $\mathscr{M}$ and an event model $\mathscr{E}$, builds the epistemic model $\mathscr{M} \otimes \mathscr{E}$ that represents the new epistemic situation after $\mathscr{E}$ has occurred in $\mathscr{M}$.

Definition 4 *Let $\mathscr{M} = (W,\{R_i\}_{i\in Ag},V)$ be an epistemic model and $\mathscr{E} = (\mathsf{E},\{\mathsf{R}_i\}_{i\in Ag},pre,post)$ be an event model. The* update product *of $\mathscr{M}$ and $\mathscr{E}$ is the epistemic model $\mathscr{M} \otimes \mathscr{E} = (W^\otimes,\{R_i^\otimes\}_{i\in Ag},V^\otimes)$, where $W^\otimes = \{(w,e) \in W \times \mathsf{E} \mid \mathscr{M},w \models pre(e)\}$, $R_i^\otimes(w,e) = \{(w',e') \in W^\otimes \mid w' \in R_i(w)$ and $e' \in \mathsf{R}_i(e)\}$, and $V^\otimes(p) = \{(w,e) \in W^\otimes \mid \mathscr{M},w \models post(e)(p)\}$.*

The update product of a pointed epistemic model $(\mathscr{M},w)$ with a pointed event model $(\mathscr{E},e)$ is $(\mathscr{M},w) \otimes (\mathscr{E},e) = (\mathscr{M} \otimes \mathscr{E},(w,e))$ if $\mathscr{M},w \models \mathrm{pre}(e)$, and it is undefined otherwise.

To finish with this section, we define the *size* of an epistemic model $\mathscr{M} = (W,\{R_i\}_{i\in Ag},V)$, denoted by $|\mathscr{M}|$, as its number of edges: $|\mathscr{M}| = \sum_{i\in Ag}|R_i|$. The size of an event model $\mathscr{E} = (\mathsf{E},\{\mathsf{R}_i\}_{i\in Ag},\mathrm{pre},\mathrm{post})$, that we note $|\mathscr{E}|$, is its number of edges plus the sizes of precondition and postcondition formulas: $|\mathscr{E}| = \sum_{i\in Ag}|\mathsf{R}_i| + \sum_{e\in\mathsf{E}}(|\mathrm{pre}(e)| + \sum_{p\in AP}|\mathrm{post}(e)(p)|)$.

3 Trees, forests and $\mathsf{CTL}^*\mathsf{K}_n$

A *tree alphabet* is a finite set of *directions* $\Upsilon = \{d_1, d_2 \ldots\}$. A Υ-*tree*, or *tree* for short when Υ is clear from the context, is a set of words $\tau \subseteq \Upsilon^+$ that is closed for nonempty prefixes, and for which there is a direction $r = \tau \cap \Upsilon$, called the *root*, such that for all $x \in \tau$, $x = r \cdot x'$ for some $x' \in \Upsilon^*$. A Υ-*forest*, or *forest* when Υ is understood, is defined likewise, except that it can have several roots. Alternatively a forest can be seen as a union of trees.

We classically allow nodes of trees and forests to carry additional information via labels: given a *labelling alphabet* Σ and a tree alphabet Υ, a Σ-*labelled* Υ-*tree*, or (Σ,Υ)-*tree* for short, is a pair $t = (\tau, \ell)$, where τ is a Υ-tree and $\ell : \tau \to \Sigma$ is a *labelling* . The notion of (Σ,Υ)-*forest* $\mathscr{U} = (u, \ell)$ is defined likewise. Note that we use forests to represent the universe (to be defined) in the semantics of $\mathsf{CTL}^*\mathsf{K}_n$, hence the notations $\mathscr{U}$ and u. Given a Υ-forest u and a node $x = d_1 \ldots d_n$ in the forest u, we define the tree u_x to which this node belongs as the "greatest" tree in the forest u that contains the node x: $u_x = \{y \in u \mid d_1 \preccurlyeq y\}$. Similarly, given a (Σ,Υ)-forest $\mathscr{U} = (u, \ell)$ and a node $x \in u$, $\mathscr{U}_x = (u_x, \ell_x)$, where u_x is as above and ℓ_x is the restriction of ℓ to the tree u_x.

The set of well-formed $\mathsf{CTL}^*\mathsf{K}_n$ formulas is given by the following grammar:

$$\begin{array}{lll} \text{State formulas:} & \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{A}\psi \mid K_i\varphi & (\text{where } p \in AP \text{ and } i \in Ag) \\ \text{Path formulas:} & \psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi \mathbf{U} \psi, & \end{array}$$

Let Υ be a finite set of directions, and let $\Sigma = 2^{AP}$ be the set of possible valuations. A $\mathsf{CTL}^*\mathsf{K}_n$ (state) formula is interpreted in a node of a (Σ,Υ)-tree, but the semantics is parameterized by, first, for each agent $i \in Ag$, a binary relation $\rightsquigarrow_i$ between finite words over Σ, and second, a forest of (Σ,Υ)-trees which we see as the *universe*. Preliminary to defining the semantics of $\mathsf{CTL}^*\mathsf{K}_n$, we let the *node word* of a node $x = d_1 d_2 \ldots d_n \in \tau$ be $w(x) = \ell(d_1)\ell(d_1 d_2)\ldots\ell(d_1 \ldots d_n) \in \Sigma^*$, made of the sequence of labels of all nodes from the root to this node. Now, given a family $\{\rightsquigarrow_i\}_{i\in Ag}$ of binary relations over Σ^*, a (Σ,Υ)-forest $\mathscr{U}$, two nodes $x, y \in \mathscr{U}$ and $i \in Ag$, we let $x \rightsquigarrow_i y$ denote that $w(x) \rightsquigarrow_i w(y)$.

A state formula of $\mathsf{CTL}^*\mathsf{K}_n$ is interpreted over a (Σ,Υ)-tree $t = (\tau, \ell)$ in a node $x \in \tau$, with an implicit universe $\mathscr{U}$ and relations $\{\rightsquigarrow_i\}_{i\in Ag}$, usually clear from the context: the notation $t, x \models \varphi$ means that φ holds at the node x of the labelled tree t. Because all inductive cases but the knowledge operators follow the classic semantics of CTL^* on trees, we only give the semantics for formulas of the form $K_i\varphi$:

$t, x \models K_i\varphi$ if for all $y \in \mathscr{U}$ such that $x \rightsquigarrow y$, $\mathscr{U}_y, y \models \varphi$ [1]

We shall use the notation $t \models \varphi$ for $t, r \models \varphi$, where r is the root of t.

Before stating the problems considered and our results, we establish in the next section a connection between DEL-generated models and regular structures, that allows us to apply automata techniques to planning problems in DEL.

4 DEL-generated models and regular structures

We first briefly recall some basic definitions and facts concerning finite state automata and transducers. A *deterministic word automaton* is a tuple $\mathscr{A} = (\Sigma, Q, \delta, q_t, F)$, where Σ is an *alphabet*, Q is a finite set of *states*, $\delta : Q \times \Sigma \to Q$ is a partial *transition function* and F is a set of *accepting* states. The *language* accepted by a word automaton $\mathscr{A}$ consists in the set of words accepted by $\mathscr{A}$, and it is classically written

[1] Recall that $\mathscr{U}_y$ is the biggest tree in $\mathscr{U}$ that contains y.

$\mathscr{L}(\mathscr{A})$. It is well known that the set of languages accepted by word automata is precisely the set of *regular* word languages. A *finite state synchronous transducer*, or *synchronous transducer* for short, is a finite word automaton with two tapes, that reads one letter from each tape at each transition. Formally, a synchronous transducer is a tuple $T = (\Sigma, Q, \Delta, q_\iota, F)$, where the components are as for word automata, except for the *transition relation* $\Delta \subseteq Q \times \Sigma \times \Sigma \times Q$. The (binary) relation recognized by a transducer T is denoted by $[T] \subseteq \Sigma^* \times \Sigma^*$. Synchronous transducers are known to recognize the set of *regular relations*, also called *synchronized rational relations* in the literature (see [8, 6, 3]). In the following, the size of a transducer T, written $|T|$, will denote the size of its transition relation: $|T| = |\Delta|$.

Definition 5 *A* relational structure *is a tuple $\mathscr{S} = (D, \{\rightsquigarrow_i\}_{i\in Ag}, V)$ where D is the (possibly infinite)* domain *of $\mathscr{S}$, for each $i \in Ag$, $\rightsquigarrow_i \subseteq D \times D$ is a binary relation and $V : AP \to 2^D$ is a valuation function. V can alternatively be seen as a set of predicate interpretations for atomic propositions in AP.*

Definition 6 *A relational structure $\mathscr{S} = (D, \{\rightsquigarrow_i\}_{i\in Ag}, V)$ is a* regular structure *over a finite alphabet Σ if its domain $D \subseteq \Sigma^*$ is a regular language over Σ, for each i, $\rightsquigarrow_i \subseteq \Sigma^* \times \Sigma^*$ is a regular relation and for each $p \in AP$, $V(p) \subseteq D$ is a regular language. Given deterministic word automata $\mathscr{A}_\mathscr{S}$ and $\mathscr{A}_p$ ($p \in AP$), as well as transducers T_i for $i \in Ag$, we say that $(\mathscr{A}_\mathscr{S}, \{T_i\}_{i\in Ag}, \{\mathscr{A}_p\}_{p\in AP})$ is a* representation *of $\mathscr{S}$ if $\mathscr{L}(\mathscr{A}_\mathscr{S}) = D$, for each $i \in Ag$, $[T_i] = \rightsquigarrow_i$ and for each $p \in AP$, $\mathscr{L}(\mathscr{A}_p) = V(p)$.*

Definition 7 *For an epistemic model $\mathscr{M} = (W, \{R_i\}_{i\in Ag}, V)$ and an event model $\mathscr{E} = (\mathsf{E}, \{\mathsf{R}_i\}_{i\in Ag}, pre, post)$, we define the family of epistemic models $\{\mathscr{M}\mathscr{E}^n\}_{n\geq 0}$ by letting $\mathscr{M}\mathscr{E}^0 = \mathscr{M}$ and $\mathscr{M}\mathscr{E}^{n+1} = \mathscr{M}\mathscr{E}^n \otimes \mathscr{E}$. Letting, for each n, $\mathscr{M}\mathscr{E}^n = (W^n, \{R_i^n\}_{i\in Ag}, V^n)$, we define the relational structure generated by $\mathscr{M}$ and $\mathscr{E}$ as $\mathscr{M}\mathscr{E}^* = (D, \{\rightsquigarrow_i\}_{i\in Ag}, V)$, where:*

- $D = \bigcup_{n\geq 0} W^n$,
- *$h \rightsquigarrow_i h'$ if there is some n such that $h, h' \in \mathscr{M}\mathscr{E}^n$ and $h R_i^n h'$, and*
- $V(p) = \bigcup_{n\geq 0} V^n(p)$.

Proposition 1 *If $\mathscr{M}$ is an epistemic model and $\mathscr{E}$ is a propositional event model, then $\mathscr{M}\mathscr{E}^*$ is a regular structure, and it admits a representation of size $2^{O(|AP|)} \cdot (|\mathscr{M}| + |\mathscr{E}|)^{O(1)}$.*

Proof Let $\mathscr{M} = (W, R, V)$ be an epistemic model, let $\mathscr{E} = (\mathsf{E}, \mathsf{R}, \mathrm{pre}, \mathrm{post})$ be a propositional event model, and let $\mathscr{M}\mathscr{E}^* = (D, \{\rightsquigarrow_i\}_{i\in Ag}, V_D)$.

Define the word automaton $\mathscr{A}_D = (\Sigma, Q, \delta, q_\iota, F)$, where $\Sigma = W \cup \mathsf{E}$, $F = \{q_\nu \mid \nu \subseteq AP\}$ and $Q = F \uplus \{q_\iota\}$. For a world $w \in W$, we define its *valuation* as $\nu(w) := \{p \in AP \mid w \in V(p)\}$. We now define δ, which is the following partial transition function:

$$\begin{array}{ll} \forall w \in W, \forall e \in \mathsf{E}, & \\ \delta(q_\iota, w) = q_{\nu(w)} & \delta(q_\iota, e) \text{ is undefined,} \\ \delta(q_\nu, w) \text{ is undefined} & \delta(q_\nu, e) = \begin{cases} q_{\nu'}, \text{ with } \nu' = \{p \mid \nu \models \mathrm{post}(e)(p)\} & \text{if } \nu \models \mathrm{pre}(e) \\ \text{undefined} & \text{otherwise.} \end{cases} \end{array}$$

It is not hard to see that $\mathscr{L}(\mathscr{A}_D) = D$, hence D is a regular language. Also, $\mathscr{A}_D$ has $2^{|AP|} + 1$ states, and each state has at most $|\mathscr{M}| + |\mathscr{E}|$ outgoing transitions, so that $|\mathscr{A}_D| = 2^{O(|AP|)} \cdot (|\mathscr{M}| + |\mathscr{E}|)$.

Concerning valuations, take some $p \in AP$. Let $\mathscr{A}_p = (\Sigma, Q, \delta, q_\iota, F_p)$, where $F_p = \{q_\nu \mid p \in \nu\}$. Clearly, $\mathscr{L}(\mathscr{A}_p) = V_D(p)$, hence $V_D(p)$ is a regular language, and $|\mathscr{A}_p| = |\mathscr{A}_D|$.

For the relations, let $i \in Ag$ and consider the one-state synchronous transducer $T_i = (\Sigma, Q', \Delta_i, q_\iota, F')$, where $Q' = \{q\}$, $q_\iota = q$, $F' = \{q\}$, and $\Delta_i = \{(q, w, w', q) \mid w R_i w'\} \cup \{(q, e, e', q) \mid e \mathsf{R}_i e'\}$. It is easy to see

that $\rightsquigarrow_i = [T_i] \cap D \times D$. Since $[T_i]$ is a regular relation and D is a regular language, $\rightsquigarrow_i$ is a regular relation recognized by $T_i' = T_D \circ T_i \circ T_D$, where T_D is a synchronous transducer that recognizes the identity relation over D (easily obtained from $\mathscr{A}_D$). This transducer is of size $|T_i'| = |T_D|^2 \cdot |T_i| = 2^{O(|AP|)} \cdot (|\mathscr{M}| + |\mathscr{E}|)^{O(1)}$. Finally, $\mathscr{M}\mathscr{E}^*$ is a regular structure that accepts $(\mathscr{A}_D, \{T_i'\}_{i \in Ag}, \{\mathscr{A}_p\}_{p \in AP})$ as a regular representation of size $2^{O(|AP|)} \cdot (|\mathscr{M}| + |\mathscr{E}|)^{O(1)}$. One can check that this is also an upper bound on the time needed to compute this representation.

□

5 Epistemic protocol synthesis

We first consider the problem of epistemic planning [7, 11] studied in the Dynamic Epistemic Logic community. Note that our formulation slightly differs from the classic one as we consider a unique event model, but both problems can easily be proved inter-reducible in linear time.

Definition 8 (Epistemic planning problem) *Given a pointed epistemic model $(\mathscr{M}_\iota, w_\iota)$, an event model $\mathscr{E}$, a set of events $\mathsf{E} \subseteq \mathscr{E}$ and a goal formula $\varphi \in \mathscr{L}^{EL}$, decide if there exists a finite series of events $e_1 \dots e_n$ in E such that $(\mathscr{M}_\iota, w_\iota) \otimes (\mathscr{E}, e_1) \otimes \dots \otimes (\mathscr{E}, e_n) \models \varphi$. The* propositional epistemic planning problem *is the restriction of the epistemic planning problem to propositional event models.*

The epistemic planning problem is undecidable [7, 1]. However, [7] proved that the problem is decidable in the case of one agent and equivalence accessibility relations in epistemic and event models. More recently, [1] and [19] proved independently that the one agent problem is also decidable for K45 accessibility relations. [19] also proved that restricting to propositional event models yields decidability of the epistemic planning problem, even for several agents and arbitrary accessibility relations.

Theorem 2 ([19]) *The propositional epistemic planning problem is decidable.*

Proposition 1 allows us to establish an alternative proof of this result, with two side-benefits. First, using automata techniques, our decision procedure can synthesize as a by-product a finite word automaton that generates exactly the (possibly infinite) set of all solution plans. Second, we obtain accurate upper-bounds on the time complexity.

For an instance $(\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi)$ of the epistemic planning problem, we define its size as the sum of its components' sizes, plus the number of atomic propositions: $|\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi| = |\mathscr{M}| + |\mathscr{E}| + |\mathsf{E}| + |\varphi| + |AP|$.

Theorem 3 *The propositional epistemic planning problem is in $k+1$-*EXPTIME *for formulas of nesting depth k. Moreover, it is possible to build in the same time a finite word automaton $\mathscr{P}$ such that $\mathscr{L}(\mathscr{P})$ is the set of all solution plans.*

Proof sketch Let $(\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi)$ be an instance of the problem. By Proposition 1 we obtain an exponential size automatic representation of the forest $\mathscr{M}\mathscr{E}^*$: the set of possible histories, as well as their valuations, are represented by a finite automaton $\mathscr{A}$, and the epistemic relations are given by finite state transducers. Because the epistemic relations are rational, we can use the powerset construction presented in [13] in the context of uniform strategies [13, 14, 12]. Indeed, this construction easily generalizes to the case of n relations, and even though in [13] it is defined on game arenas it can, in our context, be adapted to regular structures. Letting k be the maximal nesting depth of knowledge operators in φ, this construction yields an automaton $\widehat{\mathscr{A}}$ of size k-exponential in the size of $\mathscr{A}$, hence $(k+1)$-exponential in $|\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi|$, that still represents $\mathscr{M}\mathscr{E}^*$, and in which φ can be evaluated positionally. Keeping only transitions labelled by events in E, and choosing for accepting states those that verify φ, we obtain the

automaton $\mathscr{P}$ that recognizes the set of solution plans. Furthermore, solving the epistemic planning problem amounts to solving the nonemptiness problem for $\mathscr{L}(\mathscr{P})$; this can be done in time linear in the size of $\mathscr{P}$, which is $k+1$-exponential in the size of the input $(\mathscr{M}, \mathscr{E}, \mathsf{E}, \varphi)$. □

In fact, the correspondence between the DEL framework and automatic structures established in Proposition 1 allows us to solve a much more general problem than epistemic planning.

We generalize the notion of epistemic planning in three directions. First, we no longer consider finite sequences of actions but infinite ones. As a consequence, we need not stick to reachability objectives as in planning (where the aim is to reach a state of the world that verifies some formula), and we therefore allow for any epistemic temporal formula as objective, which is the second generalization. Finally, we no longer look for a single series of events, but we try to synthesize a *protocol*, *i.e.* a set of plans.

Definition 9 *Given an epistemic model $\mathscr{M}$ and an event model $\mathscr{E}$, an* epistemic protocol *is a forest $P \subseteq \mathscr{M}\mathscr{E}^*$; it is* rooted *if it is a tree.*

Definition 10 (Epistemic protocol synthesis problem) *Given an initial pointed epistemic model $(\mathscr{M}, w)$, a propositional event model $\mathscr{E}$ and a $\mathsf{CTL}^*\mathsf{K}_n$ formula φ, letting $\mathscr{U} = \mathscr{M}\mathscr{E}^*$ be the universe, decide if there is an epistemic protocol $P \subseteq \mathscr{U}$ rooted in w such that $P \models \varphi$, and synthesize such a protocol if any.*

Again making use of Proposition 1, the epistemic protocol synthesis problem can be reduced to synthesizing a uniform strategy in a game arena with regular relations between plays. This can be solved with the powerset construction from [13] and classic automata techniques for solving games with CTL^* winning condition. We finally obtain the following result.

Theorem 4 *The epistemic protocol synthesis problem is decidable. If the nesting depth of the goal formulas is bounded by k, then the problem is in* $\max(2, k+1)$-EXPTIME.

6 Discussion

We have described a connection between DEL-generated models and regular structures, which enabled us to resort to a combination of mature automata techniques and more recent ones developed for the study of uniform strategies, in order to solve planning problems in the framework of DEL. We believe that this is but a first step in applying classic automata techniques developed for temporal logics to the study of dynamic epistemic logic. As witnessed by classic works on automata-based program synthesis (see for example [15, 17]), automata techniques are well suited to tackle problems such as synthesizing plans, protocols, strategies or programs, and we believe that it should also be the case in the DEL framework; in addition the complexity of solving classic automata problems such as nonemptiness has been extensively studied, and this may help to settle the complexity of problems in DEL, such as the epistemic planning problem.

As for future work, we would like to investigate the optimality of the upper-bounds that we obtained on the time complexity of the epistemic planning problem for propositional event models, as well as for our notion of epistemic protocol synthesis. Another direction for future research concerns the latter problem: a next step would be to apply techniques from control theory and quantified μ-calculus [16] to synthesize *maximal permissive* epistemic protocols. In general such objects only exist for safety objectives, but recently a weaker notion of *permissive strategy* has been studied in the context of parity games [5]. A strategy is permissive if it contains the behaviours of all memoryless strategies, and such strategies always exist in parity games. Similar notions may be introduced for protocols with epistemic temporal objectives to capture concepts of "sufficiently permissive" protocols.

References

[1] Guillaume Aucher & Thomas Bolander (2013): *Undecidability in Epistemic Planning*. In: *IJCAI*. Available at http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6903.

[2] Guillaume Aucher & Andreas Herzig (2011): *Exploring the power of converse events*. In: *Dynamic formal epistemology*, Springer, pp. 51–74, doi:10.1007/978-94-007-0074-1_4.

[3] Pablo Barceló, Diego Figueira & Leonid Libkin (2013): *Graph Logics with Rational Relations*. *Logical Methods in Computer Science* 9(3). Available at http://dx.doi.org/10.2168/LMCS-9(3:1)2013, http://arxiv.org/abs/1304.4150.

[4] Johan van Benthem, Jelle Gerbrandy, Tomohiro Hoshi & Eric Pacuit (2009): *Merging frameworks for interaction*. *Journal of Philosophical Logic* 38(5), pp. 491–526, doi:10.1007/s10992-008-9099-x.

[5] Julien Bernet, David Janin & Igor Walukiewicz (2002): *Permissive strategies: from parity games to safety games*. *ITA* 36(3), pp. 261–275. Available at http://dx.doi.org/10.1051/ita:2002013.

[6] Jean Berstel (1979): *Transductions and context-free languages*. 4, Teubner Stuttgart, doi:10.1007/978-3-663-09367-1.

[7] Thomas Bolander & Mikkel Birkegaard Andersen (2011): *Epistemic planning for single and multi-agent systems*. *Journal of Applied Non-Classical Logics* 21(1), pp. 9–34. Available at http://dx.doi.org/10.3166/jancl.21.9-34.

[8] C.C. Elgot & J.E. Mezei (1965): *On relations defined by generalized finite automata*. *IBM Journal of Research and Development* 9(1), pp. 47–68, doi:10.1147/rd.91.0047.

[9] Malik Ghallab, Dana S. Nau & Paolo Traverso (2004): *Automated planning - theory and practice*. Elsevier.

[10] Tomohiro Hoshi & Audrey Yap (2009): *Dynamic epistemic logic with branching temporal structures*. *Synthese* 169(2), pp. 259–281. Available at http://dx.doi.org/10.1007/s11229-009-9552-6.

[11] Benedikt Löwe, Eric Pacuit & Andreas Witzel (2011): *DEL Planning and Some Tractable Cases*. In Hans P. van Ditmarsch, Jérôme Lang & Shier Ju, editors: *LORI, Lecture Notes in Computer Science* 6953, Springer, pp. 179–192. Available at http://dx.doi.org/10.1007/978-3-642-24130-7_13.

[12] Bastien Maubert & Sophie Pinchinat (2013): *Jumping Automata for Uniform Strategies*. In: *FSTTCS'13*, pp. 287–298, doi:10.4230/LIPIcs.FSTTCS.2013.287.

[13] Bastien Maubert & Sophie Pinchinat (2014): *A General Notion of Uniform Strategies*. *International Game Theory Review* 16(01), doi:10.1142/S0219198914400040.

[14] Bastien Maubert, Sophie Pinchinat & Laura Bozzelli (2013): *The Complexity of Synthesizing Uniform Strategies*. In Fabio Mogavero, Aniello Murano & Moshe Y. Vardi, editors: *SR*, *EPTCS* 112, pp. 115–122. Available at http://dx.doi.org/10.4204/EPTCS.112.17.

[15] A. Pnueli & R. Rosner (1989): *On the Synthesis of an Asynchronous Reactive Module*. In: *Proc. 16th Int. Coll. on Automata, Languages and Programming, ICALP'89, Stresa, Italy, LNCS 372*, Springer-Verlag, pp. 652–671, doi:10.1007/BFb0035790.

[16] Stéphane Riedweg & Sophie Pinchinat (2003): *Quantified Mu-Calculus for Control Synthesis*. In Branislav Rovan & Peter Vojtás, editors: *MFCS*, *Lecture Notes in Computer Science* 2747, Springer, pp. 642–651. Available at http://dx.doi.org/10.1007/978-3-540-45138-9_58.

[17] Wolfgang Thomas (1995): *On the Synthesis of Strategies in Infinite Games*. In: *STACS*, pp. 1–13. Available at http://dx.doi.org/10.1007/3-540-59042-0_57.

[18] Yanjing Wang & Guillaume Aucher (2013): *An Alternative Axiomatization of DEL and Its Applications*. In: *IJCAI*. Available at http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6802.

[19] Quan Yu, Ximing Wen & Yongmei Liu (2013): *Multi-Agent Epistemic Explanatory Diagnosis via Reasoning about Actions*. In: *IJCAI*, pp. 1183–1190. Available at http://ijcai.org/papers13/Papers/IJCAI13-178.pdf.

Partial Preferences for Mediated Bargaining

Piero A. Bonatti Marco Faella Luigi Sauro
Dept. of Electrical Engineering and Information Technologies
Università di Napoli "Federico II"
Italy

In this work we generalize standard Decision Theory by assuming that two outcomes can also be incomparable. Two motivating scenarios show how incomparability may be helpful to represent those situations where, due to lack of information, the decision maker would like to maintain different options *alive* and defer the final decision. In particular, a new axiomatization is given which turns out to be a weakening of the classical set of axioms used in Decision Theory. Preliminary results show how preferences involving complex distributions are related to judgments on single alternatives.

1 Introduction

In his pioneering work on Decision Theory [4], when delineating the fundamental properties of a preference relation $\prec$, Savage makes the following point: given two potential outcomes f and g, it cannot be the case that $f \prec g$ and $g \prec f$ at the same time. Clearly, this is logically equivalent to saying that either $f \not\prec g$ or $g \not\prec f$, which leads to three possible cases: *(i)* $f \not\prec g$ and $g \prec f$, *(ii)* $f \prec g$ and $g \not\prec f$, or *(iii)* $f \not\prec g$ and $g \not\prec f$. Then, he postulates that these three cases are the only possible judgments concerning f and g. In particular, the last case ($f \not\prec g$ and $g \not\prec f$) allegedly implies that f and g are equivalent in the sense that in any situation wherein these are the only two possible options, the decision maker does not mind delegating to coin flipping. Consequently, in classical Decision Theory (*CDT*) a very fundamental property of a preference relation is its totality.

From the theory's very start, the hidden assumptions underlying this model of an *economic man* raised some criticisms, one of the most influential of which being due to Simon:

> "This man is assumed to have knowledge of the relevant aspects of his environment which, if not absolutely complete, is at least impressively clear and voluminous. He is assumed also to have a well-organized and stable system of preferences, and a skill in computation that enables him to calculate, for the alternative courses of action that are available to him, which of these will permit him to reach the highest attainable point on his preference scale" [5].

In recent years, the massive development of e-commerce services makes Simon's criticisms even more cogent and the classical viewpoint on the economic man more and more idealistic. Often preferences result from complex trade-offs between different attributes (functionalities, cost, Quality of Service (QoS), information disclosure risks, etc.) sometimes the user has only a vague idea of. Moreover, in some cases the user actually consists of a group of persons where internal debate does not easily end up with a total preference. Finally, from a computer-science perspective, our aim could be to develop a software agent that acts in an electronic market on behalf of a real user. As we discuss in Section 2, even if the user conforms with the classical economic man, her preference relation could be so complex that it could not be entirely and efficiently injected into the software agent.

Differently from Simon, who moved towards a problem-solving perspective, in this work we challenge CDT on its playground. In particular, we provide an alternative axiomatization where two outcomes, due to the lack of information or an irreducible heterogeneity of the attributes involved, can also

F. Mogavero, A. Murano, and M.Y. Vardi (Eds.):
2nd Workshop on Strategic Reasoning 2014 (SR14)
EPTCS 146, 2014, pp. 105–111, doi:10.4204/EPTCS.146.14

be incomparable. In Section 2, we show two motivating scenarios where considering preferences as incomplete seems to be appropriate. In Section 3, we introduce the new axiomatization by emphasizing which axioms of CDT remain the same and which axioms should be replaced. The resulting theory, that we call Partial Decision Theory, consists in a weakening of CDT, that is, all the properties it satisfies are satisfied by CDT as well (but not vice versa). In Section 3, we show some general features of the new axiomatization; in particular, we argue that the proposed framework is not too weak, as it retains several desirable properties of CDT. Conclusions and future works end the paper.

2 Motivating Scenarios

In this section we introduce two scenarios where partial preferences seem to provide a more natural way to describe a decision maker, or a software agent behaving on its behalf, than total preferences.

In the first scenario, Bob wants to learn to play the piano and posts a request on a consumer-to-consumer social network. Soon after, he receives offers from two musicians, Carl and Mary. Carl provides two options: a 5 people class for 15 dollars per person or a one-to-one class for 35 dollars. Mary offers two similar options: a 3 people class for 20 dollars per person or a one-to-one class for 40 dollars. Furthermore, they both offer a trial lesson. Regarding Carl's options, Bob thinks that 5 people are too many for a class, thus he prefers the one-to-one option. On the contrary, he judges the price difference between Mary's options somewhat excessive, hence he prefers the 3 people class. If someone asks Bob *"Do you prefer Mary's 3 people class or Carl's one-to-one class?"*, Bob will probably answer *"I do not know, I first have to attend the trial lessons"*. Notice that this is different from saying that the two options are equivalent, because in that case Bob would simply flip a coin and choose one of them. On the contrary, it is more natural to think that these options are initially *incomparable* and Bob will use the trial lessons to disambiguate them.

More generally, in absence of complete information it might be difficult for an individual to figure out a coherent total order over the bids and choose in a single step one of them. On the contrary, making a decision can be viewed as a multiple step process where offers are initially filtered according to a partial preference relation. Then, depending on the resulting offers, an individual can acquire further information and possibly rank them.

In the second scenario, Alice's father has finally agreed to buy her a smartphone, and now she is browsing Ebay for possible offers. Unfortunately, the list is huge and patience is not Alice's forte. So, she would like to be assisted by a software agent to filter out undesired options. The software agent accepts constraints such as maximum cost and size, color restrictions, etc., and also a preference relation as a total order over bids. Then, according to the specified preference relation, the agent returns the best offer. Clearly, Alice's desires are influenced by several attributes such as operating system, color, weight, brand, and so on. For instance, she has a preference over operating systems in the following decreasing order: OS1, OS2 and OS3; over colors: blue, red, black, white; and over brands: Brand1, Brand2, Brand3. Furthermore, out of benevolence for her father, given a specific model the cheaper the better. However, such preferences over single attributes do not constitute a total order. Moreover, Alice cannot establish a priority over attributes, for instance she prefers a Brand3 phone with operating system OS1 to a Brand2 one with operating system OS2, but she prefers a Brand1 OS2 phone to a Brand2 OS1 one. Alice soon realizes that providing a total order to the software agent is frustrating and requires about the same effort as comparing all the offers by herself. This scenario reveals the following issue: in designing a software agent that behaves on behalf of real users, we have to take into account how users can *instruct* the agent about their own preferences. In electronic markets where the number of offers can be huge, it could be unfeasible to transfer an exact representation of users' desires into a software

agent. In this case, the agent should make do with an approximate representation of users' desires as a partial order and return a restricted list of choices from which the user can select the preferred one. As a further advantage, the user retains the ability of applying unforeseen, situation-specific knowledge and preferences that had not been formalized in advance.

3 Partial Preferences

Let $\Delta(\mathscr{A})$ be the class of all probability distributions over a countable set of alternatives $\mathscr{A}$. Given two probability distributions $f,g \in \Delta(\mathscr{A})$ and $\alpha \in [0,1]$, we denote by $\langle \alpha, f, g\rangle$ the convex combination of f and g such that $\langle \alpha, f, g\rangle(a) = \alpha f(a) + (1-\alpha)g(a)$, for all $a \in \mathscr{A}$. Moreover, for an alternative $a \in \mathscr{A}$, $[a]$ denotes the degenerate distribution that assigns probability 1 to a.

A preference relation $\preceq$ is a binary relation on $\Delta(\mathscr{A})$, subject to the following classical Decision Theory axioms: [1]

1. $f \preceq g$ or $g \preceq f$ *(totality)*;
2. if $f \preceq g$ and $g \preceq h$, then $f \preceq h$ *(transitivity)*;
3. if $f \prec g$ and $0 \leq \alpha < \beta \leq 1$, then $\langle \beta, f, g\rangle \prec \langle \alpha, f, g\rangle$;
4. if $f_1 \preceq g_1$, $f_2 \preceq g_2$, and $0 \leq \alpha \leq 1$, then $\langle \alpha, f_1, f_2\rangle \preceq \langle \alpha, g_1, g_2\rangle$;
5. if $f_1 \prec g_1$, $f_2 \preceq g_2$, and $0 \leq \alpha \leq 1$, then $\langle \alpha, f_1, f_2\rangle \prec \langle \alpha, g_1, g_2\rangle$;

where, as usual, $f \prec g$ means that $f \preceq g$ and $g \not\preceq f$.

Notice that the first two axioms force $\preceq$ to be a total (hence reflexive) transitive relation, i.e., a total preorder (also called non-strict weak order). As shown by the previous scenarios, we advocate that in several contexts some outcomes may be incomparable, meaning that $\preceq$ should be modeled as a (possibly partial) preorder. For this reason, we weaken the totality axiom in favor of one which requires reflexivity only:

1'. $f \preceq f$;

Having allowed for incomparable distributions, at first look it may seem that the deal is done. However, the obtained theory is so weak that it contemplates unrealistic preferences. The problem is that the previous axioms do not say anything about incomparable distributions which, once combined, can be then freely judged. On the contrary, it is natural to think that, to some extent, the incomparability between distributions persists also when they are combined.

Assume for example that f and g are incomparable and let $0 \leq \alpha < \beta \leq 1$. According to the axioms above, it is possible that $\langle \alpha, f, g\rangle \prec \langle \beta, f, g\rangle$. This looks inappropriate: given that I cannot compare f and g, why should I strictly prefer one combination of f and g over another? This leads to a further axiom:

6. if $0 \leq \alpha \leq 1$, and $\langle \alpha, f_1, f_2\rangle \prec \langle \alpha, g_1, g_2\rangle$, then there exist $j,k \in \{1,2\}$ such that $f_j \prec g_k$.

Intuitively, a distribution f of the type $\langle \alpha, f_1, f_2\rangle$ can be seen as a random choice (a.k.a. a *lottery*) which picks f_1 with probability α and f_2 with probability $1-\alpha$. Comparing f with another distribution g of the type $\langle \alpha, g_1, g_2\rangle$ encompasses comparing four possible draws: (f_1, g_1), (f_1, g_2), (f_2, g_1), and (f_2, g_2).[2] If there is no draw in which the second component is strictly better than the second, Axiom 6 requires that g is not strictly preferred to f. Notice that one could easily come up with more stringent

[1] Here, we borrow the formulation presented in [3].

[2] With probabilities α^2, $\alpha(1-\alpha)$, $\alpha(1-\alpha)$ and $(1-\alpha)^2$, respectively.

conditions on the persistence of incomparability, for instance by requiring that a majority of draws favors the second component over the first one. We instead propose a rather weak requirement, in the form of Axiom 6, which supports a wide range of preference relations, while still ensuring a number of interesting properties, which are the subject of Section 4.

We call *Partial Decision Theory* (PDT) the new set of axioms 1'-6. Notice that Axiom 6 can be easily derived in classical Decision Theory, consequently PDT is a weakening of classical Decision Theory, in the sense that all preference relations satisfying classical Decision Theory also satisfy PDT. The converse does not hold, as witnessed by the "empty" preference relation, i.e., the relation that considers incomparable all distinct distributions.

As seen above, Axiom 6 has been motivated by analyzing which preferences between $f = \langle \alpha, f_1, f_2 \rangle$ and $g = \langle \alpha, g_1, g_2 \rangle$ are admissible on the basis of the preferences on the possible draws (f_1, g_1), (f_1, g_2), (f_2, g_1), and (f_2, g_2). In Table 1 we perform such an analysis extensively. In particular, for each entry, the left-hand side $\bowtie_1 \bowtie_2 \bowtie_3 \bowtie_4$, with $\bowtie \in \{\sim, \prec, \succ, \lesssim\!\!\!\gtrsim\}$, is a consistent combination $f_1 \bowtie_1 g_1$, $f_1 \bowtie_2 g_2$, $f_2 \bowtie_3 g_1$, and $f_2 \bowtie_4 g_2$, whereas the right-hand side shows which preference relations between f and g are consistent with PDT. For example, the first entry is the case $f_1 \sim g_1$, $f_1 \sim g_2$, $f_2 \sim g_1$, and $f_2 \sim g_2$, then according with Axiom 4, $f \sim g$ is the only possibility. Conversely, in some other cases (e.g. $\prec \lesssim\!\!\!\gtrsim \succ \lesssim\!\!\!\gtrsim$) no axiom can be applied, consequently f can be in any relationship with g.

Table 1 provides a close look on how PDT behaves and hence it can be a good starting point to debate whether and how it can be extended or modified. For example, notice that in some cases f and g are comparable even if some of the underlying draws are not (e.g. due to Axiom 5, $\prec \lesssim\!\!\!\gtrsim \lesssim\!\!\!\gtrsim \sim$ results in $f \prec g$). Somewhat conversely, f and g can be incomparable even if all the underlying draws are comparable (e.g. $\prec \succ \prec \succ$ admits $f \lesssim\!\!\!\gtrsim g$). Finally, f and g are never forced to be incomparable, even if $f_1 \lesssim\!\!\!\gtrsim g_1$, $f_1 \lesssim\!\!\!\gtrsim g_2$, $f_2 \lesssim\!\!\!\gtrsim g_2$, and $f_2 \lesssim\!\!\!\gtrsim g_2$.

4 Properties of Partial Preferences

In the following, given two distributions f and g, we write $f \sim g$ for $f \preceq g$ and $g \preceq f$ (i.e., equivalence), and we write $f \lesssim\!\!\!\gtrsim g$ for $f \npreceq g$ and $g \npreceq f$ (i.e., incomparability). First, we show that two relevant properties of classical Decision Theory continue to hold in PDT. Given two distributions $f, g \in \Delta(\mathscr{A})$, we write $f \to g$ in case there exist $\varepsilon > 0$ and two alternatives a_1 and a_2 such that *(i)* $[a_1] \prec [a_2]$, *(ii)* $g(a_1) = f(a_1) - \varepsilon$, $g(a_2) = f(a_2) + \varepsilon$, and *(iii)* for all $a \neq a_1, a_2$, $g(a) = f(a)$. When $f \to g$, g can be obtained from f by shifting a positive amount of probability from an alternative a_1 to a strictly preferred alternative a_2. Then, denote by $f \Rightarrow g$ the transitive closure of $\to$.

Theorem 1 *Let $f, g \in \Delta(\mathscr{A})$. If $f \Rightarrow g$ then $f \prec g$.*

Proof. It suffices to show that $f \to g$ implies $f \prec g$. Assume that for some a_1 and a_2, where $[a_1] \prec [a_2]$, there exists $\varepsilon > 0$ such that $f(a_1) = g(a_1) + \varepsilon$ and $f(a_2) = g(a_2) - \varepsilon$. Let $\gamma = g(a_1) + g(a_2) = f(a_1) + f(a_2)$, $\alpha = \frac{f(a_2)}{\gamma}$ and $\beta = \frac{g(a_2)}{\gamma}$. Then, g can be written as $\langle \gamma, g', h \rangle$ and f as $\langle \gamma, f', h \rangle$, where h is a probability distribution such that $h(a_1) = h(a_2) = 0$, $g' = \langle \beta, [a_2], [a_1] \rangle$ and $f' = \langle \alpha, [a_2], [a_1] \rangle$. Since $[a_1] \prec [a_2]$ and $\alpha < \beta$, Axiom 3 implies that $f' \prec g'$. Then, due to Axiom 5, $f \prec g$. ∎

Another property that can be proved in PDT is that if each alternative coming out from a distribution f is dominated by all the alternatives from g, then $f \preceq g$. Preliminarily, given a distribution $f \in \Delta(\mathscr{A})$, the *support* of f, $supp(f) = \{a \in \mathscr{A} \mid f(a) > 0\}$, is the set of alternatives to which f assigns positive probability.

Theorem 2 *Let $f, g \in \Delta(\mathscr{A})$ be such that, for all $a \in supp(f)$ and $a' \in supp(g)$, $[a] \preceq [a']$. Then, $f \preceq g$.*

Table 1: Extensive analysis of PDT

Proof. We first show that for all $a \in supp(f)$, $[a] \preceq g$. The proof is by induction on the cardinality n of $supp(g)$ where the case $n = 1$ is trivial. Assume $n > 1$, then g can be written as $\langle \alpha, [a'], g' \rangle$, where a' is a generic alternative from $supp(g)$ and $g'(a') = 0$. Clearly, the distribution $[a]$ can also be written as $\langle \alpha, [a], [a] \rangle$. By assumption $a \preceq a'$ and, since the cardinality of g' is $n-1$, by induction $[a] \preceq g$. Then, by applying Axiom 4 we have $[a] \preceq g$.

Now, the proof is by induction on the cardinality m of $supp(f)$ where the base case $m = 1$ has been proved above. Assume $m > 1$ and let $a \in supp(f)$, then f can be written as $\langle \alpha, [a], f_{-a} \rangle$, where $f_{-a}(a) = 0$ and $f_{-a}(a') = \frac{f(a')}{1-f(a)}$ for all $a' \in supp(f) \setminus \{a\}$. We already proved that $[a] \preceq g$ and by induction hypothesis it holds also that $f_{-a} \preceq g$. Then, by writing g as $\langle \alpha, g, g \rangle$ and applying Axiom 4 we have that $f \preceq g$. ∎

From Theorem 2 it is immediate to show that distributions over equivalent alternatives are equivalent themselves. What if some of the alternatives are incomparable? We show that their distributions are either equivalent or incomparable, as due to Axiom 6 no strict preference can be derived.

Lemma 1 *Let $f, g \in \Delta(\mathscr{A})$ be such that, for all $a \in supp(f)$ and $a' \in supp(g)$, $a \not\lesssim a'$. Then, either $f \not\lesssim g$ or $f \sim g$.*

Proof. Let $n_f = |supp(f)|$ and $n_g = |supp(g)|$, we proceed by induction on $n_f + n_g$. If $n_f + n_g = 2$, the thesis is obviously true. Otherwise, assume w.l.o.g. that $n_f > 1$ and let $a \in supp(f)$. We can write f as $\langle f(a), [a], f_{-a} \rangle$, where $f_{-a}(a) = 0$ and $f_{-a}(a') = \frac{f(a')}{1-f(a)}$ for all $a' \in supp(f) \setminus \{a\}$. Since the support of f_{-a} is smaller than the one of f, we can apply the inductive hypothesis to the pair f_{-a}, g, obtaining that either $f_{-a} \not\lesssim g$ or $f_{-a} \sim g$. We can also apply the inductive hypothesis to the pair $[a], g$, obtaining that $[a] \not\lesssim g$ or $[a] \sim g$. Assume by contradiction that $f \bowtie g$ for some $\bowtie \in \{\prec, \succ\}$. By Axiom 6, it holds $[a] \bowtie g$ or $f_{-a} \bowtie g$, which is a contradiction. ∎

Finally, the following generalization of Lemma 1 shows that alternatives that are either incomparable or equivalent lead to distributions that are themselves either incomparable or equivalent.

Theorem 3 *Let $f, g \in \Delta(\mathscr{A})$ be such that, for all $a \in supp(f)$ and $a' \in supp(g)$, either $a \sim a'$ or $a \not\lesssim a'$. Then, $f \not\lesssim g$ or $f \sim g$.*

Proof. The proof is very similar to the one of Lemma 1, where only the base case is affected by the weakened assumption. ∎

5 Conclusions

In this work we challenged the customary decision-theoretic assumption of totality of the preference relations, on the basis of two real-world scenarios. We proposed a weakening of the classical theory and proved that it retains several desirable properties, while allowing for incomparable alternatives.

Partial preferences have been already employed in procurement auctions. In [2] second-price auctions have been generalized by considering a bid domain representing information disclosures and two ad hoc partial preference relations have been defined for modeling the sensitivity of the disclosed data. Then, in [1] the previous framework has been extended for modeling also service cost, QoS and functional differences, etc. Somewhat surprisingly, it has been shown that extending second price auctions to partial preferences does not yield truthful mechanisms, since overbidding may be profitable in some contexts. This means that, in general, partial preferences may significantly change the theoretical properties of a mechanism and the axiomatization presented in this work enables to uniformly employ them in the field of Mechanism Design and estimate their impact.

References

[1] P.A. Bonatti, M. Faella, C. Galdi & L. Sauro (2013): *Auctions for Partial Heterogeneous Preferences*. In: *Proc. of MFCS: Mathematical Foundations of Compuer Science*, *Lecture Notes in Computer Science* 8087, Springer, pp. 183–194, doi:10.1007/978-3-642-40313-2_18.

[2] Piero A. Bonatti, Marco Faella, Clemente Galdi & Luigi Sauro (2011): *Towards a Mechanism for Incentivating Privacy*. In Vijay Atluri & Claudia Díaz, editors: *ESORICS*, *Lecture Notes in Computer Science* 6879, Springer, doi:10.1007/978-3-642-23822-2_26.

[3] R.B. Myerson (1997): *Game Theory: Analysis of Conflict*. Harvard University Press.

[4] L. J. Savage (1954): *The Foundations of Statistics*. John Wiley and Sons, New York.

[5] Herbert Simon (1955): *A Behavioral Model of Rational Choice*. *The Quarterly Journal of Economics* 69(1), pp. 99–118, doi:10.2307/1884852.