Proceedings of the

3rd International Workshop on Strategic Reasoning

Oxford, United Kingdom, September 21-22, 2015

Edited by:
Julian Gutierrez, Fabio Mogavero, Aniello Murano, Michael Wooldridge

Preface

This volume contains the papers presented at SR 2015: the 3rd International Workshop on Strategic Reasoning held on September 20-21, 2015 in Oxford.

Strategic reasoning is one of the most active research areas in the multi-agent system domain. The literature in this field is extensive and provides a plethora of logics for modelling 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, and automatic players capable of beating expert human adversary, just to cite a few. All these examples share the challenge of developing novel theories, tools, and techniques for agent-based reasoning that take into account the likely behaviour of adversaries. The *SR international workshop* aims to bring together researchers working on different aspects of strategic reasoning in computer science and artificial intelligence, both from a theoretical and a practical point of view.

This year SR has four invited talks:

1. Coalgebraic Analysis of Equilibria in Infinite Games.
   Samson Abramsky (University of Oxford).
2. In Between High and Low Rationality.
   Johan van Benthem (University of Amsterdam and Stanford University).
3. Language-based Games.
   Joseph Halpern (Cornell University).
4. A Revisionist History of Algorithmic Game Theory.
   Moshe Vardi (Rice University).

We also have four invited tool presentations and eleven contributed papers. Each submission to SR 2015 was evaluated by three reviewers for quality and relevance to the topics of the workshop. We would like to acknowledge the people and institutions who contributed to the success of this edition of SR. We thank 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 the members of the Organizing Committee for their hard work in making sure that the workshop could be successfully organised as well as the EasyChair organization for supporting all tasks related to the selection of contributions and production of the proceedings. We gratefully acknowledge the financial support of the ERC Advanced Grant 291528 ("RACE") at the University of Oxford and the Artificial Intelligence journal. Finally, we acknowledge the support of the Department of Computer Science of the University of Oxford.

September, 2015
Oxford, UK.

Julian Gutierrez
Fabio Mogavero
Aniello Murano
Michael Wooldridge

Table of Contents

Coalgebraic Analysis of Equilibria in Infinite Games 1
Samson Abramsky and Viktor Winschel

Language-Based Games 2
Joseph Halpern, Adam Bjorndahl and Rafael Pass

In Between High and Low Rationality 3
Johan van Benthem

A Revisionist History of Algorithmic Game Theory 4
Moshe Vardi

Substructural modal logic for optimal resource allocation 5
Gabrielle Anderson and David Pym

Games with Communication: from Belief to Preference Change 6
Guillaume Aucher, Bastien Maubert, Sophie Pinchinat and Francois Schwarzentruber

Epistemic Game Theoretical Reasoning in History Based Models 7
Can Baskent and Guy McCusker

Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications 8
Petr Čermák, Alessio Lomuscio and Aniello Murano

P-Automata for Markov Decision Processes 9
Souymodip Chakraborty and Joost-Pieter Katoen

Energy Structure and Improved Complexity Upper Bound for Optimal Positional Strategies in Mean Payoff Games 10
Carlo Comin and Romeo Rizzi

Simulating cardinal payoffs in Boolean games 11
Egor Ianovski and Luke Ong

Dealing with imperfect information in Strategy Logic 12
Sophia Knight and Bastien Maubert

An Arrow-based Dynamic Logic of Norms 13
Louwe B. Kuijer

The risk of divergence 14
Pierre Lescanne

Preference Refinement in Normative Multi-agent System 15
Xin Sun

Workshop Chairs

Julian Gutierrez	University of Oxford
Fabio Mogavero	University of Oxford
Aniello Murano	University of Naples
Michael Wooldridge	University of Oxford

Program Committee

Natasha Alechina	University of Nottingham
Julian Bradfield	University of Edinburgh
Krishnendu Chatterjee	Institute of Science and Technology
Vojtech Forejt	Oxford University
Valentin Goranko	Stockholm University
Erich Grädel	RWTH Aachen University
Julian Gutierrez	University of Oxford
Paul Harrenstein	University of Oxford
Andreas Herzig	IRIT-CNRS
Orna Kupferman	Hebrew University
Salvatore La Torre	Università degli studi di Salerno
Jérôme Lang	LAMSADE
Alessio Lomuscio	Imperial College London
Enrico Marchioni	University of Oxford
Nicolas Markey	LSV, CNRS & ENS Cachan
John-Jules Meyer	Utrecht University
Fabio Mogavero	University of Oxford
Aniello Murano	University of Naples
Andrzej Murawski	University of Warwick
Luke Ong	University of Oxford
Sophie Pinchinat	IRISA Rennes
Nikos Tzevelekos	Queen Mary University of London
Wiebe Van Der Hoek	University of Liverpool
Igor Walukiewicz	CNRS, LaBRI
Michael Wooldridge	University of Oxford
Thomas Ågotnes	University of Bergen

Local Arrangements Committee

Vojtech Forejt	University of Oxford
Julian Gutierrez	University of Oxford
Paul Harrenstein	University of Oxford
Enrico Marchioni	University of Oxford
Fabio Mogavero	University of Oxford
Giuseppe Perelli	University of Oxford
Michael Wooldridge	University of Oxford

Additional Reviewers

Doyen, Laurent
Gardy, Patrick
Ianovski, Egor
Malacaria, Pasquale
Schwarzentruber, Francois

Invited Tool Presentations

- EAGLE. Alexis Toumi, University of Oxford.
- PRISM. Clemens Wiltsche, University of Oxford.
- PRALINE. Romain Brenguier, Université Libre de Bruxelles.
- MCMAS. Alessio Lomuscio, Imperial College London.

Invited Speakers

- Samson Abramsky. University of Oxford.
- Johan van Benthem. University of Amsterdam and Stanford University.
- Joseph Halpern. Cornell University.
- Moshe Y. Vardi. Rice University.

Workshop Webpage

https://sites.google.com/site/sr2015homepage/

1

Invited Talk

Coalgebraic Analysis of Equilibria in Infinite Games

Samson Abramsky
University of Oxford
Oxford, UK

Viktor Winschel
ETH Zurich
Zurich, Switzerland

We present a novel coalgebraic formulation of infinite economic non-cooperative games. We define the infinite trees of the extensive representation of the games as well as the strategy profiles by possibly infinite systems of corecursive equations. Subgame perfect equilibria are defined and proved using a novel proof principle of predicate coinduction which is related to Kozen's metric coinduction. We characterize all subgame perfect equilibria for the dollar auction game. The economically interesting feature is that in order to prove these results we do not need to rely on continuity assumptions on the payoffs which amount to discounting the future. This suggests that coalgebras support a more adequate treatment of infinite-horizon models in game theory and economics.

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

2

Invited Talk

Language-based games

Adam Bjorndahl
Carnegie Mellon University*
Pittsburgh, PA
adam.bjornedahl@gmail.com

Joseph Y. Halpern
Cornell University
Ithaca, NY
halpern@cs.cornell.edu

Rafael Pass
Cornell University
New York City, NY
rafael@cs.cornell.edu

1 Introduction

In a classical, normal-form game, an *outcome* is a tuple of strategies, one for each player, and players' preferences are formalized by utility functions defined on the set of all such outcomes. This framework thereby hard-codes a single conception of how players represent the world insofar as their preferences are concerned.

The motivating idea of the present work is to relax this rigidity in a systematic way by using *language* as the foundation of preference. Roughly speaking, we assume that what the players care about is captured by some *underlying language*, with utility defined on *descriptions* in that language. Classical game theory can be viewed as the special case where the underlying language can talk only about outcomes. In general, however, the language can be as rich or poor as desired.

In the colloquial sense of the word, the role of "language" in decision making and preference formation can hardly be overstated. It is well known, for example, that presenting alternative medical treatments in terms of survival rates versus mortality rates can produce a marked difference in how those treatments are evaluated, even by experienced physicians [7]. More generally, one of the core insights of *prospect theory* [4]—that subjective value depends not (only) on facts about the world but on how those facts are *presented* (as gains or losses, dominated or undominated options, etc.)—can be viewed as a kind of language-sensitivity. We celebrate 10th and 100th anniversaries specially, and make a big deal when the Dow Jones Industrial Average crosses a multiple of 1,000, all because we happen to work in a base 10 number system (i.e., our language puts special emphasis on multiples of 10 that would be absent, for example, in a hexadecimal system). Furthermore, we often assess likelihoods using words like "probable", "unlikely", or "negligible", rather than numeric representations, and when numbers are used, we tend to round them [6]. Much of the motivation and conceptual appeal of our approach stems from observations like these: defining preferences in terms of language provides a direct avenue for formalizing such intuitions about how people think.

Of special interest is the general phenomenon of *coarseness* or *categoricity*. Theories of rational decision making are often couched in the formalism of continuous mathematics, but the world is not always a continuous place, at least as far as preferences are concerned. Consumers tend to ignore, for example, the difference in price between \$3.98 and \$3.99, but take seriously (or even exaggerate) the difference between \$3.99 and \$4.00. Similarly, although degrees of belief are often formalized using probability measures, a coarser representation can be more appropriate for reasoning about human choice and inference (see [8], [9], [6]). We show, for instance, that the Allais paradox [1] can be resolved simply and intuitively when belief is represented discretely, rather than on a continuum.

Coarseness in the underlying language—cases where there are fewer descriptions than there are actual differences to describe—provides a natural and powerful way of capturing such phenomena, offering

*Work on this paper was mainly done while the author was at Cornell University

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

insight into a variety of puzzles and paradoxes of human decision making. Moreover, it allows for a unified analysis of coarseness as it pertains both to preferences and to beliefs, traditionally distinct domains of decision making. This is accomplished using languages expressive enough to talk about beliefs, a technique that is of interest in its own right.

Classically, beliefs are relevant to decision making insofar as they determine *expected utility*. But beliefs can also themselves be considered as objects of preference: one might wish to model players who feel guilt, wish to surprise their opponents, or are motivated by a desire to live up to what is expected of them. *Psychological game theory*, beginning with the work of Geanakoplos, Pearce, and Stachetti [3] and expanded by Battigalli and Duwfenberg [2], is an enrichment of the classical setting meant to capture such preferences and motivations. In a similar vein, the notion of *reference-dependent preferences* developed by Köszegi and Rabin [5], building on prospect theory, formalizes phenomena such as loss-aversion by augmenting players' preferences with an additional sense of gain or loss derived by comparing the actual outcome to what was expected.

With the appropriate choice of language, our approach subsumes these: an underlying language that includes beliefs allows us to capture psychological games, while a language that distinguishes expected from actual outcomes allows us to represent reference-dependent preferences. Moreover, in each of these frameworks, modeling coarse beliefs provides insight and opportunities lacking in the continuous setting. Much of this paper is an elaboration and justification of this point.

The central concept we develop in this paper is that of a *language-based game*, where utility is defined not on outcomes but on *situations*. As noted, a situation can be conceptualized as a collection of statements about the game; intuitively, each statement is a description of something that might be relevant to a player's preferences, such as whether or not Alice believes that Bob will play a certain strategy. Of course, this notion crucially depends on just what counts as an admissible description. The set of all admissible descriptions—what we refer to as the *underlying language* of the game—is a key component of our model. Since utility is defined on situations, and situations are sets of descriptions taken from the underlying language, a player's preferences can depend, in principle, on anything expressible in this language, but nothing more. Succintly: players can prefer one state of the world to another if and only if they can *describe* the difference between the two in the underlying language. From a technical standpoint, this paper makes three major contributions. First, we define a generalization of classical game theory and demonstrate its versatility in modeling a wide variety of strategic scenarios, focusing in particular on psychological and reference-dependent effects. Second, we provide a formal representation of coarse beliefs in a game-theoretic context. This exposes an important insight: a discrete representation of belief, often conceptually and technically easier to work with than its continuous counterpart, is sufficient to capture psychological phenomena that have heretofore been modeled only in a continuous framework. Moreover, as we show by example, utilities defined over coarse beliefs provide a natural way of capturing some otherwise puzzling behavior. Third, we provide novel equilibrium analyses for a broad class of language-based games that do not depend on continuity assumptions as do those of, for example, Geanakoplos et al. [3]. In particular, our main theorem demonstrates that if the underlying language satisfies certain natural "compactness" assumptions, then every game over this language admits rationalizable strategies. By contrast, even under such compactness assumptions, not every game admits a Nash equilibrium.

This paper originally appeared in the *Theoretical Aspects of Rationality and Knowledge: Proc. Fourteenth Conference (TARK 2013)*; the full paper, which expands and on all the points above and gives numerous examples, can be found at https://www.cs.cornell.edu/home/halpern/papers/lbg.pdf.

2 Acknowledgements

Bjorndahl was supported in part by NSF grants IIS-0812045, CCF-1214844, DMS-0852811, and DMS-1161175, and ARO grant W911NF-09-1-0281. Halpern was supported in part by NSF grants IIS-0812045, IIS-0911036, and CCF-1214844, by AFOSR grant FA9550-08-1-0266, by the Multidisciplinary University Research Initiative (MURI) program administered by the AFOSR under grant FA9550-12-1-0040, and by ARO grant W911NF-09-1-0281. Pass was supported in part by an Alfred P. Sloan Fellowship, a Microsoft Research Faculty Fellowship, NSF Awards CNS-1217821 and CCF-1214844, NSF CAREER Award CCF-0746990, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2-0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

References

[1] M. Allais (1953): *Le comportement de l'homme rationel devant le risque: critique de l'École Americaine*. *Econometrica* 21, pp. 503–546.

[2] P. Battigalli & M. Dufwenberg (2009): *Dynamic psychological games*. *Journal of Economic Theory* 144, pp. 1–35.

[3] J. Geanakoplos, D. Pearce & E. Stacchetti (1989): *Psychological games and sequential rationality*. *Games and Economic Behavior* 1(1), pp. 60–80.

[4] D. Kahneman & A. Tversky (1979): *Prospect theory, an analysis of decision under risk*. *Econometrica* 47(2), pp. 263–292.

[5] B. Kőszegi & M. Rabin (2006): *A model of reference-dependent preferences*. *The Quarterly Journal of Economics* CXXI, pp. 1133–1165.

[6] C. F. Manski & F. Molinari (2010): *Rounding probabilistic expectations in surveys*. *Journal of Business and Economic Statistics* 28(4), pp. 219–231.

[7] B. J. McNeil, S. J. Pauker, H. C. Sox Jr. & A. Tversky (1982): *On the elicitation of preferences for alternative therapies*. *New England Journal of Medicine* 306, pp. 1259–1262.

[8] G. A. Miller (1956): *The magical number seven, plus or minus two: Some limits on our capacity for processing information*. *Psychological Review* 63(2), pp. 81–97.

[9] S. Mullainathan (2002): *Thinking through categories*. Unpublished manuscript; available at www.haas.berkeley.edu/groups/finance/cat3.pdf.

3

Invited Talk

In Between High and Low Rationality

Johan van Benthem
University of Amsterdam and Stanford University

Strategic social behavior may be held in place by highly sophisticated reasoning, but its stability may also result from a simple iterated imitation and reward structure. The same two perspectives can be taken with respect to other aspects of social life, including the origins of morality. We will explore this tension by taking a look at the interface of classical and evolutionary game theory from a logician's perspective.

References

[1] Johan van Benthem (2014): *Logic in Games*. MIT Press.

[2] Johan van Benthem, Sujata Ghosh & Rineke Verbrugge, editors (2015): *Modeling Strategic Reasoning: Logics, Games and Communities*. Lecture Notes in Computer Science 8972, Springer.

[3] Brian Skyrms (1990): *The Dynamics of Rational Deliberation*. Harvard University Press.

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

4

Invited Talk

A Revisionist History of Algorithmic Game Theory

Moshe Y. Vardi
Rice University
Department of Computer Science
Houston, TX 77251-1892, USA
vardi@cs.rice.edu

A key feature of current Theoretical Computer Science (TCS) is the division between Volume-A TCS, focused on algorithms and complexity, and Volume-B TCS, focused on logic and formal models. Algorithmic Game Theory (AGT), introduced independently by several authors around 2000, is considered today as one of the main topics of Volume-B TCS, and plays a major role in the two premier North American TCS Conferences FOCS (Symposium on Foundations of Computer Science) and STOC (Symposium on Theory of Computing)which focus almost exclusively on Volume-A TCS.

In this revisionist history of AGT, I will show that, contrary to popular perception, AGT was studied by logicians 40 years before it was "discovered" by computer scientists, and has been a part of Volume-B TCS at least a decade before it became a part of Volume-A TCS. I will sketch the differences between Volume-A AGT and Volume-B AGT, and call for an integrated approach to AGT.

References

[1] Guy Avni, Orna Kupferman & Tami Tamir (2014): *Network-Formation Games with Regular Objectives*. In Anca Muscholl, editor: *FOSSACS*, *Lecture Notes in Computer Science* 8412, Springer, pp. 119–133.

[2] Julius R. Büchi & Lawrence H. Landweber (1969): *Solving sequential conditions by finite-state strategies*. *Trans. AMS* 138, pp. 295–311.

[3] Krishnendu Chatterjee, Laurent Doyen, Sumit Nain & Moshe Y. Vardi (2014): *The Complexity of Partial-Observation Stochastic Parity Games with Finite-Memory Strategies*. In Anca Muscholl, editor: *FOSSACS*, *Lecture Notes in Computer Science* 8412, Springer, pp. 242–257.

[4] Alonzo Church (1962): *Logic, arithmetics, and automata*. In: *Proc. Int. Congress of Mathematicians*, Institut Mittag-Leffler, pp. 23–35.

[5] Yuri Gurevich & Leo Harrington (1982): *Trees, Automata, and Games*. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard & Lawrence H. Landweber, editors: *STOC*, ACM Press, pp. 60–65.

[6] Marcin Jurdzinski (1998): *Deciding the Winner in Parity Games is in UP $\cap$ co-UP*. *Inf. Process. Lett.* 68(3), pp. 119–124.

[7] Amir Pnueli & Roni Rosner (1989): *On the Synthesis of a Reactive Module*. In: *POPL*, ACM Press, pp. 179–190.

[8] Michael O. Rabin (1972): *Automata on infinite objects and Churchs problem*. *Amer. Mathematical Society*.

[9] Tim Roughgarden (2010): *Algorithmic game theory*. *Commun. ACM* 53(7), pp. 78–86.

[10] Moshe Y. Vardi (1985): *Automatic Verification of Probabilistic Concurrent Finite-State Programs*. In: *FOCS*, IEEE Computer Society, pp. 327–338.

[11] Moshe Y. Vardi (1995): *Alternating Automata and Program Verification*. In Jan van Leeuwen, editor: *Computer Science Today: Recent Trends and Developments*, *Lecture Notes in Computer Science* 1000, Springer, pp. 471–485.

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

5

Regular Paper

Substructural modal logic for optimal resource allocation

Gabrielle Anderson
University College London, UK
gabrielle.anderson@ucl.ac.uk

David Pym
University College London, UK
d.pym@ucl.ac.uk

We introduce a substructural modal logic for reasoning about (optimal) resource allocation in models of distributed systems. The underlying logic is a variant of the modal logic of bunched implications, and based on the same resource semantics, which is itself closely related to concurrent separation logic. By considering notions of cost, strategy, and utility, we are able to formulate characterizations of Pareto optimality, best responses, and Nash equilibrium within resource semantics.

1 Introduction

Mathematical modelling and simulation modelling are fundamental tools of engineering, science, and social sciences such as economics, and provide decision-support tools in management. The components of distributed systems (as described, e.g., in [9]) are typically modelled using various algebraic structures for the structural components — location, resource, and process — and probability distributions to represent stochastic interactions with the environment. A key aspect of modelling distributed systems is resource allocation. For example, when many processes execute concurrently, they compete for resources.

A common desire of system designers, managers, and users is to determine, if possible, the optimal allocation of resources required in order to solve a specific problem or deliver a specific service. The notion of optimality of resource allocation is a central topic in economics, where game theory plays a significant role. For all elementary notions from economics required for this short paper, including ideas from utility theory and game theory, a suitable source is [20].

Building on a mathematical systems and security modelling framework — described in, for example, [8, 6, 7], which builds on ideas in [2] and which has been widely deployed (e.g., [15, 1, 5, 3, 4]) — we sketch the development of a systems modelling framework that provides a theory of (optimal) resource allocation.

The key systems components of our resource semantics-based framework (which in turn builds on BI and its resource semantics [17, 18, 10, 8, 6]) are the following: *environment* (within which the system resides), *locations* (the architecture of the system), *resources* (that are manipulated — e.g., consumed, created, moved — by the system), and *processes* (that operate the system and deliver services). We integrate these components into an algebra of locations, resources, and processes that is defined by an operational semantics [8, 7] with a judgement of the form $L,R,E \xrightarrow{a} L',R',E'$ in which the process E evolves by action a, using resources R at locations L, to become the process E', able to evolve further using the resources R' at locations L'. A key component of this operational semantics is a (partial) *modification function*,

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

$\mu : Actions \times Resources \times Locations \rightharpoonup Resources \times Locations$, that specifies the effects of actions on resources and locations.

Properties of systems, including optimality properties, can be expressed logically. Specifically, we make use of a substructural modal logic [8, 7] that is naturally associated with the process algebra above in the Hennessy–Milner sense [12, 16, 8] — that is, it is defined by a (truth-functional) satisfaction relation of the form $L, R, E \models \phi$, for logical formulae ϕ — with transitions between worlds defined by the operational semantics.

For the purposes of this paper, however, we make two simplifications. First, we elide locations, which can be coded in terms of resources if necessary. Second, we neglect the structure of processes, using modification functions to describe the effects of actions on processes. Thus we are able to define a logic with a satisfaction relation between resource states R and formulae ϕ (i.e., $R \models \phi$) in which the meaning of formulae involving action modalities, such as $\langle a \rangle \phi$, is given by transitions as specified by $\mu(a, R)$.

To this logic we add, in Section 4, a simple account of utility, building on simple notions of strategy and cost that we introduce in Section 3. Then, in Section 4, we consider a range of examples about resource allocation and optimality, including Pareto optimality, best responses, and Nash equilibrium. We begin by introducing, in Section 2, resource semantics.

2 Resource semantics and modal logic for systems modelling

We present our resource model and semantics, along with its key technical properties. We define resources, actions, and an operational semantics for resources. We define our notion of bisimulation, and note that resource composition forms a congruence with respect to the bisimulation relation. We sketch a modal logic, and describe how it can be used for systems modelling.

First, we introduce our notion of resource, following [8, 7].

Definition 1 (Resource monoid)**.** *A resource monoid is a structure $\boldsymbol{R} = (\boldsymbol{R}, \circ, e)$ with carrier set $\boldsymbol{R}$, commutative partial binary operation $\circ : \boldsymbol{R} \times \boldsymbol{R} \rightharpoonup \boldsymbol{R}$, and unit $e \in \boldsymbol{R}$.* □

We assume a commutative monoid, **Act**, of actions, freely generated from a set of atomic actions. The actions correspond to the events of the system.

Definition 2 (Actions)**.** *Let **Act** be the free commutative monoid formed by combinations of atomic actions, with operation $\cdot$ and unit 1. Let ab denote $a \cdot b$.* □

We set up a function that describes how actions transform resources.

Definition 3 (Modification function)**.** *A modification function is a partial function $\mu : \boldsymbol{Act} \times \mathbf{R} \rightharpoonup \mathbf{R}$ such that, for all resources $R, S \in \mathbf{R}$ and actions $a, b, c \in \boldsymbol{Act}$:*

- *If $\mu(a,R)$, $\mu(b,S)$, and $R \circ S$ are all defined, then $\mu(a,R) \circ \mu(b,S)$ and $\mu(ab, R \circ S)$ are both defined, and $\mu(ab, R \circ S) = \mu(a,R) \circ \mu(b,S)$ holds;*
- *If $R \circ S$ and $\mu(c, R \circ S)$ are defined, then there exist $a, b \in \boldsymbol{Act}$ such that $c = ab$, and $\mu(a,R)$ and $\mu(b,S)$ are both defined;*
- *$\mu(1,R) = R$.* □

If $\mu(a,R)$ is defined, then we say that action a is defined on resource R. We can use the partiality of the resource monoid, along with the modification function, to model straightforwardly key examples in systems modelling [8, 7], such as the following:

Example 4 (Semaphores)**.** *Suppose a resource monoid* $(\{s,e\},\circ,e)$*, where* $s\circ s$ *is undefined. Let* a *be an action. We define a modification function* μ *such that* $\mu(a,s)=s$*. Note that* μ *is undefined for any values that are neither specified explicitly nor required by properties of Definition 3. We then have that, for all resources* $R\in\boldsymbol{R}$*,* $\mu(aa,R)$ *is not defined. The resource* s *acts like a semaphore, in that only one access action* a *can be performed at any given time.* □

From a resource monoid, action monoid, and modification function, we derive a transition relation. If the modification function is defined for an action a on a resource R, and $\mu(a,R)=S$, then we say that there exists a transition $R\xrightarrow{a}S$, and that S is a successor of R. A notion of *bisimulation* between resources is defined in the standard way.

Definition 5 (Bisimulation)**.** *A bisimulation is a relation* $\mathscr{R}$ *such that, for all* $R\mathscr{R}S$*, then, for all actions* $a\in\boldsymbol{Act}$*,*

- *if* $R\xrightarrow{a}R'$*, then there exists* S' *such that* $S\xrightarrow{a}S'$ *and* $R'\mathscr{R}S'$*, and*
- *if* $S\xrightarrow{a}S'$*, then there exists* R' *such that* $R\xrightarrow{a}R'$ *and* $R'\mathscr{R}S'$*.* □

Let $\sim\subseteq\mathbf{R}\times\mathbf{R}$ be the union of all bisimulations. The union of any two bisimulations is also a bisimulation. Hence $\sim$ is well defined, and a bisimulation. In this simple setting, bisimulation equivalence is the same as trace equivalence, but that is not generally true in the more general location-resource-process framework, of which this is an example.

We can now obtain a key property: that bisimulation is a congruence; that is, an equivalence relation that is respected by the composition operator.

Lemma 6 (Bisimulation congruence)**.** *The relation* $\sim$ *on resources is a congruence for the operation* $\circ$*: if* $R_1\sim S_1$*,* $R_2\sim S_2$*, and* $R_1\circ R_2$ *and* $S_1\circ S_2$ *are defined, then* $R_1\circ R_2\sim S_1\circ S_2$*.*

Proof. A straightforward argument, similar to many others. □

We can use a substructural modal logic of resources to reason about our models (of distributed systems). The logic freely combines classical propositional logic with action modalities, in the style of Hennessy–Milner logic [12, 8] or dynamic logic [11], and with BI's multiplicatives [17]. Worlds are given by the resources R of a resource monoid. The classical connectives are defined with respect to a fixed world in the usual way: $R\models\bot$ never, $R\models\phi_1\vee\phi_2$ iff $R\models\phi_1$ or $R\models\phi_2$, and $R\models\neg\phi$ iff $R\not\models\phi$, with satisfying truth $\top=\neg\bot$ and conjunction satisfying $\phi_1\wedge\phi_2$ = $\neg(\neg\phi_1\vee\neg\phi_2)$, so that, in its resulting semantics, a resource R is shared by the conjuncts.

Transitions between worlds, used to define the action modalities, are given by modifications:

$$R\models\langle a\rangle\phi \quad \text{iff} \quad \text{there exists } R\xrightarrow{a}R' \text{ such that } R'\models\phi$$

giving the possible truth of ϕ after the action a (with necessity satisfying $[a]\phi=\neg\langle a\rangle\neg\phi$).

The substructural connectives — key to the analysis of resource usage in BI [17, 18, 10] and Separation Logic [13, 19], including the Frame Rule, where the specific resource semantics of a program's stack/heap is analysed — use the monoidal structure of resources to separate properties of different parts of a given model:

$$R\models\phi_1*\phi_2 \quad \text{iff} \quad \text{there exist } R_1 \text{ and } R_2\text{, where } R\sim R_1\circ R_2\text{, such that } R_1\models\phi_1 \text{ and } R_2\models\phi_2$$

with the corresponding implication, $-\!\!*$, given as the right adjoint to $*$.

Recall Example 4 (semaphores). We can now formally state the property that the action aa cannot be performed on each of the resources in the monoid. The formula $\phi = \neg(\langle aa\rangle\top)$ denotes that there is no transition for the action aa. As $\mu(aa,e)$ and $\mu(aa,s)$ are not defined, we have that $e \not\models \langle aa\rangle\top$ and $s \not\models \langle aa\rangle\top$. We then straightforwardly have that $e \models \phi$ and $s \models \phi$. Note that, as $e \not\sim s$, the equivalence classes generated by $\sim$ are singleton sets, consisting of each of the two resources. We can also state that, on each resource of the monoid, there is no binary decomposition such that each of the two parts can perform an a action. This property is represented by the formula $\psi = \neg(\langle a\rangle\top * \langle a\rangle\top)$. The only S and T such that $e = S \circ T$ are $S = T = e$. The only S and T such that $s = S \circ T$ are $S = s$ and $T = e$, or $S = e$ and $T = s$. For each of these possible binary decompositions, at least one of the two parts cannot perform an a action, and hence at least one of the two parts does not satisfy $\langle a\rangle\top$. Hence, $e \models \psi$ and $s \models \psi$.

3 Strategies and cost

We address non-determinism in the transition systems generated by our resource semantics, as introduced in the previous section. We introduce a notion of cost, that represents the preferences of an entity (or agent) in a system. We describe how to systematically determine the cost associated with a resource. We conclude with a brief example.

The transition systems generated by our resource semantics can be non-deterministic, in the sense that multiple actions can be defined on a given resource.

Example 7. *Take a resource monoid* $(\{0,\ldots,10\}\times\{0,\ldots,10\},\circ,(0,0))$*, where* $(m_1,m_2)\circ(n_1,n_2)=(m_1+n_1,m_2+n_2)$ *only if* m_1 *or* m_2 *is* 0 *and* n_1 *or* n_2 *is* 0 *(and is undefined otherwise). Suppose actions* p *and* c*. Let* $\mu(p,(m,n))=(m,n+1)$*, if* $n\le 9$*, and* $\mu(c,(m+1,n))=(m,n)$*. Then, for the resource* $(2,0)$*, the actions* p *and* c *are both defined and, in the generated transition system, there is non-determinism between the distinct, non-unit, actions,* p *and* c*.* □

When evolving such non-deterministic transition systems, it is necessary to have a method to decide between possible options. A strategy can be used to determine, for a given resource, which possible action is preferred.

Definition 8 (Strategies)**.** *A strategy is a total function* $\sigma : \mathbf{R} \to \boldsymbol{Act}$ *such that, for all resources* $R,S\in\boldsymbol{R}$*, if* $R\sim S$*, then* $\sigma(R)=\sigma(S)$ *and* $\mu(\sigma(R),R)$ *and* $\mu(\sigma(R),S)$ *are defined.* □

Example 9. *We can define a strategy to resolve the non-determinism we saw in Example 7. Let* σ *be a function such that, if* $1\le m$*, then* $\sigma((m,n))=c$*, and* $\sigma((m,n))=p$*, otherwise. This strategy chooses the* c *action, whenever possible, and chooses the* p *action otherwise.* □

The resource semantics approach to distributed systems modelling abstracts away from the entities that make decisions, and their mechanisms for doing so. A mechanism for resolving choices can be re-introduced into the models through strategies: it does not, however, represent the goals and interests of the entities making the choices. We can model the decision-making-entities' preferences through the use of a map from actions to the rationals. These numbers are interpreted as measures of an agent's level of happiness in the given states [20].

Definition 10 (Action payoff function)**.** *An action payoff function is a partial function $v : \mathbf{Act} \rightharpoonup \mathbb{Q}$ s.t. $v(1) = 0$ and, for all $a, b \in \mathbf{Act}$, if $v(a)$ and $v(b)$ are defined, then $v(ab) = v(a) + v(b)$.* □

Note that it is possible to have that $v(ab)$ is defined, but that $v(a)$ and $v(b)$ are not defined (c.f., Example 18). We use different action payoff functions to represent the preferences of different decision-making entities. Fix an action payoff function v, a strategy σ, and let δ be some rational number in the open interval $(0,1)$. We can then straightforwardly extended the notion of preference over actions to preferences over resources.

Definition 11 (Resource payoff function)**.** *A resource payoff function is a partial function $u_{v,\sigma,\delta} : \mathbf{R} \rightharpoonup \mathbb{Q}$ such that*

$$u_{v,\sigma,\delta}(R) = \begin{cases} v(a) + \delta \times u_{v,\sigma,\delta}(\mu(a,R)) & \text{if } \sigma(R) = a, \text{ and } v(a) \text{ and } u_{v,\sigma,\delta}(\mu(a,R)) \text{ are defined} \\ \text{undefined} & \text{otherwise.} \end{cases} \qquad \Box$$

The value that can be accumulated from actions performed at resources reachable in the future are worth less than value that can be accumulated immediately. The discount factor δ is used to discount future accumulated values. In the case that the set $\mathbf{R}$ is finite, we generate a finite set of simultaneous equations which can be solved using the methods described in [14]. Henceforth, we assume that all resource monoids have finite carrier sets.

Lemma 12. *For all action payoff functions v, strategies σ, and discount factors δ, if $\sigma(R) = 1$, then $u_{v,\sigma,\delta}(R) = 0$.*

Proof. By Definitions 3 and 10, we have that $\mu(1,R) = R$ and $v(1) = 0$. By Definition 11, we have that $u_{v,\sigma,\delta}(R) = 0 + \delta \times u_{v,\sigma,\delta}(R)$. As $(1-\delta) \neq 0$, we have that $u_{v,\sigma,\delta}(R) = 0$. □

Example 13. *We can now determine payoffs for various resources in Example 9 (which relies on Example 7). This is a simplification of a distributed systems example, presented fully in Example 16. Let v be an action payoff function such that $v(p) = -1$ and $v(c) = 3$, and $\delta = 0.8$. We then have that*

$$\begin{array}{rcl} u_{v,\sigma,\delta}((0,0)) & = & 0 \\ u_{v,\sigma,\delta}((1,0)) & = & 3 + 0.8 \times u_{v,\sigma,\delta}((0,0)) \\ & = & 3. \end{array} \qquad \begin{array}{rcl} u_{v,\sigma,\delta}((2,0)) & = & 3 + 0.8 \times u_{v,\sigma,\delta}((1,0)) \\ & = & 5.4 \end{array}$$

With a different strategy, and the same action payoff, discount factor, and underlying systems model, different payoffs can be achieved. □

4 A modal logic of resources and utilities

We define a modal predicate logic, MBIU, for expressing properties of resources and their utility. Building directly on [8, 6], we define, in Figure 1, a semantics for MBIU in terms of the transition relation of a resource monoid, action monoid, and modification function, and its corresponding bisimulation relation.

Let term variables be denoted x, y, etc., and action variables be denoted α, β, etc.. The action terms of MBIU, building on actions a, b, c, etc., are formed according to the grammar

R	$\vDash$	$\mathrm{p}(t_1,\ldots,t_n)$	iff	$t_1^{\mathscr{U}(R)},\ldots,t_n^{\mathscr{U}(R)}$ are defined and $(t_1^{\mathscr{U}(R)},\ldots,t_n^{\mathscr{U}(R)},R)\in\mathscr{V}(\mathrm{p})$
R	$\vDash$	$t_1=t_2$	iff	$t_1^{\mathscr{U}(R)}$ and $t_2^{\mathscr{U}(R)}$ are defined and $t_1^{\mathscr{U}(R)}=t_2^{\mathscr{U}(R)}$
R	$\vDash$	$s_1=s_2$	iff	$s_1^{\mathscr{U}(R)}=s_2^{\mathscr{U}(R)}$
R	$\vDash$	$\bot$	never	
R	$\vDash$	$\top$	always	
R	$\vDash$	$\phi_1\vee\phi_2$	iff	$R\vDash\phi_1$ or $R\vDash\phi_2$
R	$\vDash$	$\phi_1\wedge\phi_2$	iff	$R\vDash\phi_1$ and $R\vDash\phi_2$
R	$\vDash$	$\neg\phi$	iff	$R\nvDash\phi$
R	$\vDash$	$\phi_1\to\phi_2$	iff	$R\vDash\phi_1$ implies $R\vDash\phi_2$
R	$\vDash$	I	iff	$R\sim e$
R	$\vDash$	$\phi_1*\phi_2$	iff	there exist R_1, R_2, with $R\sim R_1\circ R_2$, such that $R_1\vDash\phi_1$ and $R_2\vDash\phi_2$
R	$\vDash$	$\phi_1 -\!\!* \phi_2$	iff	for all S, $S\vDash\phi_1$ implies $R\circ S\vDash\phi_2$
R	$\vDash$	$\langle s\rangle\phi$	iff	there exist a, R' such that $s^{\mathscr{U}(R)}=a$, $R\xrightarrow{a}R'$, and $R'\vDash\phi$
R	$\vDash$	$[s]\phi$	iff	for all a, R', $s^{\mathscr{U}(R)}=a$ and $R\xrightarrow{a}R'$ implies $R'\vDash\phi$
R	$\vDash$	$\exists\alpha.\phi$	iff	there exists $a\in\mathbf{Act}$ such that $R\vDash\phi[a/\alpha]$
R	$\vDash$	$\forall\alpha.\phi$	iff	for all $a\in\mathbf{Act}$, $R\vDash\phi[a/\alpha]$
R	$\vDash$	$\exists x.\phi$	iff	there exists $q\in\mathbb{Q}$ such that $R\vDash\phi[q/x]$
R	$\vDash$	$\forall x.\phi$	iff	for all $q\in\mathbb{Q}$, $R\vDash\phi[q/x]$

Figure 1: Satisfaction Relation for MBIU

$s ::= a \mid \alpha \mid s\diamond s$, where a ranges over **Act** and α ranges over action variables. Closed action terms are those that contain no variables. Fix a set of action payoff functions $\mathbf{V}$.

Let q be rational, $\mathtt{u_v}$ be a non-logical symbol denoting the resource payoff function $u_{v,\sigma,\delta}$ corresponding to an action payoff function $v\in\mathbf{V}$ (for a strategy and discount factor that are fixed in the interpretation of the logic). Let $\mathtt{v}(s)$ be the valuation of some action term, for some action payoff function $v\in\mathbf{V}$. Let the numerical terms, denoted t, t', etc., be formed according to the grammar $t ::= x \mid q \mid \mathtt{u_v} \mid \mathtt{v}(s) \mid t+t \mid t\times t$. Let closed terms be those that contain no variables.

We assume a set Pred of predicate symbols, each with a given arity n, with elements denoted p, q, etc.. Then, the formulae of MBIU are given by the following grammar:

$$\begin{array}{rcl}\phi & ::= & \mathrm{p}(t,\ldots,t) \mid t=t \mid s=s \mid \bot \mid \top \mid \phi\vee\phi \mid \phi\wedge\phi \mid \neg\phi \mid \phi\to\phi \\ & & \mid I \mid \phi*\phi \mid \phi -\!\!*\, \phi \\ & & \mid \langle s\rangle\phi \mid [s]\phi \\ & & \mid \exists\alpha.\phi \mid \forall\alpha.\phi \mid \exists x.\phi \mid \forall x.\phi,\end{array}$$

where $|\mathrm{p}|=n$, $(t,\ldots,t)$ is an n-tuple of terms, $=$ is syntactic equality of the rationals, and t, s, x, and α range over terms, action terms, term variables, and action variables, respectively.

The (additive) modalities are the standard *necessarily* and *possibly* connectives familiar from modal logics, in particular Hennessy–Milner-style logics for process algebras [12, 16]. As such, they implicitly use meta-theoretic quantification to make statements about reachable resources. Multiplicative modalities can also be defined [8, 7]. The connectives $*$ and $-\!\!*$ are the multiplicative conjunction (with unit I) and implication (right-adjoint to $*$), respectively.

We define how atomic predicates are interpreted with respect to resources in Figure 1. Let

ϕ, ψ, etc. denote predicate formulae. The quantifiers $\exists\alpha$ and $\forall\alpha$ bind occurrences of action variables within predicate formulae and the modalities, and $\exists x$ and $\forall y$ bind occurrences of term variables within predicate formulae. Closed formulae contain no free term variables. The formula $\phi[q/x]$ is the formula formed by the (capture-avoiding) substitution of q for the term variable x that is free in ϕ. The formula $\phi[a/\alpha]$ is defined similarly.

The mathematical structure in which we interpret MBIU is the cartesian product of the set $\bigcup_{n\in\mathbb{N}}\mathbb{Q}^n$ of finite tuples of elements of the rationals and the set $\mathbf{R}$ of resources. In an interpretation, we fix a strategy σ and a discount factor δ. Recall that each resource generates a transition structure, via the modification function. An interpretation is given with respect to a particular resource R, and is written as $\mathscr{U}(R)$. The denotations of rationals and their addition and multiplication are the obvious ones in $\mathbb{Q}$. The denotation of the symbol $\mathtt{u_v}$ is given by $u_{v,\sigma,\delta}(R)$, as specified in Definition 11. Note that the corresponding interpretation of $\mathtt{u_v}$ is a constant, at a given resource R, and is given with respect to the fixed strategy and discount factor. The denotation of actions are themselves. The denotation of $\diamond$ is action composition $\cdot$.

Recall the bisimulation relation $\sim$. A set Σ of finite tuples of elements of the rational numbers and resources is said to be $\sim$-closed if it satisfies the property that, for all resources R and S, and for all rational numbers $q_1,\ldots,q_n$, $(q_1,\ldots,q_n,R)\in\Sigma$ and $R\sim S$ implies $(q_1,\ldots,q_n,S)\in\Sigma$. Let $\mathscr{P}_\sim(\bigcup_{n\in\mathbb{N}}\mathbb{Q}^n\times\mathbf{R})$ be the set of all $\sim$-closed sets of the cartesian product of the set of finite tuples of rational numbers and the set of resources. A valuation is a function $\mathscr{V}:\mathrm{Pred}\to\mathscr{P}_\sim(\bigcup_{n\in\mathbb{N}}\mathbb{Q}^n\times\mathbf{R})$, together with a fixed strategy and dicount factor. Every valuation extends in a canonical way to an interpretation for closed MBIU-formulae, the satisfaction relation for which is indicated in Figure 1. A model for MBIU consists of the resource monoid, action monoid, and modification function, together with such an interpretation. Satisfaction in a given model is then denoted $R\vDash\phi$, read as 'for the given model, the resource R has property ϕ', and is defined as in Figure 1.

An alternative formulation of MBIU with intuitionistic additives (cf. [17, 8]) can be taken if desired. Its used in modelling applications remains to be explored in future work.

We can now formally describe payoff properties of resources, in the following sense:

Example 14. *Recall Examples 7, 9, and 13. The formula*

$$\phi=\exists x,y.(\langle p\rangle\mathtt{u_v}=x)\wedge(\langle c\rangle\mathtt{u_v}=y)\wedge(v(p)+(\delta\times x)<v(c)+\delta\times y)$$

denotes that it is possible to perform actions p and c, and that the payoff obtained by performing p is less than that obtained by performing c. Note that $u_{v,\sigma,\delta((2,1))}=5.4$ and $u_{v,\sigma,\delta((1,0))}=3$. As a result, we have that $(2,0)\vDash\phi$. $\square$

To obtain some key theoretical properties of our resource modelling framework, we require some additional properties. When we perform a composition of resources, it is necessary to take account of the partiality of the composition operator. As a result, we shall also require the following $\circ$-$\sim$-*closed* property of resource monoids. A resource monoid is $\circ$-$\sim$-closed if, for all resources R_1, S_1, R_2, $S_2\in\mathbf{R}$, if $R_1\sim S_1$, $R_2\sim S_2$, and $R_1\circ R_1$ are defined, then $S_1\circ S_2$ is defined. Henceforth, all resource monoids are assumed to be $\circ$-$\sim$-closed. When we interpret the payoff of resources, it is necessary to take account of bisimilarity. A model is payoff-$\sim$-closed if, for all $v\in\mathbf{V}$, $R,S\in\mathbf{R}$, $R\sim S$ and $u_{v,\sigma,\delta}(R)$ is defined implies that $u_{v,\sigma,\delta}(S)$ is defined and $u_{v,\sigma,\delta}(R)=u_{v,\sigma,\delta}(S)$. From this point onwards, all models are assumed to be payoff-$\sim$-closed.

With this set-up, we can prove the Hennessy–Milner soundness and completeness theorem. The soundness direction of the Hennessy–Milner completeness theorem — operational equivalence implies logical equivalence — requires the congruence property.

Theorem 15. *$R \sim S$ iff, for any model of MBIU and all ϕ, $R \vDash \phi$ iff $S \vDash \phi$.*

Proof. For soundness — operational equivalence implies logical equivalence — by induction over the structure of the formulae, using Theorem 6 and the satisfaction relation. Completeness — logical equivalence implies operational equivalence — follows [8, 7]. □

Theorem 15 provides basic assurance that the logic is well formulated, and supports the formulation of proof systems and reasoning tools, such as model checking.

5 Examples and optimality

To illustrate the logical set-up we have introduced, we begin with a classic example from distributed systems modelling: mutual producer–consumer. We then explain, using a generic example, how our set-up can be used to express Pareto optimality. This example leads naturally into a discussion of game-theoretic examples and concepts. We consider here the prisoner's dilemma, the best-response property, and Nash equilibrium.

Example 16 (Mutual producer–consumer)**.** *A classic example of distributed systems modelling is distributed coordination without mutual exclusion, the most common form of which is that of the producer–consumer system [7, Section 2.3.5]. In such a scenario, one entity generates work that another entity can handle at a later point. We modify this slightly to the scenario with two entities, where each entity can generate work for, and consume work from, the other.*

We extend Example 7. Suppose a resource monoid $(\{0,\ldots,10\} \times \{0,\ldots,10\}, \circ, (0,0))$, where $(m_1,m_2) \circ (n_1,n_2) = (m_1+n_1, m_2+n_2)$ if either m_1 or m_2 is 0 and either n_1 or n_2 is 0.

The elements of the resource monoid are pairs of natural numbers, where the first element of the pair denotes the number of work packages that the first entity can consume, and the second element of the pair denotes the number of work packages that the second entity can consume.

Suppose actions p_1, p_2, c_1, and c_2, where $\mu(p_1,(m,n)) = (m,n+1)$ if $n \leq 9$, $\mu(c_1,(m+1,n)) = (m,n)$, $\mu(p_2,(m,n)) = (m+1,n)$ if $m \leq 9$, and $\mu(c_2,(m,n+1)) = (m,n)$. The p_1 action denotes production of a work package by the first entity for the second entity, and the c_1 action denotes the consumption of a work package by the first entity. The p_2 and c_2 actions have the obvious converse denotations.

Consider the situation where the processes 'profit' from the consumption of work packages, and must 'pay' to create work packages. A pair of possible payoff functions v_1 and v_2, for the two entities, which represents this situation is $v_1(p_1) = -1$, $v_1(c_1) = 3$, $v_1(p_2) = 0$, $v_1(c_2) = 0$ $v_2(p_1) = 0$, $v_2(c_1) = 0$, $v_2(p_2) = -2$, and $v_2(c_2) = 4$.

Let σ be a function such that, if $1 \leq m$ and $1 \leq n$, then $\sigma((m,n)) = c_1c_2$, if $1 \leq m$, then $\sigma((m,0)) = c_1$, if $1 \leq n$, then $\sigma((0,n)) = c_2$, and $\sigma((0,0)) = p_1p_2$. Let the discount factor δ be 0.8. *Consider the unit resource,* $(10,0)$. *As there are only work packages available for the first entity, the actions defined on the resource are the consume action c_1, the produce action p_1, and the unit. Each entity incurs costs by performing a produce action, which only benefits the other entity. We have $v_1(p_1) + \delta \times u_{v_1,\sigma,\delta}(10,1) \approx -1 + \delta * 13.4 \approx 9.7$, $v_1(c_1) + \delta \times u_{v_1,\sigma,\delta}(9,0) \approx$*

13.4, $v_2(p_1) + \delta \times u_{v_2,\sigma,\delta}(10,1) = 0 + 0.8 * 4 = 3.2$, *and* $v_2(c_1) + \delta \times u_{v_2,\sigma,\delta}(9,0) = 0$. *The action* c_1 *gains the most for the first entity and* p_1 *gains the most for the second.* □

For either action, it is not possible to swap to an alternative action that makes one of the entities better off, without making the other entity worse off. This notion is called *Pareto optimality*.

Definition 17 (Pareto optimality)**.** *A state R is Pareto optimal if there exists an action a such that, for all other actions b, if some entity (weakly) prefers that action b be performed, then there is some other agent that strongly prefers that action a be performed. Formally, the state R is Pareto optimal if, for entities with payoff functions* $v_1, \ldots, v_n$,

$$R \models \exists \alpha . \forall \beta . (\neg(\beta = \alpha)) \rightarrow$$

$$\begin{pmatrix} \forall x,x'.\exists y,y'. \\ \quad ((\langle\alpha\rangle \mathtt{u}_{\mathtt{v}_1} = x) \wedge (\langle\beta\rangle \mathtt{u}_{\mathtt{v}_1} = x') \wedge (x \leq x')) \rightarrow \\ \quad\quad ((\langle\alpha\rangle \mathtt{u}_{\mathtt{v}_2} = y) \wedge (\langle\beta\rangle \mathtt{u}_{\mathtt{v}_2} = y') \wedge (y' < y)) \\ \quad\quad \vee \ldots \vee \\ \quad\quad ((\langle\alpha\rangle \mathtt{u}_{\mathtt{v}_n} = y) \wedge (\langle\beta\rangle \mathtt{u}_{\mathtt{v}_n} = y') \wedge (y' < y)) \end{pmatrix} \vee \ldots \vee \begin{pmatrix} \forall x,x'.\exists y,y'. \\ \quad ((\langle\alpha\rangle \mathtt{u}_{\mathtt{v}_n} = x) \wedge (\langle\beta\rangle \mathtt{u}_{\mathtt{v}_n} = x') \wedge (x \leq x')) \rightarrow \\ \quad\quad ((\langle\alpha\rangle \mathtt{u}_{\mathtt{v}_1} = y) \wedge (\langle\beta\rangle \mathtt{u}_{\mathtt{v}_1} = y') \wedge (y' < y)) \\ \quad\quad \vee \ldots \vee \\ \quad\quad ((\langle\alpha\rangle \mathtt{u}_{\mathtt{v}_{n-1}} = y) \wedge (\langle\beta\rangle \mathtt{u}_{\mathtt{v}_{n-1}} = y') \wedge (y' < y)) \end{pmatrix}$$

We abbreviate the above formula as $PO(v_1, \ldots, v_n)$. □

In Example 16, the resource $(10,0)$ is Pareto optimal, witnessed by both the actions p_1 and c_1. Note that optimality is defined in terms of actions; this is as, here, we take seriously the representation of actions that *perform* resource allocations. A transition is then an (actively performed) resource allocation.

One field in which notions of optimality have been studied significantly is that of games and decision theory. We can model games in our resource semantics. A classic decision-making example from game theory is the prisoner's dilemma.

Example 18 (Prisoner's dilemma)**.** *Two individuals have been arrested, and are kept separately, so that they cannot collude in their decision making. Each is offered the choice of attempting to 'defect', and give evidence against their partner, or to 'collaborate', and say nothing. If one person collaborates and the other defects, then the collaborating partner goes to jail for a long time, and the defecting partner goes free. If both people defect, then they both go to jail for a moderate time. If both people collaborate, then they both go to jail for a short time.*

Suppose a resource monoid $(\{r_1, r_2, r_{1,2}, e\}, \circ, e)$, *where* $r_1 \circ r_2 = r_{1,2}$. *The* r_1 *resource denotes a resource where the first person can make a choice, the* r_2 *resource denotes a resource where the second person can make a choice, and the* $r_{1,2}$ *resource denotes a resource where both people can make a choice at the same time. Suppose actions* c_1, d_1, c_2, *and* d_2, *where* $\mu(c_1, r_1) = \mu(d_1, r_1) = e$, $\mu(c_2, r_2) = \mu(d_2, r_2) = e$, *and* $\mu(c_1c_2, r_{1,2}) = \mu(c_1d_2, r_{1,2}) = \mu(d_1c_2, r_{1,2}) = \mu(d_1d_2, r_{1,2}) = e$. *The* c_1 *action denotes collaboration by the first person, and the* d_1 *action denotes defection by the person. The* c_2 *and* d_2 *actions have the obvious denotations for the second person. We make use of the trivial strategy* $\sigma(R) = 1$. *The action payoff functions* v_1 *and* v_2 *for the two people are* $v_1(c_1c_2) = -2$, $v_1(c_1d_2) = -6$, $v_1(d_1c_2) = 0$, $v_1(d_1d_2) = -4$, $v_2(c_1c_2) = -2$, $v_2(c_1d_2) = 0$, $v_2(d_1c_2) = -6$, *and* $v_2(d_1d_2) = -4$. *Hence, if the first person collaborates and the second defects, then the first person receives six years in prison (cost* $v_1(c_1d_2) = -6$*), while the second receives no time in prison (cost* $v_2(c_1d_2) = 0$*).* □

We can define notions of *best response* and *Nash equilibrium*.

Example 19 (Best response). *An action a is a best response for a given entity to a particular choice of action b by another entity, at a given resource, if the (former) entity has no other action c available to it such that the action cb is defined on the resource and the entity (strongly) prefers cb to ab. Formally, a is the best response to action b at resource R if*

$$\begin{aligned} R \quad \models \quad & \forall\alpha.\exists x,y.\Big(\big((\langle a\rangle\top \wedge \langle\alpha\rangle\top) * (\langle b\rangle\top)\big) \wedge \big([ab](\mathtt{u_v} = x) \wedge [\alpha b](\mathtt{u_v} = y)\big)\Big) \\ & \to \big((v(\alpha b) + \delta \times y) \leq (v(ab) + \delta \times x)\big). \end{aligned}$$

□

We abbreviate the above formula, denoting that a is the best response to action b for the agent whose payoff function is v, as $BR(a,b,v)$. In the prisoner's dilemma example, the best response for the first agent to the action c_2 is d_1, and $BR(d_1,c_2,v_1)$ holds.

We generalize this notation slightly, so that we write $BR(a,b_1,\ldots,b_n,v)$ to denote that a_1 is the best response the the composite action $b_1\ldots b_n$, for the payoff function v. Formally,

$$\begin{aligned} R \quad \models \quad & \forall\alpha.\exists x,y.\Big(\big((\langle a\rangle\top \wedge \langle\alpha\rangle\top) * (\langle b_1\ldots b_n\rangle\top)\rangle\big) \wedge \big([ab_1\ldots b_n](\mathtt{u_v} = x) \wedge [\alpha b_1\ldots b_n](\mathtt{u_v} = y)\big)\Big) \\ & \to \big((v(\alpha b_1\ldots b_n) + \delta \times y) \leq (v(ab_1\ldots b_n) + \delta \times x)\big). \end{aligned}$$

Here, for simplicity, we suppress all issues concerned with the structure of the composite action $b_1\ldots b_n$: In general, a process-theoretic treatment, allowing control over the presumed nature of the concurrent composition, can be given [8, 7]. Now we can express Nash equilibrium.

Example 20 (Nash equilibrium). *A state R is a Nash equilibrium for a set of entities* $I = \{1,\ldots,n\}$ *if there is a collection of actions* $a_1, \ldots, a_n$ *such that, for each entity* $i \in I$ *with payoff function* v_i*, the action* a_i *is the best response to the composition of actions* a_j*, where* $j \in I \setminus \{i\}$.

Formally, the state R is a Nash equilibrium if

$$R \models \exists\alpha_1\ldots\alpha_n\,.BR(\alpha_1,\alpha_2,\ldots,\alpha_n,v_1) \wedge \ldots \wedge BR(\alpha_n,\alpha_1,\ldots,\alpha_{n-1},v_n).$$

□

We abbreviate the above formula as $NE(v_1,\ldots,v_n)$. In the prisoner's dilemma example, the Nash equilibrium is the state $r_{1,2}$, witnessed by the actions d_1 and d_2, for payoff functions v_1 and v_2, and the property $NE(v_1,v_2)$ holds.

6 Discussion

Notice, in the examples of Section 5, the key role played in the formulae *BR* by the multiplicative conjunction, $*$. Used with the additives, it allows the separation of the resources allocated locally to different actions (the *a*s and *b*s) to be enforced when required whilst allowing utility properties of the overall system to be expressed relative to the overall resources, as required.

In a richer set-up, retaining explicit process structure — recall the discussion of Section 1 — the trace leading to the optimal and equilibrium states, together with its history of resource usage, would be represented explicitly (though at some technical cost in the development). Presentation of this richer view is deferred to another occasion.

By developing such a view we should be able to incorporate the analysis of utility and optimality presented here into the widely deployed systems and security modelling tools established in, for example, [8, 6, 7], with deployments described in, for example, [15, 1, 5, 3, 4].

References

[1] Y. Beres, D. Pym & S. Shiu (2010): *Decision Support for Systems Security Investment*. In: *Proc. 5th BDIM*, IEEE Xplore, pp. 118–125.

[2] G. Birtwistle (1987): *Discrete event modelling on Simula*. Springer.

[3] T. Caulfield & D. Pym (2015): *Improving Security Policy Decisions with Models*. *To appear, IEEE Security and Privacy*.

[4] T. Caulfield & D. Pym (2015): *Modelling and Simulating Systems Security Policy*. In: *To appear, Proc. 8th. SIMUTools*, ACM Digital Library.

[5] T. Caulfield, D. Pym & J. Williams (2014): *Compositional Security Modelling: Structure, Economics, and Behaviour*. *LNCS* 8533, pp. 233–245.

[6] M. Collinson, B. Monahan & D. Pym (2010): *Semantics for Structured Systems Modelling and Simulation*. In: *Proc. SIMUTools*, ICST, Brussels, Belgium, pp. 34:1–34:10.

[7] M. Collinson, B. Monahan & D. Pym (2012): *A Discipline of Mathematical Systems Modelling*. College Publications.

[8] M. Collinson & D. Pym (2009): *Algebra and Logic for Resource-based Systems Modelling*. *Mathematical Structures in Computer Science* 19(5), pp. 959–1027.

[9] G. Coulouris, J. Dollimore & T. Kindberg (2000): *Distributed Systems: Concepts and Design*, 3rd edition. Addison Wesley.

[10] D. Galmiche, D. Méry & D. Pym (2005): *The Semantics of BI and Resource Tableaux*. *Mathematical Structures in Computer Science* 15(6), pp. 1033–1088.

[11] D. Harel, J. Tiuryn & D. Kozen (2000): *Dynamic Logic*. MIT Press, Cambridge, MA, USA.

[12] M. Hennessy & G. Plotkin (1980): *On Observing Nondeterminism and Concurrency*. *Lecture Notes in Computer Science* 85, pp. 299–308.

[13] S. Ishtiaq & P. O'Hearn (2001): *BI as an Assertion Language for Mutable Data Structures*. In. *Proc. 28th POPL*, ACM, pp. 14–26.

[14] J.-B. Jeannin, D. Kozen & A. Silva (2013): *Language Constructs for Non-well-Founded Computation*. In: *Proc. 22nd ESOP*, Springer-Verlag Berlin, Heidelberg, pp. 61–80.

[15] Hewlett-Packard Laboratories: *Towards a Science of Risk Analysis*. Available at `http://www.hpl.hp.com/news/2011/oct-dec/security_analytics.html`.

[16] R. Milner (1989): *Communication and Concurrency*. Prentice Hall, New York.

[17] P. O'Hearn & D. Pym (1999): *The Logic of Bunched Implications*. *Bulletin of Symbolic Logic* 5(2), pp. 215–244.

[18] D. Pym, P. O'Hearn & H. Yang (2003): *Possible Worlds and Resources: The Semantics of BI*. *Theoretical Computer Science* 315(1), pp. 257–305.

[19] J. Reynolds (2002): *Separation Logic: A Logic for Shared Mutable Data Structures*. In: *Proc. 17th LICS*, IEEE, pp. 55–74.

[20] Y. Shoham & K. Leyton-Brown (2008): *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, New York, NY, USA.

6

Regular Paper

Games with Communication: from Belief to Preference Change

Guillaume Aucher
IRISA - INRIA / Université de Rennes 1

Bastien Maubert
LORIA - CNRS / Université de Lorraine, Nancy

Sophie Pinchinat
IRISA - / Université de Rennes 1

François Schwarzentruber
IRISA - ENS Rennes

In this work we consider simple extensive-form games with two players, Player **A** and Player **B**, where Player **B** can make announcements about his strategy. Player **A** has then to revise her preferences about her strategies, so as to better respond to the strategy she believes Player **B** will play. We propose a generic framework that combines methods and techniques from belief revision theory and social choice theory to address this problem. Additionally, we design a logic that Player **A** can use to reason and decide how to play in such games.

1 Introduction

Communication between players is a notion that arises naturally in a variety of contexts in game theory, and that led to the theory of games where players can communicate [5, 6, 12]. We are interested in non-cooperative games with two players, say Player **A** and **B**, in which Player **B** makes announcements about his strategy, before the game starts. Just as the *cheap talks* in [5], this preliminary communication round does not directly affect the payoffs of the game.

We illustrate our research problem with a classic example from [12] in which communication between players improves the payoff of both players. The extensive form game is described in Figure 1. Player **A** can go left or right. If **A** goes left, she gets 1$ and **B** gets 0$. If **A** goes right, player **B** can in turn choose to go left or right. If **B** goes left, he gets 100$ and **A** gets 0$, if **B** goes right both get 99$. The solution given by the classic backward induction algorithm, which relies on the hypothesis that players are rational, is the following: **A** thinks that if she goes right, **B** will go left to maximize his payoff, and **A** will get 0$. Therefore, **A** prefers to move left, and gets 1$.

On the other hand, let us assume that the players communicate and trust each other, and that **B** tells **A**: “If you move right, I will move right”. As a consequence, **A** thinks she would better move right since she would collect 99$ instead of 1$: as such, **A** has revised her preferences about her own strategies.

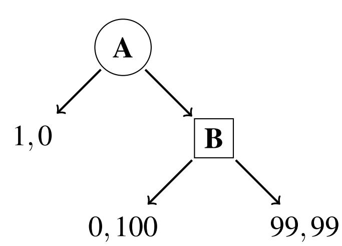


Figure 1: Motivating example

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

Notice that in this example, **B**'s announcement could have been reflected by pruning the game, in the spirit of Public Announcement Logic [10]: we could have removed the moves (in the example, just one) of **B** that do not conform to his announcement, in this very case by ruling out his left move, and have recomputed a strategy of **A** by backward induction in the pruned game.

However, the pruning technique, although attractive in practice, has some serious limitations. First, we cannot guarantee that in any game, every announcement of **B** amounts to pruning the game, in particular those relying on conditional statements. Second, **B** can make a series of successive announcements, possibly conflicting each other. In that case, **A** will need to aggregate these announcements in order to revise her beliefs on what **B** will play. This phenomenon cannot be represented straightforwardly by means of a series of destructive prunings of the game, and we propose to work on the level of **B**'s strategies instead.

Preliminary announcements can be motivated by various reasons, such as trying to coordinate with the other player or to mislead him in order to get a better payoff. After these announcements, Player **A** needs to revise her strategy so as to better respond to what Player **B** announces she will play. Notice that depending on the context, the confidence Player **A** has on Player **B**'s commitment about his annoucements varies widely. In this work, like in belief revision theory [7], we assume that Player **A** always trusts Player **B**'s last announcement, which has also priority over the previous announcements.

The question we consider is the following:

> How can Player **A** take into account the announcements of Player **B** about his strategy in order to update her preferences on her strategies?

This question can be decomposed into:

> **Question 1:** How can Player **A** revise her beliefs about Player **B**'s preferences on his strategies?
>
> **Question 2:** How can Player **A** update her preference about her strategies on the basis of these beliefs?

Regarding Question 1, we propose to apply classical belief-revision techniques[1] to represent what **A** believes about **B**'s strategy and update these beliefs when **B** makes announcements. There exist several ways to perform this update/revision, but our approach aims at remaining as general as possible by not selecting a particular one, and by leaving the choice to peak the update mechanism that reflects how trustworthy **B**'s announcements are considered.

The main originality of our contribution lies in the solution we offer for Question 2, by combining techniques and methods from game theory and from social choice theory [2]: informally, each possible strategy of **B** is seen as a voter, who votes for strategies of **A** according to the payoff **A** would obtain in the play defined by both strategies. Individual votes are then aggregated to define the new preferred strategy of **A**. Here again we do not choose a particular type of ballot nor a precise aggregation method, but rather leave it open and free to be set according to the kind of strategy one wants to obtain: for instance, one that has best average payoff against **B**'s most plausible strategies, or one that is most often a best response.

The paper is organized as follows. In Section 2, we set up the mathematical framework we use to model games and communication/announcements. In Section 3, we develop the solution to the revision of beliefs, and in Section 4 we expose our proposal for the revision of preferences. Based on the developped setting, we propose in Section 5 a logic that Player **A** can use to reason and decide how to play. Section 6 illustrates our framework on a more complex example.

[1] Typically, **A** initially believes that **B** will play one of the strategies given by the classical backward-induction algorithm. Then **B** may announce a piece of information that is in contradiction with this belief, which thus needs to be revised.

2 Games and announcements

We consider two-player extensive-form games in which at each decision node two distinct moves are available. A *finite rooted binary tree* (simply called *tree* from now on) is a prefix-closed finite set $T \subset \{0,1\}^*$. Elements of T are called *nodes*, ε is the *root*, if $w \cdot a \in T$, with $a \in \{0,1\}$, then w is called the *parent* of $w \cdot a$ and $w \cdot a$ is called the *left* (resp. *right*) *child* of w if $a = 0$ (resp. $a = 1$). If a node has no child, it is a *leaf*, otherwise it is an *interior node*. A tree is called *complete* if every interior node has exactly two children. If T, T' are trees such that $T \subseteq T'$, we say that T is a *subtree* of T'.

A *game* between **A** and **B** is a tuple $G = (T, \nu_{\mathbf{A}}, \nu_{\mathbf{B}})$ where T is a complete tree, and if we note $L \subseteq T$ the set of leaves of T, then $\nu_{\mathbf{A}} : L \to \mathbb{N}$ is the *utility function* for **A**, $\nu_{\mathbf{B}} : L \to \mathbb{N}$ is the utility function for **B**. Interior nodes are partitioned between nodes of **A** ($N_{\mathbf{A}}$) and those of **B** ($N_{\mathbf{B}}$), such that $T = N_{\mathbf{A}} \uplus N_{\mathbf{B}} \uplus L$.

Given a game $G = (T, \nu_{\mathbf{A}}, \nu_{\mathbf{B}})$, a *strategy*[2] for **A** (resp. **B**) is a subtree $\sigma_{\mathbf{A}}$ (resp. $\sigma_{\mathbf{B}}$) of T such that every node in $\sigma_{\mathbf{A}} \cap N_{\mathbf{A}}$ (resp. $\sigma_{\mathbf{B}} \cap N_{\mathbf{B}}$) has exactly one child, and every node in $\sigma_{\mathbf{A}} \cap N_{\mathbf{B}}$ (resp. $\sigma_{\mathbf{B}} \cap N_{\mathbf{A}}$) has exactly two children. Two strategies $\sigma_{\mathbf{A}}$ and $\sigma_{\mathbf{B}}$ define a unique path, hence a unique leaf in the tree T, that we shall write $\sigma_{\mathbf{A}}\hat{}\sigma_{\mathbf{B}}$. We note $\Sigma_{\mathbf{A}}$ and $\Sigma_{\mathbf{B}}$ the set of all strategies for **A** and **B**, respectively.

For a strategy $\sigma_{\mathbf{A}} \in \Sigma_{\mathbf{A}}$, we define its *value* $\mathrm{val}(\sigma_{\mathbf{A}})$ as the minimum utility it can bring about for **A**: $\mathrm{val}(\sigma_{\mathbf{A}}) := \min_{w \in L} \nu_{\mathbf{A}}(w)$. The value of a strategy for Player **B** is defined likewise.

The language Player **B** uses to make the announcements about his strategies is the bimodal language $\mathscr{L}_2$, the syntax of which is:

$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \Diamond_i \psi$$

where $p \in \{turn_{\mathbf{A}}, turn_{\mathbf{B}}\}$ and $i \in \{0,1\}$.

For $i \in \{0,1\}$, we write $\top$ for $\neg(p \wedge \neg p)$, $\Box_i \psi$ for $\neg\Diamond_i \neg\psi$, $\Box\varphi$ for $\Box_0 \varphi \wedge \Box_1 \varphi$, and $move_i$ for $\Diamond_i \top$, meaning that the strategy at this point chooses direction i.

Example 1. For instance, in the example of Figure 1, the strategy of **B** consisting in playing the action leading to 99, 99 is $\Diamond_1 \Diamond_1 \top$.

Given a game $G = (T, \nu_{\mathbf{A}}, \nu_{\mathbf{B}})$, a strategy σ can be seen as a Kripke structure with two relations (one for left child, one for right child). The valuations of propositions $turn_{\mathbf{A}}$ and $turn_{\mathbf{B}}$ are given by the partition between positions of Player **A** and Player **B**. Formally, the truth conditions are defined inductively as follows:

$$\begin{array}{lll}
\sigma, w \models turn_a & \text{if} & w \in N_a,\ a \in \{\mathbf{A}, \mathbf{B}\} \\
\sigma, w \models \neg\psi & \text{if} & \sigma, w \not\models \psi \\
\sigma, w \models \psi \wedge \psi' & \text{if} & \sigma, w \models \psi \text{ and } \sigma, w \models \psi' \\
\sigma, w \models \Diamond_i \psi & \text{if} & w \cdot i \in \sigma \text{ and } \sigma, w \cdot i \models \psi
\end{array}$$

3 Belief revision: from announcements to beliefs

We now represent the beliefs **A** has about what **B** is more likely to play, and how these beliefs evolve as **B** makes new announcements.

From a purely semantic point of view, the framework of belief revision theory [1, 8] can be roughly described as follows. Given a universe $\mathscr{U}$ of possible worlds, a player ranks each possible world via a *ranking function* $\kappa : \mathscr{U} \to \mathbb{N}$, also called *belief state*, such that $\kappa^{-1}(0) \neq \emptyset$. This ranking induces a plausibility preorder between possible worlds: among two possible worlds, the one with the lowest rank

[2] To be precise these are reduced strategies, but they are sufficient for what we present here.

is considered to be more plausible than the other by the player. Given a ranking function κ, the set of *most plausible worlds* for the player is the set $\kappa^{-1}(0)$.

The impact of a new piece of information on these beliefs is modelled by a *revision function* which takes a ranking function together with the new information, and returns the revised ranking function that induces the new belief state of the player. Many such revision functions exist in the literature, that correspond amongst other things to various degrees in the trust put in the received information, the reluctance to modify one's beliefs, etc (see *e.g.* [11]). Formally, if one chooses say formulas of propositional logic PL to represent new pieces of information, a revision function is a binary function $*: (\mathscr{U} \to \mathbb{N}) \times PL \to (\mathscr{U} \to \mathbb{N})$, and given $F \in PL$, a belief state κ is changed into $\kappa * F$.

In our framework, the universe $\mathscr{U} = \Sigma_{\mathbf{B}}$ is the set of Player **B**'s strategies, and the new pieces of information are modal formulas of $\mathscr{L}_2$, representing **B**'s announcements about his strategy. For a belief state κ, $\kappa^{-1}(0)$ is then what **A** believes **B** is the most likely to play. Initially, we assume that **A** has an *a priori* belief, represented by κ_0, that may for example arise from the very values of the strategies:

$$\kappa_0(\sigma_{\mathbf{B}}) := \max_{\sigma'_{\mathbf{B}} \in \Sigma_{\mathbf{B}}} \mathrm{val}(\sigma'_{\mathbf{B}}) - \mathrm{val}(\sigma_{\mathbf{B}}) \tag{1}$$

The revision function signature is now $(\Sigma_{\mathbf{B}} \to \mathbb{N}) \times \mathscr{L}_2 \to (\Sigma_{\mathbf{B}} \to \mathbb{N})$, and we can use any kind of revision function. For example here, we present the classic *moderate revision* [9, 11], written $*_m$, and defined by: for a belief state κ, $\psi \in \mathscr{L}_2$ and $\sigma \in \Sigma_{\mathbf{B}}$,

$$(\kappa *_m \psi)(\sigma) = \begin{cases} \kappa(\sigma) - \min_{\sigma' \models \psi} \kappa(\sigma') & \text{if } \sigma \models \psi \\ \max_{\sigma' \models \psi} \kappa(\sigma') + 1 + \kappa(\sigma) & \\ - \min_{\sigma' \not\models \psi} \kappa(\sigma') & \text{if } \sigma \not\models \psi \end{cases}$$

The moderate revision makes all the possible worlds that verify the announcement ψ more believed than those which do not; it preserves the original order of preference otherwise.

4 Voting: from beliefs to preferences

The belief Player **A** has about **B**'s strategy induces some preference over **A**'s strategies. We describe a mechanism that, given a belief state κ, computes a *preference set* $\mathscr{P}_\kappa \subseteq \Sigma_{\mathbf{A}}$. This preference set is made of all the strategies that should be preferred by **A** if she believes that **B** will play a strategy in $\kappa^{-1}(0)$. This mechanism relies on voting systems.

A plethora of different voting systems have been proposed and studied [4], verifying different properties one may want a voting system to verify (majority criterion, Condorcet criterion etc). Since we are interested in quantitative outcomes, we argue that a relevant choice is to use a *cardinal voting system* [13]. In a cardinal voting system, a voter gives each candidate a *rating* from a set of grades; we take here grades in $\mathbb{N}$. Take a set of n candidates, $C = \{c_1, \ldots, c_n\}$, and a set of m voters, $V = \{v_1, \ldots, v_m\}$. A *ballot* is a mapping $b : C \to \mathbb{N}$ and a *voting correspondence* is a function $r^C : (C \to \mathbb{N})^m \to 2^C \backslash \{\emptyset\}$ that takes a vector $(b_1, b_2, \ldots, b_m)$ of ballots (one for each voter) and returns a nonempty set of *winning* candidates[3]. In this work we take as an example the *range voting system*, but the method is generic and any other cardinal voting system can be used. Range voting works as follows: for each candidate, we sum the grades obtained in the different ballots, and the set of winners is the set of candidates who share

[3] It is called a voting rule if there is a unique winner.

the highest overall score: if b_i is voter i's ballot, for $i \in \{1,\dots,m\}$, r^C is defined by

$$r^C(b_1,\dots,b_m) := \operatorname*{argmax}_{c\in C} \sum_{i=1}^{m} b_i(c).$$

We aim at electing the strategies of Player **A** that she should prefer with regard to the most plausible strategies of Player **B**. Therefore, the set of candidates consists in Player **A**'s possible strategies ($C = \Sigma_{\mathbf{A}}$), and each of Player **B**'s most plausible strategie is seen as a voter ($V = \kappa^{-1}(0)$). We assume that Player **A** prefers strategies that in average give her the best payoff, which leads us to define ballots as follows. For each strategy $\sigma_{\mathbf{B}} \in \kappa^{-1}(0)$, we let $b_{\sigma_{\mathbf{B}}}$ be the ballot that assigns to each $\sigma_{\mathbf{A}} \in \Sigma_{\mathbf{A}}$ the payoff of **A** in the play $\sigma_{\mathbf{A}}\hat{\ }\sigma_{\mathbf{B}}$, that is $b_{\sigma_{\mathbf{B}}}(\sigma_{\mathbf{A}}) = \nu_{\mathbf{A}}(\sigma_{\mathbf{A}}\hat{\ }\sigma_{\mathbf{B}})$. In other words, each voter ranks the candidates according to the corresponding payoff for Player **A**. The voting system aggregates these "individual" preferences in order to obtain a "collective" preference $\mathscr{P}_\kappa$ against all strategies of $\kappa^{-1}(0)$, defined by:

$$\mathscr{P}_\kappa := r^C(b_{\sigma^1_{\mathbf{B}}},\dots,b_{\sigma^m_{\mathbf{B}}}), \text{ whenever } \kappa^{-1}(0) = \{\sigma^1_{\mathbf{B}},\dots,\sigma^m_{\mathbf{B}}\}.$$

Remark 1. We could use more of the information we have by letting all strategies in $\Sigma_{\mathbf{B}}$ vote, and weigh their votes according to their respective plausibility.

5 A logic for strategies, announcements and preferences

We present the formal language $\mathscr{L}_{SAP}$, where SAP stands for "Strategies, Announcements and Preferences", to reason about Player **A**'s preferences concerning her strategies, and how these evolve while Player **B** makes announcements about his strategy. The syntax of $\mathscr{L}_{SAP}$ is the following:

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi\wedge\varphi \mid P_{\mathbf{A}}\varphi \mid [\psi!]\varphi$$

where $\psi \in \mathscr{L}_2$.

The formula $P_{\mathbf{A}}\varphi$ reads as 'φ holds in all the preferred strategies of Player **A**'; $[\psi!]\varphi$ reads as 'φ holds after Player **B** announces that her strategy satisfies ψ'.

$\mathscr{L}_{SAP}$ formulas are evaluated in models of the form $(\kappa,\sigma_{\mathbf{A}})$, where κ is the *belief state* of Player **A** and $\sigma_{\mathbf{A}} \in \Sigma_{\mathbf{A}}$ is the strategy **A** is considering. The truth conditions are given inductively as follows:

$$\begin{array}{lll}
(\kappa,\sigma_{\mathbf{A}}) \models \psi & \text{if} & (\sigma_{\mathbf{A}},\varepsilon) \models \psi \\
(\kappa,\sigma_{\mathbf{A}}) \models \neg\varphi & \text{if} & (\kappa,\sigma_{\mathbf{A}}) \not\models \varphi \\
(\kappa,\sigma_{\mathbf{A}}) \models \varphi\wedge\varphi' & \text{if} & (\kappa,\sigma_{\mathbf{A}}) \models \varphi \text{ and } (\kappa,\sigma_{\mathbf{A}}) \models \varphi' \\
(\kappa,\sigma_{\mathbf{A}}) \models P_{\mathbf{A}}\varphi & \text{if} & \text{for all } \sigma'_{\mathbf{A}} \in \mathscr{P}_\kappa,\ (\kappa,\sigma'_{\mathbf{A}}) \models \varphi \\
(\kappa,\sigma_{\mathbf{A}}) \models [\psi!]\varphi & \text{if} & (\kappa *_m \psi,\sigma_{\mathbf{A}}) \models \varphi
\end{array}$$

6 Example

Consider the game in Figure 2. By backward induction, we get that **B** chooses r, **A** thus chooses γ, **B** chooses L, and finally **A** chooses Γ, obtaining 60$ while **B** gets nothing. **B** would therefore like **A** to change her mind and play Δ on the first move, so that he can play L and get 100. The problem is that if he announces that he will do so, then **A** will stick to her strategy, as she will know that changing it will give her a payoff of 50 instead of 60. So **B** announces, instead, that he commits to play either L, or R

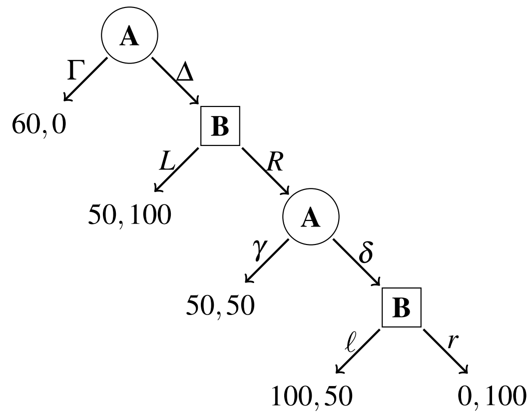


Figure 2: Second example game

and then ℓ (we note this strategy $R\ell$), but not Rr. This announcement can be described by the following $\mathscr{L}_2$-formula:

$$\psi = \Box(turn_{\mathbf{B}} \to move_0) \vee \Box\Box\Box(turn_{\mathbf{B}} \to move_0)$$

Consider now the following $\mathscr{L}_{SAP}$-formula:

$$\varphi = turn_{\mathbf{A}} \wedge P_{\mathbf{A}} move_0 \wedge [\psi!] P_{\mathbf{A}} move_1$$

φ expresses that it is Player **A**'s turn to play, and that in all her preferred strategies she goes left (*i.e.* she plays Γ), but in case Player **B** announces ψ, Player **A** prefers to play differently, namely moving right.

Now, considering this game, moderate revision, range voting, with the initial belief ranking κ_0 of Equation (1) on Page 4, and any strategy $\sigma_{\mathbf{A}} \in \Sigma_{\mathbf{A}}$, one can check that indeed we have:

$$(\kappa_0, \sigma_{\mathbf{A}}) \models \varphi$$

This is because going right ensures **A** a better mean-payoff against **B**'s most plausible strategies after the announcement ψ, which are L and Rl. However, consider now the classic plurality voting system, where each voter only gives one voice to its preffered candidate (here, the one that ensures **A** the best outcome), and where the winner is the one with most votes for him. This amounts to electing **A**'s strategy that is most often a best response against **B**'s most plausible strategies. Using this instead of range voting system, one can verify that after the announcement, the vote results into a tie, with strategy Γ of **A** obtaining one vote (from **B**'s strategy L), and strategy $\Delta\delta$ receiving the other one (from strategy Rl). Therefore, $P_{\mathbf{A}} move_1$ does not hold in the state resulting from the announcement, so that we have:

$$(\kappa_0, \sigma_{\mathbf{A}}) \not\models \varphi$$

7 Conclusion

Our work contributes to the study of games with communication. We have defined a generic framework that uses belief revision techniques to take into account communication, and voting for choosing strategies to play. A specific revision function and voting system may characterize the behavior of Player **A** (trustful, optimistic, etc), and the kind of strategies she wants (best mean payoff, most often best-response…). Investigating the theoretical properties of the agent's behavior in terms of combinations of revision and voting mechanisms is left for future work.

References

[1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985.

[2] K.J. Arrow and A.K. Sen. *Handbook of social choice and welfare*, volume 19. North Holland, 2002.

[3] C. Baier, J.P. Katoen, et al. *Principles of model checking*, volume 26202649. MIT press, 2008.

[4] S.J. Brams and P.C. Fishburn. Voting procedures. *Handbook of social choice and welfare*, 1:173–236, 2002.

[5] Vincent P Crawford and Joel Sobel. Strategic information transmission. *Econometrica: Journal of the Econometric Society*, pages 1431–1451, 1982.

[6] Joseph Farrell and Matthew Rabin. Cheap talk. *The Journal of Economic Perspectives*, pages 103–118, 1996.

[7] Peter Gärdenfors. *Knowledge in Flux (Modeling the Dynamics of Epistemic States)*. Bradford/MIT Press, Cambridge, Massachusetts, 1988.

[8] Peter Gärdenfors and Hans Rott. Belief revision. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 35–132. 1995.

[9] A.C. Nayak. Iterated belief change based on epistemic entrenchment. *Erkenntnis*, 41(3):353–390, 1994.

[10] Jan Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.

[11] H. Rott. Shifting priorities: Simple representations for twenty-seven iterated theory change operators. *Towards mathematical philosophy*, pages 269–296, 2009.

[12] J. van Benthem. *Logical dynamics of information and interaction*. Cambridge University Press, 2011.

[13] S. Vasiljev. Cardinal voting: the way to escape the social choice impossibility. *Available at SSRN 1116545*, 2008.

7

Regular Paper

Epistemic Game Theoretical Reasoning in History Based Models

Can Başkent
Department of Computer Science, University of Bath
c.baskent@bath.ac.uk

Guy McCusker
Department of Computer Science, University of Bath
g.a.mccusker@bath.ac.uk

In this work, we use Parikh and Ramanujam's history based temporal-epistemic models to reason about various epistemic game theoretical issues. First, we introduce a modal operator to express subjective preferences to history based models, and present an analysis of the Prisoners' Dilemma in this framework. Finally, we extend Aumann's celebrated agree-to-disagree result to history based models.

... you act, and you know why you act, but you don't know why you know that you know what you do.

The Name of the Rose, Umberto Eco

1 Introduction

1.1 Motivation

History based structures, proposed by Parikh and Ramanujam [16], suggest a formal framework which lies between process models, interpreted systems and propositional dynamic logics. They have been used to model epistemic messages and communication between agents, deontic obligations and the relation between obligations and knowledge [16, 14, 15]. Moreover, history based models are technically similar to interpreted systems [7, 14]. Epistemic and temporal reasoning in history based models depend on a sequence of events, called *history*.

In this work, we consider history based structures from a game theoretical point of view with some applications. In order to achieve this, we first make history-based models more *game-theory friendly* by introducing a preference modality. Then, we apply our extended formalism to a fundamental game, which is the Prisoners' Dilemma, and show how history based models can be helpful to compute the equilibrium. The choice of prisoners' dilemma is not arbitrary. Because in this game, the epistemology of the agents play a central role and the way their knowledge is formalized bear some similarities to some other formalisms of epistemic games. Building on this observation, we use history based game models to present an iteration of Aumann's well-known "agree-to-disagree" theorem.

The overall goal of this research agenda is to introduce more expressive formalism for the analysis of various foundational game theoretical issues. These issues include security games, epistemic games and how they depend on the *history* of the game and how we can *read off* strategies from such a model. We achieve this by discussing these topics in a model where histories are taken as the basic elements of the model and by introducing a modal preference relation.

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

1.2 Basic Logical Structure

Different from Kripke models, history based models are constructed by using a given set of events and agents. Events can be seen as actions or moves which vary over time and affect the knowledge of the agents. In such a model, agents' epistemic capacities differ from local and global perspectives. When a history is considered as a sequence of events, it is important to tell apart which events were carried out by which agents, and which agents can see which events, and how all this affects the knowledge and preferences of the agents.

Similar attempts have been made to apply history based models to deontic and epistemic issues [15, 14]. However, in that body of work, game theoretical reasoning was never clear or of prime importance which left many interesting phenomena outside its boundaries. In this preliminary work, we take the first step to formalize epistemic games with their histories and start from history based models. For this aim of ours, we first introduce preferences. Let us proceed step by step in our formalism.

History based structures are constructed by using a fix set of events $\mathbf{E}$ and agents $\mathbf{A}$. A finite set of events is denoted as $\mathbf{E}^*$, and for each agent i, $\mathbf{E}_i \subseteq \mathbf{E}$ is the set of events which are "seen" or "accessible" by the agent i. A finite sequence of events from $\mathbf{E}$ is denoted by lowercase h, whereas a possibly infinite sequence of events is denoted by uppercase H. We call them both *histories*.

We denote the concatenation of finite history h with (possibly infinite) history H by hH. For a set of events $\mathbf{E}$, $\mathscr{H}^*_{\mathbf{E}}$ denotes the set of all finite histories with events from $\mathbf{E}$ and $\mathscr{H}_{\mathbf{E}}$ denotes the set of all histories, finite and infinite, with events from $\mathbf{E}$. By $\mathscr{H}$, we denote any set of histories. Given two histories H, H', $H \leq H'$ denotes that H is a prefix of H'. We denote the length of finite h with $\mathsf{len}(h)$. For a history H, H_t denotes that $H_t \leq H$ with $\mathsf{len}(H_t) = t$.

We define *global history* as a sequence of events, finite or infinite, where a *local history* is the history of a particular agent. For any set of histories $\mathscr{H}$, the set $\mathsf{FinPre}(\mathscr{H})$ denotes the set of finite prefixes of the histories in $\mathscr{H}$. A set of histories $\mathscr{H}$ is called a *protocol* if it is closed, under set inclusion, for all prefixes. In other words, in order for a history to make sense, its prefixes should be included in the model, and there should be no jumps.

Now we can discuss temporal and epistemic operators in this framework. Given an agent i and a global history H, the agent i can only access some of H. For two histories H, H', if the agent can access to the same parts of H and H', then H and H' are indistinguishable for i. Then, a function $\lambda_i : \mathsf{FinPre}(H) \to \mathbf{E}^*_i$ is called a *locality function* for agent i and a global history H. Based on locality functions, the epistemic indistinguishability $\sim_i$ for agent i is defined between two histories H, H' as follows: If $H \sim_i H'$, then $\lambda_i(H) = \lambda_i(H')$.

The locality function as given above is rather general. For that reason, we impose some conditions on it [14]. First, we assume that agents' clock is consistent with the global clock, that is all agents share the same clock. Second, $\lambda_i(H)$ is embeddable in H, that is the events in $\lambda_i(H)$ appear in H in the same order. In other words, "agents are not wrong on about the events that they witness" [ibid].

For obvious reasons, $\sim_i$ is an equivalence relation. Thus, the epistemic logic of history based structures is the standard multi-agent epistemic logic $\mathbf{S5}_n$.

Given a set $\mathbf{P}$ of propositional letters, the syntax of history based models can be given as follows in the Backus - Naur form where $p \in \mathbf{P}$ and $i \in \mathbf{A}$.

$$\varphi := \mathrm{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid \bigcirc\varphi \mid \varphi U \varphi$$

The epistemic modality for agent i is K_i and the operator $\bigcirc$ is the next-time modality. We call U the *until operator*.

A history based model M is given as a tuple $M = \{\mathscr{H}, \mathbf{E}_1, \ldots, \mathbf{E}_n, \lambda_1, \ldots, \lambda_n, \mathscr{V}\}$ where $\mathscr{V}$ is a valuation function which is defined in the standard fashion as follows: $V : \mathsf{FinPre}(\mathscr{H}) \mapsto \wp(\mathbf{P})$.

History based models semantically evaluates formulas at history-time pairs. At history H and time t, the satisfaction of a formula φ is denoted as $H, t \models \varphi$, and defined inductively as follows.

$$
\begin{array}{lll}
H,t \models_M p & \text{iff} & H_t \in V(p), \\
H,t \models_M \neg\varphi & \text{iff} & H,t \not\models_M \varphi, \\
H,t \models_M \varphi \wedge \psi & \text{iff} & H,t \models_M \varphi \text{ and } H,t \models_M \psi, \\
H,t \models_M \bigcirc\varphi & \text{iff} & H,t+1 \models_M \varphi, \\
H,t \models_M K_i\varphi & \text{iff} & \forall H' \in \mathscr{H} \text{ and } H_t \sim_i H'_t \text{ implies } H',t \models_M \varphi, \\
H,t \models_M \varphi U \psi & \text{iff} & \exists k \geq t \text{ such that } H,k \models_M \psi \text{ and } \forall l, t \leq l < k \text{ implies } H,l \models_M \varphi.
\end{array}
$$

The dual of the epistemic modality will be denoted with L_i and defined in the usual way. The expression $M \models \varphi$ denotes the truth of φ in a history based model M, independent from the current history and time-stamp.

The axioms for history based models are given as follows.

- All tautologies of propositional logic,
- $K_i(\varphi \to \psi) \to (K_i\varphi \to K_i\psi)$,
- $K_i\varphi \to \varphi \wedge K_iK_i\varphi$,
- $\neg K_i\varphi \to K_i\neg K_i\varphi$,
- $\bigcirc(\varphi \to \psi) \to (\bigcirc\varphi \to \bigcirc\psi)$,
- $\bigcirc\neg\varphi \leftrightarrow \neg\bigcirc\varphi$,
- $\varphi U \psi \leftrightarrow \psi \vee (\varphi \wedge \bigcirc(\varphi U \psi))$.

The rules of inference are modus ponens, and normalization for all three modalities:

- $\models \varphi, \varphi \to \psi \therefore \models \psi$,
- $\models \varphi \therefore \models K_i\varphi$,
- $\models \varphi \therefore \models \bigcirc\varphi$,
- $\models \varphi \to (\neg\psi \wedge \bigcirc\varphi) \therefore \models \varphi \to \neg(\varphi' U \psi)$.

Additional axioms can be introduced to history based models to formalize variety of properties including perfect recall and no learning [14]. It is also important to note that the above axiomatization does not include any axioms that govern a possible interaction between the epistemic and temporal modalities. The reason for this is the fact that the former quantifies over histories (up to a fixed t) whereas the latter ranges over the time stamp only. However, as we argued earlier, further temporal and epistemic conditions can be forced by introducing various interaction axioms.

History based models combine epistemic and temporal modalities in a complex way and they are closely related to *runs* [7]. Furthermore, histories and runs can be translated to each other effectively [14]. However, it still remains an unexplored direction to use history based models for game theoretical purposes. We will illustrate it in due time.

Now, from a modal logical point of view, the immediate question is how bisimulations can be defined within the context of history based models where we focus on events/actions as opposed to possible worlds/states and possess complex temporal modalities such as the until modality.

Definition 1.1. For history based models M, M', a bisimulation $\bowtie$ between M and M' is a tuple $\bowtie = (\bowtie_0, \bowtie_1)$ where $\bowtie_0 \subseteq M \times M'$ and $\bowtie_1 \subseteq M^2 \times M'^2$ such that
Propositional base case:

- If $H, t \bowtie_0 H', t'$, then H, t and H', t' satisfy the same propositional variable,

Temporal forth case:

- If $H,t \bowtie_0 H',t'$ and $t < u$, then there is u' in M' such that $t' < u'$, $H,u \bowtie_0 H',u'$ and $(H,t),(H,u) \bowtie_1 (H',t'),(H',u')$,
- If $(H,t),(H,u) \bowtie_1 (H',t'),(H',u')$ and if there is v' with $t' < v' < u'$, then there exists v such that $t < v < u$ and $H,v \bowtie_0 H',v'$,

Temporal back case:

- If $H,t \bowtie_0 H',t'$ and $t' < u'$, then there is u in M such that $t < u$, $H,u \bowtie_0 H',u'$ and $(H,t),(H,u) \bowtie_1 (H',t'),(H',u')$,
- If $(H,t),(H,u) \bowtie_1 (H',t'),(H',u')$ and if there is v with $t < v < u$, then there exists v' such that $t' < v' < u'$ and $H,v \bowtie_0 H',v'$,

Epistemic forth case:

- If $H,t \bowtie_0 H',t'$ and $H_t \sim_i K_l$, then there is K',l' in M' such that $K,l \bowtie_0 K',l'$ and $H'_{t'} \sim_i K'_{l'}$,

Epistemic back case:

- If $H,t \bowtie_0 H',t'$ and $H'_{t'} \sim_i K_{l'}$, then there is K,l in M such that $K,l \bowtie_0 K',l'$ and $H_t \sim_i K_l$,

In the above definition, the *interval bisimulations* we defined in the temporal cases are needed for the until modality, as the until modality is essentially an interval process equivalence. This definition clarifies how history based models can simulate state-based models or interpreted systems, and how different histories can be identified to form bisumulations. Based on this definition, the following theorem follows immediately.

Theorem 1.2. *For history based models M,M', if $M \bowtie M'$, then they satisfy the same formula.*

Proof. For the epistemic case see [3], for the temporal case see [11]. □

2 Adding Preferences

History based models provide sufficient tools to formalize simple epistemic games. If games are considered as formal representations of interactive situations in which agents make rational decisions, such decisions then must rely on those agents' subjective preferences. Moreover, these subjective preferences may change depending on what stage of the game the players are in and how far ahead in the game they have progressed. In short, preferences depend on the game history. This is the motivation behind introducing subjective preferences into history based models.

For an agent i, and possibly infinite histories H,H', the expression $H \preceq_i H'$ denotes that "the agent i (weakly) prefers H' to H". The preference relation will be taken as a pre-order satisfying reflexivity and transitivity [2, 10].

We can amend the syntax of the logic of history based models with the modal operator $\Diamond_i \varphi$ which expresses that there is a history which is at least as good as the current one and satisfies φ for agent i. We specify the semantics of this new modality as follows.

$$H,t \models \Diamond_i \varphi \;\text{ iff }\; \exists H'.H \preceq_i H' \;\text{ and }\; H',t \models \varphi$$

The dual of the above modality is denoted by $\Box_i$ with the following semantics: $H,t \models \Box_i \varphi$ whenever $\forall H'.H \preceq_i H' \to H',t \models \varphi$.

Notice that this formalism compares histories as opposed to propositions. For a history based model M, the formula $M \models \varphi \to \Diamond_i \psi$ denotes that the agent i prefers ψ to φ. In other words, each φ has an alternative history which is at least as good as the current one and satisfies ψ.

The additional axioms and rules of inference for the **S4** preference modality can be given as follows.

- $\varphi \to \Diamond_i \varphi$,
- $\Diamond_i \Diamond_i \varphi \to \Diamond_i \varphi$,

The additional rule of inference for the preference modality is the expected one.

- $\models \varphi \mathrel{.\!\!\cdot\!\!.} \models \Box_i \varphi$

We call the logic of history based structures with preferences as HBPL after *history based preference logic*. HBPL can be supplemented with various additional axioms to express some other interactive epistemic, temporal and game theoretical properties. Here we consider a few.

Connectedness of Preferences The connectedness property for the preference relation suggests that any two histories are comparable. Therefore, it can be formalized as $\forall H, H'. H \preceq_i H' \vee H' \preceq_i H$. The modal axiom that corresponds to it is the following axiom: $\Box_i(\Box_i \varphi \to \psi) \vee \Box_i(\Box_i \psi \to \varphi)$. This renders the frame with preference modality as a total pre-order.

Epistemic Perfect Recall The agents with perfect recall retain knowledge once they acquired it. The standard axiom for this property is given as follows: $K_i \bigcirc \varphi \to \bigcirc K_i \varphi$. It is rather easy to show that this axiom is valid in HBPL. Given an arbitrary history H and a time-stamp t, we start with assuming $H, t \models K_i \bigcirc \varphi$. Our aim is to show that $\bigcirc K_i \varphi$ holds at H, t. Now, by definition, $\forall H'.(H \sim_i H' \to H', t \models \bigcirc \varphi)$. Unfolding the temporal modality gives $\forall H'.(H \sim_i H' \to H', t+1 \models \varphi)$. Now, we can fold back, but this time starting with the epistemic modality. By definition, we first obtain $H, t+1 \models K_i \varphi$, which produces $H, t \models \bigcirc K_i \varphi$. Thus, $K_i \bigcirc \varphi \to \bigcirc K_i \varphi$ is valid in HBPL.[1]

Preferential Perfectness By *preferential perfectness*, we mean that agents do not change their preferences in time. Consider the scheme $\Box_i \bigcirc \varphi \to \bigcirc \Box_i \varphi$. It is also easy to show that this scheme is valid in HBPL, so we skip it.

Epistemic Rationality By a slight abuse of terminology we will call the axiom scheme $\Diamond_i L_i \varphi \to L_i \Diamond_i \varphi$ as the Church-Rosser axiom. The frames of HBPL which satisfies the *Church-Rosser* Property enjoys the following condition:

$$\begin{tikzcd}
H' \arrow[r, "\preceq_i"] & H'' \arrow[d, dotted, "\sim_i"] \\
H \arrow[u, "\sim_i"] \arrow[r, dotted, "\preceq_i"] & K
\end{tikzcd}$$

If $H \sim_i H'$ and $H' \preceq_i H''$, then there exists a history K such that $H \preceq_i K$ and $H'' \sim_i K$.

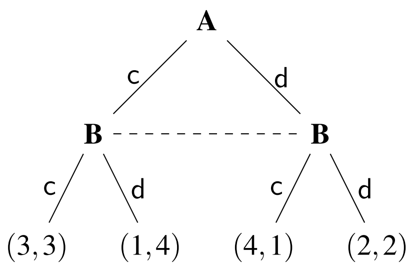
Consider the dual axiom scheme $K_i \Box_i \varphi \to \Box_i K_i \varphi$. This is valid in HBPL. Similar to above, consider $H, t \models K_i \Box_i \varphi$. Then by definition, $\forall H'.(H \sim_i H' \to H', t \models \Box_i \varphi)$. This reduces to $\forall H', H''(H \sim_i H' \wedge H' \preceq_i H'' \to H'', t \models \varphi)$. By the Church-Rosser Property, then there exists a history K such that $H \preceq_i K$ and $H'' \sim_i K$. So, by definition, $K, t \models K_i \varphi$. Thus, $H, t \models \Box_i K_i \varphi$, which shows the validity of the axiom scheme in question.

Various other combinations of the modalities, such as $\Box_i K_i \bigcirc \varphi \to \bigcirc \Box_i K_i \varphi$ or $K_i \Box_i \bigcirc \varphi \to \bigcirc K_i \Box_i \varphi$ remain valid in HBPL. Similarly, various commutativity properties of the modalities, such as $K_i K_j \varphi \leftrightarrow K_j K_i \varphi$, can be examined in order to shed light to epistemic interaction of the agents.

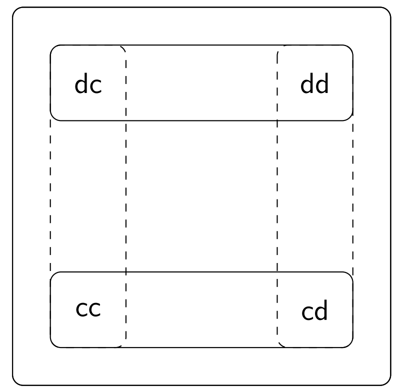
[1] However, as van der Meyden showed, the axiom $K_i \bigcirc \varphi \to \bigcirc K_i \varphi$ is not sufficient to establish the completeness of frames with respect to perfect recall [13, 12]. The additional axiom required for this task is a complicated one: $K_i \varphi_1 \wedge \bigcirc (K_i \varphi_2 \wedge \neg K_i \varphi_3) \to \neg K_i \neg((K_i \varphi_1) U ((K_i \varphi_2) U \neg \varphi_3))$.

3 Case Study: An Epistemic Analysis of the Prisoner's Dilemma

Viewed as histories with imposed subjectives preferences, HBPL is helpful in formalizing epistemic games. As an application, we consider how HBPL computes best responses in Prisoners' Dilemma (PD, for short).



(a) *Extensive form representation*



(b) *Equivalence classes of histories*

Figure 1: Prisoners' Dilemma

Let us consider PD in its extensive normal form where the utility pair (u_A, u_B) denotes the utility of the players **A** and **B** respectively. Epistemic indistinguishability of the states for player **B** is denoted by the dashed line given in Figure 1a. Based on the extensive normal form, we reproduce the epistemic model of PD below where agents' knowledge is represented by the equivalence classes in the standard way in Figure 1b [2]. In the history xy, the first event denotes Player **A**'s move while the second one denotes Player **B**'s move. Also, due to the utilities associated with the players at the possible end states of the game, we have $\mathsf{cc} \preceq_B \mathsf{cd}$ and $\mathsf{dc} \preceq_B \mathsf{dd}$. Similarly, $\mathsf{cc} \preceq_A \mathsf{dc}$ and $\mathsf{cd} \preceq_A \mathsf{dd}$. The HBPL model for PD can easily be read off from Figures 1a and 1b, hence skipped.

We define best response relation for agent i in a two-player game as follows where $-i$ denotes the players other than i.

$$BR_i = \sim_{-i} \cap \preceq_i$$

By a slight abuse of notation, we will use the same notation to denote the intersection modality. Put informally, in this context, best response for an agent is a move that is *indistinguishable* by the opponent yet more preferable for the agent himself.

Now let us see how we can verify the best responses of the players. Recall that for both players, the best response is defect (the move d). What follows is a direct computation of best responses for each players based on the game history and the subjective preferences of the players. Since PD is a one-shot game, we use a fixed-time stamp t.

We start with Player **A**.

$\mathsf{cc}, t \not\models BR_A$	since	there is dc such that $\mathsf{dc} \sim_B \mathsf{cc}$ and $\mathsf{cc} \preceq_A \mathsf{dc}$
$\mathsf{cd}, t \not\models BR_A$	since	there is dd such that $\mathsf{dd} \sim_B \mathsf{cd}$ and $\mathsf{cd} \preceq_A \mathsf{dd}$
$\mathsf{dc}, t \models BR_A$	since	there is no compatible history with these properties. The only alternative cc fails to bring a higher utility
$\mathsf{dd}, t \models BR_A$	since	there is no compatible history with these properties. The only alternative cd fails to bring a higher utility

Similarly for player **B**:

$\mathsf{cc}, t \not\models BR_B$	since	there is cd such that $\mathsf{cd} \sim_A \mathsf{cc}$ and $\mathsf{cc} \preceq_A \mathsf{cd}$
$\mathsf{dc}, t \not\models BR_B$	since	there is dd such that $\mathsf{dd} \sim_A \mathsf{dc}$ and $\mathsf{dc} \preceq_A \mathsf{dd}$
$\mathsf{cd}, t \models BR_B$	since	there is no compatible history with these properties. The only alternative cc fails to bring a higher utility
$\mathsf{dd}, t \models BR_B$	since	there is no compatible history with these properties. The only alternative dc fails to bring a higher utility

Based on the above analysis, Nash equilibrium can be observed at dd which is the state where neither of the agents can unilaterally benefit by diverging from. If **A** diverges, then the history cd is obtained which is not preferable for him. Similarly, if **B** diverges, then the history dc is obtained which is not preferable for him either. Thus, dd is the Nash equilibrium of PD.

It can be noticed that we have not discussed *strategies* in HBPL. Therefore, the Nash equilibrium in HBPL is simply a game history formed if the players follow a particular equilibrium strategy constructed with respect to their best responses. Therefore, the equilibrium is expressed in terms of a game history.

This is a model of prisoner's dilemma in HBPL.

4 Case Study: Set Based Analysis of Histories and Decisions - An Agree-to-Disagree Result

The above analysis of PD considered epistemic states of the game as sequences of moves, or histories. However, there was an additional layer of formalization on top of the histories, which considered the *structure* of histories and their relation to each other in the form of equivalence classes. We can now develop this idea further, and relate it to a well-known and foundational result in epistemic game theory.

Aumann's celebrated agree-to-disagree theorem is a mile-stone in epistemic game theory [1]. Several iteration of the agree-to-disagree result have been given in the literature [4]. In this section, we take one of such variations, which is due to Dov Samet, and apply it to history based models. Samet's model uses a non-probabilistic model together with a set algebra where the knowledge is formalized using a set operator [17]. In that case, Aumann's original statement of the theorem becomes a special case of Samet's generalized formalism.

Our application of HBPL to agree-to-disagree theorem serves two goals. First, it shows the versatility of HBPL by considering sets of histories as equivalence classes. Second, it shows that it is possible to introduce two different levels of complexity to epistemic games. The first level of complexity deals with the game play and constructs a history which includes the moves of all players and the local knowledge of players. The second level of complexity, on the other hand, provides a global view of the model by forming equivalence classes of histories introducing additional structure. In HBPL, unlike Kripkean models, we can read off the epistemics of agents from the histories directly. This is one of the major advantages of using history based models.

However, notice that HBPL evaluates truth at time stamps. The truth of a formula depends both on the history and where we are at the history. Nevertheless, the epistemic modality and the preference modality in HBPL does not quantify over the temporal parameter. For that reasons, in what follows we assume that the time stamp is fixed and the same for all agents, for simplicity.

Let us now start with defining some standard epistemic operators following [17].

Definition 4.1. For a given set of agents **A** and a formula φ, we define $E_{\mathbf{A}}\varphi$ which reads "everyone in **A** knows φ". Formally, $E_{\mathbf{A}}\varphi = \bigwedge_{i \in \mathbf{A}} K_i \varphi$.

We define the common knowledge operator $C_{\mathbf{A}}\varphi$ which reads "φ is common knowledge among $\mathbf{A}$" as follows

$$C_{\mathbf{A}}\varphi = E_{\mathbf{A}}\varphi \wedge E^2_{\mathbf{A}}\varphi \wedge \cdots \wedge \ldots E^m_{\mathbf{A}}\varphi \wedge \ldots$$

where $E^1_{\mathbf{A}} = E_{\mathbf{A}}\varphi$ and $E^{k+1}_{\mathbf{A}}\varphi = E_{\mathbf{A}}E^k_{\mathbf{A}}\varphi$, for $k \geq 1$.

The epistemic indistinguishability relation $\sim_i$ for agent i makes it possible to redefine history based models as epistemic set models in a way that we can compare agents' knowledge relative to a given protocol [17]. In order to achieve this, we define a set valued function which takes a set and returns a partition in that set that belongs to the agent. Given a protocol $\mathscr{H}$, we define $\kappa_i : 2^{\mathscr{H}} \mapsto 2^{\mathscr{H}}$. For simplicity, we will consider sets of finite histories, and denote the sets of histories with bold letters such as $\mathbf{h}, \mathbf{h}'$ etc. In this model, for each agent, there exists a partitioning of the given protocol $\mathscr{H}$.

Now, in a given model, let π_i denote the agent i's partitioning of the protocol $\mathscr{H}$. That is, for each i, there exists equivalence classes of histories in $\mathscr{H}$. Similarly, $\pi_i(h)$ denotes the partition for agent i that contains h. In other words, for an agent at history h, the histories in $\pi_i(h)$ are *indistinguishable*.

Now, we define $\kappa_i(\mathbf{h}) = \{h : \pi_i(h) \subseteq \mathbf{h}\}$. Simply put, for a set of histories $\mathbf{h}$, the set $\kappa_i(\mathbf{h})$ includes all the histories h whose partitions are contained in $\mathbf{h}$. The operator κ_i is a set valued operator which will express agent's knowledge. In order to achieve this, we stipulate that κ_i satisfies the following three properties, for given sets of histories $\mathbf{h}, \mathbf{h}'$ [7].

1. $\kappa_i(\mathbf{h} \cap \mathbf{h}') = \kappa_i(\mathbf{h}) \cap \kappa_i(\mathbf{h}')$
2. $\kappa_i(\mathbf{h}) \subseteq \mathbf{h}$
3. $-\kappa_i(\mathbf{h}) = \kappa_i(-\kappa_i(\mathbf{h}))$

where $-$ denotes the set theoretical complement. The above three property makes κ_i an epistemic operator where the first condition corresponds to normality, the second one to veridicality and the last one to introspection in the traditional sense. Similarly, a common knowledge operator $\mathfrak{c}$ can be defined for sets of histories to express the common knowledge modality $C_{\mathbf{A}}$.

Extending the preference relation in HBPL, it is possible to compare agents' knowledge relative to each other, given a set of histories.

Definition 4.2. Define the set of histories $[j > i]^{\mathscr{H}}$ in which agent j is at least as knowledgeable as agent i with respect to a given set of protocols $\mathscr{H}$ as follows.

$$[j > i]^{\mathscr{H}} := \bigcap_{\mathbf{h} \in 2^{\mathscr{H}}} -\kappa_i(\mathbf{h}) \cup \kappa_j(\mathbf{h})$$

By a slight abuse of notation, we will denote the proposition whose extension is the set $[j > i]^{\mathscr{H}}$ by the same symbol.

Since our epistemic model is based on equivalence classes and partitions, it is possible to compare agents' knowledge based on their partitions. The following lemma expresses the fact that the finer the partitions, the more the epistemic knowledge.

Lemma 4.3 ([17])**.** *$h \in [j > i]^{\mathscr{H}}$ iff $\pi_j(h) \subseteq \pi_i(h)$.*

Proof. Let $h \in [j > i]^{\mathscr{H}}$. For $\mathbf{h} = \pi_i(h)$, and by the above definition, we have $h \in -\kappa_i(\pi_i(h)) \cup \kappa_j(\pi_i(h))$. By definition, $\kappa_i(\pi_i(h)) = \pi_i(h)$ and also $h \in \pi_i(h)$. Thus, $h \in \kappa_j(\pi_i(h))$. Then, by definition of κ, $\pi_j(h) \subseteq \pi_i(h)$.

For the converse direction, let κ, $\pi_j(h) \subseteq \pi_i(h)$. Suppose for some set of histories $\mathbf{h}$, we have $h \in \kappa_i(\pi_i(\mathbf{h}))$. Then, by definition, $\pi_i(h) \subseteq \mathbf{h}$. By the initial assumption, we also have $\pi_j(h) \subseteq \mathbf{h}$ which means that $h \in \kappa_j(\pi_i(\mathbf{h}))$. Thus, for each $\mathbf{h} \in 2^{\mathscr{H}}$, $h \in -\kappa_i(\mathbf{h}) \cup \kappa_j(\mathbf{h})$. Hence, $h \in [j > i]^{\mathscr{H}}$. $\square$

Another interesting lemma suggested by Samet shows how the comparison ordering of agents' knowledge and epistemic partitions relate to each other. Let us prove it for HBPL.

Lemma 4.4 ([17])**.** *$h \in \kappa_i([j > i])$ iff $\pi_i(h) = \bigcup_{h' \in \pi_i(h)} \pi_j(h')$.*

Proof. The proof directly follows from the definitions.

$h \in \kappa_i[j > i]$ amounts to $\pi_i(h) \subseteq [j > i]$ by definition. By the first lemma, this statement holds if and only if for each $h' \in \pi_i(h)$ we have $\pi_j(h') \subseteq \pi_i(h)$. This is equivalent to $\pi_i(h) = \bigcup_{h' \in \pi_i(h)} \pi_j(h')$. □

Next, we define a decision function $\delta_i : \mathscr{H} \mapsto D$ for a protocol $\mathscr{H}$, agent i and any set of decisions D. The vector $\delta = (\delta_1, \ldots, \delta_n)$ is called a decision profile for n agents. In this context, we consider D as *any* set of decisions, not necessarily probabilistic or propositional. Now, for a decision $d \in D$, we define the proposition $[\delta_i = d]^{\mathscr{H}}$ with the following set as its extension.

$$[\delta_i = d]^{\mathscr{H}} = \{H \in \mathscr{H} : \delta_i(H) = d \text{ for all } H \in \mathscr{H}\}$$

Similarly, we will use $[\delta_i = d]^{\mathscr{H}}$ to denote both the set and the proposition, if no confusion arises from the context. If obvious, we will drop the superscript.

We assume each agent knows his decision [17]. In our notation, this amounts to the following statement $[\delta_i = d]^{\mathscr{H}} \subseteq \kappa_i([\delta_i = d]^{\mathscr{H}})$. In other words, agents agree with those agents who know better. Let us put it formally and more carefully as follows.

Definition 4.5. $\kappa_i([j > i]^{\mathscr{H}} \cap [\delta_j = d]^{\mathscr{H}}) \subseteq [\delta_i = d]^{\mathscr{H}}$.

Sure Thing Principle suggests that if an agent j is at least as knowledgable as another agent i, and if j's decision is d, then i's decision is also d.

If the knowledge comparison is an intuitive order, this means that there can be postulated some agents that know less than all the other agents. Now, an agent i is called *an epistemic dummy* if all the agents are at least as knowledgeable as i. Dummy agents can be introduced to decision making process if they do not upset the sure thing principle. The following notion incorporates dummy agents into the sure thing principle.

Definition 4.6. A decision profile $\mathbf{d}$ in a model with a protocol $\mathscr{H}$ with n agents is expandable if for any additional epistemic dummy i, there exists a decision profile $\mathbf{d}'$ which satisfies the sure thing principle.

It is important to stipulate that for an expandable decision profile $\mathbf{d}$ and dummy agent i, $\mathbf{d}$ and $\mathbf{d}'$ agree on the decisions of agents who are not dummies. Expandable decision profiles play an important role for the following theorem, which we adopt from [17].

Theorem 4.7. *If δ is an expandable decision profile in a model with a protocol $\mathscr{H}$ with n agents, then for any decisions $d_1, \ldots, d_n$ in D which are not identical, $C(\bigwedge_{i \leq n}[\delta_i = d_i]^{\mathscr{H}})$ is nowhere satisfiable, in other words $\mathfrak{c}(\bigcap_{i \leq n}[\delta_i = d_i]^{\mathscr{H}}) = \emptyset$.*

Proof. First, we will construct an epistemic dummy agent. Call him $n+1$. Now, define π_{n+1} as the finest partition which is coarser than any of the partitions π_i for $1 \leq i \leq n$. Then, the epistemic set operator κ_{n+1} based on the partition π_{n+1} is the common knowledge operator $C_{\mathbf{A}}$ [7].

Also $\kappa_{n+1}([j > n+1]) = \mathscr{H}$ as κ_{n+1} is common knowledge operator and $[j > n+1] = \mathscr{H}$ for each agent j for $1 \leq j \leq n$. This shows that $n+1$ is an epistemic dummy.

Now, for an expandable decision profile δ, there is δ_{n+1} such that $(\delta_1, \ldots, \delta_{n+1})$ satisfies the sure thing principle.

We will now prove the contrapositive. For this, let $h \in \mathfrak{c}(\bigcap_i[\delta_i = d_i])$. We showed that κ_{n+1} is the common knowledge operator. So, let $h \in \kappa_{n+1}(\bigcap_i[\delta_i = d_i])$.

Since κ operator satisfies the property that $\kappa(\mathbf{h} \cap \mathbf{h}') = \kappa(\mathbf{h}) \cap \kappa(\mathbf{h}')$, we have $h \in \bigcap_i \kappa_{n+1}([\delta_i = d_i])$. Therefore, for each j, $h \in \kappa_{n+1}([\delta_j = d_j])$.

By definition, π_{n+1} is coarser than π_j for any j, and $\pi_{n+1}(h) = \bigcup_{h' \in \pi_{n+1}(h)} \pi_j(h')$. By the second lemma, we observe that $h \in \kappa_{n+1}([j > i])$.

Now, we have $h \in \kappa_{n+1}([\delta_j = d_j])$ and $h \in \kappa_{n+1}([j > i])$, so that we can apply the sure thing principle to obtain $h \in [\delta_{n+1} = d_j]$ for each j. Therefore, all the decisions d_j are identical to $\delta_{n+1}(h)$. This is also why we need an epistemic dummy agent.

Thus, if the common knowledge is not an empty set, the decisions of the agents coincide.

This proves the theorem. □

So far, we have adopted Aumann's well-known theorem to history based structures via Samet's formalism [1, 17]. What is more interesting is, via our proof, the result can be extended to runs and function based knowledge structures, and expands the domain of applicability of Aumann's theorem [5, 8, 6].

Now, it is worth mentioning the potential future applications of above results. First, Theorem 4.7 provides some good handles for systems security policies. In systems' security, it can obviously be seen that attacker's and defender's decisions cannot be the same for a successful attack. Also, it is not enough that they will have different decisions, those decisions cannot be commonly known among them. The theorem specifies under which conditions, agents' decisions which are not identical cannot be common knowledge. If they are common knowledge, then some agents *cannot agree to disagree* [1].

Also, it is noteworthy that the decision set D above is given arbitrarily. Therefore, it seems possible to choose a probability measure to precise the decisions of the agents in a way close to the original set up of the theorem by Aumann [1]. Such a set-up would facilitate the introduction of probabilistic issues and mixed strategies into HBPL, which we leave to future work.

Finally, set based approaches to histories relate HBPL to topological spaces where agents' indistinguishable histories may form an open set. In such a formalization, topological transformations and paths might help us to transform histories in a continuous and knowledge-preserving fashion.

5 Conclusion

History based models provide a natural formalism for epistemic logic. In this work, we extended the standard framework by introducing modal preferences in order to reason about subjective preferences and epistemic games, and made a connection between logic and games via history based models. This opens up a broad spectrum of theoretical and applied fields for future work including process algebras, preference logics, deontological games and topologies.

History based models also seem to provide a richer understanding for agents' rationality by introducing various tools for explicate agents' decision and preferences based on the progress of the game, preferences and the time. This potential can easily be extended to a broader and utilitarian analysis of history based games, which we leave for future work.

In this preliminary work, apart from introducing a conceptual development, we argued that HBPL fits rather well within the current research on epistemic game theory, modal logic and logic of games, and provides a new and broad framework.

Acknowledgement The epigraph is taken from [9].

References

[1] Robert J. Aumann (1976): *Agreeing to Disagree*. The Annals of Statistics 4(6), pp. 1236–9.

[2] Johan van Benthem (2014): *Logic in Games*. MIT Press.

[3] Patrick Blackburn, Maartijn de Rijke & Yde Venema (2001): *Modal Logic*. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press.

[4] Boudewijn de Bruin (2010): *Explaining Games: The Epistemic Programme in Game Theory*. Synthese Library 346, Springer.

[5] Ronald Fagin (1994): *A Quantitative Analysis of Modal Logic*. The Journal of Symbolic Logic 59(1), pp. 209–252.

[6] Ronald Fagin, John Geanakoplos, Joseph Y. Halpern & Moshe Y. Vardi (1999): *The Hierarchical Approach to Modeling Knowledge and Common Knowledge*. International Journal of Game Theory 28, pp. 331–365.

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

[8] Ronald Fagin, Joseph Y. Halpern & Moshe Y. Vardi (1991): *A Model-Theoretic Analysis of Knowledge*. Journal of the Association for Computing Machinery 38(2), pp. 382–428.

[9] Joseph Y. Halpern & Ronald Fagin (1989): *Modelling Knowledge and Action in Distributed Systems*. Distributed Computing 3(4), pp. 159–177.

[10] S. O. Hanson (2001): *Preference Logic*. In Dov Gabbay & F. Guenthner, editors: Handbook of Philosophical Logic, 4, Kluwer, pp. 319–393.

[11] Natasha Kurtonina & Maarten de Rijke (1997): *Bisimulations for Temporal Logic*. Journal of Logic, Language and Information 6(4), pp. 403–425.

[12] Ron van der Meyden (1994): *Axioms for Knowledge and Time in Distributed Systems with Perfect Recall*. In: Proceedings of IEEE Symposium on Logic in Computer Science, pp. 448–457.

[13] Ron van der Meyden & Ka shu Wong (2003): *Complete Axiomatization for Reasoning About Knowledge and Branching Time*. Studia Logica 75(1), pp. 93–123.

[14] Eric Pacuit (2007): *Some Comments on History Based Structures*. Journal of Applied Logic 5(4), pp. 613–24.

[15] Eric Pacuit, Rohit Parikh & Eva Cogan (2006): *The Logic of Knowledge Based Obligation*. Synthese 149(2), pp. 311–341.

[16] Rohit Parikh & R. Ramanujam (2003): *A Knowledge Based Semantics of Messages*. Journal of Logic, Language and Information 12(4), pp. 453 – 467.

[17] Dov Samet (2010): *Agreeing to Disagree: The Non-probabilistic Case*. Games and Economic Behavior 69(1), pp. 169–174.

8

Regular Paper

Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications*

Petr Čermák Alessio Lomuscio
Imperial College London
London, UK

Aniello Murano
Università degli Studi di Napoli Federico II
Napoli, Italy

Strategy Logic (SL) has recently come to the fore as a useful specification language to reason about multi-agent systems. Its one-goal fragment, or SL[1G], is of particular interest as it strictly subsumes widely used logics such as ATL*, while maintaining attractive complexity features. In this paper we put forward an automata-based methodology for verifying and synthesising multi-agent systems against specifications given in SL[1G]. We show that the algorithm is sound and optimal from a computational point of view. A key feature of the approach is that all data structures and operations on them can be performed on BDDs. We report on a BDD-based model checker implementing the algorithm and evaluate its performance on the fair process scheduler synthesis.

1 Introduction

A concern in the deployment of autonomous multi-agent systems (MAS) is the limited availability of efficient techniques and toolkits for their verification. The problem is compounded by the fact that MAS require ad-hoc techniques and tools. This is because, while reactive systems are typically specified purely by temporal properties, MAS are instead described by statements expressing a number of typical AI concepts including knowledge, beliefs, intentions, and abilities.

Some progress in this direction has been achieved in the past decade. For example, several efficient techniques are now available for the verification of MAS against temporal-epistemic languages [22, 27, 16, 21]. Some of these have been implemented into fully-fledged model checkers [11, 14, 18].

Less attention has so far been given to the verification of properties expressing cooperation and enforcement [19, 15]. While the underlying logics have been thoroughly investigated at theoretical level [3], tool support is more sketchy and typically limited to alternating-time temporal logic (ATL) [1]. A number of recent papers [8, 24] have however pointed out significant limitations of ATL when used in a MAS setting. One of these is the syntactic impossibility of referring explicitly to what particular strategies a group of agents ought to use when evaluating the realisability of temporal properties in a MAS. Being able to do so would enable us to express typical MAS properties, including strategic game-theoretic considerations for a group of agents in a cooperative or adversarial setting.

In response to this shortcoming, *Strategy logic* (SL), a strict extension of any logic in the ATL hierarchy, has recently been put forward [24]. In SL, strategies are explicitly referred to by using first-order quantifiers and bindings to agents. Sophisticated concepts such as Nash equilibria, which cannot be expressed in ATL, can naturally be encoded in SL.

Given this, a natural and compelling question that arises is whether *automatic and efficient* verification methodologies for MAS against SL specifications can be devised. The answer to this is negative in general: model checking systems against SL specifications is NONELEMENTARYSPACE-HARD [25], thereby hindering any concrete application on large systems. It is therefore of interest to investigate

*This is an expository contributions based on [6].

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

whether computationally attractive methodologies can be put forward for fragments of SL. The only contributions we are aware of in this regard are [4, 12], where model checking MAS against a *memoryless* fragment of SL combined with epistemic modalities was studied. Although a tool was released, memoryless strategies severely constrain the expressivity of the formalism.

To overcome this difficulty, we here put forward a technique for the verification and synthesis of MAS against specifications given in *One-Goal Strategy Logic*, or SL[1G], an expressive variant of SL. We claim there are several advantages of choosing this setting. Firstly, and differently from full SL, strategies in SL[1G] are *behavioural* [25]. A consequence of this is that they can be synthesised automatically, as we show later. Secondly, SL[1G], in the perfect recall setting we here consider, retains considerable expressiveness and is strictly more expressive than any ATL variant, including ATL*. Thirdly, the complexity of the model checking problem is the same as that for ATL*, thereby making its verification attractive.

The rest of the paper is organised as follows. In Section 2 we recall the logic SL[1G], introduce the model checking and synthesis problems and a few related concepts. In Section 3 we put forward practical algorithms for the model checking and synthesis of MAS against SL[1G] specifications. We also show that these are provably optimal when considered against the theoretical complexity known for the problem. In Section 4 we show that the algorithms are amenable to symbolic implementation with binary-decision diagrams and present an experimental model checker implementing the algorithms discussed. We evaluate its performance on the fair process scheduler synthesis. We conclude in Section 5 where we also point to further work.

2 One-Goal Strategy Logic

In this section we introduce some basic concepts and recall SL[1G], a syntactic fragment of SL, introduced in [25].

Underlying Framework. Differently from other treatments of SL, which were originally defined on concurrent game structures, we here use *interpreted systems*, which are commonly used to reason about knowledge and strategies in multi-agent systems [10, 19]. An interpreted system is a tuple $\mathscr{I} = \langle (L_i, Act_i, P_i, t_i)_{i \in Agt}, I, h \rangle$, where each agent $i \in Agt$ is modelled in terms of its set of *local states* L_i, set of *actions* Act_i, *protocol* $P_i : L_i \rightarrow 2^{Act_i}$ specifying what actions can be performed at a given local state, and *evolution function* $t_i : L_i \times Act \rightarrow L_i$ returning the next local state given the current local state and a joint action for all agents.

The set of *global states* G of the whole system consists of tuples of local states for all agents. As a special subset G contains the set I of *initial global* states. The labelling function h maps each *atomic proposition* $p \in AP$ to the set of global states $h(p) \subseteq G$ in which it is true. *Joint actions* Act are tuples of local actions for all the agents in the system; *shared actions* in the set $Act_A \triangleq \bigcap_{i \in A} Act_i$ are actions for the agents $A \subseteq Agt$; The *global protocol* $P : G \rightarrow 2^{Act}$ and *global evolution function* $t : G \times Act \rightarrow G$, which are composed of their local counterparts P_i and t_i, complete the description of the evolution of the entire system.

Syntax of SL[1G]. SL has been introduced as a powerful formalism to reason about sophisticated cooperation concepts in multi-agent systems [24]. Formally, it is defined as a syntactic extension of the logic LTL by means of an *existential strategy quantifier* $\langle\langle x \rangle\rangle \varphi$, a *universal strategy quantifier* $[[x]]\varphi$, and an *agent binding* operator $(a, x)\varphi$. Intuitively, $\langle\langle x \rangle\rangle \varphi$ is read as *"there exists a strategy x such that φ holds"*, $[[x]]\varphi$ is its dual, and $(a, x)\varphi$ stands for *"bind agent a to the strategy associated with the variable x in φ"*. In SL[1G], these three new constructs are merged into one rule $\wp\flat\varphi$, where $\wp$ is a quantification prefix over strategies (e.g. $[\![x]\!]\langle\!\langle y\rangle\!\rangle[\![z]\!]$) and $\flat$ is a binding prefix (e.g. $(a,x)(b,y)(c,x)$). As this limits the use of strategy quantifiers and bindings in SL, SL[1G] is less expressive than SL [25]. Nevertheless, it still strictly subsumes commonly considered logics for strategic reasoning such as ATL*. Additionally, several attractive features of ATL* hold in SL[1G], including the fact that satisfiability and model checking are 2EXPTIME-COMPLETE [25]. Crucially, SL[1G] can be said to be *behavioural*, that is the choice of a strategy for a group of agents at a given state depends only on the history of the game and the actions performed by other agents. This is in contrast with the *non-behavioural* aspects of SL in which strategy choices depend on other agents' actions in the future or in counterfactual games. In summary, SL[1G] is strictly more expressive than ATL*, whereas it retains ATL*'s elementary complexity of key decision problems, including the strategy synthesis problem.

To define formally the syntax of SL[1G], we first introduce the concepts of a quantification and binding prefix [25].

A *quantification prefix* over a set of variables $V \subseteq Var$ is a finite word $\wp \in \{\langle\!\langle x\rangle\!\rangle, [\![x]\!] \mid x \in V\}^{|V|}$ of length $|V|$ such that each variable $x \in V$ occurs in $\wp$ exactly once. $QPre_V$ denotes the set of all quantification prefixes over V. $QPre = \bigcup_{V\subseteq Var} QPre_V$ is the set of all quantification prefixes. A *binding prefix* over a set of variables $V \subseteq Var$ is a finite word $\flat \in \{(i,x) \mid i \in Agt \wedge x \in V\}^{|Agt|}$ of length $|Agt|$ such that each agent $i \in Agt$ occurs in $\flat$ exactly once. $BPre_V$ denotes the set of all binding prefixes over V. $BPre = \bigcup_{V\subseteq Var} BPre_V$ is the set of all binding prefixes. Similarly to first-order languages, we also use $\mathsf{free}(\varphi)$ to represent the *free agents and variables* in a formula φ. Formally, $\mathsf{free}(\varphi) \subseteq Agt \cup Var$ contains *(i)* all agents having no binding after the occurrence of a temporal operator and *(ii)* all variables having a binding but no quantification.

Definition 1 (SL[1G] Syntax)**.** SL[1G] formulas *are built inductively from the set of atomic propositions AP, strategy variables Var, agents Agt, quantification prefixes QPre, and binding prefixes BPre, by using the following grammar, with $p \in AP$, $x \in Var$, $a \in Agt$, $\flat \in BPre$, and $\wp \in QPre$:*

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \wp\flat\varphi$$

where $\wp \in QPre_{\mathsf{free}(\flat\varphi)}$.

The conditions on $\wp$ and $\flat$ ensure that $\wp\flat\varphi$ is an SL[1G] sentence, i.e. , it does not have any free agents or variables.

Semantics of SL[1G]. We assume perfect recall and complete information. So agents have full memory of the past and complete information of the global state they are in. Note that allowing incomplete information would make the logic undecidable [9], whereas SL[1G] with incomplete information and imperfect recall is equivalent to a proper fragment of the logic SLK already studied in [4].

To establish the truth of a formula, the set of strategies over which a variable can range needs to be determined. For this purpose we introduce the set $\mathsf{sharing}(\varphi,x)$ representing the agents sharing the variable x within the formula φ. Also, we make use of the general concepts of *path*, *track*, *play*, *strategy*, and *assignment* for agents and variables. We refer to [25] for a detailed presentation. Intuitively, a strategy identifies paths in the model on which a formula needs to be checked. So, for each track (i.e. a finite prefix of a path), a strategy determines which action has to be performed by a variable, possibly shared by a set of agents. More formally, given an SL[1G] formula φ, for each variable x in φ, the strategy $f : Trk \rightharpoonup Act_{\mathsf{sharing}(\varphi,x)}$ determines the action to be taken by agents in $\mathsf{sharing}(\varphi,x)$.

Given an interpreted system $\mathcal{I}$ having a set of global states G, a global state $g \in G$, and an assignment χ defined on $\mathsf{free}(\varphi)$, we write $\mathcal{I},\chi,g \models \varphi$ to represent that the SL[1G] formula φ holds at g in $\mathcal{I}$ under χ. The satisfaction relation for SL[1G] formulas is inductively defined by using the usual LTL interpretation for the atomic propositions, the Boolean connectives $\neg$ and $\wedge$, as well as the temporal operators X, F, G, and U. The inductive cases for the strategy quantification $\langle\langle x\rangle\rangle$ and the agent binding (a,x) are given as follows. The cases for universal quantification $[[x]]$ are omitted as they can be given as the dual of the existential ones.

- $\mathscr{I},\chi,g \models \langle\langle x\rangle\rangle\varphi$ iff there is a strategy f for the agents in $\mathsf{sharing}(\varphi,x)$ such that $\mathscr{I},\chi[x\mapsto f],g \models \varphi$ where $\chi[x\mapsto f]$ is the assignment equal to χ except for the variable x, for which it assumes the value f.
- $\mathscr{I},\chi,g \models (x,a)\varphi$ iff $\mathscr{I},\chi[a\mapsto\chi(x)],g \models \varphi$, where $\chi[a\mapsto\chi(x)]$ denotes the assignment χ in which agent a is bound to the strategy $\chi(x)$.

Model Checking and Strategy Synthesis. The model checking problem is about deciding whether an SL[1G] formula holds in a certain model. Precisely, given an interpreted system $\mathscr{I}$, an initial global state g_0, an SL[1G] formula φ and an assignment χ defined on $\mathsf{free}(\varphi)$, the *model checking problem* concerns determining whether $\mathscr{I},\chi,g_0 \models \varphi$.

Synthesis can be further used as a witness for the model checking problem as it allows to construct the strategies the agents need to perform to make the formula true. This amounts to deciding which action has to be taken by each shared variable. More formally, let $\mathscr{I}$ be an interpreted system and φ an SL[1G] formula. W.l.o.g., assume φ to be a so called *principal sentence*[1] of the form $\wp\flat\psi$, with $\wp \in QPre_{\mathsf{free}(\flat\psi)}$, $\wp = \wp(0)\cdot\wp(1)\cdots\wp(|\wp|-1)$, and $\flat \in BPre$. Additionally assume that there exists an integer $0 \le k < |\wp|$ such that for each $0 \le j < k$ there exists a strategy f_j for variable $\wp_v(j)$ shared by agents in $\mathsf{sharing}(\flat\psi,\wp_v(j))$. Then, *strategy synthesis* amounts to defining the strategy $f_k : Trk \rightharpoonup Act_{\mathsf{sharing}(\flat\psi,\wp_v(k))}$ for variable $\wp_v(k)$ such that if $\mathscr{I},\chi,g \models \wp_{\ge k}\flat\psi$, then $\mathscr{I},\chi[\wp_v(k)\mapsto f_k],g \models \wp_{>k}\flat\psi$, where χ is an assignment defined on $\{\wp_v(j) \mid 0 \le j < k\}$ such that for all $0 \le j < k$ we have $\chi(\wp_v(j)) \triangleq f_j$, $\wp_{\ge k} \triangleq \wp(k)\cdots\wp(|\wp|-1)$, and $\wp_{>k} \triangleq \wp(k+1)\cdots\wp(|\wp|-1)$.

3 Symbolic Model Checking SL[1G]

We now introduce a novel algorithm for model checking an interpreted system $\mathscr{I}$ against an arbitrary SL[1G] sentence φ. For simplicity we assume that φ is a *principal sentence* of the form $\wp\flat\psi$.

Our aim is to find the set of all global reachable states $\|\varphi\|_{\mathscr{I}} \subseteq G$ at which the SL[1G] sentence φ holds, i.e. $\|\varphi\|_{\mathscr{I}} \triangleq \{g \in G \mid \mathscr{I},\emptyset,g \models \varphi\}$. We proceed in a recursive manner over the structure of φ: According to SL[1G] syntax, ψ is a formula which combines atoms AP and *direct principal subsentences* of the form $\varphi' = \wp'\flat'\psi'$ using only Boolean and temporal connectives. Since φ' is also an SL[1G] principal sentence, we can recursively calculate $\|\varphi'\|_{\mathscr{I}}$; then replace φ' in φ with a new atom $p_{\varphi'} \in AP$; and finally update the assignment with $h(p_{\varphi'}) \triangleq \|\varphi'\|_{\mathscr{I}}$. This allows us to consider the simpler problem of model checking an SL[1G] *basic principal sentence* $\varphi = \wp\flat\psi$ where ψ is an LTL formula. Our general procedure is as follows:

1. We construct a *deterministic parity automaton* $\mathfrak{P}^{\psi}_{\mathscr{I}}$ equivalent to the LTL formula ψ.
2. We construct a two-player *formula arena* $\mathscr{A}^{\wp\flat}_{\mathscr{I}}$ representing the global state space G and the interdependency of strategies in the prefix $\wp\flat$.
3. We combine $\mathscr{A}^{\wp\flat}_{\mathscr{I}}$ and $\mathfrak{P}^{\psi}_{\mathscr{I}}$ into an infinite two-player *parity game* $\mathfrak{G}^{\wp\flat\psi}_{\mathscr{I}}$. Solving the parity game yields its winning regions and strategies, which can in turn be used to calculate $\|\varphi\|_{\mathscr{I}}$ and the strategies in $\wp$.

[1] If this is not the case, one can simply add one quantifier and agent binding for each agent without changing the semantics as φ is a sentence.

We shall now expand on each of the steps above.

Formula automaton. The first step of our algorithm is the standard construction of a deterministic parity automaton $\mathfrak{P}^{\psi}_{\mathscr{I}}$ equivalent to the underlying LTL formula ψ. This is usually performed in three steps: *(i)* ψ is converted to a non-deterministic generalised Büchi automaton $\mathfrak{A}^{\psi}_{\mathscr{I}}$ via standard translation [29]; *(ii)* $\mathfrak{A}^{\psi}_{\mathscr{I}}$ is translated to an equivalent non-deterministic Büchi automaton $\mathfrak{B}^{\psi}_{\mathscr{I}}$ by adding a counter for fairness constraints [29]; *(iii)* $\mathfrak{B}^{\psi}_{\mathscr{I}}$ is transformed into a deterministic parity automaton $\mathfrak{P}^{\psi}_{\mathscr{I}} = (S, s_I, \delta, c)$ with a non-empty set of states S, an initial state $s_I \in S$, a transition function $\delta : S \times G \to S$, and a colouring function $c : S \to \mathbb{N}$. While the third step is typically done using Safra's construction [28], we perform the determinisation using a recently put forward procedure [26] instead, which is amenable to a symbolic implementation. It is worth pointing out that the recursive step (replacing direct principal subsentences φ' with atoms $p_{\varphi'}$) can be incorporated as an extra case of the standard translation in the first step.

As an example, consider the simple interpreted system $\mathscr{I}_{\mathrm{RPS}}$ in Figure 1a with agents $Agt \triangleq \{1,2\}$ representing the *Rock-Paper-Scissors game*. The global states of the system are $G \triangleq \{g_{\mathrm{g}}, g_1, g_2\}$ meaning "game", "player 1 won", and "player 2 won", respectively. The actions available to both players are: $Act_1 \triangleq Act_2 \triangleq \{\mathrm{r}, \mathrm{p}, \mathrm{s}, \mathrm{i}\}$ meaning "rock", "paper", "scissors", and "idle". Finally, the atoms p_1 and p_2 encode that player 1 and player 2 won, respectively. The assignment is defined as $h(p_1) \triangleq \{g_1\}$ and $h(p_2) \triangleq \{g_2\}$. Furthermore, consider the SL[1G] basic principal sentence $\gamma \triangleq [\![x]\!]\langle\!\langle y\rangle\!\rangle(1,x)(2,y)\,\mathsf{G}\,[\neg p_1 \wedge \neg p_2]$ which expresses that *"Whichever action player 1 performs, there exists an action for player 2 such that neither player will ever win"*. The corresponding deterministic parity automaton $\mathfrak{P}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$ constructed using the three-step procedure described in the previous paragraph is shown in Figure 1b.

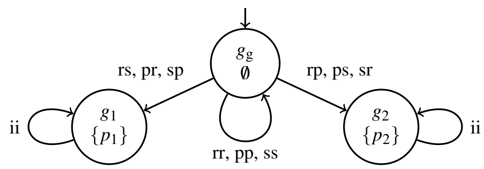
Formula arena. The second step of the algorithm involves building a two-player formula arena $\mathscr{A}^{\wp\flat}_{\mathscr{I}} = (V_0, V_1, E)$, which encodes the state space of the interpreted system $\mathscr{I}$ and the interdependency of strategies in the prefix $\wp\flat$. The vertices V of $\mathscr{A}^{\wp\flat}_{\mathscr{I}}$ are pairs $(g,d) \in G \times Dec^{\wp\flat}_{\mathscr{I}}$ of global reachable states and lists of actions such that for all $0 \le k < |d|$ we have $d(k) \in \bigcap_{i \in \mathsf{sharing}(\flat\top, \wp_v(k))} P_i(l_i(g))$, where $Dec^{\wp\flat}_{\mathscr{I}} \triangleq \bigcup_{\ell=0}^{|\wp|} \prod_{k=0}^{\ell-1} Act_{\mathsf{sharing}(\flat\top, \wp_v(k))}$ and $l_i(g)$ is the local state of agent i in g. The existential player vertices $V_0 \subseteq V$ are vertices $(g,d) \in V$ such that $|d| < |\wp|$ and $\wp(|d|)$ is an existential strategy quantifier. Conversely, the universal player vertices are $V_1 = V \setminus V_0$. The edge relation $E \subseteq V \times V$ is defined as:

$$\begin{aligned} E \triangleq & \{((g,d),(g,d\cdot a)) \in V \times V \mid |d| < |\wp|\} \cup \\ & \left\{\left((g,d),\left(t(g,d^{Act}),[]\right)\right) \in V \times V \,\middle|\, |d| = |\wp|\right\} \end{aligned}$$

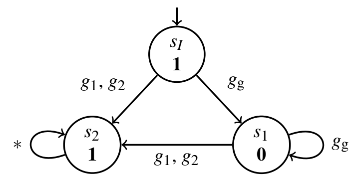
where $d^{Act} \in Act$ is a joint action such that for all $0 \le k < |\wp|$ and $i \in \mathsf{sharing}(\flat\top, \wp_v(k))$ we have $\mathsf{act}_i(d^{Act}) = d(k)$.

Intuitively, the existential (universal) player represents all existential (universal) quantifiers in the quantification prefix $\wp$. Equivalently, the two players correspond to the existential-universal partition of Agt. The game starts in some vertex $(g,[])$. The players take turns to select actions $d(0), \ldots, d(|\wp|-1)$ for the quantifiers $\wp(0), \ldots, \wp(|\wp|-1)$. The decision d then determines the joint action of all agents d^{Act} and a temporal transition to $\left(t(g,d^{Act}),[]\right)$ is performed. This pattern is repeated forever. The formula arena $\mathscr{A}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$ of the Rock-Paper-Scissors game interpreted system $\mathscr{I}_{\mathrm{RPS}}$ for the SL[1G] formula γ introduced earlier is shown in Figure 1d. Observe that the three grey blobs in $\mathscr{A}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$ correspond to the three global reachable states in Figure 1a.

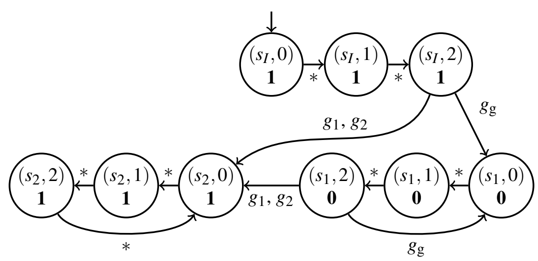
We now consider a pseudo-LTL game $\mathfrak{L}^{\wp\flat\psi}_{\mathscr{I}}$ based on the arena $\mathscr{A}^{\wp\flat}_{\mathscr{I}}$. We define an infinite path $\pi \in V^{\omega}$ in $\mathfrak{L}^{\wp\flat\psi}_{\mathscr{I}}$ to be winning for the existential player iff the LTL formula ψ holds along the underlying infinite path $\pi_{\mathscr{I}} \in G^{\omega}$ in $\mathscr{I}$.



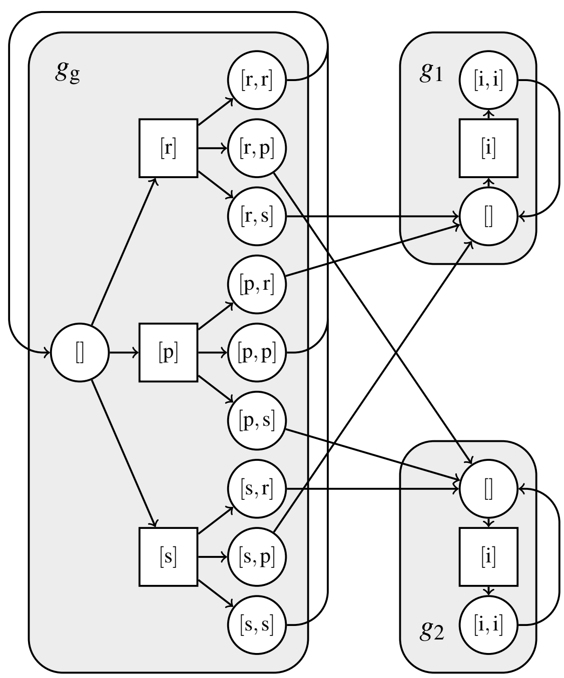
(a) Interpreted system $\mathscr{I}_{\mathrm{RPS}}$.



(b) Deterministic parity automaton $\mathfrak{P}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$.



(c) Delayed automaton $\mathfrak{D}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$.



(d) Formula arena $\mathscr{A}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$. The existential and universal player states are represented by squares and circles respectively.

Figure 1: Interpreted system $\mathscr{I}_{\mathrm{RPS}}$, parity automaton $\mathfrak{P}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$, delayed automaton $\mathfrak{D}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$, and formula arena $\mathscr{A}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$ of the Rock-Paper-Scissors game and the SL[1G] basic principal sentence $\gamma \triangleq [\![x]\!]\langle\!\langle y\rangle\!\rangle(1,x)(2,y)\,\mathsf{G}\,[\neg p_1 \wedge \neg p_2]$.

Lemma 1. *An* SL[1G] *principal sentence* $\wp\flat\psi$ *holds at a global state* $g \in G$ *in an interpreted system* $\mathscr{I}$ *iff the vertex* $(g, []) \in V$ *is winning for the existential player in the pseudo-*LTL *game* $\mathfrak{L}^{\wp\flat\psi}_{\mathscr{I}}$ *defined above.*

Proof. This follows from the fact that SL[1G] model checking can be reduced to solving a so-called *dependence-vs-valuation game* [25] in which the existential player chooses a specific dependence map $\theta : \left(\llbracket\wp\rrbracket \to \bigcup_{i\in Agt} Act_i\right) \to \left(\wp \to \bigcup_{i\in Agt} Act_i\right)$ for $\wp$ over actions in the current global state $g \in G$ and then the universal player chooses a valuation $v : \llbracket\wp\rrbracket \to \bigcup_{i\in Agt} Act_i$. The combination $\theta(v) : \wp \to \bigcup_{i\in Agt} Act_i$ assigns actions to all variables and determines the next state $g' \in G$. Instead of choosing the whole dependence map and valuation at once, the players in $\mathfrak{L}^{\wp\flat\psi}_{\mathscr{I}}$ assign actions to strategies one by one for each quantifier. Furthermore, the order of the players' moves in $\mathfrak{L}^{\wp\flat\psi}_{\mathscr{I}}$ game ensures that the independence constraints of θ are satisfied. Hence, our claim follows. □

We shall next explain how this pseudo-LTL game can be converted to a standard parity game.

Combined game. In order to construct the combined parity game, the solving of which is equivalent to model checking the basic principal sentence $\wp\flat\psi$, we need to combine the formula automaton $\mathfrak{P}^{\psi}_{\mathscr{I}}$ and the formula arena $\mathscr{A}^{\wp\flat}_{\mathscr{I}}$ because $\mathfrak{P}^{\psi}_{\mathscr{I}}$ represents the winning condition of the pseudo-LTL game $\mathfrak{L}^{\wp\flat\psi}_{\mathscr{I}}$. However, we cannot simply take their product because, informally, they work at different, albeit constant, "speeds". While $\mathfrak{P}^{\psi}_{\mathscr{I}}$ performs a temporal transition at every step, it takes exactly $|\wp|+1$ turns before a different underlying global state (grey blob in Figure 1d) is reached by $\mathscr{A}^{\wp\flat}_{\mathscr{I}}$. To cater for this asynchrony, we can make the parity automaton "wait" for $|\wp|+1$ steps before each actual transition. We do this by extending the state of $\mathfrak{P}^{\psi}_{\mathscr{I}}$ with a simple counter from 0 to $|\wp|$. The resulting delayed (deterministic parity) automaton $\mathfrak{D}^{\gamma}_{\mathscr{I}_{\mathrm{RPS}}}$ for the basic principal sentence γ introduced earlier is shown in Figure 1c.

The delayed automaton $\mathfrak{D}^{\wp\psi}_{\mathscr{I}}$ accepts precisely those paths in the formula arena $\mathscr{A}^{\wp\flat}_{\mathscr{I}}$ which are winning for the existential player. Hence, by combining the two structures, we obtain the combined parity game $\mathfrak{G}^{\wp\flat\psi}_{\mathscr{I}} \triangleq ((V_0 \times S, V_1 \times S, E_{\mathfrak{G}}), c_{\mathfrak{G}})$ with edge relation and colouring function defined as $E_{\mathfrak{G}} \triangleq \{((g,d,s),(g',d',s')) \in (V \times S) \times (V \times S) \mid E((g,d),(g,d)) \wedge \delta_{\mathfrak{D}}((s,|d|),g)\}$ and $c_{\mathfrak{G}}((g,d,s)) \triangleq c(s)$ respectively, where $\delta_{\mathfrak{D}}$ is the transition function of the delayed automaton.

Model Checking. Model checking of an SL[1G] principal sentence can finally be performed by solving the corresponding combined parity game (e.g. using Zielonka's algorithm [30]) as formalised by the following lemma:

Lemma 2. *Let* $\wp\flat\psi$ *be an* SL[1G] *principal sentence,* $g \in G$ *a global state in an interpreted system* $\mathscr{I}$*, and* (W_0, W_1) *the winning regions of the combined parity game* $\mathfrak{G}^{\wp\flat\psi}_{\mathscr{I}}$*.* $\wp\flat\psi$ *holds at g (i.e.* $\mathscr{I}, \emptyset, g \models \wp\flat\psi$*) iff the vertex* $(g, [], s_I)$ *is in the winning region of the existential player (i.e.* $(g, [], s_I) \in W_0$*).*

Proof. Our claim follows directly from Lemma 1 and the correctness of the determinisation procedure. □

Strategy Synthesis. The formula arena encodes the effects and interdependency of agents' actions. Therefore, the solution, i.e., the winning strategies, of the combined parity game can be used for strategy synthesis.

Lemma 3. *Let* $\wp\flat\psi$ *be an* SL[1G] *principal sentence,* $\mathscr{I}$ *an interpreted system,* (w_0, w_1) *the winning strategies of the combined parity game* $\mathfrak{G}^{\wp\psi\flat}_{\mathscr{I}}$*,* $0 \le k < |\wp|$ *an integer, and* $f_0, \ldots, f_{k-1}$ *strategies for variables* $\wp_v(0), \ldots, \wp_v(k-1)$*. Then the strategy* $f_k : Trk \to Act_{\mathsf{sharing}(\flat\psi, \wp_v(k))}$ *is defined for all tracks* $\pi \in Trk$ *implicitly as* $\hat{w}((\mathsf{last}(\pi), [f_0(\pi), \ldots, f_{k-1}(\pi)], \delta(s_I, \pi_{\le|\pi|-2}))) = (\mathsf{last}(\pi), [f_0(\pi), \ldots, f_{k-1}(\pi), \underline{f_k(\pi)}], \delta(s_I,$

$\pi_{\leq|\pi|-2}))$ *where* $\delta(s_I, \pi_{\leq|\pi|-2}) \triangleq \delta(\ldots\delta(s_I, \pi(0))\ldots, \pi(|\pi|-2))$ *and* $\hat{w}: V \to V$ *is a total function such that* $w_0 \cup w_1 \subseteq \hat{w}$.

Proof. The correctness follows from the structure of the formula arena $\mathscr{A}_{\mathscr{I}}^{\flat\wp}$. See [5] for more details. □

Optimality. The theoretical complexity of SL[1G] model checking is 2EXPTIME-COMPLETE with respect to the size of the formula and P-COMPLETE with respect to the size of the model [25]. Given this, we show that our algorithm has *optimal time complexity*:

Lemma 4. *Let* φ *be an arbitrary* SL[1G] *sentence and* $\mathscr{I}$ *an interpreted system. Our algorithm calculates the set of all global states* $\|\varphi\|_{\mathscr{I}} \subseteq G$ *satisfying* φ *in time* $|\mathscr{I}|^{2^{O(|\varphi|)}}$.

Proof. Let us first consider an arbitrary SL[1G] basic principal sentence $\wp\flat\psi$. The automata $\mathfrak{A}_{\mathscr{I}}^{\psi}$, $\mathfrak{B}_{\mathscr{I}}^{\psi}$, $\mathfrak{P}_{\mathscr{I}}^{\psi}$, and $\mathfrak{D}_{\mathscr{I}}^{\wp\flat\psi}$ have $O(2^{|\psi|})$, $2^{O(|\psi|)}$, $2^{2^{O(|\psi|)}}$, and $|\wp| \times 2^{2^{O(|\psi|)}}$ states. Moreover, both parity automata have $2^{O(|\psi|)}$ colours. The arena $\mathscr{A}_{\mathscr{I}}^{\flat\wp}$ and game $\mathfrak{G}_{\mathscr{I}}^{\wp\flat\psi}$ have $O(|I|^{|\wp|})$ and $|I|^{|\wp|} \times 2^{2^{O(|\psi|)}}$ states. Given the number of states and colours, the game can be solved in time $|I|^{2^{O(|\wp\flat\psi|)}}$ [13].

We model check φ in a recursive bottom-up manner as explained earlier. Hence, at most $|\varphi|$ SL[1G] basic principal sentences of size at most $|\varphi|$ need to be checked. If φ is *not* a principal sentence, it must be a Boolean combination of principal sentences, the results of which we can combine using set operations. Thus, the model checking time is $|\varphi| \times |I|^{2^{O(|\varphi|)}} + |\varphi| \times |\mathscr{I}| = |I|^{2^{O(|\varphi|)}}$ and our claim follows. □

Note that SL[1G] subsumes ATL*, which has the same model checking complexity [17]. Hence, our algorithm is also optimal for ATL* model checking. Moreover, the same complexity result applies to SL[1G] and, consequently, ATL* strategy synthesis.

4 Implementation and Experimental Results

We implemented the algorithm presented in the previous section as part of the new experimental model checker MCMAS-SL[1G]. The tool, available from [20], takes as input the system in the form of an ISPL file [18] describing the agents in the system, their local states, actions, protocols, and evolution functions as well as the SL[1G] specifications to be verified. Upon invocation MCMAS-SL[1G] calculates the set of reachable states, encoded as BDDs, and then checks whether each specification holds in the system. If requested, all quantified strategies in all formulas are synthesised (together with their interdependencies). While MCMAS-SL[1G] is built from the existing open-source model checker MCMAS [18] and shares some of its algorithms, it necessarily differs from MCMAS in several key components, including a more complex encoding of the model, as described in the previous section, as well as the novel procedure for computing the sets of states satisfying SL[1G] formulas.

Evaluation. To evaluate the proposed approach, we present the experimental results obtained for the problem of fair process scheduler synthesis. The experiments were run on an Intel® Core™ i7-3770 CPU 3.40GHz machine with 16GB RAM running Linux kernel version 3.8.0-35-generic. Table 1 reports the performance observed when *synthesising* a process scheduler satisfying the following SL[1G] specification which asserts absence of starvation [23]:

$$\phi \triangleq \xi \bigwedge_{i=1}^{n} \mathsf{G}\left(\langle \mathrm{wt}, i\rangle \to \mathsf{F}\neg\langle \mathrm{wt}, i\rangle\right)$$

algo.	processes n	possible states	reachable states	reachability time	model checking time (s)					memory (MB)
					total	automaton	arena	game	solve	
unoptimised	2	72	9	0.00	0.09	0.00	0.00	0.02	0.06	4.44
	3	432	21	0.00	10.11	0.01	0.02	1.06	9.02	15.14
	4	2592	49	0.01	631.11	0.33	0.26	86.47	544.04	41.80
	5	15552	113	0.02	29593.56	5.86	2.61	4323.81	25261.15	2792.22
	6	93312	257	0.02	*out of memory*					
optimised	2	72	9	0.00	0.12	0.00	0.00	0.02	0.10	4.52
	3	432	21	0.00	6.39	0.00	0.01	0.65	5.72	14.54
	4	2592	49	0.01	338.16	0.00	0.24	23.33	314.57	40.70
	5	15552	113	0.02	6131.43	0.00	2.65	444.06	5684.69	306.41
	6	93312	257	0.02	85976.57	0.00	38.27	8012.11	77925.96	2688.93

Table 1: Verification results for the fair process scheduler synthesis.

where $\xi \triangleq \langle\langle x\rangle\rangle [\![y_1]\!] \cdots [\![y_n]\!] (\mathrm{Sched}, x)(1, y_1) \cdots (n, y_n)$ is a prefix and $\langle \mathrm{wt}, i\rangle$ denotes that process $1 \leq i \leq n$ is waiting for the resource.

We ran the experiments with two different versions of the SL[1G] model checking algorithm: an unoptimised one (described in the previous section) and an optimised one. Given an SL[1G] principal sentence of the form $\wp(\psi_0 \wedge \psi_1 \wedge \cdots \wedge \psi_{n-1})$, the optimised algorithm determinises each conjunct ψ_i with $0 \leq i < n$ separately, i.e. it constructs the delayed automata $\mathfrak{D}^{\wp\psi_0}_{\mathscr{I}}, \mathfrak{D}^{\wp\psi_1}_{\mathscr{I}}, \ldots, \mathfrak{D}^{\wp\psi_{n-1}}_{\mathscr{I}}$. The resulting combined game $\mathfrak{G}^{\wp\psi}_{\mathscr{I}} \triangleq \mathscr{A}^{\wp}_{\mathscr{I}} \times \prod_{i=0}^{n-1} \mathfrak{D}^{\wp\psi_i}_{\mathscr{I}}$ is a *generalised parity game* [7]. The reasoning behind this optimisation is that the size of the deterministic automata is doubly exponential in the size of the LTL formulas. Hence, separate determinisation may lead to much smaller combined games.

In the experiments, MCMAS-SL[1G] synthesised correct strategies using both versions of the algorithm. The results show that the main performance bottlenecks are the construction and solution of the combined parity game; this is in line with the theoretical complexity results reported in the proof of Lemma 4. We can observe that separate determinisation has indeed a significant impact on performance in terms of both time and memory footprint, thereby allowing us to reason about more processes. Note that the relative speedup increases with the number of processes with gains quickly reaching an order of magnitude and more.

We tested the tool on various other scalable scenarios [5]. When verifying a fixed-size formula, the tool has efficiently handled systems with 10^5 reachable global states. This is approximately an order of magnitude worse than the MCMAS's performance on plain ATL specifications. This is because the expressiveness of SL[1G] requires a richer encoding for the models, as discussed earlier. We are not aware of any other tool capable of verifying specifications richer than plain ATL under the assumptions of perfect recall. Therefore, we cannot compare our results to any other in the literature.

5 Conclusions

Most approaches put forward over the past ten years for the verification of MAS are concerned with temporal-epistemic properties so to assess the evolution of the knowledge of the agents over time. Considerably less attention has been devoted so far to the problem of establishing what strategic properties agents in a system have. We are aware of two lines of research concerning this. The first concerns the verification of MAS against ATL specifications [2, 19, 15]; the second pertains to the verification of systems against an observational fragment of SL to which epistemic modalities are added [4, 12]. As argued in the literature, the first line is limited by the fact that ATL specifications are not sufficiently rich to refer to strategies explicitly. The second direction suffers from the weakness of the observational fragments analysed as they cannot account for the perfect recall abilities normally assumed in a strategic setting.

In this paper we attempted to overcome both difficulties above and put forward a fully symbolic approach to the verification of MAS against specifications in SL[1G], a rich behaviourally fragment of SL. We showed the algorithm developed is provably optimal and built a BDD-based checker to support it. The experimental results obtained point to the feasibility of the practical verification problem for MAS against SL[1G] specifications. Since SL[1G] strictly subsumes ATL*, an important byproduct of the work presented is the fact that it also constitutes the first verification toolkit for ATL*. A further key innovative feature of our approach is that it does not only support verification, but also strategy synthesis. This enables us to use the synthesis engine for developing controllers or automatic planners in a MAS context. We leave this to further work.

Acknowledgments. The research described in this paper was partly supported by the EPSRC Research Project "Trusted Autonomous Systems" (grant EP/I00520X/1) and FP7 EU project 600958-SHERPA).

References

[1] R. Alur, T. A. Henzinger & O. Kupferman (2002): *Alternating-Time Temporal Logic*. Journal of the ACM 49(5), pp. 672–713.

[2] Rajeev Alur, Luca de Alfaro, Radu Grosu, Thomas A Henzinger, Minsu Kang, Christoph M Kirsch, Rupak Majumdar, Freddy Mang & Bow-Yaw Wang (2001): *jMocha: A model checking tool that exploits design structure*. In: Proceedings of the 23rd International Conference on Software Engineering (ICSE'01), IEEE, pp. 835–836.

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

[4] P. Čermák, A. Lomuscio, F. Mogavero & A. Murano (2014): *MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications*. In: Proceedings of the 26th International Conference on Computer Aided Verification (CAV'14), LNCS 8559, Springer, pp. 524–531.

[5] Petr Čermák (2014): *A Model Checker for Strategy Logic*. Master's thesis, Department of Computing, Imperial College London, UK.

[6] Petr Cermák, Alessio Lomuscio & Aniello Murano (2015): *Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications*. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA., AAAI Press, pp. 2038–2044.

[7] K. Chatterjee, T. Henzinger & N. Piterman (2006): *Generalized Parity Games*. Technical Report UCB/EECS-2006-144, University of California, Berkeley. Available at `http://chess.eecs.berkeley.edu/pubs/318.html`.

[8] Krishnendu Chatterjee, Thomas A. Henzinger & Nir Piterman (2010): *Strategy logic*. Inf. Comput. 208(6), pp. 677–693.

[9] Catalin Dima & Ferucio Laurentiu Tiplea (2011): *Model-checking ATL under imperfect information and perfect recall semantics is undecidable*. CoRR abs/1102.4225.

[10] R. Fagin, J. Y. Halpern, Y. Moses & M. Y. Vardi (1995): *Reasoning about Knowledge*. MIT Press, Cambridge.

[11] P. Gammie & R. van der Meyden (2004): *MCK: Model Checking the Logic of Knowledge*. In: Proceedings of 16th International Conference on Computer Aided Verification (CAV'04), LNCS 3114, Springer, pp. 479–483.

[12] X. Huang & R. van der Meyden (2014): *Symbolic Model Checking Epistemic Strategy Logic*. In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*, AAAI Press, pp. 1426–1432.

[13] M. Jurdziński (2000): *Small Progress Measures for Solving Parity Games*. In: *Proceedings of the 17th Symposium on Theoretical Aspects of Computer Science (STACS'00)*, LNCS 1770, Springer, pp. 290–301.

[14] M. Kacprzak, W. Nabialek, A. Niewiadomski, W. Penczek, A. Pólrola, M. Szreter, B. Wozna & A. Zbrzezny (2008): *VerICS 2007 - a Model Checker for Knowledge and Real-Time*. *Fundamenta Informaticae* 85(1-4), pp. 313–328.

[15] M. Kacprzak & W. Penczek (2005): *Fully Symbolic Unbounded Model Checking for Alternating-time Temporal Logic*. *Autonomous Agents and Multi-Agent Systems* 11(1), pp. 69–89.

[16] P. Kouvaros & A. Lomuscio (2013): *A Cutoff Technique for the Verification of Parameterised Interpreted Systems with Parameterised Environments*. In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, AAAI Press, pp. 2013–2019.

[17] F. Laroussinie, N. Markey & G. Oreiby (2007): *On the Expressiveness and Complexity of ATL*. In: *Proceedings of the 10th International Conference on Foundations of Software Science and Computational Structures (FOSSACS'07)*, LNCS 4423, Springer, pp. 243–257.

[18] A. Lomuscio, H. Qu & F. Raimondi (2009): *MCMAS: A Model Checker for the Verification of Multi-Agent Systems*. In: *Proceedings of the 21th International Conference on Computer Aided Verification (CAV'09)*, LNCS 5643, Springer, pp. 682–688.

[19] A. Lomuscio & F. Raimondi (2006): *Model checking knowledge, strategies, and games in multi-agent systems*. In: *Proceedings of the 5th international joint conference on Autonomous agents and multiagent systems (AAMAS'06)*, ACM Press, pp. 161–168.

[20] *MCMAS-SL[1G] – A Model Checker for the Verification of One-Goal Strategy Logic Specifications*. Http://vas.doc.ic.ac.uk/software/tools/, 2014.

[21] A. Meski, W. Penczek, M. Szreter, B. Wozna-Szczesniak & A. Zbrzezny (2014): *BDD versus SAT-based bounded model checking for the existential fragment of linear temporal logic with knowledge: algorithms and their performance*. *Autonomous Agents and Multi-Agent Systems* 28(4), pp. 558–604.

[22] R. van der Meyden & H. Shilov (1999): *Model checking knowledge and time in systems with perfect recall*. In: *Proceedings of the 19th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'99)*, LNCS 1738, Hyderabad, India, pp. 432–445.

[23] F. Mogavero, A. Murano & L. Sauro (2013): *On the Boundary of Behavioral Strategies*. In: *Proceedings of the 28th Annual IEEE/ACM Symposium on Logic in Computer Science (LICS'13)*, IEEE, pp. 263–272.

[24] F. Mogavero, A. Murano & M. Y. Vardi (2010): *Reasoning About Strategies*. In: *Proceedings of the 30th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'10)*, LIPIcs 8, pp. 133–144.

[25] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y Vardi (2014): *Reasoning About Strategies: On the Model-Checking Problem*. *ACM Trans. Comput. Logic* 15(4), pp. 1–47.

[26] A. Morgenstern & K. Schneider (2008): *From LTL to Symbolically Represented Deterministic Automata*. In: *Proceedings of the 9th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'08)*, LNCS 4905, Springer, pp. 279–293.

[27] F. Raimondi & A. Lomuscio (2005): *Automatic verification of multi-agent systems by model checking via OBDDs*. *Journal of Applied Logic* Vol 5(2), pp. 235–251.

[28] S. Safra (1988): *On the complexity of ω-automata*. In: *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pp. 319–327.

[29] K. Schneider (2004): *Verification of Reactive Systems*. Texts in Theoretical Computer Science, Springer.

[30] W. Zielonka (1998): *Infinite games on finitely coloured graphs with applications to automata on infinite trees*. *Theoretical Computer Science* 200(1–2), pp. 135–183.

9

Regular Paper

P-Automata for Markov Decision Processes

Souymodip Chakraborty, Joost-Pieter Katoen
RWTH Aachen University, 52056 Aachen, Germany

P-automata provide an automata-theoretic approach to probabilistic verification. Similar to alternating tree automata accepting labelled transition systems, p-automata accept labelled Markov chains (MCs). This paper proposes an extension of p-automata that accept the set of all MCs (modulo bisimulation) obtained from a Markov decision process under its schedulers.

1 Introduction

Model checking of μ-calculus [**?**] formulas on a Kripke structure (or labelled translation system) is a well studied method for verifying the correctness of discrete state systems [**?**]. The problem entails whether every execution (infinite tree) of a Kripke structure satisfies a given μ-calculus formula. The satisfiability problem for μ-calculus, on the other hand, is to decide whether there exists an infinite tree which satisfies a given μ-calculus formula. Both these problems are algorithmically feasible, and the key method is the translation to alternating tree automata [**?**].

The notion of p-automata was introduced in [**?**] to provide a similar automata-theoretical foundation for the verification of probabilistic systems as alternating tree automata provide for Kripke structures. As alternating tree automata describe a complete framework for abstraction with respect to branching-time logic like, μ-calculus, CTL and CTL* [**?**], p-automata similarly give a unifying framework for different probabilistic logics.

Every p-automaton defines a set of labeled Markov chains, that is, a p-automaton reads an entire Markov chain as input and it either accepts the Markov chain or rejects it. Analogous to alternating tree automata where acceptance of a Kripke structure is decided by solving 2-player games [**?**], the acceptance of a labelled Markov chain by a p-automaton is decided by solving *stochastic* 2-player games. In this paper we revisit p-automata defined by [**?**] and extend it with a new construct for representing Markov decision processes. We view a Markov decision process (MDP) as a set of Markov chains defined by different schedulers and use the extended p-automata to represent this set. Modeling MDPs as p-automata allows us to define a automata theoretical framework for abstraction of MDPs.

The main contribution of this paper is as follows: We extend the p-automata with a construct that captures the non-determinism in the choice of probability distribution. This allows us to model Markov decision processes as p-automata. We show that the extended p-automata are closed under bisimulation, union and intersection, (though, in contrast to [**?**], the language is no longer closed under negation). We show that the language of the p-automaton obtained from an MDP accepts exactly those Markov chains that are bisimilar to the Markov chains induced by the schedulers of the MDP. In the rest of the paper, when referring to p-automata we will assume the extended p-automata (as defined in Definition **??**), unless the contrary is stated explicitly.

The paper is organised as follows. In Section 2, we mention some important definitions and preliminaries. In section 3 and 4, we introduce the p-automata and define the acceptance game. In Section 5, we describe the embedding of an MDP as a p-automaton and conclude in Section 6. Details of some of the proofs are present in the appendix.

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

2 Preliminaries and definitions

Let X^Y be the set of functions from the set Y to the set X. For $\varphi \in X^Y$ let $\mathsf{img}(\varphi) \subseteq X$ be the image and $\mathsf{dom}(\varphi) = Y$ be the domain of φ. The set of probability distributions over set X is denoted by $\mathscr{D}_X$ where $\boldsymbol{d} \in \mathscr{D}_X$ iff $\boldsymbol{d} \in \mathbb{R}_+^X$ and $\boldsymbol{d}^T \cdot \boldsymbol{1} = 1$ ($\mathbb{R}_+$ is the set of non-negative reals). For $\mu \in \mathscr{D}_X$, let $\mathsf{supp}(\mu) = \{x \in X \mid \mu(x) > 0\}$ be the support of distribution μ.

Definition 1. *A Markov chain (MC) M is a quintuple (S, P, AP, L, s_{in}) where S is a (countable) set of states, $P(s) \in \mathscr{D}_S$ for all $s \in S$, AP is a set of atomic propositions, $L : S \to 2^{AP}$ is a labeling function, and $s_{in} \in S$ is the initial state (Figure **??**).*

An *infinite path* σ through MC M is a sequence of states $\sigma = \{\sigma_i\}_{i \geq 0}$, where for all $i \geq 0$, $P(\sigma_i, \sigma_{i+1}) > 0$. Let $\mathsf{path}(s)$ denote the set of (finite or infinite) paths starting from state s. For a path σ, let $\sigma{\downarrow}$ denote the last state of σ if this exists (i.e., if σ is finite) and $|\sigma|$ denote the length of σ. Let $\mathsf{succ}(s) = \{t \mid P(s,t) > 0\}$ be the successors of state s. A probability measure on sets of infinite paths is obtained in a standard way. Let $(\Omega_s, \mathscr{F}, \mathrm{Pr})$ be the Borel σ-algebra where Ω_s is the set of infinite paths from state s, $\mathscr{F}$ is the smallest σ-field on cylinder sets of Ω_s, and Pr is the probability measure on $\mathscr{F}$, for a finite path σ, $\mathrm{Pr}(\sigma) = \prod_{0 < i \leq |\sigma|} P(\sigma_{i-1}, \sigma_i)$ [**?**].

Definition 2. *A* Markov decision process *(MDP) D is a quintuple $(S, \Delta, AP, L, s_{in})$ where S, AP, L, and s_{in} are as before, and $\Delta : S \to 2^{\mathscr{D}_S}$ such that $\Delta(s)$ is a finite set of distributions. (Figure **??**) We assume S and $\Delta(s)$ for each $s \in S$ to be finite (unless the contrary is explicitly specified).*

A finite path of an MDP is a sequence of states $\sigma = \sigma_0 \ldots \sigma_n$ such for each $0 < i \leq n$ $\sigma_i \in \mathsf{supp}(\mu)$ for some $\mu \in \Delta(\sigma_{i-1})$. Let $\mathsf{path}(s)$ be the set of (finite and infinite) paths from the state s. Let $\mathsf{succ}(s) = \{t \mid t \in \bigcup_{\mu \in \Delta(s)} \mathsf{supp}(\mu)\}$ be the set of successors of s. As usual, we use *schedulers* to resolve the possible non-determinism in a state.

Definition 3. *A* scheduler *of MDP $D = (S, \Delta, AP, L, s_{in})$ is a function $\eta : S^+ \to \mathscr{D}_{\mathscr{D}_S}$ with $\eta(\sigma) \in \mathscr{D}_{\Delta(\sigma\downarrow)}$. The scheduler η induces the MC $D_\eta = (S^+, P, AP, L', s_{in})$ with $L'(\sigma) = L(\sigma{\downarrow})$, and $P(\sigma, \sigma \cdot t) = \sum_{\mu \in \Delta(\sigma\downarrow)} \eta(\sigma)(\mu) \cdot \mu(t)$.*

These schedulers are history-dependent and randomized. Let $\mathsf{HR}(D)$ denote the set of history-dependent randomized schedulers of MDP D.

Definition 4. *Let MC $M = (S, P, AP, L, s_{in})$. The equivalence relation $\mathscr{R} \subseteq S \times S$ is a* probabilistic bisimulation *[**?**] iff for every $(s, s') \in \mathscr{R}$ it holds:*

1. *$L(s) = L(s')$, and*
2. *for every $C \in S/\mathscr{R}$, we have $\sum_{t \in C} P(s,t) = \sum_{t' \in C} P(s', t')$.*

Let $\sim$ denote the largest probabilistic bisimulation on S. The MCs M_1 and M_2 are probabilistically bisimilar, denoted $M_1 \sim M_2$, if $s_{in}^1 \sim s_{in}^2$ in the disjoint union of M_1 and M_2.

Definition 5. *A stochastic game G is a tuple $(V, E, V_0, V_1, V_p, P, \Omega)$, where (V, E) is a finite directed graph and (V_0, V_1, V_p) is a partition of V. V_0 is the set of Player 0 configurations, V_1 is the set of Player 1 configurations and V_p is the set of stochastic (or probabilistic) configurations. P is a probability transition function $P : V_p \to \mathscr{D}_V$ and $\Omega \subseteq V$ is a set of accepting configurations. A path (also called a play) in the graph (V, E) is winning for Player 0 if it is finite and ends in Player 1 configuration, or it is infinite and ends in a suffix of configurations in Ω. Otherwise, that play is winning for Player 1.*

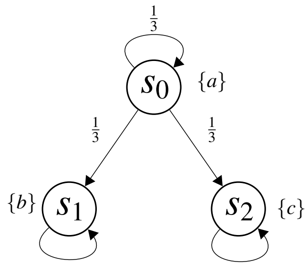


Figure 1: A Markov chain M, with $S = \{s_0, s_1, s_2\}$ $P(s_0,s_0) = P(s_0,s_1) = P(s_0,s_2) = \frac{1}{3}$, $P(s_1,s_1) = P(s_2,s_2) = 1$.

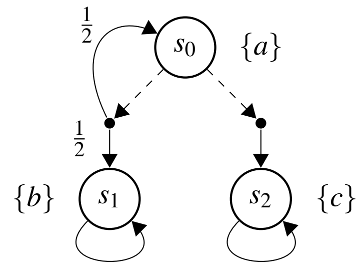


Figure 2: A Markov decision process D. $\Delta(s_0) = \{\mu_1, \mu_2\}$, where $\mu_1(s_0) = \frac{1}{2}, \mu_1(s_1) = \frac{1}{2}$ and $\mu_2(s_2) = 1$.

A stochastic game is called a *weak stochastic game* iff for all maximal connected components (MSCC) C in (V,E), either $C \subseteq \Omega$ or $C \cap \Omega = \emptyset$. On the other hand, if $V_p = \emptyset$ then it is called a *weak game*. A *strategy* of a Player 0 is a function $\gamma : V^* \times V_0 \to \mathscr{D}_V$, with $\gamma(w \cdot u)(v) > 0$ implies $(u,v) \in E$. A play $w = v_0 v_1 \ldots$ is consistent with strategy γ if for every $i \geq 0$, $v_i \in V_0$ implies $\gamma(v_0 \ldots v_i)(v_{i+1}) > 0$. Strategies of Player 1 are defined similarly. Let Υ and Π be the set of all strategies for Player 0 and Player 1, respectively. A player 0 strategy γ is memoryless iff $\gamma(w \cdot v) = \gamma(w' \cdot v)$, for any $w, w' \in V^*$, and it pure iff $\gamma : V^* \times V_0 \to V$, (similarly definitions applies to strategies of player 1).

A pair of strategies $(\gamma, \pi) \in \Upsilon \times \Pi$ of a game G determines an MC $M^{\gamma,\pi}$ (configurations without an out-going transition are made absorbing) whose paths are plays of G according to γ, π. The measure of the set of winning plays of Player 0 starting from a configuration c in $M^{\gamma,\pi}$ is denoted by $\mathsf{val}_0^{\gamma,\pi}(c)$. We have $\mathsf{val}_1^{\gamma,\pi}(c) = 1 - \mathsf{val}_0^{\gamma,\pi}(c)$. The $\mathsf{val}_0(c) = \sup_{\gamma \in \Upsilon} \inf_{\pi \in \Pi} \mathsf{val}_0^{\gamma,\pi}(c)$ and $\mathsf{val}_1(c) = \sup_{\pi \in \Pi} \inf_{\gamma \in \Upsilon} \mathsf{val}_1^{\gamma,\pi}(c)$. If a strategy achieves these values then it is called *optimal*.

Theorem 1. *[**?**, **?**, **?**] Let G be a stochastic game and c be one of its configurations. Then G is determined, that is $\mathsf{val}_0(c) + \mathsf{val}_1(c) = 1$. If G is finite and weak, then optimal strategies for both players exist and they are memoryless and pure. If G is a stochastic weak game, then the problem whether $\mathsf{val}_0(c)$ greater than a given quantity $v \in \mathbb{Q}$ can be decided in $\mathsf{NP} \cap$co-NP, and if G is weak game then $\mathsf{val}_0(c) = 1$ can be decided in linear time.*

The theorem extends to cases where some configurations have predefined values in $[0,1]$.

3 Weak p-automata

In this section we extend p-automata, as defined in [**?**] with a new operator $\oplus$.

Definition 6 (Boolean formulas on T)**.** *Let T be any arbitrary set, then $B^+(T)$ is the set of positive boolean formulas generated by the following syntax:*

$$\varphi ::= \; t \mid \mathit{true} \mid \mathit{false} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \tag{1}$$

where $t \in T$.

The closure of $\varphi \in B^+(T)$ is defined as $\mathsf{cl}(\varphi)$, where $\varphi \in \mathsf{cl}(\varphi)$ and if $\varphi_1 \circ \varphi_2 \in \mathsf{cl}(\varphi)$ then $\varphi_1, \varphi_2 \in \mathsf{cl}(\varphi)$, for $\circ \in \{\wedge, \vee\}$. Let Q be any set of states, the following sets are derived from Q:

$$\begin{array}{llll} \|Q\|_> & = \{\|q\|_{\bowtie p} \mid q \in Q, \bowtie \in \{\geq, >\}, p \in [0,1] \cap \mathbb{Q}\} & \|Q\|^* & = \{*(t_1,\ldots,t_n) \mid n \in \mathbb{N}, \forall i, t_i \in \|Q\|_>\} \\ \|Q\|^\vee & = \{\vee(t_1,\ldots,t_n) \mid n \in \mathbb{N}, \forall i, t_i \in \|Q\|_>\} & \|Q\|^\oplus & = \{\oplus(r_1,\ldots,r_n) \mid n \in \mathbb{N}, \forall i, r_i \in \|Q\|^*\} \\ \|Q\| & = \|Q\|^* \cup \|Q\|^\vee \cup \|Q\|^\oplus & & \end{array}$$

We will call the elements of $\|Q\|_>$ guarded states and elements of $\|Q\|^\oplus$ terms. For brevity, we will write $*(t|t \in X)$ for $*(t_1,\ldots,t_n)$ where $X = \{t_1,\ldots,t_n\}$, (similarly for $\varphi \in \|Q\|^\oplus$ or $\|Q\|^\vee$). For $\varphi = *(\|q_1\|_{\bowtie_1 p_1},\ldots,\|q_n\|_{\bowtie_n p_n})$ (or $\vee(\|q_1\|_{\bowtie_1 p_1},\ldots,\|q_n\|_{\bowtie_n p_n})$), let the set of guarded states be $\mathsf{gs}(\varphi) = \{q_1,\ldots,q_n\}$. If $\varphi = \oplus(r_1,\ldots,r_n)$ then the set of terms is $\mathsf{tm}(\varphi) = \{r_1,\ldots,r_n\}$. In particular, if $|\mathsf{tm}(\varphi)| = 1$ then $\varphi = \oplus(r)$ is the same as r where $r = *(t_1,\ldots,t_n)$. Thus, we consider $\|Q\|^*$ a special case of $\|Q\|^\oplus$.

We will see subsequently that, $\varphi \in \|Q\|^*$ represents the different probabilistic branches, whereas $\varphi \in \|Q\|^\oplus$ represents the non-determinism among the possible probabilistic branching $r \in \mathsf{tm}(\varphi)$.

Definition 7. *A p-automaton A is a tuple $(Q,\Sigma,\delta,\varphi_{in},F)$, where Q is a finite set of states, Σ is a finite alphabet (2^{AP}), $\delta : Q\times\Sigma \to B^+(Q\cup\|Q\|)$ is the transition function, $\varphi_{in} \in B^+(\|Q\|)$ is an initial condition, and $F \subseteq Q$ is an accepting set of states.*

As a convention, p-automata have states, MC have locations, and weak stochastic games have configurations. We will make the following simplification, from hereon we assume that for each $\varphi \in \|Q\|^\oplus$, if a state $q \in \mathsf{gs}(r)$ and $q \in \mathsf{gs}(r')$, where $r,r' \in \mathsf{tm}(\varphi)$ then $r = r'$. A p-automaton $A = (Q,\Sigma,\delta,\varphi_{in},F)$ defines a labeled directed graph $G_A = (Q',E,E_b,E_u)$ (called the *game graph*):

$$
\begin{array}{lcl}
Q' & = & Q \cup cl(\delta(Q,\Sigma)) \\
E & = & \{(\varphi_1 \wedge \varphi_2, \varphi_i) | \varphi_i \in Q' \setminus Q, 1 \le i \le 2\} \;\cup\; \{(q,\delta(q,\sigma)) \mid q \in Q, \sigma \in \Sigma\} \\
 & & \{(\varphi_1 \vee \varphi_2, \varphi_i) | \varphi_i \in Q' \setminus Q, 1 \le i \le 2\} \\
E_u & = & \{(\varphi \wedge q, q), (q \wedge \varphi, q), (\varphi \vee q, q), (q \vee \varphi, q) \mid \varphi \in Q', q \in Q\} \\
E_b & = & \{(\varphi,q) \mid \varphi \in \|Q\|^\vee, q \in \mathsf{gs}(\varphi)\} \;\cup\; \{(\varphi,q) \mid \varphi \in \|Q\|^\oplus, q \in \mathsf{gs}(\mathsf{tm}(\varphi))\}
\end{array}
$$

where $\delta(Q,\Sigma) = \{\delta(q,\sigma) \mid q \in Q \text{ and } \sigma \in \Sigma\} \cup \{\varphi_{in}\}$.

Example 1. *Let the p-automaton $A = (Q,\Sigma,\delta,\varphi,F)$ be defined as follows: $Q = \{q_1,\ldots,q_5\}$, $\Sigma = \{a,b,c\}$, $\varphi = \oplus(*(\|q_1\|_{\ge\frac{1}{2}},\|q_5\|_{\ge\frac{1}{2}}),*(\|q_2\|_{\ge 1}))$, $\delta(q_1,a) = *\|q_3\|_{\ge 1}$, $\delta(q_2,a) = *\|q_4\|_{\ge 1}$, $\delta(q_3,b) = *\|q_3\|_{\ge 1}$, $\delta(q_4,c) = *\|q_4\|_{\ge 1}$, $\delta(q_5,a) = \varphi$ and $F = Q$. The game graph is shown in Figure* **??**.

We add markings on the edges to distinguish them. Edges in E_u and E are unmarked and are called *unbounded* and *simple* transitions, respectively. Edge $(\varphi,q) \in E_b$ is called a *bounded* transition and is marked with $\oplus$ if $\varphi \in \|Q\|^\oplus$, else it is marked with $\vee$. Two formulas $\varphi,\varphi' \in Q'$ are related as $\varphi \preceq_A \varphi'$ iff there is a path from φ to φ' in G_A, and let $\preceq_A \cap \preceq_A^{-1}$ be defined as $\equiv_A$. The equivalence class $[\![\varphi]\!]$ of φ with respect to $\equiv_A$ forms a maximal strongly connected component (MSCC) in G_A. An MSCC is *bounded* iff every edge in an MSCC of G_A, is either in $E \cup E_b$, and an MSCC is *unbounded* iff every edge of the MSCC is in $E \cup E_u$.

Definition 8 (uniform weak p-automata)**.** *A p-automaton A is called* uniform *if: 1.) Every MSCC of G_A is either bounded or unbounded. 2.) For every bounded MSCC marked edges are either all marked with $\oplus$ or (exclusively) with $\vee$. 3.) The set of equivalence classes $\{[\![\varphi]\!] \mid \varphi \in Q'\}$ is finite. 4.) For every symbol σ and $\varphi = \oplus(r_1,\ldots,r_n)$, either every $q \in r_i$, $\delta(q,\sigma) \in \mathscr{B}^+(\|Q\|)$ or every $q \in r_i$, $\delta(q,\sigma) \in \mathscr{B}^+(Q)$. A (not necessarily uniform) p-automaton A is called* weak *if for all $q \in Q$, either $[\![q]\!] \cap Q \subseteq F$ or $[\![q]\!] \cap F = \emptyset$.*

In the rest of the paper we will only consider uniform weak p-automata.

4 Acceptance games

Let $A = (Q,\Sigma,\delta,\varphi_{in},\Omega)$ be a p-automaton and $M = (S,P,L,\mathsf{AP},s_{in})$ be an MC. The acceptance of M by A depends on the results of a sequence of (stochastic) weak games. Let $\Phi = Q \cup \mathsf{cl}(\delta(Q,\Sigma))$ be the set of formulas appearing in the vertices of the game graph G_A. Consider the partial order

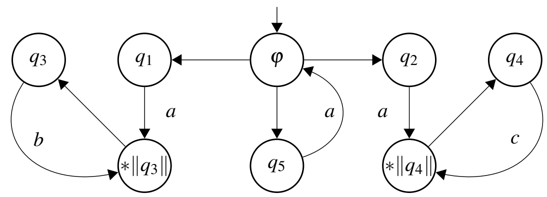


Figure 3: Game graph G_A without unbounded edges.

$$
\boxed{\begin{array}{ll}
V_0^{T,\varphi} = & \{\langle T,\varphi\rangle\} \cup \{\langle T',\varphi',v\rangle \in R_{T,\varphi}\times \mathsf{Val}_{T,\varphi} \mid \bot \neq \mathsf{val}(T',\varphi') < v\} \cup \\
& \{\langle T',\varphi_1\vee\varphi_2,v\rangle \in R_{T,\varphi}\times \mathsf{Val}_{T,\varphi} \mid \mathsf{val}(T',\varphi_1\vee\varphi_2) = \bot\} \\
V_1^{T,\varphi} = & \{\langle T,\varphi,f\rangle \mid f\in \mathscr{F}^{\oplus}_{T,\varphi}\} \cup \{\langle T',\varphi',v\rangle \in R_{T,\varphi}\times \mathsf{Val}_{T,\varphi} \mid \bot \neq \mathsf{val}(T',\varphi') \geq v\} \cup \\
& \{\langle T',\varphi_1\wedge\varphi_2,v\rangle \in R_{T,\varphi}\times \mathsf{Val}_{T,\varphi} \mid \mathsf{val}(T',\varphi_1\wedge\varphi_2) = \bot\} \\
E^{T,\varphi} = & \{(\langle T,\varphi\rangle,\langle T,\varphi,f\rangle) \mid f\in \mathscr{F}^{\oplus}_{T,\varphi}\} \cup \{(\langle T',\varphi_1\circ\varphi_2,v\rangle,\langle T',\varphi_i,v\rangle) \mid \circ\in\{\wedge,\vee\}, 1\leq i\leq 2, \\
& \qquad\qquad (T',\varphi_1\circ\varphi_2,v) \in R_{T,\varphi}\times \mathsf{Val}_{T,\varphi}, \mathsf{val}(T',\varphi_1\circ\varphi_2)=\bot\} \cup \\
& \{(\langle T',\varphi',v\rangle,\langle T',\varphi'\rangle) \mid T'\in \mathsf{succ}(T), \varphi'\in[\varphi], v\in \mathsf{Val}_{T,\varphi}, \mathsf{val}(T,\varphi')=\bot\} \cup \\
& \{(\langle T,\varphi,f\rangle,\langle T',\delta(q,\sigma),f(q,T')\rangle) \mid T'\subseteq \mathsf{succ}(T), q\in I_\varphi, f(q,T')>0,\} \cup \\
& \{(\langle T,\varphi,f\rangle,\langle \{s'\},\delta(q,\sigma),f(q,s')\rangle) \mid s'\in \mathsf{succ}(T), q\in I_\varphi, f(q,T')>0, \delta(q,\sigma)\in \mathscr{B}^+(Q)\}
\end{array}}
$$

Table 1: Acceptance game $\mathsf{G}(M,[\varphi])$, Case 1. $\sigma = L(T)$.

$(\Phi\backslash \equiv_A, \leq_A)$ where $[\varphi] \leq_A [\varphi]'$ iff $\varphi \preceq_A \varphi'$. Let $T\subseteq S$ non-empty set of locations, where for all $s,s'\in T$, $L(s)=L(s')=\sigma$. We assign σ to $L(T)$. For a formula $\varphi\in\Phi$, $\mathsf{val}(T,\varphi)$ is calculated for each MSCC $[\varphi]$ inductively according to the partial order $\leq_A$. $\mathsf{val}(T,\varphi)$ is the value $\mathsf{val}_0(T,\varphi)$ of Player 0 in the game $\mathsf{G}(M,[\varphi]) = (V,E,V_0,V_1,V_p,P,\Omega)$ (defined below). When calculating $\mathsf{val}(T,\varphi)$, the value of $\mathsf{val}(T',\varphi')$ is pre-calculated for every $\varphi'\in[\varphi']$, such that $[\varphi]\leq_A[\varphi']$. Initially, we set $\mathsf{val}(T,\varphi)=\bot$. Depending on the MSCC we have the following cases:

Case 1. Let $[\varphi]$ be a non-trivial *bounded* MSCC where marked edges have marking $\oplus$. For $\varphi = \oplus(r_1,\ldots,r_n)$, let $I_\varphi = \{q \mid q\in \mathsf{gs}(r), r\in \mathsf{tm}(\varphi)\}$, and $p_{i,q}$ be the probability bound on the state q in the term r_i, i.e., $r_i = *(\|q\|_{\geq p_{i,q}} \mid q\in \mathsf{gs}(r_i))$. Consider any non empty subset of states of the Markov chain, $T\subseteq S$, such that for any $s,s'\in T, L(s)=L(s')$. Let the label of every state of T be σ. We define the set $R_{T,\varphi}$, which is the set of successor configurations of $\langle T,\varphi\rangle$, and $\mathsf{Val}_{s,\varphi}$, which is the set of possible values of $\mathsf{val}(T,\varphi)$. We need to enforce that the value of every state of $\mathsf{val}(T,\varphi)$ is well defined. Thus, if $\mathsf{val}(T',\varphi)\neq \mathsf{val}(T'',\varphi)$, then for all sets $T\supseteq T'\cup T''$, $\mathsf{val}(T,\varphi)=0$.

$$
\begin{aligned}
R_{T,\varphi} &= \bigcup_{q\in I_\varphi} \{(T',\varphi') \mid T'\in \mathsf{succ}(T) \text{ and } \varphi'\in \mathsf{cl}(\delta(q,L(T)))\} \\
\mathsf{Val}_{T,\varphi} &= \{0,1\}\cup\{\mathsf{val}(T',\varphi') \mid \langle T',\varphi'\rangle \in R_{T,\varphi}, \mathsf{val}(T',\varphi')\neq\bot\}
\end{aligned}
\tag{2}
$$

Observe, $R_{T,\varphi}$ is finite and hence $\mathsf{Val}_{T,\varphi}\subseteq\mathbb{Q}$ is also finite. Let $\mathscr{F}^{\oplus}_{T,\varphi}$ be a set of functions $I_\varphi\times S\to \mathsf{Val}_{T,\varphi}$ where $f\in\mathscr{F}^{\oplus}_{T,\varphi}$ iff there exists a $\boldsymbol{d}\in\mathscr{D}_{\mathsf{tm}(\varphi)}$ and $\{a_{q,s'}\}_{q\in I_\varphi, s'\in S}\in R^{I_\varphi\times S}$ such that:

$$
\forall q,\forall s\in T\in I_\varphi: \sum_{s'\in succ(s)} a_{q,s'} f(q,s')P(s,s') \geq p_{i,q}\boldsymbol{d}_{r_i}, \text{ and } \forall s'\in \mathsf{succ}(s): \sum_{q\in I_\varphi} a_{q,s'} = 1 \tag{3}
$$

$\boldsymbol{d}$ and $\{a_{q,s'}\}$ are called *witness* of the function f. Note that, the set $\mathscr{F}^{\oplus}_{s,\varphi}$ is finite, because both the domain and the range are finite sets (but can be exponential in size). The game $\mathsf{G}(M,[\varphi]) = (V,V_0,V_1,V_p,E,P,\Omega)$

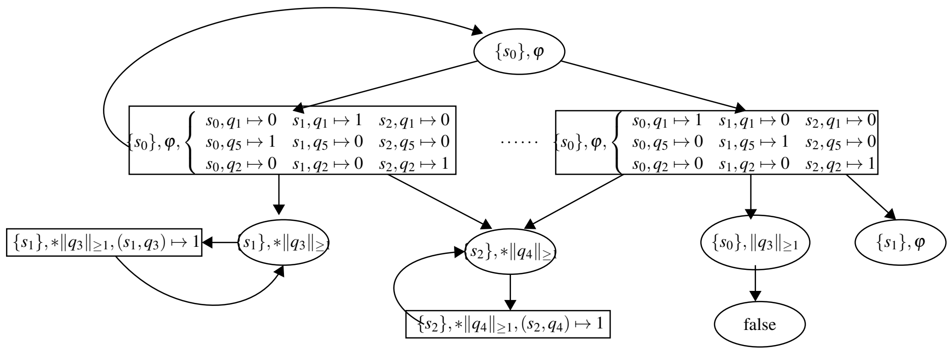


Figure 4: 2-player game (case 1.) generated by p-automaton A in Example **??** and MC M in Figure **??**. The oval states are Player 0 states and the rectangle states are Player 1 states. State $\langle\{s_1\},\varphi\rangle$ belongs to another game and $\mathsf{val}(\{s_1\},\varphi)$ has been pre-computed.

is defined as follows:

$$V_0 \;=\; \bigcup_{T,\varphi} V_0^{T,\varphi} \quad V_1 = \bigcup_{T,\varphi} V_1^{T,\varphi} \quad V_p = \emptyset \quad E \;=\; \bigcup_{T,\varphi} E^{T,\varphi} \qquad \Omega \;=\; \emptyset \text{ or } V$$

where $V_0^{T,\varphi}, V_1^{T,\varphi}$, and $E^{T,\varphi}$ are defined in Table **??**, and $\Omega = V$ if for some $q \in [t]$, $q \in F$ else $\Omega = \emptyset$. Starting from the configuration $\langle T,\varphi\rangle$, the game progresses as follows: At $\langle T,\varphi\rangle$, Player 0 selects a function $f \in \mathscr{F}^{\oplus}_{T,\varphi}$ (i.e., there exist witnesses $\{a_{q,s'}\}$ and $\boldsymbol{d}$). Player 1 can select any subset $T' \subseteq \mathsf{succ}(T)$, such that for every state $s' \in T'$ there is a $q \in I_\varphi$, such that $f(s',q) = 1$ and $\delta(q,a) \in \mathscr{B}^+(\|Q\|)$. Or, it can select $T' = \{s'\}$, where $f(s',q) > 0$ and $\delta(q,\sigma)' \in \mathscr{B}^+(Q)$. Thus, Player 1 can move to $\langle T',\delta(q,\sigma),v\rangle$, where $v = f(s',q)$ for $s' \in T$. A winning play of the game (see Figure **??**) for Player 0 is determined by the following rules:

a. A finite play reaches a configuration $\langle T',\varphi',v\rangle$ such that $\mathsf{val}(s',\varphi') \neq \bot$, that is the value of the configuration $\langle s',\varphi'\rangle$, was already determined. Player 0 wins if $v \leq \mathsf{val}(T',\varphi')$ else player 1 wins. Observe again that configuration $\langle T',\varphi',v\rangle$ is a player 1 configuration if $\bot \neq v \leq \mathsf{val}(T',\varphi)$ and a player 0 configuration if $\bot \neq v > \mathsf{val}(T',\varphi')$.

b. If at $\langle T',\varphi',v\rangle$, $\mathsf{val}(T',\varphi') = \bot$ then the play continues with $\langle T',\varphi'\rangle$. An infinite play is winning if it satisfies the weak acceptance condition Ω. That is, if the play stays in V then player 0 wins if and $V \subseteq \Omega$ else player 1 wins.

Case 2. Let $[\varphi]$ be a nontrivial MSCC such that every transition in the graph G_A belonging to $[\varphi]$ are not in E_u and not marked $\oplus$. Details are present in the appendix.

Case 3. Let $[\varphi]$ be a nontrivial MSCC such that all the transitions in $[\varphi]$ of G_A are in $E_u \cup E$. This gives rise to a weak stochastic game.

$$\begin{array}{lll} V = \{\langle\{s\},\varphi'\rangle \mid \{s\} \in S \text{ and } \varphi' \in [\varphi]\} & V_0 = \{\langle\{s\},\varphi_1 \vee \varphi_2\rangle \in V\} & V_p = (S \times Q) \cap V \\ V_1 = \{\langle\{s\},\varphi_1 \wedge \varphi_2\rangle \in V\} & P(\langle s,q\rangle,\langle s',\delta(q,L(s))\rangle) = P(s,s') & \Omega = \emptyset \text{ or } V \end{array}$$

where Ω is V if some q in $[\varphi]$ is in F else $\Omega = \emptyset$.

$$E = \{(\langle\{s\},\varphi_1 \wedge \varphi_2\rangle, \langle\{s\},\varphi_i\rangle) \in V \times V \mid 1 \leq i \leq 2\} \cup \{(\langle\{s\},\varphi_1 \vee \varphi_2\rangle, \langle\{s\},\varphi_i\rangle) \in V \times V \mid 1 \leq i \leq 2\}$$
$$\cup \{(\langle\{s\},q\rangle, \langle s', \delta(q, L(s))\rangle) \in V \times V \mid P(s,s') > 0\}$$

By Theorem **??**. a value $\mathsf{val}_0(s,\varphi)$ of any configuration $\langle s,\varphi\rangle \in V$ exists. We set $\mathsf{val}(\{s\},\varphi)$ to this value.

Case 4. Let $[\varphi]$ be a trivial MSCC. It is handled as one of the above cases. The value of the configurations $\mathsf{val}(s,\varphi)$ is obtained from the $\mathsf{val}(s',\varphi')$ which have already been calculated in $\mathsf{G}(M,[\varphi'])$.

M is accepted by A, iff $\mathsf{val}(\{s_{in}\},\varphi_{in}) = 1$. The language of A, $\mathscr{L}(A) = \{M : A \text{ accepts } M\}$.

The p-automata defined here has two notable difference than p-automata in [**?**]. First is the syntactic difference due to the presence of formula $\oplus(\varphi_1,\ldots,\varphi_n)$. Second is the semantic difference were we deal with sets of states of the Markov chains for a bounded MSCC (case 1.). This is crucial for proving correctness of Theorem **??**. For unbounded MSCC the description of the acceptance game is same as the original definition.

The number of configurations of the weak game $\mathsf{G}(M,[\varphi])$ is exponential in the size of $[\varphi]$ and Markov chain, when $[\varphi]$ is bounded (case 1.). It is exponential in the size of automaton due to the different function $f \in \mathscr{F}^{\oplus}_{s,\varphi}$. Since, weak games can be solved in polynomial time in the size of the game (and the weak stochastic game can be solved in NP$\cap$co-NP), the problem whether a finite Markov chain is accepted by a p-automaton can be decided in exponential time.

Next we show that the language of a extended p-automaton is closed under probabilistic bi-simulation.

Proposition 1. *For a p-automaton A and MCs M_1 and M_2 with $M_1 \sim M_2$, $M_1 \in \mathscr{L}(A)$ iff $M_2 \in \mathscr{L}(A)$.*

Proof. Let $M_1 = (S_1, P, \mathsf{AP}, L, s_{1,in})$ and $M_2 = (S_2, P, \mathsf{AP}, L, s_{2,in})$, with S_1 disjoint from S_2. Let $A = (Q, \Sigma, \delta, \varphi_{in}, \Omega)$, G_1 and G_2 be the acceptance game for MCs M_1 and M_2, respectively. We show that for each configurations $\langle T_1,\varphi\rangle$ and $\langle T_2,\varphi\rangle$ in G_1 and G_2, respectively, if for every $s_1 \in T_1$ there exists a $s_2 \in T_2$ such that $s_1 \sim s_2$ and vice-versa, then $\mathsf{val}(T_1,\varphi) = \mathsf{val}(T_2,\varphi)$. Towards this end, we will construct a winning strategy for player 0 in G_2 from the game G_1 and vice-versa. The details are present in the appendix. □

Theorem 2. *The language of p-automata is closed under union, intersection, and bisimulation.*

Proof. Closure under union and intersection follows from the presence of $\vee$ and $\wedge$, respectively in the syntax. Closure under bisimulation follows from Proposition **??**. □

5 Embedding MDP

In this section we will embed an MDP into an p-automaton. Let $D = (S, \Delta, \mathsf{AP}, L, s_{in})$ be an MDP.

Definition 9 (p-automata for an MDP)**.** *The p-automaton $A_D = (Q, \Sigma, \delta, \varphi_{in}, \Omega)$ is defined as follows:* [1]

$$Q = S \times S \quad ; \quad \Omega = Q \quad ; \quad \delta((s,s'), L(s)) = \varphi_{s'} \text{ and } \delta((s,s'),\sigma) = \mathit{false} \qquad \text{if } \sigma \neq L(s)$$
$$\varphi_{in} = \oplus(r_i \mid \mu_i \in \Delta(s_{in}), r_i = *(\|(s_{in},\mu_i,s')\|_{\geq \mu_i(s')} \mid \mu_i(s') > 0))$$
$$\varphi_s = \oplus(r_i \mid \mu_i \in \Delta(s) \text{ and } r_i = *(\|(s,\mu_i,s')\|_{\geq \mu_i(s')} \mid \mu_i(s') > 0))$$

[1] It could be the case that there is some state $q \in Q$ which a guarded state of more than one term of a formula $\varphi \in \|Q\|^{\oplus}$. This can be resolved by renaming and introducing new states.

Example 2. *The MDP in the Figure* **??** *is embedded in the automaton A defined in the Example* **??** *and the MC of Figure* **??** *is accepted by A.*

Theorem 3. *Let D be an MDP and A_D be its p-automaton. a.) For every scheduler η, $D_\eta \in \mathscr{L}(A_D)$ and b.) for every MC $M \in \mathscr{L}(A_D)$ there exists a $\eta \in \mathsf{HR}(D)$ such that $M \sim D_\eta$.*

Proof. We first show that if for any η, $D_\eta \in L(A_D)$, and then we show that if a Markov chain $M \in L(A_D)$, then there exists a scheduler η such that $M \sim D_\eta$.

- Let the MDP D be $(S,\Delta,\Sigma,L,s_{in})$. We will first show that for any scheduler $\eta \in \mathsf{HR}(D)$, $D_\eta = (S^+,\Sigma,P',L,s_0)$ is in $\mathscr{L}(A_D)$. We need to show that $\mathsf{val}(\{s_0\},\varphi_{s_0}) = 1$. We first prove that for any state $w \in S^+$ of D_η, with $w{\downarrow} = s$ the value $\mathsf{val}(\{w\},\varphi_s) = \mathsf{val}(\{w{\cdot}u\},\varphi_u)$ whenever $P'(w,w{\cdot}u) > 0$. Player 0 at the configuration $\langle\{w\},\varphi_s\rangle$ chooses function $f \in \mathscr{F}^{\oplus}_{\{w\},\varphi_s}$, such that the witness are as follows: $\boldsymbol{d} = \eta(w)$, $a_{q,w\cdot u} = 1$ and $f(q,w{\cdot}u) = 1$, where $q = (s,\mu,u)$. Observe, that there exists exactly one state $w{\cdot}u$, such that $f(w{\cdot}u,q) = 1$. Thus player can only move to configurations of the type $\langle\{w{\cdot}u\},\varphi_u\rangle$. Thus, $\mathsf{val}(\{w\},\varphi_s) = \mathsf{val}(\{w{\cdot}u\},\varphi_u)$. In an MSCC where non of the values are known, $\mathsf{val}(\{w\},\varphi_s) = 1$, because the every infinite path is winning. This shows, $\mathsf{val}(\{s_0\},\varphi_{s_0}) = 1$.

- Suppose a finite path $\langle T_0,\varphi_{s_0}\rangle,\ldots,\langle T_n,\varphi_{s_n}\rangle$ is winning for Player 1. That is at $\langle T_n,\varphi_{s_n}\rangle$ it is not the case that Player 0 can find a distribution $\boldsymbol{d}$ such that,

$$\forall r_i \in \mathsf{tm}(\varphi_{s_n})\ \forall q \in \mathsf{gs}(r_i)\ \forall s \in T_n : \sum_{s' \in \mathsf{succ}(s)} a_{q,s'} f(q,s') = p_{i,q}\boldsymbol{d}_i$$

  and for each $q \in \mathsf{gs}(\mathsf{tm}(\varphi_{s_n}))$ and any set $T' \subseteq \mathsf{succ}(T_n)$, where $\forall s' \in T' : f(q,s') = 1$, $\langle T',\varphi_{s_n}\rangle$ is winning for Player 0. Take any other (arbitrary) play $\langle T'_0,\varphi_{s_0}\rangle,\ldots,\langle T'_n,\varphi_{s_n}\rangle$ (with $T_0 = T'_0 = \{t_0\}$). Then $\langle T_0 \cup T'_0,\varphi_{s_0}\rangle,\ldots,\langle T_n \cup T'_n,\varphi_{s_n}\rangle$ is also winning for Player 1. So it is in her best interest to choose T' as large as possible

  Let $M = (T,\Sigma,P,L,t_0)$, and $M \in \mathscr{L}(A_D)$. The value of configuration $\langle\{t_0\},\varphi_{s_0}\rangle$ is 1, and assume Player 1 plays optimally, i.e., she chooses a set as large as possible. We will construct a map $\eta^\star \subseteq (S^+ \times \mathscr{D}_{\mathscr{D}_s})$. For any possible finite run, $\rho_n = \langle T_0,\varphi_{s_0}\rangle,\ldots,\langle T_n,\varphi_{s_n}\rangle$, with $T_0 = \{s_0\}$, $(s_0,\ldots,s_n,\boldsymbol{d}) \in \eta^\star$, where $\boldsymbol{d}$ is the distribution chosen by Player 0 at $\langle T_n,\varphi_{s_n}\rangle$. Since, Player 1 plays optimally, it cannot be the case that two distinct play $\rho_n = \langle T_0,\varphi_{s_0}\rangle,\ldots,\langle T_n,\varphi_{s_n}\rangle$ and $\rho'_n = \langle T'_0,\varphi_{s_0}\rangle,\ldots,\langle T'_n,\varphi_{s_n}\rangle$ exists. Thus, we see that $\eta^\star \in \mathsf{HR}(D)$.

  Now consider an unrolling of M. Thus, states of M are subsets of T^+. It suffices to show a bisimulation relation between, $D_{\eta^\star}$ and the unrolled M. Let $R \subseteq (T^+ \cup S^+) \times (T^+ \cup S^+)$ be the smallest transitive, reflexive and symmetric relation with the following property:

  - $t_0 R s_0$.
  - For each play $\rho_n = \langle T_0,\varphi_{s_0}\rangle,\ldots,\langle T_n,\varphi_{s_n}\rangle\langle T_{n+1},\varphi_s\rangle$, all $t \in T_{n+1}$, tRs.

  We will show that R is a bi-simulation relation.

  - If tRw then $L(t) = L'(w)$. If $L(t) \neq L'(w)$ then $\langle t,\varphi_{w\downarrow}\rangle$ cannot be winning for Player 0.
  - For each $q \in I_{\varphi_{w\downarrow}}$, we know, $\sum_{t' \in \mathsf{succ}(t)} P(t,t') a_{q,t'} f(q,t') = p_{q,i}\boldsymbol{d}_i$. From this we can deduce, $\sum_{t' \in C,(t',w\cdot s') \in R} P(t,t') = \sum_{w\cdot s' \in C} P'(w,w{\cdot}s')$ (see Appendix for details).

  Thus, R is a bi-simulation relation , and $M \sim D_{\eta^\star}$.

□

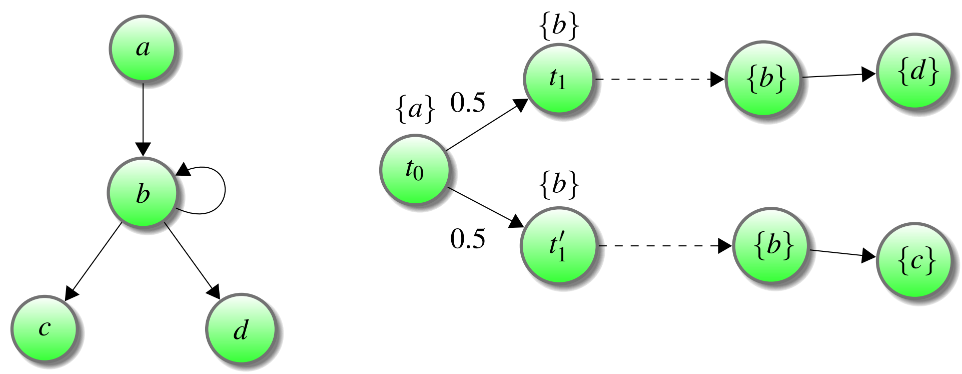


Figure 5: The MDP (left) and a Markov chain (right).

The embedding of MDP relies on the construct $\varphi \in \|Q\|^{\oplus}$. Consider the MDP in Figure **??**. At the state s_0 there are two choices of distribution. If we limit the definition of the p-automata to [**?**] then we have only disjunction (or conjunction) to define the non-determinism at the state s_0 and we cannot accept the MC in Figure **??**. We also store the subset of states T that were *induced* by the same $q \in I_\varphi$. Refer to the Figure **??**. We need to remember that states t_1 and t'_1, were induced by the same distribution. We end this section by mentioning that any PCTL formula can be embedded as a p-automaton. That is, given any PCTL formula, we can construct a p-automaton such that the set models of the formula is exactly the language of the automaton [**?**].

6 Conclusion

We have presented an extension of the p-automata [**?**], and used it to represent the set of MCs which are bisimilar to the MCs induced by the schedulers of an MDP. We have seen that the languages of the p-automata are closed under bi-simulation (union and intersection, trivially). We have addressed the issue of non-determinism of the probability distribution, observed in the concluding remark of [**?**], and shown that the p-automata are powerful enough to represent various probabilistic systems and logics. Even though the acceptance is still EXPTIME, the number configuration has become also exponential in the size of the Markov chain. Unfortunately, the simulation relation gives only a sound characterize language inclusion. It would be interesting to investigate well behaving fragments for which the simulation relation exactly characterizes language inclusion.

Acknowledgment. Author thanks Prof. Nils Pitermann for his fruitful directions.

7 Bibliography

References

[1] Andrea Bianco & Luca de Alfaro (1995): *Model Checking of Probabalistic and Nondeterministic Systems*. In: *Foundations of Software Technology and Theoretical Computer Science, 15th Conference*, pp. 499–513.

[2] Patrick Billingsley: *Probability and Measure*.

[3] T. Brazdil, V. Brozek, V. Forejt & A. Kucera (2006): *Stochastic games with branching-time winning objectives*. In: *Logic in Computer Science, 2006 21st Annual IEEE Symposium on*, pp. 349–358.

[4] Krishnendu Chatterjee, Marcin Jurdziński & Thomas A. Henzinger (2004): *Quantitative Stochastic Parity Games*. In: *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '04, pp. 121–130.

[5] Anne Condon (1992): *The Complexity of Stochastic Games*. *Information and Computation* 96, pp. 203–224.

[6] E.Allen Emerson, Charanjit S. Jutla & A.Prasad Sistla (2001): *On model checking for the μ-calculus and its fragments*. *Theoretical Computer Science* 258(12), pp. 491 – 522.

[7] Hans Hansson & Bengt Jonsson (1994): *A logic for reasoning about time and reliability*. *Formal Aspects of Computing* 6(5), pp. 512–535.

[8] Michael Huth, Nir Piterman & Daniel Wagner (2012): *p-Automata: New foundations for discrete-time probabilistic verification*. *Performance Evaluation* 69(78), pp. 356 – 378. Selected papers from {QEST} 2010.

[9] Dexter Kozen (1983): *Results on the propositional μ-calculus*. *Theoretical Computer Science* 27(3), pp. 333 – 354. Special Issue Ninth International Colloquium on Automata, Languages and Programming (ICALP) Aarhus, Summer 1982.

[10] Kim G. Larsen & Arne Skou (1991): *Bisimulation through probabilistic testing*. *Information and Computation* 94(1), pp. 1 – 28.

[11] Donald A. Martin (1975): *Borel Determinacy*. *Annals of Mathematics* 102(2), pp. pp. 363–371.

[12] Roberto Segala & Nancy Lynch (1995): *Probabilistic Simulations for Probabilistic Processes*. *Nordic J. of Computing* 2(2), pp. 250–273.

[13] Thomas Wilke (2001): *Alternating tree automata, parity games, and modal μ-calculus*. *Bull. Belg. Math. Soc. Simon Stevin* 8(2), pp. 359–391.

10

Regular Paper

Energy Structure and Improved Complexity Upper Bound for Optimal Positional Strategies in Mean Payoff Games

– Extended Abstract*–

Carlo Comin
Department of Mathematics
University of Trento
Trento, Italy
carlo.comin@unitn.it

Romeo Rizzi
Department of Computer Science
University of Verona
Verona, Italy
romeo.rizzi@univr.it

This paper studies structural and algorithmic aspects concerning optimal positional strategies in Mean Payoff Games. In particular, a pseudo-polynomial $O(|V|^2|E|W)$ time algorithm for Optimal Strategy Synthesis in MPGs is provided. This sharpens by a factor $\log(|V|W)$ the best previously known pseudo-polynomial upper bound due to Brim, et al. (2011). The improvement hinges on a suitable description of optimal positional strategies in terms of reweighted games and small energy-progress measures. In addition, we present an energy decomposition theorem describing the whole space of all optimal positional strategies in terms of so-called extremal small energy-progress measures.

Keywords: Mean Payoff Games, Optimal Positional Strategies, Optimal Strategy Synthesis, Energy Games, Small Energy-Progress Measures, Pseudo-Polynomial Time.

1 Introduction and Preliminaries

A *Mean-Payoff Game* (MPG) is a two-player infinite game played on an *arena* $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$, where $G^\Gamma = \langle V, E, w\rangle$ is a finite weighted directed graph whose vertices are partitioned in two classes, V_0 and V_1, according to the player to which they belong [5, 9]. It is also assumed that G^Γ has no sink, i.e., that for every $v \in V$ there exists $v' \in V$ such that $(v, v') \in E$. Moreover, our weights are assumed to be integers, i.e., $w : E \to \mathbf{Z}$. At the beginning of the game a pebble is placed on some vertex of the arena Γ. Then, the two players named Player 0 and Player 1 move the pebble ad infinitum along the arcs. In more detail, assuming the pebble is currently on Player 0's vertex v, then she chooses an arc (v, v') going out of v and the pebble goes to the destination vertex v'. Similarly, assuming the pebble is currently on Player 1's vertex, then it is his turn to choose an outgoing arc. In order to play well, Player 0 wants to maximize the limit inferior of the long-run average weight, that is to maximize the $\liminf_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} w(v_i, v_{i+1})$, whereas Player 1 wants to minimize the $\limsup_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} w(v_i, v_{i+1})$.

Ehrenfeucht and Mycielski proved in [5] that each vertex v admits a *value*, denoted $\mathtt{val}^\Gamma(v)$, which each player can secure by means of a *positional* (or *memoryless*) strategy, i.e., one depending only on the current vertex position and not on the previous choices. Formally, for any $i \in \{0, 1\}$, a strategy of Player i is any function $\sigma_i : V^* \times V_i \to V$ such that for every finite path $p'v$ in G^Γ, where $p' \in V^*$ and $v \in V_i$, it holds $(v, \sigma_i(p', v)) \in E$. A strategy σ_i of Player i is *positional* (or *memoryless*) if $\sigma_i(p, v_n) = \sigma_i(p', v'_m)$ for every finite paths $pv_n = v_0 \dots v_{n-1}v_n$ and $p'v'_m = v'_0 \dots v'_{m-1}v'_m$ in G^Γ such that $v_n = v'_m \in V_i$. The set of all the positional strategies of Player i is denoted by Σ_i^M. Solving an MPG consists in computing the optimal

*This work was supported by the *Department of Computer Science, University of Verona, Italy* under PhD grant "Computational Mathematics and Biology" on a co-tutelle agreement with the *Université Paris-Est in Marne-la-Vallée, Paris, France*.

values of all vertices (*Value Problem*) and, for each player, a positional strategy that secures such values to that player (*Optimal Strategy Synthesis*). The corresponding decision problem lies in $\mathsf{NP} \cap \mathsf{coNP}$ [9], and it was later shown by Jurdziński [7] to be recognizable with unambiguous polynomial time non-deterministic Turing machines, thus falling within the $\mathsf{UP} \cap \mathsf{coUP}$ complexity class. The problem of devising efficient algorithms for Optimal Strategy Synthesis in MPGs has been studied extensively in the literature. The first milestone was the algorithm of Zwick and Paterson [9], which was the first deterministic procedure for computing values, and optimal positional strategies securing them, within a pseudo-polynomial time. In particular, Zwick and Paterson established an $O(|V|^3|E|W)$ bound for the time complexity of the Value Problem, as well as an $O(|V|^4|E|W \log(|E|/|V|))$ bound for that of Optimal Strategy Synthesis [9]. Here and in the rest of this paper, $W := \max_{e\in E} |w_e|$. Recently, several research efforts have been spent in studying quantitative extensions of infinite games on graphs for modeling quantitative aspects of reactive systems [2–4]. In that context quantities may represent, for example, the power usage of an embedded component or the buffer size of a networking element [3]. These studies unveiled interesting connections with MPGs and they recently led to the design of faster algorithms for the strategy synthesis. For instance, Brim, et al. [3] devised a faster deterministic algorithm for solving both Optimal Strategy Synthesis and the Value Problem in MPGs within a pseudo-polynomial $O(|V|^2|E|W \log(|V|W))$ time and polynomial space. To the best of our knowledge, this is the tightest pseudo-polynomial upper bound on the time complexity of Optimal Strategy Synthesis in MPGs which is currently known. Table 1 offers a summary of results.

Table 1: Time Complexity of the main Algorithms for solving Optimal Strategy Synthesis in MPGs.

Algorithm	Optimal Strategy Synthesis	Value Problem
This paper	$O(\lvert V\rvert^2\lvert E\rvert W)$	$O(\lvert V\rvert^2\lvert E\rvert W)$
[3]	$O(\lvert V\rvert^2\lvert E\rvert W \log(\lvert V\rvert W))$	$O(\lvert V\rvert^2\lvert E\rvert W \log(\lvert V\rvert W))$
[9]	$\Theta(\lvert V\rvert^4\lvert E\rvert W \log \frac{\lvert E\rvert}{\lvert V\rvert})$	$\Theta(\lvert V\rvert^3\lvert E\rvert W)$

In particular, the algorithm from [3] is based on a reduction to the so-called Energy Games (EGs), whose decision problem was shown in [2, 4] to be log-space equivalent to that of MPGs. Recall that the decision problem for MPGs asks, given any vertex $v \in V$, to decide whether or not $\mathtt{val}^\Gamma(v) \geq 0$. The corresponding *winning regions* are denoted $\mathscr{W}_0 = \{v \in V \mid \mathtt{val}^\Gamma(v) \geq 0\}$ and $\mathscr{W}_1 = V \setminus \mathscr{W}_0$. On the other side, in EGs, given an initial credit $c \in \mathbf{N}$ and a vertex $v \in V$, Player 0 wins the game starting from v if she can maintain the sum of the encountered weights always non-negative, i.e., if $c + \sum_{i=0}^{j} w(v_i, v_{i+1}) \geq 0$ for all $j \geq 0$; otherwise, the winner is Player 1. The decision problem for EGs asks whether there exists an initial credit c^* for which Player 0 wins from a given starting position vertex v. The corresponding winning regions are also denoted $\mathscr{W}_0$ and $\mathscr{W}_1$. In more detail, Bouyer, et al. [2] and Brim, et al. [3] related the decision problems of MPGs and EGs by means of reweighted EGs. Here, for any weight function $w' : E \to \mathbf{Z}$, the *reweighting* of $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ with respect to w' is defined as $\Gamma^{w'} := \langle V, E, w', (V_0, V_1)\rangle$.
Proposition 1 ([2, 3])**.** *Let $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ be an arena. For all threshold $\nu \in \mathbf{Z}$, for all vertices $v \in V$, Player 0 has a strategy in the MPG Γ that secures value at least ν from v if and only if for some initial credit Player 0 has a winning strategy from v in the reweighted EG $\Gamma^{w-\nu} = \langle V, E, w-\nu, (V_0, V_1)\rangle$.*

Indeed, it is known that memoryless strategies are sufficient for EGs and that Player 0 essentially needs to ensure that all cycles that can be formed by Player 1 have a non-negative total weight. In [3], EGs were solved efficiently using the notion of *small energy-progress measure* (SEPM). These are bounded, non-negative, integer-valued functions that impose local conditions to ensure global properties on the arena, in particular, witnessing that Player 0 has a way to enforce conservativity (i.e., non-negativity of cycles) in the resulting game graph. Recovering standard notation, see e.g. [3], let us denote $\mathscr{C}_\Gamma = \{n \in \mathbf{N} \mid n \leq |V|W\} \cup \{\top\}$ and let $\preceq$ be the total order on $\mathscr{C}_\Gamma$ defined as: $x \preceq y$ if and only if either $y = \top$ or $x, y \in \mathbf{N}$ and $x \leq y$. Moreover, let us consider the operator $\ominus : \mathscr{C}_\Gamma \times \mathbf{Z} \to \mathscr{C}_\Gamma$ defined as follows: if $a \neq \top$ and $a - b \leq |V|W$, then $a \ominus b = \max(0, a-b)$; otherwise, $a \ominus b = \top$. Given an EG Γ on vertex set $V = V_0 \cup V_1$, a function $f : V \to \mathscr{C}_\Gamma$ is a SEPM for Γ if and only if the following two conditions are met:

1. if $v \in V_0$, then $f(v) \succeq f(v') \ominus w(v,v')$ for *some* $(v,v') \in E$;
2. if $v \in V_1$, then $f(v) \succeq f(v') \ominus w(v,v')$ for *all* $(v,v') \in E$.

The values of a SEPM, i.e., the elements of the image set $f(V)$, are named *energy-levels* of f. It is worth to denote $V_f := \{v \in V \mid f(v) \neq \top\}$, i.e., the set of vertices having finite energy in f. Given a SEPM f and any vertex $v \in V_0$, an arc $(v,v') \in E$ is said to be *compatible with* f whenever $f(v) \succeq f(v') \ominus w(v,v')$; moreover, a positional strategy $\sigma_0 \in \Sigma_0^M$ of Player 0 is *compatible with* f whenever for all $v \in V_0$, if $\sigma_0(v) = v'$ then $(v,v') \in E$ is an arc compatible with f. Concerning compatible positional strategies, a key observations made in [3] is the following, which was used to reduce Optimal Strategy Synthesis in MPGs to the computation of SEPMs in EGs.

Proposition 2 ([3]). *Given an MPG $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ and a threshold $\nu \in \mathbf{Z}$, let $f : V \to \mathscr{C}_\Gamma$ be a SEPM for the reweighted EG $\Gamma^{w-\nu} = \langle V, E, w-\nu, (V_0, V_1)\rangle$. All strategies $\sigma_0 \in \Sigma_0^M$ of Player 0 that are compatible with f in the EG $\Gamma^{w-\nu}$ secure to Player 0 a payoff at least ν from all $v \in V_f$ in the MPG Γ.*

Given $\nu \in \mathbf{Z}$, Proposition 2 provides a sufficient condition for a positional strategy to ensure payoff at least ν. Notice that such condition is expressed in terms of SEPMs in reweighted EGs. In some sense, Proposition 2 points out an interesting connection between positional strategies in MPGs and SEPMs in reweighted EGs. Still, the counter-example given in Section 2 shows that the converse statement of Proposition 2 is not true generally. In fact, given an MPG Γ, a threshold $\nu \in \mathbf{Z}$ and a fixed SEPM $f : V \to \mathscr{C}_\Gamma$ for the EG $\Gamma^{w-\nu}$, there exist (in general) positional strategies of Player 0 that secure a payoff at least ν from all $v \in V_f$ in the MPG Γ but that are not compatible with f. For this reason, Proposition 2 doesn't provide yet a complete description of the whole space of all optimal positional strategies. Such a description, if it existed, could be viewed as the *"energy structure"* of the space of all optimal positional strategies in MPGs. We thought that the search for a description of the space of all optimal positional strategies in MPGs in terms of SEPMs in reweighted EGs was an interesting subject of inquiry.

In addition, as shown in [3], if f and g are SEPMs for the EG Γ, then so is the function h defined as follows: $h(v) = \min\{f(v), g(v)\}$ for every $v \in V$. This fact allows one to consider the *least SEPM* of Γ, namely, the (unique) SEPM $f^* : V \to \mathscr{C}_\Gamma$ such that, for any other SEPM $g : V \to \mathscr{C}_\Gamma$ of Γ, it holds that $f^*(v) \preceq g(v)$ for every $v \in V$. The algorithm devised by Brim, et al. [3] for solving EGs is known as *Value-Iteration* [3]. On input any EG Γ, the Value-Iteration aims to compute the least SEPM f^* of Γ. This simple procedure basically relies on a lifting operator δ. Let us denote *in* and *out neighbourhoods* by $\texttt{pre}(\cdot)$ and $\texttt{post}(\cdot)$, respectively. Given $v \in V$, the *lifting operator* $\delta(\cdot, v) : [V \to \mathscr{C}_\Gamma] \to [V \to \mathscr{C}_\Gamma]$ is defined as $\delta(f, v) := g$, where:

$$g(u) := \begin{cases} f(u) & \text{if } u \neq v \\ \min\{f(v') \ominus w(v,v') \mid v' \in \texttt{post}(v)\} & \text{if } u = v \in V_0 \\ \max\{f(v') \ominus w(v,v') \mid v' \in \texttt{post}(v)\} & \text{if } u = v \in V_1 \end{cases}$$

Given any function $f : V \to \mathscr{C}_\Gamma$, we say that f is *inconsistent* in v whenever: 1. $v \in V_0$ and for all $v' \in \texttt{post}(v)$ it holds $f(v) \prec f(v') \ominus w(v,v')$ or 2. $v \in V_1$ and there exists $v' \in \texttt{post}(v)$ such that

$f(v) \prec f(v') \ominus w(v,v')$. To start with, the Value-Iteration initializes f to the constant zero function: i.e., $f(v) \leftarrow 0$ for every $v \in V$. Furthermore, the procedure maintains a list L of vertices that witness an inconsistency of f. Initially, $v \in V_0 \cap \mathsf{L}$ if and only if all arcs going out of v are negative, while $v \in V_1 \cap \mathsf{L}$ if and only if v is the source of at least one negative arc. Checking all these conditions takes time $O(|E|)$.

As long as the list L is nonempty, the algorithm picks a vertex v from L and performs the following:

1. Apply the lifting operator $\delta(f,v)$ to f in order to resolve the inconsistency of f in v;
2. Insert into L all vertices $u \in \texttt{pre}(v) \setminus \mathsf{L}$ witnessing a new inconsistency due to the increase of $f(v)$. Here, the same vertex can't occur twice in L, i.e., the list never contains duplicate vertices.

The algorithm terminates when L is empty. This concludes the description of the Value-Iteration.

As shown in [3], the update of L following an application of the lifting operator $\delta(f,v)$ requires $O(|\texttt{pre}(v)|)$ time. Moreover, a single application of the lifting operator $\delta(\cdot,v)$ takes $O(|\text{post}(v)|)$ time at most. This implies that the algorithm halts within $O(|V|\,|E|\,W)$ time (the reader is referred to [3] in order to grasp all the details). The procedure lends itself to the following basic generalization, which turns out to best suit our technical needs later on. Let f^* be the least SEPM of the EG Γ. Recall that, as a first step, the Value-Iteration initializes f to be the constant zero function. Here, we remark that it is not necessary to do that really. Indeed, it is sufficient to initialize f to any function f_0 which bounds f^* from below, that is to say, to initialize f^* to any $f_0 : V \to \mathscr{C}_\Gamma$ such that $f_0(v) \preceq f^*(v)$ for every $v \in V$. Soon after, L can be managed in the natural way, i.e., insert v into L if and only if f_0 is inconsistent in v. This initialization still requires $O(|E|)$ time and it doesn't affect the correctness nor the time complexity of the procedure.

So, in the rest of this paper, we assume to dispose of a procedure `Value-Iteration()`, which takes as input an EG $\Gamma = \langle V, W, w, (V_0, V_1)\rangle$ *and* an *initial function* f_0 that bounds from below the least SEPM f^* of the EG Γ (i.e., such that $f_0(v) \preceq f^*(v)$ for every $v \in V$). Then, `Value-Iteration()` outputs the least SEPM f^* of the EG Γ within $O(|V|\,|E|\,W)$ time and, for this, it employs $O(|V|)$ working space [3].

Contributions.

- The first contribution of this paper is a deterministic $O(|V|^2|E|\,W)$ time algorithm for solving Optimal Strategy Synthesis and the Value Problem in MPGs. The best previously known deterministic procedure has time complexity $O(|V|^2|E|\,W\log(|V|\,W))$ and it is due to Brim, et al. [3]. In this way, we improve the best previously known deterministic pseudo-polynomial time upper bound by a factor $\log(|V|\,W)$. The result is summarized in the following theorem.

  Theorem 1. *There exists an algorithm for solving Optimal Strategy Synthesis and the Value Problem within $O(|V|^2|E|\,W)$ time and $O(|V|)$ space on any input MPG $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$.*

  In order to prove Theorem 1, we point out a characterization of the values and a suitable description of optimal strategies in terms of SEPMs in reweighted EGs. In particular, we show that the optimal value $\texttt{val}^\Gamma(v)$ of any vertex v is actually the unique rational number for which v *transits* from being winning for Player 0 to being winning for Player 1, with respect to certain reweightings of the original arena. This will be clarified in Theorem 2 of Section 3. Moreover, concerning optimal positional strategies, we show that an optimal positional strategy for any vertex v of Player 0 is given by any arc (v,v') which is compatible with certain SEPMs of the same reweighted arenas mentioned above. This will be clarified in Theorem 3 of Section 3. These observations are simple and their proofs rely on elementary arguments. We believe that these findings contribute to clarifying the relation between optimal strategies, values, and SEPMs in reweighted EGs, with respect to some previous literature [2–4]. Indeed, they allow us to prove Theorem 1.

- The second contribution is an energy decomposition theorem, i.e., Theorem 4 in Section 5, which describes the whole space of all the optimal positional strategies of MPGs in terms of so-called *extremal* SEPMs in reweighted EGs. This describes the *"energy structure"* of the whole space of all optimal positional strategies in MPGs, as it allows for a disjoint-set decomposition (of that space) which is expressed in terms of certain SEPMs (that we named *extremal* SEPMs).

2 An Example

An example MPG Γ_{ex} is depicted in Fig. 1. It is not difficult to see that $\texttt{val}^{\Gamma_{\text{ex}}}(v) = \nu = 1$ for every $v \in V$. Indeed, Γ_{ex} contains only two cycles, i.e., the left one $C_L = [A,B,C,D]$ and the right cycle $C_R = [F,G]$, and notice they satisfy $\frac{w(C_L)}{|C_L|} = \frac{w(C_R)}{|C_R|} = 1$.

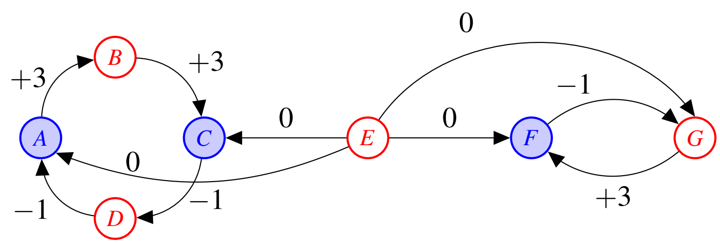


Figure 1: An example MPG $\Gamma_{\text{ex}} = \langle V, \mathsf{E}, w, (V_0, V_1)\rangle$. Here, $V = \{A,B,C,D,E,F,G\}$ and $\mathsf{E} = \{(A,B,+3),(B,C,+3),(C,D,-1),(D,A,-1),(E,A,0),(E,C,0),(E,F,0),(E,G,0),$ $(F,G,-1),(G,F,+3)\}$. Also, $V_0 = \{B,D,E,G\}$ is colored in red, while $V_1 = \{A,C,F\}$ is filled in blue.

It is not difficult to compute the least SEPM f^* for the *reweighted* EG $\Gamma_{\text{ex}}^{w-\nu} = \Gamma_{\text{ex}}^{w-1}$, this can easily be done with the Value-Iteration algorithm of [3]. Taking the reweighting $w \rightsquigarrow w-1$ into account, it turns out that $f^*(A) = f^*(G) = 0$, $f^*(E) = 1$, $f^*(B) = f^*(D) = f^*(F) = 2$, and $f^*(C) = 4$.

We now argue that $\Gamma_{\texttt{ex}}$ shows that neither the converse statement of Proposition 2 nor that of Theorem 3 hold generally. For this, let us focus on the least SEPM f^* of the reweighted EG Γ_{ex}^{w-1}. To begin, notice that the only vertex of $\Gamma_{\texttt{ex}}$ at which Player 0 really has a choice is E. It is clear that every arc going out of E is optimal in the MPG Γ_{ex}. Still, observe that (E,C) and (E,F) are not compatible with f^* in Γ_{ex}^{w-1}, only (E,A) and (E,G) are. For instance, the positional strategy $\sigma_0 \in \Sigma_0^M$, defined as $\sigma_0(E) := F$, $\sigma_0(B) := C$, $\sigma_0(D) := A$, $\sigma_0(G) := F$, ensures a payoff $\texttt{val}^{\Gamma_{\text{ex}}}(v) = 1$ for every $v \in V$, but it is not compatible with the least SEPM f^* of Γ_{ex}^{w-1} (because $f^*(E) = 1 < 3 = 2 \ominus -1 = f^*(F) \ominus w(E,F)$).

3 Values and Optimal Positional Strategies from Reweightings

Optimal Values and Farey Sequences. Let $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ be an MPG. To begin with, it is well known (see e.g. [3]) that each optimal value $\texttt{val}^{\Gamma}(v)$, for every $v \in V$, is contained within the following set of rational numbers:

$$S_\Gamma = \Big\{ \frac{N}{D} \;\Big|\; D \in [1, |V|] \cap \mathbf{Z},\, N \in [-DW, DW] \cap \mathbf{Z} \Big\}.$$

Let us introduce some notation in order to handle S_Γ in a way that is suitable for our purposes. Firstly, we shall write every $\nu \in S_\Gamma$ as $\nu = i + F$, where $i = i(\nu) = \lfloor \nu \rfloor$ and $F = F(\nu) = \nu - i$. Notice that $i \in [-W, W] \cap \mathbf{Z}$ and that F is a non-negative rational number having denominator at most $|V|$.

As a consequence, it is natural to consider the *Farey sequence* $\mathscr{F}_n$ of order $n = |V|$. This is the increasing sequence of all irreducible fractions from the rational interval $[0,1] \cap \mathbf{Q}$ with denominators less than or equal to n. In the rest of the paper $\mathscr{F}_n$ denotes the following sorted set:

$$\mathscr{F}_n = \left\{ \frac{N}{D} \;\Big|\; 0 \leq N \leq D \leq n, \gcd(N,D) = 1 \right\}.$$

Farey sequences have numerous and interesting properties, in particular, many algorithms for generating the entire sequence $\mathscr{F}_n$ in time $O(n^2)$ are known in the literature [6]. Moreover, efficient algorithms (i.e., with a running time sublinear in n) are known for computing the size and the j-th term of the sequence, on the fly without generating it entirely [8]. Notice that the above mentioned quadratic running time is optimal, in fact $\mathscr{F}_n$ has $s(n) = \frac{3n^2}{\pi^2} + O(n \ln n) = \Theta(n^2)$ terms, see e.g. [6].

Throughout the paper, we shall assume that $F_0, \ldots, F_{s-1}$ is an ascending ordering of $\mathscr{F}_n$, so that $\mathscr{F}_n = \{F_j\}_{j=0}^{s-1}$ and $F_j < F_{j+1}$ for every j. Also notice that $F_0 = 0$ and $F_{s-1} = 1$.

For example, $\mathscr{F}_5 = \left\{0, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, 1\right\}$.

At this point, we remark that S_Γ can thus be rewritten in the following manner:

$$S_\Gamma = [-W, W) \cap \mathbf{Z} + \mathscr{F}_{|V|} = \left\{ i + F_j \;\big|\; i \in [-W, W) \cap \mathbf{Z},\, j \in [0, s-1] \cap \mathbf{Z} \right\}.$$

The above represention of S_Γ, in terms of Farey sequences, turns out to be convenient in a while. Indeed, we shall consider reweighted games of the form Γ^{w-i-F_j}, for some $(i + F_j) \in S_\Gamma$. Notice that the corresponding weight function $w' : E \to \mathbf{Q} : e \mapsto w(e) - i - F_j$ is rational valued, while we required that the weights of the arenas are always integers. To overcome this issue, it is sufficient (for our purposes) to re-define Γ^{w-i-F_j} scaling all of its weights by a factor equal to the denominator of F_j, namely, to re-define: $\Gamma^{w-i-F_j} := \Gamma^{D(w-i)-N}$, where $N, D \in \mathbf{N}$ are such that $F_j = N/D$ and $\gcd(N,D) = 1$. Observe that this scaling operation doesn't change the winning regions of the corresponding games, and it has the significant advantage of allowing for a discussion strictly based on arenas with integer weights only.

Optimal Values from Reweightings. In the following theorem we observe that the value $\mathtt{val}^\Gamma(v)$ of any vertex $v \in V$ is actually the unique rational number for which v *transits* from being winning for Player 0 to being winning for Player 1, with respect to certain reweightings of the original arena.

Theorem 2. *Given an MPG $\Gamma = \langle V, E, w, (V_0, V_1) \rangle$, let us consider the following reweightings:*

$$\Gamma_{i,j} := \Gamma^{w-i-F_j}, \text{ for any } i \in [-W, W] \cap \mathbf{Z} \text{ and } j \in [1, s-1] \cap \mathbf{Z},$$

where $s = |\mathscr{F}_{|V|}|$ and F_j is the j-th term of the Farey sequence $\mathscr{F}_{|V|}$. Then, the following holds:

$$\mathtt{val}^\Gamma(v) = i + F_{j-1} \text{ if and only if } v \in \mathscr{W}_0(\Gamma_{i,j-1}) \cap \mathscr{W}_1(\Gamma_{i,j}).$$

3.1 Optimal Positional Strategies from Reweightings

In this section we point out a sufficient condition, for a positional strategy of Player 0 to ensure an optimal payoff from any starting vertex position, which is expressed in terms of SEPM in reweighted EGs.

Theorem 3. *Let $\Gamma = \langle V, E, w, (V_0, V_1) \rangle$ be an MPG. For each vertex $v \in V_0$, consider the reweighted EG $\Gamma_v := \Gamma^{w-\mathtt{val}^\Gamma(v)}$. Let $f_v : V \to \mathscr{C}_{\Gamma_v}$ be any SEPM of the EG Γ_v such that $V_f = \mathscr{W}_0(\Gamma_v) = V$, and let v'_{f_v} denote any vertex out of $v \in V$ such that $(v, v'_{f_v}) \in E$ is compatible with f_v in the EG Γ_v.*

Now, consider the positional strategy $\sigma_0^* \in \Sigma_0^M$ defined as follows:

$$\sigma_0^*(v) := v'_{f_v} \quad \text{for every } v \in V_0.$$

Then, σ_0^* is an optimal positional strategy for Player 0 in the MPG Γ.

Remark 1. *Theorem 3 holds, in particular, when f is the least SEPM f^* of the EG Γ_v. In fact, it is not difficult to show that $V_{f^*} = \mathscr{W}_0(\Gamma_v) = V$ always holds for the least SEPM f^* of Γ_v.*

4 An $O(|V|^2|E|W)$ time Algorithm for Optimal Strategy Synthesis

The present section describes an algorithm for solving Optimal Strategy Synthesis and the Value Problem of MPGs within time $O(|V|^2|E|W)$ on any input $\langle V,E,w,(V_0,V_1)\rangle$. As mentioned, $W = \max_{e\in E}|w_e|$.

In order to describe our algorithm in a suitable way, let us first mention some notation.

Given an MPG $\Gamma = \langle V,E,w,(V_0,V_1)\rangle$, we will consider once again the following reweightings:

$$\Gamma_{i,j} := \Gamma^{w-i-F_j}, \text{ for any } i \in [-W,W]\cap\mathbf{Z} \text{ and } j \in [0,s-1]\cap\mathbf{Z},$$

where $s = |\mathscr{F}_{|V|}|$ and F_j is the j-th term of the Farey sequence $\mathscr{F}_{|V|}$.

Assuming $F_j = N_j/D_j$ for some $N_j, D_j \in \mathbf{N}$, we thus consider the following weight functions, for any $i \in [-W,W]\cap\mathbf{Z}$ and $j \in [0,s-1]\cap\mathbf{Z}$:

$$\begin{aligned} w_{i,j} =& w - i - F_j = w - i - \frac{N_j}{D_j}; \\ w'_{i,j} =& D_j\, w_{i,j} = (w-i)\,D_j - N_j. \end{aligned}$$

In fact, recall that $\Gamma_{i,j}$ is thus $\Gamma_{i,j} := \Gamma^{w'_{i,j}}$, which is an arena with integer weights. We also remark that, since $F_0 < \ldots < F_{s-1}$ is an ordered rational sequence, then the corresponding weight functions $w_{i,j}$ can be ordered in a natural way, i.e., $w_{-W,1} > w_{-W,2} > \ldots > w_{W-1,s-1} > \ldots > w_{W,s-1}$. In the rest of this section, we denote by $f^*_{w'_{i,j}} : V \to \mathscr{C}_{\Gamma_{i,j}}$ the least SEPM of the EG $\Gamma_{i,j}$. Moreover, we say that the function $f^*_{i,j} : V \to \mathbf{Q}$, defined as $f^*_{i,j}(v) := \frac{1}{D_j} f^*_{w'_{i,j}}(v)$ for every $v \in V$, is the *rational scaling* of $f^*_{w'_{i,j}}$.

4.1 Description of the Algorithm

We now describe a procedure named `solve_MPG()`, whose pseudocode is given below in Algorithm 1. It takes as input an arena $\Gamma = \langle V,E,w,(V_0,V_1)\rangle$, and it aims to return a tuple $\langle \mathscr{W}_0, \mathscr{W}_1, \nu, \sigma^*_0\rangle$ such that: $\mathscr{W}_0$ and $\mathscr{W}_1$ are the winning regions of Player 0 and Player 1 in the MPG Γ (respectively), $\nu : V \to S_\Gamma$ is a map sending each starting position $v \in V$ to its optimal value, i.e., $\nu(v) = \mathtt{val}^\Gamma(v)$, and finally, $\sigma^*_0 : V_0 \to V$ is an optimal positional strategy for Player 0 in Γ.

The intuition underlying Algorithm 1 is that of considering the following sequence of weights:

$$w_{-W,1} > w_{-W,2} > \ldots > w_{-W,s-1} > w_{-W+1,1} > w_{-W+1,2} > \ldots > w_{W-1,s-1} > \ldots > w_{W,s-1}$$

where the key idea is that to rely on Theorem 2 at each one of these steps, thus testing whether a *transition of winning regions* has occurred. Stated otherwise, the idea is to test, for each vertex $v \in V$, whether v is winning for Player 1 with respect to the current weight $w_{i,j}$, and to recall whether or not v was winning for Player 0 with respect to the immediately preceding weight $w_{\mathtt{prev}(i,j)}$ in the sequence above.

If such transition actually occurs, say for some $\hat{v} \in \mathscr{W}_0(\Gamma_{\mathtt{prev}(i,j)}) \cap \mathscr{W}_1(\Gamma_{i,j})$, then the procedure can easily compute the value $\mathtt{val}^\Gamma(\hat{v})$ by relying on Theorem 2, and it can also compute an optimal positional strategy, provided that $\hat{v} \in V_0$, by relying on Theorem 3 and Remark 1 in this case.

Each one of these phases, in which we look at transitions of winning regions, is dubbed *Scan Phase*.

An in-depth description of the algorithm and of its pseudocode now follows.

- **Initialization Phase.** To start with, the algorithm performs an initialization phase. At line 1, Algorithm 1 initializes the output variables $\mathscr{W}_0$ and $\mathscr{W}_1$ to be empty sets. Notice that, within the pseudocode, the variables $\mathscr{W}_0$ and $\mathscr{W}_1$ represent the winning regions of Player 0 and Player 1, respectively; also recall that the variable ν represents the optimal values of the input MPG Γ, and the variable σ_0^* represents an optimal positional strategy for Player 0 in the input MPG Γ. Secondly, at line 2, an array variable $f : V \to \mathscr{C}_\Gamma$ is initialized to $f(v) = 0$ for every $v \in V$; throughtout the computation, the variable f represents a SEPM. Next, at line 3, the greatest absolute weight value W is assigned as $W = \max_{e \in E} |w_e|$, an auxiliary weight function w' is initialized as $w' = w + W$, and a "denominator" variable is initialized as $D = 1$. Concluding the initialization phase, at line 4 the size (i.e., the total number of terms) of $\mathscr{F}_{|V|}$ is computed and assigned to the variable s. This size can be computed very efficiently with the algorithm devised by Pawlewicz and Pătraşcu in [8].

Algorithm 1: Solving Optimal Strategy Synthesis and the Value Problem in MPGs.

Procedure *solve_MPG*(Γ)

input : an MPG $\Gamma = \langle V, E, w, (V_0, V_1) \rangle$.

output: a tuple $(\mathscr{W}_0, \mathscr{W}_1, \nu, \sigma_0^*)$ such that: $\mathscr{W}_0$ and $\mathscr{W}_1$ are the winning regions of Player 0 and Player 1 (respectively) in the MPG Γ; $\nu : V \to S_\Gamma$ is a map sending each starting position $v \in V$ to its corresponding optimal value, i.e., $\nu(v) = \mathtt{val}^\Gamma(v)$; $\sigma_0^* : V_0 \to V$ is an optimal positional strategy for Player 0 in the MPG Γ.

// Init Phase
1 $\mathscr{W}_0 \leftarrow \emptyset$; $\mathscr{W}_1 \leftarrow \emptyset$;
2 $f(v) \leftarrow 0 \ \forall\, v \in V$;
3 $W \leftarrow \max_{e \in E} |w_e|$; $w' \leftarrow w + W$; $D \leftarrow 1$;
4 $s \leftarrow$ compute the size $|\mathscr{F}_{|V|}|$ of the Farey sequence $\mathscr{F}_{|V|}$; // e.g. with the algorithm given in [8]
// Scan Phases
5 **for** $i = -W$ ***to*** W **do**
6 $\quad F \leftarrow 0$;
7 $\quad$**for** $j = 1$ ***to*** $s - 1$ **do**
8 $\quad\quad$ prev_$f \leftarrow f$;
9 $\quad\quad$ prev_$w \leftarrow \frac{1}{D} w'$;
10 $\quad\quad$ prev_$F \leftarrow F$;
11 $\quad\quad$ $F \leftarrow$ generate the j-th term of the Farey sequence $\mathscr{F}_{|V|}$; // e.g. with the algorithm of [8]
12 $\quad\quad$ $N \leftarrow$ numerator of F;
13 $\quad\quad$ $D \leftarrow$ denominator of F;
14 $\quad\quad$ $w' \leftarrow (w - i)D - N$;
15 $\quad\quad$ $f \leftarrow \frac{1}{D}\,\mathtt{Value\text{-}Iteration}(\Gamma^{w'}, \lceil D\,\mathrm{prev\_}f \rceil)$;
16 $\quad\quad$ **for** $v \in V$ **do**
17 $\quad\quad\quad$ **if** *prev_f*$(v) \neq \top$ ***and*** $f(v) = \top$ **then**
18 $\quad\quad\quad\quad$ $\nu(v) \leftarrow i + \mathtt{prev\_}F$; // set optimal value ν
19 $\quad\quad\quad\quad$ **if** $\nu(v) \geq 0$ **then**
20 $\quad\quad\quad\quad\quad$ $\mathscr{W}_0 \leftarrow \mathscr{W}_0 \cup \{v\}$; // v is winning for Player 0
21 $\quad\quad\quad\quad$ **else**
22 $\quad\quad\quad\quad\quad$ $\mathscr{W}_1 \leftarrow \mathscr{W}_1 \cup \{v\}$; // v is winning for Player 1
23 $\quad\quad\quad\quad$ **if** $v \in V_0$ **then**
24 $\quad\quad\quad\quad\quad$ **for** $u \in \mathit{post}(v)$ **do**
25 $\quad\quad\quad\quad\quad\quad$ **if** *prev_f*$(v) \succeq$ *prev_f*$(u) \ominus$ *prev_w*(v,u) **then**
26 $\quad\quad\quad\quad\quad\quad\quad$ $\sigma_0^*(v) \leftarrow u$; **break**;
27 **return** $(\mathscr{W}_0, \mathscr{W}_1, \nu, \sigma_0^*)$

- **Scan Phases.** After initialization, the procedure performs multiple *scan phases*. Each one of those is *indexed* by a pair of integers (i,j), where $i \in [-W,W]\cap \mathbf{Z}$ (at line 5) and $j \in [1,s-1]\cap \mathbf{Z}$ (at line 7). Thus, the index i goes from $-W$ to W, and for each i, the index j goes from 1 to $s-1$.

Hence, at each step, we say that the algorithm goes through the (i,j)-th scan phase. For each scan phase, we would also need to consider the *previous* scan phase, so that the *previous index* $\texttt{prev}(i,j)$ is defined as follows: the predecessor of the first index is $\texttt{prev}(-W,1)=(-W,0)$; if $j>1$, then $\texttt{prev}(i,j)=(i,j-1)$; finally, if $j=1$ and $i>-W$, then $\texttt{prev}(i,j)=(i-1,s-1)$.

At the (i,j)-th scan phase, the algorithm considers the rational number $z_{i,j}\in S_\Gamma$ defined as $z_{i,j}:=i+F[j]$, where $F[j]$ denotes the j-th term of the Farey sequence $\mathscr{F}_{|V|}$. For each j, $F[j]$ can be computed very efficiently with the algorithm of Pawlewicz and Pătraşcu [8]. Notice that, since $F[0]<\ldots<F[s-1]$ is a monotonically increasing sequence, in this way the values $z_{i,j}$ are scanned in ascending order as well. At this point, the procedure aims to compute the rational scaling $f:=f^*_{i,j}=\frac{1}{D_j}f^*_{w'_{i,j}}$ of the least SEPM $f^*_{w'_{i,j}}$. This computation is really at the heart of the algorithm and it goes from line 8 to line 15. To start with, at line 8 and line 9, the previous rational scaling $f^*_{\texttt{prev}(i,j)}$ and the previous weight function $w_{\texttt{prev}(i,j)}$ (i.e., those considered during the previous scan phase) are saved into the auxiliary variables $\texttt{prev\_}f$ and $\texttt{prev\_}w$.

Remark. Since the values $z_{i,j}$ are scanned in an ascending order of magnitude, then $\texttt{prev\_}f=f^*_{\texttt{prev}(i,j)}$ bounds from below $f^*_{i,j}$. That is, it holds $\texttt{prev\_}f(v)=f^*_{\texttt{prev}(i,j)}(v)\preceq f(v)$ for every $v\in V$. The underlying intuition, at this point, is that of computing the energy-levels of $f=f^*_{i,j}$ firstly by initializing them to the energy-levels of the previous scan phase, i.e., $\texttt{prev\_}f=f^*_{\texttt{prev}(i,j)}$, and then to update them monotonically upwards by executing the Value-Iteration algorithm for EGs. □

Further details of this pivotal step now follow. Firstly, since the Value-Iteration has been designed to work with integer numerical weights only [3], then the weights $w_{i,j}=w-z_{i,j}$ have to be scaled from $\mathbf{Q}$ to $\mathbf{Z}$: this is performed in a standard way, from line 12 to line 15, by considering the numerator N and the denominator D of $F[j]$, and then by setting $w'_e=(w_e-i)D-N$ for every $e\in E$. Moreover, the initial energy-levels are also scaled up from $\mathbf{Q}$ to $\mathbf{Z}$ by considering the values: $\lceil D\,\texttt{prev\_}f(v)\rceil$, for every $v\in V$ (line 15). At this point the least SEPM of $\Gamma^{w'}$ is computed, at line 15, by invoking the procedure $\texttt{Value-Iteration}(\Gamma^{w'},\lceil D\,\texttt{prev\_}f\rceil)$, that is, by executing on input $\Gamma^{w'}$ the Value-Iteration algoritm with initial energy-levels: $\lceil D\,\texttt{prev\_}f(v)\rceil$ for every $v\in V$. Soon after that, the energy-levels have to be scaled back from $\mathbf{Z}$ to $\mathbf{Q}$; so that, in summary, the energy-levels becomes $f=f^*_{i,j}=\frac{1}{D}\texttt{Value-Iteration}(\Gamma^{w'},\lceil D\,\texttt{prev\_}f\rceil)$ at line 15.

Once $f=f^*_{i,j}$ is finally determined, then for each $v\in V$ the condition $v\in\mathscr{W}_0(\Gamma_{\texttt{prev}(i,j)})\cap\mathscr{W}_1(\Gamma_{i,j})$ is checked at line 17: it is not difficult to show that, in order to do that, it is sufficient to test whether both $\texttt{prev\_f}(v)\neq\top$ and $f(v)=\top$ hold on v.

If $v\in\mathscr{W}_0(\Gamma_{\texttt{prev}(i,j)})\cap\mathscr{W}_1(\Gamma_{i,j})$ holds, then the algorithm relies on Theorem 2 in order to assign the optimal value as follows: $\nu(v)=\texttt{val}^\Gamma(v)=z_{\texttt{prev}(i,j)}$ (line 18). If $\nu(v)\geq 0$, then v is added to the winning region $\mathscr{W}_0$ at line 20. Otherwise, $\nu(v)<0$ and v is added to $\mathscr{W}_1$ at line 22.

To conclude, from line 23 to line 27, the algorithm proceeds as follows: if $v\in V_0$, then it computes an optimal positional strategy $\sigma^*_0(v)$ for Player 0 at v in Γ: this is done by testing for each $u\in\texttt{post}(v)$ whether (v,u) is an arc compatible with $\texttt{prev\_}f$ in $\Gamma_{\texttt{prev}(i,j)}$; namely, whether $\text{prev\_}f(v)\succeq\text{prev\_}f(u)\ominus\text{prev\_}w(v,u)$ for $u\in\texttt{post}(v)$. If (v,u) is found to be compatible with $\texttt{prev\_}f$ at that point, then $\sigma^*_0(v)\leftarrow u$ gets assigned and then (v,u) becomes part of the returned optimal positional strategy. Indeed, the correctness of such an assignment relies on Theorem 3 and Remark 1.

This concludes the description of the scan phases and also that of Algorithm 1.

4.2 Correctness and Complexity

The following proposition asserts that Algorithm 1 is correct and it points out time and space complexity.

Proposition 3. *Assume that Algorithm 1 is invoked on some input MPG $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$, and that it returns $(\mathscr{W}_0, \mathscr{W}_1, \nu, \sigma_0^*)$ as output. Then, $\mathscr{W}_0$ and $\mathscr{W}_1$ are the winning regions of Player 0 and Player 1 in the MPG Γ (respectively), the map $\nu : V \to S_\Gamma$ satisfies $\nu(v) = \mathtt{val}^\Gamma(v)$ for every $v \in V$, and $\sigma_0^* : V_0 \to V$ is an optimal positional strategy for Player 0 in the MPG Γ.*

Moreover, Algorithm 1 always halts within $O(|V|^2|E|W)$ time and it employs $O(|V|)$ working space.

5 Energy Structure of the Space of Optimal Positional Strategies

This section presents a complete decomposition of the space of all optimal positional strategies in MPGs, which is expressed in terms of so-called *extremal* SEPMs in reweighted EGs. In a sense, the result complements both Proposition 2 and Theorem 3. To begin with, let $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ be an MPG, given a positional strategy $\sigma_0 \in \Sigma_0^M$ of Player 0, we denote by $\mathtt{val}^\Gamma_{\sigma_0}(v)$ the payoff value ensured by σ_0 when the game starts from $v \in V$. We aim to study the space of all optimal positional strategies of Γ, i.e.,

$$\mathrm{opt}(\Sigma_0^M) = \left\{\sigma_0 \in \Sigma_0^M \mid \forall\, v \in V \;\; \mathtt{val}^\Gamma_{\sigma_0}(v) = \mathtt{val}^\Gamma(v)\right\},$$

for this, no loss of generality occurs [1] if we assume, for some $\nu \in \mathbf{Q}$, that $\mathtt{val}^\Gamma(v) = \nu$ for every $v \in \Gamma$.

In order to proceed, we need to introduce some further definitions.

Given any $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ and $\sigma_0 \in \Sigma_0^M$, let $G^\Gamma_{\sigma_0} := (V, E', w')$ be the graph obtained from Γ by deleting all those arcs that are not consistent with σ_0, namely, let $E' := \{(u, v) \in E \mid u \in V_0 \text{ and } v = \sigma_0(u)\} \cup \{(u, v) \in E \mid u \in V_1\}$ where all arcs $e \in E'$ are weighted as in Γ, i.e., $w' : E' \to \mathbf{Z} : e \mapsto w(e)$.

Furthermore, let $\pi^*_{G^\Gamma_{\sigma_0}}$ be the *least feasible potential* [2] of $G^\Gamma_{\sigma_0}$.

Definition 1 (Energy-Tight Positional Strategies)**.** *Let $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ and let $f : V \to \mathscr{C}_\Gamma$ be a SEPM for the EG Γ. Define $\Delta_0^{\Gamma, f} \subseteq \Sigma_0^M$ to be the family of all and only those positional strategies of Player 0 in Γ such that $\pi^*_{G^\Gamma_{\sigma_0}}$ coincides with f pointwisely, i.e.,*

$$\Delta_0^{\Gamma, f} := \left\{\sigma_0 \in \Sigma_0^M \;\middle|\; \forall\, v \in V \;\; \pi^*_{G^\Gamma_{\sigma_0}}(v) = f(v)\right\}.$$

In what follows we introduce the notion of *Energy Basis* for the space $\mathtt{opt}(\Sigma_0^M)$.

Definition 2 (Energy Basis)**.** *Let $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ be an MPG such that, for some $\nu \in \mathbf{Q}$, it holds $\mathit{val}^\Gamma(v) = \nu$ for every $v \in V$. Moreover, let $\mathscr{B} = \{f_1, \ldots, f_k\}$ be a family of SEPMs for the reweighted EG $\Gamma^{w-\nu}$. We say that $\mathscr{B}$ is an* Energy Basis *for the MPG Γ if the following disjoint-set decomposition holds for the space of all optimal positional strategies:*

$$\mathit{opt}(\Sigma_0^M) = \bigsqcup_{f \in \mathscr{B}} \Delta_0^{\Gamma^{w-\nu}, f}.$$

[1] No generality is lost here thanks to the *"Ergodic Partition"* (EP) property of MPGs, (see e.g. Theorem 7.4 in [1]). The EP property allows one to partition Γ into several domains $\{\Gamma_{|C_i}\}_i$ each one satisfying, for some rational $\nu_i \in \mathbf{Q}$, the following condition: $\mathtt{val}^{\Gamma_{|C_i}}(v) = \nu_i$ for every vertex $v \in C_i$. Then, one can study the structure of the space $\mathtt{opt}(\Sigma_0^M)$ within each C_i independently with respect to the others C_j for $j \neq i$.

[2] When $G = (V, E, w)$ is a weighted directed graph, a *feasible potential* for G is any function $\pi : V \to \mathscr{C}_G$ such that $\pi(u) \succeq \pi(v) \ominus w(u, v)$ for every $u \in V$ and every $v \in \mathtt{post}(u)$. The *least* feasible potential $\pi^* = \pi^*_G$ is the (unique) feasible potential such that, for any other feasible potential π, it holds $\pi^*(v) \preceq \pi(v)$ for every $v \in V$. Notice that, given $G = (V, E, w)$ in input, the Bellman-Ford algorithm can be used to produce π^*_G in polynomial $O(|V|\,|E|)$ time.

The following theorem asserts the actual decomposition of $\mathtt{opt}(\Sigma_0^M)$ in terms of SEPMs in EGs.

Theorem 4. *Let $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ be an MPG such that, for some $\nu \in \mathbf{Q}$, it holds $\mathtt{val}^{\Gamma}(v) = \nu$ for every $v \in V$. Then, the MPG Γ admits one and only one Energy Basis, which is denoted by $\mathscr{B}^* = \mathscr{B}^*_{\Gamma}$.*

Example. Consider the MPG $\Gamma_{\mathtt{ex}}$, as defined in Fig. 1. Then, $\mathscr{B}^*_{\Gamma_{\mathtt{ex}}} = \{f^*, f_1, f_2\}$, where f^* is the least SEPM of the reweighted EG $\Gamma_{\mathtt{ex}}^{w-1}$, and where the following holds: $f_1(A) = f_2(A) = f^*(A) = 0$; $f_1(B) = f_2(B) = f^*(B) = 2$; $f_1(C) = f_2(C) = f^*(C) = 4$; $f_1(D) = f_2(D) = f^*(D) = 2$; $f_1(F) = f_2(F) = f^*(F) = 2$; $f_1(G) = f_2(G) = f^*(G) = 0$; finally, $f^*(E) = 1$, $f_1(E) = 3$, $f_2(E) = 5$.

Each element $f \in \mathscr{B}^*$ is said to be an *extremal* SEPM, and the following properties hold on it.

Proposition 4. *Let $\Gamma = \langle V, E, w, (V_0, V_1)\rangle$ be an MPG such that, for some $\nu \in \mathbf{Q}$, it holds $\mathtt{val}^{\Gamma}(v) = \nu$ for every $v \in V$. Let $\mathscr{B}^*_{\Gamma}$ be the Energy Basis of the MPG Γ. Let $f : V \to \mathscr{C}_{\Gamma}$ be a SEPM for the reweighted EG $\Gamma^{w-\nu}$. Then, the following three properties are equivalent:*

1. *$f \in \mathscr{B}^*_{\Gamma}$;*
2. *$V_f = \mathscr{W}_0(\Gamma^{w-\nu}) = V$ and $\Delta_0^{\Gamma^{w-\nu}, f} \neq \emptyset$;*
3. *There exists $\sigma_0 \in \mathtt{opt}(\Sigma_0^M)$ such that $\pi^*_{G^{\Gamma^{w-\nu}}_{\sigma_0}}(v) = f(v)$ for every $v \in V$.*

Conclusion. In this work, we presented an improved pseudo-polynomial $O(|V|^2|E|W)$ bound on the time complexity of Optimal Strategy Synthesis in MPGs. In addition, we provided an energy decomposition theorem describing the whole space of all optimal positional strategies in terms of SEPMs in reweighted EGs. We ask whether it is possible to improve over the $O(|V|^2|E|W)$ bound. Also, in future works, it would be interesting to study further properties enjoyed by $\mathscr{B}^*$ and by the extremal SEPMs.

References

[1] H. Björklund, S. Sandberg & S. Vorobyov (2004): *Memoryless determinacy of parity and mean payoff games: a simple proof*. Theoretical Computer Science 310(1–3), pp. 365 – 378.

[2] P. Bouyer, U. Fahrenberg, K. Larsen, N. Markey & J. Srba (2008): *Infinite Runs in Weighted Timed Automata with Energy Constraints*. In Franck Cassez & Claude Jard, editors: Formal Modeling and Analysis of Timed Systems, Lecture Notes in Computer Science 5215, Springer Berlin Heidelberg, pp. 33–47.

[3] 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.

[4] A. Chakrabarti, L. de Alfaro, T. Henzinger & M. Stoelinga (2003): *Resource Interfaces*. In Rajeev Alur & Insup Lee, editors: Embedded Software, Lecture Notes in Computer Science 2855, Springer Berlin Heidelberg, pp. 117–133.

[5] A. Ehrenfeucht & J. Mycielski (1979): *Positional strategies for mean payoff games*. International Journal of Game Theory 8(2), pp. 109–113.

[6] R.L. Graham, D.E. Knuth & O. Patashnik (1994): *Concrete Mathematics: A Foundation for Computer Science*, 2nd edition. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[7] Marcin Jurdziński (1998): *Deciding the winner in parity games is in $\mathtt{UP} \cap \mathtt{co\text{-}UP}$*. Information Processing Letters 68(3), pp. 119 – 124.

[8] Jakub Pawlewicz & Mihai Pătraşcu (2009): *Order Statistics in the Farey Sequences in Sublinear Time and Counting Primitive Lattice Points in Polygons*. Algorithmica 55(2), pp. 271–282.

[9] U. Zwick & M. Paterson (1996): *The Complexity of Mean Payoff Games on Graphs*. Theoretical Computer Science 158, pp. 343–359.

11

Regular Paper

Simulating cardinal payoffs in Boolean games to prove hardness

Egor Ianovski Luke Ong
University of Oxford
Oxford, UK
egor.ianovski@cs.ox.ac.uk luke.ong@cs.ox.ac.uk

Boolean games are a succinct representation of strategic games with a logical flavour. While they have proved to be a popular formalism in the multiagent community, a commonly cited shortcoming is their inability to express richer utilities than success or failure. In addition to being a modelling limitation, this parsimony of preference has made proving complexity bounds difficult. We address the second of these issues by demonstrating how cardinal utilities can be simulated via expected utility. This allows us to prove that RATIONALNASH and IRRATIONALNASH are NEXP-hard.

1 Introduction

Since their introduction in [20], Boolean games have acquired a wide following (e.g. [13, 7, 15, 16, 23, 22, 19]). Their success is perhaps due to the fact that they strike a happy middle ground between succinctness of representation and intuitive play—one need neither write down the normal form of a strategic game explicitly, nor scratch one's head as to why a mysterious machine churns out the utilities that it does. That said, a commonly cited concern with the framework is the extremely simple preferences one is restricted to—a player in a Boolean game either satisfies his formula or he does not, he is either perfectly happy or about to compose the next Gloomy Sunday. This has sparked various attempts to introduce richer preferences into the Boolean games framework, be it through the medium of a richer language ([24, 8, 25]), taxation schemes ([23, 34, 31]), or by adding weights to the players' formulae ([2, 26]).

Computational issues about Boolean games were first considered in [13] and have since been studied from a number of angles, including player dependency ([5, 29]), restrictions on player goals ([7, 15]), and the complexity of coalition formation ([14, 6, 28]). Until the work of [21], however, the focus has been exclusively on problems concerning pure equilibria; this was a serious restriction, as Boolean games are a representation of strategic games, and mixed strategies are first class citizens of that framework. One may wonder whether the two phenomena are related: were complexity results about mixed equilibria difficult to prove because we did not have any numbers to play with? After all, in the case of explicitly represented games the study of binary valued, or win-lose games required novel constructions to get around the fact that players can only win or lose ([9, 10, 4]).

This is the focus of the current work. We introduce a useful family of gadget games in Section 4 which mimic cardinal preferences by giving the players a fixed probability of winning the gadget game. This allows us to construct proofs that RATIONALNASH and IRRATIONALNASH, the problems of determining whether a game has a rational or an irrational equilibrium respectively (first studied by [3, 4]), are NEXP-hard in Section 5.

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

2 Strategic and Boolean games

Definition 2.1. A *strategic game* is a triple $(N,\{S_1,\dots,S_n\},\{u_1,\dots,u_n\})$. N is a finite set of *players*, of cardinality n. S_i is a finite set of Player i's *pure strategies*. An n-tuple of pure strategies, i.e. a member of $\mathscr{S} = S_1 \times \cdots \times S_n$ is called a *pure-strategy profile*. The function $u_i : \mathscr{S} \to \mathbb{R}$ is Player i's *utility function*.

A strategic game is called *zero-sum* just if there exists a $c \in \mathbb{R}$ such that for every $\boldsymbol{s} \in \mathscr{S}$:[1]

$$\sum_{i\in N} u_i(\boldsymbol{s}) = c.$$

■

Example 2.2. Battle of the Sexes is played by two players coordinating on a venue for a date. The choices are boxing and ballet. Player One prefers ballet and Player Two prefers boxing, but both players prefer successful coordination to choosing different venues.

This game could be represented by giving the players the strategies $S_1 = \{Box_1, Bal_1\}$ and $S_2 = \{Box_2, Bal_2\}$. Their utility functions would be given by setting $u_1(Box_1,Bal_2) = u_1(Bal_1,Box_2) = 0$, $u_1(Bal_1,Bal_2) = 2$ and $u_1(Box_1,Box_2) = 1$ for Player One, $u_2(Box_1,Bal_2) = u_2(Bal_1,Box_2) = 0$, $u_2(Bal_1,Bal_2) = 1$ and $u_2(Box_1,Box_2) = 2$ for Player Two.

Matching Pennies is played by having two players reveal a coin heads or tails up. Player Two wins if the coins have the same orientation, Player One if the orientation differs.

In the given formalism this game is $S_1 = \{H_1,T_1\}$, $S_2 = \{H_2,T_2\}$, $u_1(H_1,T_2) = u_1(T_1,H_2) = 1$, $u_1(H_1,H_2) = u_1(T_1,T_2) = 0$, $u_2(H_1,H_2) = u_2(T_1,T_2) = 1$ and $u_2(H_1,T_2) = u_2(T_1,H_2) = 0$.

This explicit representation of a strategic game, obtained by listing the graph of the utility functions, is often referred to as the *normal form* and in the two-player case it can be conveniently visualised as a table:

	Box_2	Bal_2
Box_1	$(1,2)$	$(0,0)$
Bal_1	$(0,0)$	$(2,1)$

Battle of the Sexes

	H_2	T_2
H_1	$(0,1)$	$(1,0)$
T_1	$(1,0)$	$(0,1)$

Matching Pennies

■

The most common solution concept for a strategic game is the Nash equilibrium, which is a strategy profile from which no player would opt to deviate unilaterally. In other words, the utility the player is currently getting is at least as high as in any alternative profile that differs only by that player's own choice of strategy.

Definition 2.3. A pure-strategy profile $\boldsymbol{s}$ is called a *pure-strategy equilibrium* just if for each $i \in N$, for all $s' \in S_i$:

$$u_i(\boldsymbol{s}) \geq u_i(\boldsymbol{s}_{-i}(s')),$$

where $\boldsymbol{s}_{-i}(s')$ denotes the profile obtained by replacing Player i's strategy in $\boldsymbol{s}$ with s'. ■

Example 2.4. Battle of the Sexes has two pure-strategy equilibria: either both players go to ballet or to boxing. One player will be getting a utility of 2 the other of 1, but even the player that is getting a utility of 1 would rather stay at the venue than deviate and get a utility of 0.

Matching Pennies has no pure-strategy equilibria. No matter what profile we choose, one of the players would rather switch the orientation of their coin.

[1] This is of course more correctly called a constant-sum game, but the two are functionally equivalent.

In the tabular representation the pure-strategy equilibria are precisely those cells where Player One's utility is maximal in the column, and Player Two's is maximal in the row, as the reader can verify above. ∎

The fact that even games as simple as Matching Pennies can fail to have a pure-strategy equilibrium motivates us to consider broader solution concepts. It helps to consider how one would actually play a game like Matching Pennies (e.g. Rock-Paper-Scissors) were one to find oneself stuck at a horrendous dinner party. Displaying a strong preference for either heads or tails would allow the opponent to take advantage of this, and hence the optimal way to play is to randomise. This is the concept of a mixed strategy.

Definition 2.5. Let $\mathfrak{P}(S_i)$ denote the space of probability distributions over S_i. A *mixed strategy* for Player i is a member of $\mathfrak{P}(S_i)$. The weight assigned to a pure strategy s by a mixed strategy σ, or $P(s \mid \sigma)$, is called the *strategy weight* of s.

An n-tuple of mixed strategies, $\boldsymbol{\sigma} \in \mathfrak{P}(\mathscr{S})$, is called a *mixed-strategy profile*. We extend Player i's utility function to the space of mixed-strategy profiles on the principle of expected utility. That is:

$$u_i(\boldsymbol{\sigma}) = \sum_{\boldsymbol{s} \in \mathscr{S}} u_i(\boldsymbol{s}) P(\boldsymbol{s} \mid \boldsymbol{\sigma}).$$

A mixed-strategy profile is called a *mixed-strategy equilibrium* just if for all $s' \in S_i$:

$$u_i(\boldsymbol{\sigma}) \geq u_i(\boldsymbol{\sigma}_{-i}(s')).$$

∎

Clearly a pure equilibrium is just a specific type of mixed equilibrium, so when we say "equilibrium" without qualifications we mean a mixed equilibrium.

The reader will also note that in the definition above we define a mixed-strategy equilibrium as a profile that is robust to any deviation with a *pure* strategy, and do not consider deviations with mixed strategies. There is no generality lost here—if we are considering a unilateral deviation by Player i we can fix the strategy choice of other players, leaving Player i with a linear function to maximise. Such a function will attain its maxima at the extreme points, which are precisely the pure strategies.

Example 2.6. Matching Pennies has a unique equilibrium where Player One randomises equally between H_1 and T_1, and Player Two randomises equally between H_2 and T_2. The payoff for either player is $1/2$. This is also the payoff Player One would get by playing H_1, as Player Two plays T_2 with probability $1/2$, or for playing T_1, as Player Two plays H_2 with probability $1/2$. Player One is indifferent between any deviation, and, mutatis mutandis, so is Player Two.

Battle of the Sexes has an additional equilibrium where Player One plays Bal_1 with probability $2/3$ and Box_1 with probability $1/3$, whilst Player Two plays Bal_2 with probability $1/3$ and Box_2 with probability $2/3$. The utility of either player is $2/3$—the utility Player One would get by playing Bal_1 with probability 1 ($1/3$ chance of getting 2), or Box_1 with probability 1 ($2/3$ chance of getting 1). ∎

Two foundational theorems of game theory are of interest to us here. The first, due to Nash, tells us that mixed equilibria exist.

Theorem 2.7 ([27])**.** *Every strategic game has an equilibrium in mixed strategies.*

The second (albeit first chronologically), due to von Neumann, tells us that equilibria in two-player zero-sum games have a special property.

Theorem 2.8 ([32]). *In every two-player zero-sum game there exists a v such that Player One gets a utility of v in every equilibrium. This v is called the* value *of the game.*

While the normal form has the advantage of lucidity, the representation size does not scale well; the graph of Player i's utility function is of order $O(|\mathscr{S}|^n)$—not only is this exponential in the number of players, but the polynomial dependence on the number of strategies alone is untenable in all but the most simple of games (consider checkers). As a result many concise representations of games have been studied in the literature. In this work, we are interested in one in particular.

These are Boolean games, first defined as a two-player zero-sum affair by [20] and later redefined as the multiplayer variable-sum entity we study today by [7]. A Boolean game operates by partitioning a set of propositional variables among a set of players. A player may assign *true* or *false* values to the variables under his control in the hope of satisfying his *goal formula*, a formula of propositional logic. As his goal formula may depend on variables outside of his control he cannot achieve this unilaterally, and has to reason strategically to choose the best possible truth assignment.

Definition 2.9 ([20, 7]). A *Boolean game* is a representation of a strategic game given by the triple $(N, \{\Phi_1, \ldots, \Phi_n\}, \{\gamma_1, \ldots, \gamma_n\})$. The Φ_i are mutually disjoint sets of propositional variables and each γ_i is a formula of propositional logic defined over $\biguplus_{i \in N} \Phi_i$. Player i's pure strategies are truth assignments to Φ_i, i.e. $S_i = 2^{\Phi_i}$. Player i's utility function is:

$$u_i(\boldsymbol{\nu}) = \begin{cases} 1 \text{ if } \boldsymbol{\nu} \vDash \gamma_i, \\ 0 \text{ otherwise.} \end{cases}$$

■

Example 2.10. Matching Pennies can be represented as a Boolean game in the following fashion:

$$\begin{aligned} \Phi_1 &= \{p\}, \\ \Phi_2 &= \{q\}, \\ \gamma_1 &= \neg(p \leftrightarrow q), \\ \gamma_2 &= p \leftrightarrow q. \end{aligned}$$

Battle of the Sexes cannot be represented as a Boolean game because the payoffs are not all 0 and 1. ■

A Boolean game can potentially achieve exponential succinctness—after all, we need only k variables to represent 2^k strategies. However this comes at the cost of some of the lucidity of a game in normal form—even determining whether a player ought to bother playing the game (whether γ_i is satisfiable) is NP-complete.

3 Decision problems

It now pays to introduce a specific kind of equilibrium, that is of natural interest from a computational perspective.

Definition 3.1. An equilibrium $\boldsymbol{\sigma}$ is *rational* if every strategy weight in $\boldsymbol{\sigma}$ is rational. It is *irrational* if at least one strategy weight is not. ■

Rational numbers are convenient as they allow us to reason algorithmically about a wide range of problems without delving into the representational issues concerning algebraic or computable reals. It is fortuitous, then, that a consequence of the linear programming characterisation of two-player games establishes that these have rational equilibria.

∃GUARANTEENASH	
Input:	A Boolean game G and a $v \in [0,1]_{\mathbb{Q}}$.
Output:	YES if G has a rational equilibrium $\boldsymbol{\sigma}$ for which $u_1(\boldsymbol{\sigma}) \geq v$.

RATIONALNASH	
Input:	A Boolean game G.
Output:	YES if G has a rational equilibrium.

IRRATIONALNASH	
Input:	A Boolean game G.
Output:	Yes if G has an irrational equilibrium.

Figure 1: Three decision problems for Boolean games.

Proposition 3.2 ([12])**.** *Every two-player strategic game has a rational equilibrium, and the size of this equilibrium is polynomial in the size of the normal form.*

Unfortunately, the fun stops at $n = 2$.

Proposition 3.3 ([27])**.** *There exist three-player games with rational payoffs that have only irrational equilibria.*

This is the reason behind a general trend observed in algorithmic game theory: nontrivial questions about the properties of equilibria of games in normal form tend to be NP-complete for the two-player case, as one could nondeterministically choose an equilibrium and verify the property, but NP-hard in the general case, as then the equilibria may not have finite representations (e.g. [18, 11, 4]).

∃GUARANTEENASH, in Figure 1, belongs to this class of problems applied to Boolean games. The multiplayer variant has been shown to be NEXP-hard by [21], and the two-player variant in an unpublished work. The reader will note that the way we state ∃GUARANTEENASH differs from the definition in [21], however careful observation of thr proof will show that hardness holds for this version as well. As Boolean games are a concise representation of strategic games, it is perhaps not surprising that these complexities experience an exponential jump.

RATIONALNASH however, is a very different animal—it trivialises completely in the two-player case: the answer is YES. As we shall see, hardness reasserts itself as soon as we add a third. IRRATIONALNASH is seemingly a tamer beast—we do not have a result for the two-player case in the present work, but we suspect a closer analysis of the linear programming characterisations of strategic games will reveal some connection to the degeneracy problem—but it is included due to its spiritual kinship with RATIONALNASH.

4 Games of any value

The reader will note that while the payoff for Player One in any *pure* profile in a Boolean game need be 1 or 0, there is no such restriction on *mixed* profiles—after all, the unique mixed equilibrium of Matching Pennies gives either player $1/2$. This motivates the main proof idea: we can use two-player, zero-sum gadget games of a specific value to simulate fine-grained preferences.

The key here is to interpret truth assignments numerically, and imitate arithmetic operations via propositional logic.

Definition 4.1. Let $\overline{v_m}$ represent $v_1, \ldots, v_m \in \Phi_i$. I.e., a sequence of m propositional variables, all controlled by the same player. Let $[\![\overline{v_m}]\!](\boldsymbol{v})$ represent the numeric value assigned to $\overline{v_m}$ by $\boldsymbol{v}$ in the natural fashion. I.e., v_1 is *true* is interpreted as the most significant bit of the m-bit integer $[\![\overline{v_m}]\!](\boldsymbol{v})$ being a 1.

If the truth assignment is clear from context, we omit it. ■

Proposition 4.2. *We can, in polynomial time, construct the formulae* $\boldsymbol{Less}(\overline{a_m}, \overline{b_m})$, $\boldsymbol{LessEq}(\overline{a_m}, \overline{b_m})$, $\boldsymbol{Add}(\overline{a_m}, \overline{b_m}, \overline{c_m})$ *and* $\boldsymbol{Sub}(\overline{a_m}, \overline{b_m}, \overline{c_m})$ *that are true under* $\boldsymbol{v}$ *just if* $[\![\overline{a_m}]\!](\boldsymbol{v}) < [\![\overline{b_m}]\!](\boldsymbol{v})$, $[\![\overline{a_m}]\!](\boldsymbol{v}) \leq [\![\overline{b_m}]\!](\boldsymbol{v})$, $[\![\overline{a_m}]\!](\boldsymbol{v}) + [\![\overline{b_m}]\!](\boldsymbol{v}) = [\![\overline{c_m}]\!](\boldsymbol{v})$, *and* $[\![\overline{a_m}]\!](\boldsymbol{v}) - [\![\overline{b_m}]\!](\boldsymbol{v}) = [\![\overline{c_m}]\!](\boldsymbol{v})$.

We stress that the key here is that this can be done in polynomial time, and hence the formulae are of polynomial size—the fact that such formulae exist at all is obvious from the expressive completeness of propositional logic.

Now consider the game where Player One chooses an interval of length a in $[0, b-1]_{\mathbb{N}}$ (this interval is allowed to loop around the edges), and Player Two chooses a single number in the same range. Player One wins if his interval captures Player Two's number, and loses otherwise. Clearly, the value of the game is $^a/_b$, as can be evidenced by the equilibrium where Player One randomises equally over every interval and Player Two over every number. Note also that if a and b are coprime (which we can assume because all that means is that the fraction $^a/_b$ is maximally reduced) then this equilibrium is unique.[2]

This game has a Boolean representation that is polynomial in the bit-length of b.

Proposition 4.3. *Let* $\mathfrak{G}(v)$ *be a two-player zero-sum Boolean game of value* $v \in [0,1]_{\mathbb{Q}}$. *We can construct* $\mathfrak{G}(v)$ *in time polynomial in* $|v|$.

The idea is to allow Player One to choose, via truth assignments, the start and end points of his interval and Player Two to choose her number in the same fashion. We use the $\boldsymbol{Add}$ formula to ensure Player One is picking an interval of the right length, with the $\boldsymbol{Sub}$ and $\boldsymbol{Less}$ formulae coming into play in the case of a looping interval. To ensure both players stay within $[0, b-1]_{\mathbb{N}}$, we use the $\boldsymbol{Less}$ formula.

Proof. Let $v = {^a/_b}$. We introduce the notation $\boldsymbol{x}$ to represent the sequence of the logical constants $\top$ and $\bot$ that represents the binary representation of the integer x. Terms such as $\boldsymbol{Less}(\overline{y_m}, \boldsymbol{x})$ are interpreted in the obvious way.

The desired game is the following:

$$\begin{aligned}
\Phi_1 =& \{p_1, \ldots, p_m, q_1, \ldots, q_m, s_1, \ldots, s_m, t_1, \ldots, t_m\} \\
\Phi_2 =& \{r_1, \ldots, r_m\} \\
\gamma_1 =& \big(\boldsymbol{Sub}(\overline{q_m}, \overline{p_m}, \boldsymbol{a}) \wedge \boldsymbol{LessEq}(\overline{q_m}, \boldsymbol{b-1}) \wedge \boldsymbol{LessEq}(\overline{r_m}, \overline{q_m}) \wedge \boldsymbol{LessEq}(\overline{p_m}, \overline{r_m}) \\
& \wedge \boldsymbol{Add}([\![\overline{s_m}]\!], \boldsymbol{0}, \boldsymbol{0}) \wedge \boldsymbol{Add}([\![\overline{t_m}]\!], \boldsymbol{0}, \boldsymbol{0})\big) \\
& \vee \Big(\boldsymbol{Add}(\overline{s_m}, \overline{t_m}, \boldsymbol{a})) \wedge \boldsymbol{Sub}(\overline{q_m}, \boldsymbol{0}, \overline{s_m}) \wedge \boldsymbol{Sub}(\boldsymbol{b-1}, \overline{p_m}, \overline{t_m}) \\
& \wedge \big(\boldsymbol{LessEq}(\overline{r_m}, \overline{q_m}) \vee \boldsymbol{LessEq}(\overline{p_m}, \overline{r_m})\big)\Big) \\
& \vee \boldsymbol{Less}(\boldsymbol{b-1}, \overline{r_m}).
\end{aligned}$$

The endpoints of Player One's interval are $[\![\overline{p_m}]\!]$ and $[\![\overline{q_m}]\!]$. The first disjunct deals with the case where Player One names a non-looping interval and captures Player Two's choice, $[\![\overline{r_m}]\!]$. Extraneous variables are set to false ($[\![\overline{s_m}]\!] + 0 = 0$) to ensure the equilibrium is unique.

In the case where Player One's interval is looping, we require him to tell us how much of it is on the right end ($[\![\overline{t_m}]\!]$) and on the left ($[\![\overline{s_m}]\!]$). The second disjunct verifies that $[\![\overline{t_m}]\!]$ and $[\![\overline{s_m}]\!]$ add to a, that

[2] Intuitively obvious, though the search for a proof took longer than the author would care to admit.

$[\![\overline{s_m}]\!]$ is equal to $[\![\overline{q_m}]\!]$ (that is $[\![\overline{q_m}]\!] - 0$), and that $[\![\overline{t_m}]\!]$ is the difference between $[\![\overline{p_m}]\!]$ and the end of the interval. This established, the victory is awarded to Player One should $[\![\overline{r_m}]\!]$ lie between $[\![\overline{t_m}]\!]$ and $b-1$, or 0 and $[\![\overline{s_m}]\!]$.

The last disjunct captures the case where Player Two picks a number outside of $[0, b-1]_{\mathbb{N}}$, in which case Player One is awarded the win. As the game is zero-sum, there is no need to explicitly specify Player Two's goal formula as it is simply $\neg\gamma_1$. □

While $\mathfrak{G}(v)$ is powerful and allows us to encode many things, it has a stark limitation—v has to be set ex ante. This means, among other things, that if a game has a large range of payoffs then we would need to define a new gadget game for every possible outcome; at that point it is worth asking whether we would not be better off simply using the normal form instead.

We can improve on this—it is possible to construct gadget games the value of which adapts to the state of the play, although we do not have need of these in the current work.

Proposition 4.4. *Let $\mathscr{G}(\overline{v_m})$ be a two-player zero-sum Boolean game of value $[\![\overline{v_m}]\!]/2^m$. I.e., the value of $\mathscr{G}(\overline{v_m})$ is contingent on the strategies chosen by the controller of the v_i variables. We can construct $\mathscr{G}(\overline{v_m})$ in time polynomial in m.*

The difference between this construction and $\mathfrak{G}(v)$ is that instead of verifying that the length of the interval is the numerator of v, it now has to be of length $[\![\overline{v_m}]\!](\boldsymbol{v})$. Obviously if Player One controls $\overline{v_m}$ then he would have a strong incentive to rig the dice somewhat, and within any construction this would need to be addressed by imposing restrictions on what he can do with $\overline{v_m}$ elsewhere in the game.

5 RationalNash and IrrationalNash

Theorem 5.1. RationalNash *for Boolean games is* NEXP*-hard.*

Proof. Theorem 3 in [4] gives an example of a three-player win-lose game, call it G_1', that has only irrational equilibria. Players One and Two have two strategies each, but Player Three has three, and as such the game as given does not have a Boolean representation. However, the game has the *positive utility property*: for any choice of strategies by two players, the third has a response that will yield him strictly positive utility. We can thus extend G_1' into G_1 that has a fourth strategy for Player Three, which operates as follows:

1. Player Three's payoff for choosing the fourth strategy is zero.
2. Player One and Two's payoff from any profile where Player Three chooses the fourth strategy are the same as if Player Three chose his first (this is arbitrary) strategy instead.

Observe that in no equilibrium of G_1 would Player Three attach positive weight to his fourth strategy. Thus every equilibrium of G_1 would correspond to an equilibrium of G_1', and hence be irrational, and G_1 has a Boolean representation.

Let (G_2, v) be an instance of $\exists$GuaranteeNash with three players.[3]

The idea of the construction is to let the players choose to play in G_1 or G_2 in such a way that every equilibrium involves all the players choosing the same game to play in, and that there is an equilibrium in which the players choose to play in G_2 if and only if (G_2, v) is a positive instance of $\exists$GuaranteeNash. This will prove the theorem—if $(G_2, v) \in \exists$GuaranteeNash then there is an equilibrium where the players play in G_2, and hence there is a rational-valued equilibrium (namely, the equilibrium of G_2 that

[3] The proof in [21] uses six, but the reduction to three is trivial.

guarantees Player One a payoff of v). If $(G_2, v) \notin \exists$GUARANTEENASH then every equilibrium of the game must be in G_1, and those are irrational.

The desired Boolean game G consists of four players, the three from G_1 and G_2 as well as a new O, and is defined as follows:

$$
\begin{aligned}
\Phi_{i\leq 3} &= \mathrm{var}_i(G_1) \cup \mathrm{var}_i(G_2) \cup \{\, \mathit{Choice}_i \,\} \cup \mathrm{var}_i(\mathfrak{G}_i),\\
\Phi_O &= \mathrm{var}_O(\mathfrak{G}_1(v)) \cup \mathrm{var}_O(\mathfrak{G}_2(1/2)) \cup \mathrm{var}_O(\mathfrak{G}_3(1/2)),\\
\gamma_1 &= \Big(\bigwedge_{i\leq 3} \mathit{Choice}_i \wedge \gamma_1(G_2)\Big) \vee \Big(\bigwedge_{i\leq 3} \neg\mathit{Choice}_i \wedge \gamma_1(G_1)\Big)\\
&\quad \vee \big(\neg\mathit{Choice}_1 \wedge (\mathit{Choice}_2 \vee \mathit{Choice}_2) \wedge \gamma_1(\mathfrak{G}_1(v))\big),\\
\gamma_2 &= \Big(\bigwedge_{i\leq 3} \mathit{Choice}_i \wedge \gamma_2(G_2) \vee \gamma_2(\mathfrak{G}_2(1/2))\Big) \vee \Big(\bigwedge_{i\leq 3} \neg\mathit{Choice}_i \wedge \gamma_2(G_1)\Big)\\
&\quad \vee \big(\neg\mathit{Choice}_2 \wedge (\mathit{Choice}_1 \vee \mathit{Choice}_3) \wedge \gamma_2(\mathfrak{G}_2(1/2))\big),\\
\gamma_3 &= \Big(\bigwedge_{i\leq 3} \mathit{Choice}_i \wedge \gamma_3(G_2) \vee \gamma_3(\mathfrak{G}_3(1/2))\Big) \vee \Big(\bigwedge_{i\leq 3} \neg\mathit{Choice}_i \wedge \gamma_3(G_1)\Big)\\
&\quad \vee \big(\neg\mathit{Choice}_3 \wedge (\mathit{Choice}_1 \vee \mathit{Choice}_2) \wedge \gamma_3(\mathfrak{G}_3(1/2))\big),\\
\gamma_O &= \gamma_O(\mathfrak{G}_1(v)) \wedge \gamma_O(\mathfrak{G}_2(1/2)) \wedge \gamma_O(\mathfrak{G}_3(1/2)).
\end{aligned}
$$

In the above we use $\mathrm{var}_i(G')$ to denote the set of Player i's variables in the game G', and $\gamma_i(G')$ to denote Player i's goal formula. In the gadget games $\mathfrak{G}_i(v)$, Player i is the interval-player (and hence can expect a utility of v) and Player O is the number-player.

To aid the reader in interpreting the goal formulae of players One, Two and Three, note that they consist of three mutually exclusive disjuncts. The first of these, where all the players set their Choice_i variable to *true*, represents the players agreeing to play in G_2. The second, with every Choice_i set to false, represents the players agreeing to play in G_1. The last disjunct represents the case of failed coordination; note that in that case only the players who have chosen to play in G_1 can expect to get any utility.

Suppose $(G_2, v) \in \exists$GUARANTEENASH. We claim there is a rational-valued equilibrium where Players One through Three set Choice_i to *true*, play the rational-valued equilibrium of G_2 that satisfies the payoff constraint over $\mathrm{var}_i(G_2)$, the equilibrium strategy over $\mathrm{var}_i(\mathfrak{G}_i)$ and every other variable to *false*. Player O plays the equilibrium strategy in all his gadget games.

Player O has no incentive to deviate as he wins if and only if he wins three independent games, and all those are currently in equilibrium. Player One has no incentive to deviate while Choice_1 is *true*: he is indifferent about what he does with $\mathrm{var}_1(G_1)$ and $\mathrm{var}_1(\mathfrak{G}_1(v))$ as those variables do not affect his ability to satisfy $\gamma_1(G_2)$, and he has no incentive to deviate over $\mathrm{var}_1(G_2)$ as G_2 is in equilibrium. Should he set Choice_1 to *false*, then his utility will depend only on his ability to satisfy $\gamma_1(\mathfrak{G}_1(v))$. The probability of that is v, which is at least how much he is getting in the current profile. Player Two (symmetrically, Three) has no incentive to deviate over $\mathrm{var}_2(\mathfrak{G}_2(1/2))$ or $\mathrm{var}_2(G_2)$ as those games are in equilibrium, and the variables in $\mathrm{var}_2(G_1)$ do not affect her current utility. If she deviates by setting Choice_2 to *false*, then she will be getting a utility of $1/2$; whereas in the current profile she is getting $1 - 1/2 \cdot x$, where x is her probability of losing G_2. As x is at most 1, such a deviation could do no better.

Now suppose $(G_2, v) \notin \exists$GUARANTEENASH. We claim that every equilibrium of G involves Choice_i being set to *false* with probability 1. Suppose otherwise, and consider an equilibrium $\boldsymbol{\sigma}$ in which p, q and r are the probabilities of players One, Two and Three respectively opting to play in G_2. Note that this means there is a pqr chance of the players ending up in G_2; if all three are non-zero, then that means the players must be playing an equilibrium strategy in G_2, as otherwise they could replicate the profitable deviation over $\mathrm{var}_i(G_2)$, keeping the rest of the strategy the same, and they would get the same gain in utility scaled by pqr. This being the case, observe that Player One's utility from the profile is $pqr \cdot y + (1-p)(1-(1-q)(1-r)) \cdot v + (1-p)(1-q)(1-r) \cdot x$ (payoff from G_2, from miscoordination, and from G_1), with y being strictly less than v. Player One could increase his payoff by reducing p to zero. This means that $\boldsymbol{\sigma}$ must have at least one of p, q or r equal to zero, and hence a play in G_2 can never eventuate. This established, suppose, without loss of generality, that $q \neq 0$. This gives Player Two a q chance of landing in a situation where she sets $Choice_2$ to *true* whilst the other $Choice_i$ variables are not, and hence giving her a utility of zero. This cannot be an equilibrium, as Player Two could deviate by reducing q to zero. But this means that $p = q = r = 0$, which leads us to accept that every equilibrium of G involves the players playing in G_1. As those equilibria are irrational, this proves the theorem. □

Theorem 5.2. IRRATIONALNASH *for Boolean games is* NEXP*-hard.*

Proof. Let G_1 be a Boolean game with the positive payoff property as before, and (G_2, v) an instance of $\exists$GUARANTEENASH. Unlike the previous proof, however, we assume the players of G_1 and G_2 are disjoint. The constructed Boolean game, G, has all the players of G_1 and G_2 as well as two new players C and O.

The idea of the proof here is that player C is given the power to decide whether or not the games in G_1 and G_2 will be played. If they are, he gets Player One's utility from G_2; if they are not, he gets v. This will ensure that if (G_2, v) is a positive instance then there will be an equilibrium where Player C decides to play, and the players in G_1 will ensure that this equilibrium is irrational. If (G_2, v) is a negative instance, our construction will ensure that the game will have a unique, rational equilibrium in which the component games are not played.

We give every player except O a new variable, $abstain_i$. Player C is characterised as follows:

$$\Phi_C = \{abstain_C\},$$
$$\gamma_C = (\neg abstain_C \wedge \neg abstain_1 \wedge \gamma_1(G_2)) \vee (abstain_C \wedge \gamma_C(\mathfrak{G}_C(v))).$$

That is, he can choose to abstain and get a utility of v, or if both C and Player One choose to play, C gets Player One's utility in G_2.

Player i, without loss of generality from G_1, is as follows:

$$\Phi_i = \mathrm{var}_i(G_1) \cup \{abstain_i\},$$
$$\gamma_i = (\gamma_i(G_1) \wedge \neg abstain_i \wedge \neg abstain_C) \vee \big((\bigwedge_{i \in N} abstain_i) \wedge (\bigwedge_{p \in \mathrm{var}_i(G_1)} \neg p)\big).$$

If both Player C and Player i chooses to play, then Player i gets the utility from G_1. If every player chooses to abstain and Player i sets all his variables to *false*, then Player i wins. In any other circumstance, he loses.

Player O simply plays $\mathfrak{G}_C(v)$ against C.

Suppose $(G_2, v) \in \exists$GUARANTEENASH. Consider the profile where the players in G_1 play any equilibrium of G_1, the players in G_2 the rational-valued equilibrium satisfying the payoff constraint in G_2, Player C sets $abstain_C$ to *false* and Player O plays the equilibrium strategy. This profile is in equilibrium: the players in G_1 and G_2 have no incentive to deviate within their games because they are in equilibrium, and deviating with $abstain_i$ would yield them a utility of zero because Player C is choosing to play. Player O is playing the equilibrium strategy, and Player C is getting Player One's utility—which is at least v—and should he deviate to abstaining, then v is all he would get from $\mathfrak{G}_C(v)$. This equilibrium is irrational because every equilibrium of G_1 is irrational.

Now suppose $(G_2, v) \notin \exists$GUARANTEENASH. We claim the game has a unique rational equilibrium where every player abstains, the players from G_1 and G_2 set all their variables to *false*, and C plays the equilibrium strategy of $\mathfrak{G}_C(v)$ against O. First observe that this is indeed a rational equilibrium—any Player who chooses to play will get a utility of 0 because C is abstaining, and C would get a utility of 0 from playing because Player One is abstaining.

Next, we claim that every equilibrium involves Player C abstaining with probability 1. As in the previous proof, if Player C chooses to play then the subgame G_2 would have to be in equilibrium, and Player One's (hence C's) payoff in that would be strictly less than v, which is what he could get from the gadget game against O. Given that Player C is abstaining, every i would have an incentive to abstain. This proves the theorem. □

6 Future work

Two directions for future research are apparent. The first is the case of function, or search problems as opposed to the decision problems studied so far. Precious little has been said about functions even in the broader literature on concise representations, let alone for Boolean games—the best known bounds for FINDNASH are TFNEXP and EXP-hard (not even FEXP-hard). This is something that bears addressing as for all the convenience of decision problems, the more natural algorithmic questions in game theory require a richer response than YES or NO.

The second is on approximate or probabilistic reasoning. So far the picture we have painted suggests that despite their apparent simplicity, Boolean games are every bit as hard as the seemingly much more complicated frameworks of Turing machine ([1] or circuit ([17],[30]) games. One has to wonder whether that remains true when we consider the approximation problems studied by [17] and [30].

Acknowledgements

Egor Ianovski is supported by a scholarship from the Oxford-Man Institute of Quantitative Finance.

References

[1] Carme Àlvarez, Joaquim Gabarró & Maria Serna (2005): *Pure Nash Equilibria in Games with a Large Number of Actions*. In Joanna Jędrzejowicz & Andrzej Szepietowski, editors: *Mathematical Foundations of Computer Science 2005*, *Lecture Notes in Computer Science* 3618, Springer Berlin Heidelberg, pp. 95–106.

[2] Vittorio Bilò (2007): *On Satisfiability Games and the Power of Congestion Games*. In Ming-Yang Kao & Xiang-Yang Li, editors: *Algorithmic Aspects in Information and Management*, *Lecture Notes in Computer Science* 4508, Springer Berlin Heidelberg, pp. 231–240.

[3] Vittorio Bilò & Marios Mavronicolas (2011): *Complexity of Rational and Irrational Nash Equilibria*. In Giuseppe Persiano, editor: *Algorithmic Game Theory*, *Lecture Notes in Computer Science* 6982, Springer Berlin Heidelberg, pp. 200–211, doi:$10.1007/978-3-642-24829-0_19$. Available at `http://dx.doi.org/10.1007/978-3-642-24829-0_19`.

[4] Vittorio Bilò & Marios Mavronicolas (2012): *The Complexity of Decision Problems about Nash Equilibria in Win-Lose Games*. In Maria Serna, editor: *Algorithmic Game Theory*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 37–48.

[5] Elise Bonzon, Marie-Christine Lagasquie-Schiex & Jérôme Lang (2009): *Dependencies between players in Boolean games*. *Int. J. Approx. Reasoning* 50(6), pp. 899–914.

[6] Elise Bonzon, Marie-Christine Lagasquie-Schiex & Jérôme Lang (2012): *Effectivity functions and efficient coalitions in Boolean games*. *Synthese* 187(1), pp. 73–103.

[7] Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang & Bruno Zanuttini (2006): *Boolean games revisited*. In: *Proceedings of ECAI 2006*, pp. 265–269.

[8] Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang & Bruno Zanuttini (2009): *Compact preference representation and Boolean games*. *Autonomous Agents and Multi-Agent Systems* 18(1), pp. 1–35.

[9] 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.

[10] Bruno Codenotti & Daniel Štefankovič (2005): *On the computational complexity of Nash equilibria for* $(0,1)$ *bimatrix games*. *Information Processing Letters* 94(3), pp. 145 – 150, doi:http://dx.doi.org/10.1016/j.ipl.2005.01.010. Available at `http://www.sciencedirect.com/science/article/pii/S0020019005000281`.

[11] Vincent Conitzer & Tuomas Sandholm (2008): *New complexity results about Nash equilibria*. *Games and Economic Behavior* 63(2), pp. 621–641.

[12] R. W. Cottle & G. B. Dantzig (1968): *Complementary pivot theory of mathematical programming*. *Linear Algebra and its Applications* 1, pp. 103–125.

[13] Paul E. Dunne & Wiebe van der Hoek (2004): *Representation and Complexity in Boolean Games*. In Jóse Júlio Alferes & João Leite, editors: *Logics in Artificial Intelligence*, *Lecture Notes in Computer Science* 3229, Springer-Verlag, pp. 347–359.

[14] Paul E. Dunne, Wiebe van der Hoek, Sarit Kraus & Michael Wooldridge (2008): *Cooperative Boolean games*. In: *AAMAS'08*, pp. 1015–1022.

[15] Paul E. Dunne & Michael Wooldridge (2012): *Towards tractable Boolean games*. In: *Proceedings of AAMAS '12*, pp. 939–946.

[16] Ulle Endriss, Sarit Kraus, Jérôme Lang & Michael Wooldridge (2011): *Designing incentives for Boolean games*. In: *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '11, pp. 79–86.

[17] Lance Fortnow, Valentine Kabanets, Russell Impagliazzo & Christopher Umans (2005): *On the complexity of succinct zero-sum games*. In: *IEEE Conference on Computational Complexity*, pp. 323–332.

[18] Itzhak Gilboa & Eitan Zemel (1989): *Nash and correlated equilibria: Some complexity considerations*. *Games and Economic Behavior* 1(1), pp. 80–93.

[19] John Grant, Sarit Kraus, Michael Wooldridge & Inon Zuckerman (2011): *Manipulating Boolean games through communication*. In: *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, AAAI Press, pp. 210–215.

[20] Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer & Cees Witteveen (2001): *Boolean games*. In: *Proceedings of TARK 2001*, pp. 287–298.

[21] Egor Ianovski & Luke Ong (2014): *EGuaranteeNash for Boolean Games Is NEXP-Hard*. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. Available at `http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/8002`.

[22] Sarit Kraus & Michael Wooldridge (2012): *Delegating Decisions in Strategic Settings*. In: *ECAI 2012*, pp. 468–473.

[23] Vadim Levit, Tal Grinshpoun, Amnon Meisels & Ana L. C. Bazzan (2013): *Taxation search in Boolean games*. In: *AAMAS*, pp. 183–190. Available at `http://dl.acm.org/citation.cfm?id=2484952`.

[24] Thomas Lukasiewicz & Azzurra Ragone (2009): *A Combination of Boolean Games with Description Logics for Automated Multi-Attribute Negotiation*. In: *Description Logics*.

[25] Enrico Marchioni & Michael Wooldridge (2014): *Łukasiewicz Games*. In: *Proceedings of the Thirteenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2014)*.

[26] Marios Mavronicolas, Burkhard Monien & Klaus W. Wagner (2007): *Weighted Boolean Formula Games*. In Xiaotie Deng & FanChung Graham, editors: *Internet and Network Economics*, *Lecture Notes in Computer Science* 4858, Springer Berlin Heidelberg, pp. 469–481.

[27] John Nash (1951): *Non-Cooperative Games*. *Annals of Mathematics* 54(2), pp. 286–295.

[28] Matei Popovici & Ciprian Dobre (2012): *A Game-theoretic Approach to Cooperation in Multi-agent Systems*. In: *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, WIMS '12, ACM, New York, NY, USA, pp. 54:1–54:4, doi:10.1145/2254129.2254197. Available at `http://doi.acm.org/10.1145/2254129.2254197`.

[29] Luigi Sauro & Serena Villata (2013): *Dependency in Cooperative Boolean Games*. *Journal of Logic and Computation* 23(2), pp. 425–444.

[30] Grant R. Schoenebeck & Salil Vadhan (2012): *The Computational Complexity of Nash Equilibria in Concisely Represented Games*. *ACM Trans. Comput. Theory* 4(2), pp. 4:1–4:50.

[31] Paolo Turrini (2013): *Endogenous Boolean games*. In: *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, AAAI Press, pp. 390–396.

[32] John von Neumann (1928): *Zur Theorie der Gesellschaftsspiele*. *Mathematische Annalen* 100(1), pp. 295–320.

[33] John von Neumann & Oscar Morgenstern (1947): *Theory of Games and Economic Behavior*, 2nd edition. Princeton University Press.

[34] Michael Wooldridge (2012): *Bad equilibria (and what to do about them)*. In: *ECAI 2012*, pp. 6–11.

12

Regular Paper

Dealing with imperfect information in Strategy Logic

Sophia Knight
LORIA - CNRS / Université de Lorraine
Nancy, France
sophia.knight@gmail.com

Bastien Maubert
LORIA - CNRS / Université de Lorraine
Nancy, France
bastien.maubert@gmail.com

We propose an extension of Strategy Logic (SL), in which one can both reason about strategizing under imperfect information and about players' knowledge. One original aspect of our approach is that we do not force strategies to be uniform, i.e. consistent with the players' information, at the semantic level; instead, one can express in the logic itself that a strategy should be uniform. To do so, we first develop a "branching-time" version of SL with perfect information, that we call BSL, in which one can quantify over the different outcomes defined by a partial assignment of strategies to the players; this contrasts with SL, where temporal operators are allowed only when all strategies are fixed, leaving only one possible play. Next, we further extend BSL by adding distributed knowledge operators, the semantics of which rely on equivalence relations on partial plays. The logic we obtain subsumes most strategic logics with imperfect information, epistemic or not.

1 Introduction

Over the past decade, investigation of logical systems for studying strategic abilities has thrived in the areas of artificial intelligence and multi-agent systems. However, there is still no satisfying logical framework to model, specify and analyze such systems. One of the proposals most studied so far is Alternating-time Temporal Logic (ATL) [1], in which one can specify what objectives coalitions of agents can achieve. Several extensions were introduced (ATL^*, game logics...), but all of these logics fail to model non-cooperative situations where agents follow individual objectives. It is well known that studying this kind of situation requires solution concepts from game theory, such as Nash equilibria, that cannot be expressed in ATL or its extensions.

To address this shortcoming, Chatterjee, Henzinger and Piterman recently introduced Strategy Logic (SL) [9]. This logic subsumes all extensions of ATL, and because it considers strategies as first-order citizens in the language, it can express fundamental game-theoretic concepts such as Nash Equilibria or dominated strategies. SL has recently been extended and intensively studied [19, 17, 18]. Relevant fragments enjoying nice computational characteristics have been identified. In particular, the syntactic fragment SL[1G] (One-Goal Strategy Logic) is strictly more expressive than ATL^*, but not computationally more expensive [17].

However, despite its great expressiveness, there is one fundamental feature of most real-life situations that SL lacks, which is imperfect information. An agent has imperfect information if she does not know the exact state of the system at every moment, but only has access to an approximation of it. Considering agents with imperfect information raises two major theoretical issues. The first one concerns strategizing under imperfect information. Indeed, in this context an agent's strategy must prescribe the same choice in all situations that are indistinguishable to the agent. Such strategies are called *uniform strategies*, and this requirement deeply impacts the task of computing strategies [21]. The second main theoretical challenge relates to uncertainty, deeply intertwined with imperfect information, and it consists of representing and reasoning about agents' knowledge. Over the past decades, much effort has been put into devising logical systems that address this issue, first in static settings [10] and later adding dynamics [11, 5].

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

Concerning ATL, many variants have been introduced that deal with imperfect information [12, 14, 22, 13]. Some of these numerous logics deal with strategizing under imperfect information, some with reasoning about knowledge; because it is not natural to reason about the knowledge of agents with imperfect information without treating the strategic aspects accordingly, as argued in [14], some treat both aspects. But there still remain a number of logics that do so, and that essentially differ in the semantics of the strategic operator: how much memory do agents have? should agents simply have a strategy to achieve some goal? Or should they *know that there is* a strategy to do so? Or *know a strategy* that works? The two last notions are usually referred to as *de dicto* and *de re* strategies, respectively [14].

About SL, very few works have considered imperfect information. [2] and [8] propose epistemic extensions of SL, but they do not require strategies to be uniform *i.e.* being consistent with the agents' information. In [3], an epistemic strategy logic is proposed in which uniform strategies are considered, and interestingly the *de re* semantics of the strategic operator can be expressed in the *de dicto* semantics, providing some flexibility. However, how much memory strategies use, and whether they should be uniform or not, still has to be hardwired in the semantics.

In this work, we propose yet another epistemic strategy logic, with the purpose of getting rid of the constraint of enforcing what kind of strategies are to be used at the semantic level. To do so, we first develop a "branching-time" version of SL with perfect information. In SL, temporal operators are allowed only when all strategies are fixed, leaving only one possible play. We relax this constraint by introducing a *path quantifier*, which quantifies over the different outcomes defined by a partial assignment of strategies to the agents. This enables the comparison of the various outcomes of a strategy. Because it will be important, for instance to express the uniformity of a strategy, to consider all the possible outcomes of a strategy assigned to an agent *a*, we need a way to remove in an assignment the bindings of all agents but *a*. We thus introduce an *unbinding operator*. We call the resulting logic Branching-time Strategy Logic (BSL), and we prove by providing linear translations in both directions that it has the same expressive power and same computational complexity as SL. We also present a variant of BSL, called BSL^+, which can in addition refer to the actions chosen by each agent at each moment, and we conjecture that it is strictly more expressive than SL and BSL. Next, we define our Epistemic Strategy Logic (ESL) by further extending BSL with distributed knowledge operators, the semantics of which rely on equivalence relations on partial plays. We do not change the semantics of the strategy quantifier to require them to be uniform, or *de re*, or *de dicto*, or *memoryless*, but we rather show that all of these properties of strategies can be expressed in the language, which thus subsumes most, if not all, the variants of epistemic strategic logics with imperfect information that we know about.

The paper is structured as follows. In Section 2 we recall the models, syntax and semantics of SL. In Section 3, we define BSL and BSL^+, and we prove that SL and BSL are equiexpressive. We then present ESL in Section 4, where we also show how it can express various classic properties of strategies. We conclude and discuss future work in Section 5. Some proofs are omitted by lack of space.

2 Preliminaries

Let AP be a countable non-empty set of *atomic propositions*, Ag a non-empty finite set of *agents* and Ac a non-empty finite set of *actions*. We let $Dc = Ac^{Ag}$ be the set of possible *decisions*. For $d \in Dc$ and $a \in Ag$, $d(a)$ is the action taken by Agent a in decision d.

2.1 Concurrent game structures

A *concurrent game structure* (CGS) is a tuple $G = (Q, \delta, s_\iota, \mu)$, where Q is a countable non-empty set of *states*, $\delta : Q \times Dc \to Q$ is a *transition function*, s_ι is the *initial state* and $\mu : Q \to 2^{AP}$ is a *valuation function*. A *path* is an infinite word $\pi = s_0(d_1, s_1) \ldots \in Q \cdot (Dc \times Q)^\omega$ such that for all $i \geq 0$, $s_{i+1} = \delta(s_i, d_{i+1})$, and an *initial path* ρ is a finite prefix of a path. In the following, we shall write $s_0 d_1 s_1 \ldots$ instead of $s_0(d_1, s_1)\ldots$, and similarly for initial paths. For a state s we denote by $Paths_\omega(s)$ (resp. $Paths_*(s)$) the set of paths (resp. initial paths) that start in s, *i.e.* for which $s_0 = s$. We also let $Paths_\omega$ (resp. $Paths_*$) be the set of all paths (resp. initial paths). For a path $\pi = s_0 d_1 s_1 \ldots$, for $i, j \geq 0$, we let $\pi[i] := s_i$, $\pi_{\leq i} := s_0 \ldots d_i s_i$, $\pi_{\geq i} := s_i d_{i+1} s_{i+1} \ldots$ and $\pi[i, j] := s_i d_{i+1} \ldots d_j s_j$. For an initial path $\rho = s_0 \ldots d_n s_n$, $\mathrm{last}(\rho) := s_n$ is its last state and $|\rho| := n$ is the index of its last state. Given two initial paths $\rho = s_0 d_1 s_1 \ldots d_n s_n$ and $\rho' = s'_0 d'_1 s'_1 \ldots d'_m s'_m$ such that $s_n = s'_0$, we let $\rho \cdot \rho' := s_0 d_1 s_1 \ldots d_n s_n d'_1 s'_1 \ldots d'_m s'_m$ be their concatenation.

A *strategy* is a total function $\sigma : Paths_* \to Ac$ that assigns an action to each initial path, and we let Str be the set of all strategies. Also, given a strategy σ and an initial path $\rho \in Paths_*$ ending in state s, we define the ρ*-translation* of σ as the strategy σ^ρ such that for all initial paths $\rho' \in Paths_*(s)$, $\sigma^\rho(\rho') := \sigma(\rho \cdot \rho')$, and for all initial paths $\rho' \in Paths_*(s')$ with $s' \neq s$, $\sigma^\rho(\rho') = \sigma(\rho')$.

Let Var be a countably infinite set of *variables*. An *assignment* is a partial function $\chi : Ag \cup Var \rightharpoonup Str$, assigning to each agent and variable in its domain a strategy. For an assignment χ, an agent a and a strategy σ, $\chi[a \mapsto \sigma]$ is the assignment of domain $dom(\chi) \cup \{a\}$ that maps a to σ and is equal to χ on the rest of its domain, and similarly for $\chi[x \mapsto \sigma]$ where x is a variable; also, $\chi[a \mapsto ?]$ is the assignment of domain $dom(\chi) \setminus \{a\}$, on which it is equal to χ. Given an assignment χ and a state s, we define the *outcome* of χ in s, written $\mathrm{Out}(s, \chi)$, as the set of paths $\pi = s_0 d_1 s_1 \ldots$ such that $s_0 = s$, and for all $k \geq 0$, for every agent a in the domain of χ, $d_{k+1}(a) = \chi(a)(\pi_{\leq k})$. We say that an assignment χ is *complete* if it assigns a strategy to each agent, *i.e.* $Ag \subseteq dom(\chi)$. Given an assignment χ and an initial path ρ ending in state s, we define the ρ-translation of χ as the assignment χ^ρ such that $dom(\chi^\rho) = dom(\chi)$, and for all $l \in dom(\chi^\rho)$, $\chi^\rho(l) := \chi(l)^\rho$ (l being either a variable or an agent).

Finally, we want (some of) our logics to be able to talk about the precise actions taken by agents. To do so, we consider the following set of *action propositions*: $AcP := \{p^a_c \mid c \in Ac \text{ and } a \in Ag\}$, and we let $AP^+ := AP \uplus AcP$. In the following, we will therefore always assume that CGSs are *unfolded*, such that each state s is reached by one unique transition through some decision d_s, except the initial state s_ι which has no incoming transition. We can thus extend the valuation function μ into μ^+ as follows: $\mu^+(s_\iota) := \mu(s_\iota)$, and for every state $s \neq s_\iota$, $\mu^+(s) := \mu(s) \cup \{p^a_{d_s(a)} \mid a \in Ag\}$.

2.2 Strategy Logic

We recall the syntax and semantics of Strategy Logic (SL). First, the set of formulas in SL is given by the following grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi \mid \langle\langle x \rangle\rangle\varphi \mid (a, x)\varphi$$

where $p \in AP$, $x \in Var$ and $a \in Ag$.

Notice that SL-formulas cannot talk about agents' actions.

We define $\top$ as $p \vee \neg p$. Dual operators can be defined as usual: $\bot := \neg\top, \varphi \wedge \varphi' := \neg(\neg\varphi \vee \neg\varphi'), \varphi\mathbf{R}\varphi' := \neg(\neg\varphi\mathbf{U}\neg\varphi')$ and $[\![x]\!]\varphi := \neg\langle\langle x \rangle\rangle\neg\varphi$, and we also define the classic temporal operators "eventually" and "always": $\mathbf{F}\varphi := \top\mathbf{U}\varphi$, and $\mathbf{G}\varphi := \varphi\mathbf{U}\bot$. Recall that $\langle\langle x \rangle\rangle$ is the *strategy quantifier*, and (a,x) is the *binding operator*: $\langle\langle x\rangle\rangle\varphi$ reads as "there exists a strategy x such that φ", and $(a,x)\varphi$ reads as "φ holds after agent a is bound to the strategy denoted by x".

For a formula φ, $Free(\varphi) \subseteq Var$ is the set of *free variables* in φ, *i.e.* the set of variables x that occur in φ without being under the scope of some quantification $\langle\langle x\rangle\rangle$. In the following, given a formula φ, an *assignment for* φ refers to an assignment χ such that $Free(\varphi) \subseteq dom(\chi)$.

Let φ be an SL-formula. Given a CGS $G = (Q,\delta,s_\iota,\mu)$, an assignment χ for φ and a state $s \in Q$, the semantics of φ in G with assignment χ at state s is defined inductively as follows:

$G,\chi,s \models_{\mathsf{SL}} p$ if $p \in \mu(s)$
$G,\chi,s \models_{\mathsf{SL}} \neg\varphi$ if $G,\chi,s \not\models_{\mathsf{SL}} \varphi$
$G,\chi,s \models_{\mathsf{SL}} \varphi \vee \varphi'$ if $G,\chi,s \models_{\mathsf{SL}} \varphi$ or $G,\chi,s \models_{\mathsf{SL}} \varphi'$
$G,\chi,s \models_{\mathsf{SL}} \langle\langle x\rangle\rangle\varphi$ if there exists $\sigma \in Str(s)$ such that $G,\chi[x \mapsto \sigma],s \models_{\mathsf{SL}} \varphi$
$G,\chi,s \models_{\mathsf{SL}} (a,x)\varphi$ if $G,\chi[a \mapsto \chi(x)],s \models_{\mathsf{SL}} \varphi$

If, in addition, χ is complete, then

$G,\chi,s \models_{\mathsf{SL}} \mathbf{X}\varphi$ if $G,\chi^{\pi_{\leq 1}},\pi[1] \models_{\mathsf{SL}} \varphi$, where π is the only path in $\mathrm{Out}(s,\chi)$
$G,\chi,s \models_{\mathsf{SL}} \varphi\mathbf{U}\varphi'$ if there is $i \geq 0$ such that, letting π be the only path in $\mathrm{Out}(s,\chi)$, $G,\chi^{\pi_{\leq i}},\pi[i] \models_{\mathsf{SL}} \varphi'$, and for all $0 \leq j < i$, $G,\chi^{\pi_{\leq j}},\pi[j] \models_{\mathsf{SL}} \varphi$.

Finally, we define an *SL-sentence* to be an SL-formula φ such that $Free(\varphi) = \emptyset$ and every temporal operator in φ is under the scope of a binding for each agent.

3 Branching-time Strategy Logic

We now present a first extension of Strategy Logic. In SL, temporal operators are allowed only when every agent has been assigned a strategy, which leaves only one possible outcome. Here we relax this constraint: a temporal formula can be evaluated on the outcome of a *partial* strategy assignment. The outcome of such an assignment is a tree that contains all paths corresponding to all possible completions of the assignment, which is why we use the path quantification of branching-time temporal logic. We also add the *unbinding* operator as considered in *e.g.* [16], making it possible to unbind an agent from its strategy. We first show that the logic thus obtained, called BSL, has the same expressivity as SL, by providing linear translations in both directions. The unbinding operator is thus just convenient syntactic sugar. Then we further extend BSL by allowing it to refer to actions taken by agents, and obtain the logic BSL^+ that, we postulate, is strictly more expressive than SL and BSL. BSL has two advantages: first, the semantics is slightly cleaner than that of SL, as it is defined for all formulas and all assignments; second, the unbinding operator makes it possible to easily express that we unbind an agent, at no complexity cost. Finally, because it can explicitly refer to actions and consider outcomes of partial assignments, it is possible in BSL^+ to express properties of strategies, such as being memoryless or uniform, as we show in Section 4.

3.1 Syntax

The syntax of BSL adds two operators to SL. First, the *path quantifier*, borrowed from classic branching-time temporal logics: $\mathbf{E}\psi$ intuitively reads as "there exists an outcome of the currently fixed strategies in which φ holds". Second, the *unbinding operator*: $(a,?)\varphi$ means "φ holds after Agent a has been unbound from her strategy, if any". We define two variants, one (BSL) where formulas cannot talk about the actions taken by the agents, and one (BSL^+) where they can. Also, as for CTL^*, we find it convenient to distinguish between state and path formulas. Finally, the set of BSL-formulas (resp. BSL^+-formulas) is the set of state formulas given by the following grammar:

$$\begin{array}{ll} \text{State formulas:} & \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle x\rangle\rangle\varphi \mid (a,x)\varphi \mid (a,?)\varphi \mid \mathbf{E}\psi \\ \text{Path formulas:} & \psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi, \end{array}$$

where $p \in AP$ (resp. $p \in AP^+$), $x \in Var$ and $a \in Ag$.

Observe that $\mathsf{BSL} \subset \mathsf{BSL}^+$. In addition to the shorthand defined in Section 2.2, we also define the dual of the path quantifier: $\mathbf{A}\varphi := \neg\mathbf{E}\neg\varphi$. Finally, we write BSL^+_ψ (resp. BSL_ψ) for the set of BSL^+ (resp. BSL) path formulas.

3.2 Semantics

State formulas are evaluated in a state of (the unfolding of) a CGS, and path formulas in paths. Since BSL is a syntactical fragment of BSL^+, it is enough to define the latter's semantics.

Let $\varphi \in \mathsf{BSL}^+$ be a state formula (resp. let $\psi \in \mathsf{BSL}^+_\psi$ be a path formula), and let $G = (Q, \delta, q_\iota, \mu)$ be a CGS. Let $s \in G$ be a state, $\pi \in Paths_\omega$ a path, and let χ be an assignment for φ (resp. for ψ). The semantics of BSL^+ is defined inductively as follows:

$$\begin{array}{lll}
G,\chi,s \models_{\mathsf{BSL}} p & \text{if} & p \in \mu^+(s) \\
G,\chi,s \models_{\mathsf{BSL}} \neg\varphi & \text{if} & G,\chi,s \not\models_{\mathsf{BSL}} \varphi \\
G,\chi,s \models_{\mathsf{BSL}} \varphi \vee \varphi' & \text{if} & G,\chi,s \models_{\mathsf{BSL}} \varphi \text{ or } G,\chi,s \models_{\mathsf{BSL}} \varphi' \\
G,\chi,s \models_{\mathsf{BSL}} \langle\langle x\rangle\rangle\varphi & \text{if} & \text{there exists } \sigma \in Str \text{ such that } G,\chi[x \mapsto \sigma],s \models_{\mathsf{BSL}} \varphi \\
G,\chi,s \models_{\mathsf{BSL}} (a,x)\varphi & \text{if} & G,\chi[a \mapsto \chi(x)],s \models_{\mathsf{BSL}} \varphi \\
G,\chi,s \models_{\mathsf{BSL}} (a,?)\varphi & \text{if} & G,\chi[a \mapsto ?],s \models_{\mathsf{BSL}} \varphi \\
G,\chi,s \models_{\mathsf{BSL}} \mathbf{E}\psi & \text{if} & \text{there exists } \pi \in \mathrm{Out}(s,\chi) \text{ such that } G,\chi,\pi \models_{\mathsf{BSL}} \psi \\
 & & \\
G,\chi,\pi \models_{\mathsf{BSL}} \varphi & \text{if} & G,\chi,\pi[0] \models_{\mathsf{BSL}} \varphi \\
G,\chi,\pi \models_{\mathsf{BSL}} \neg\psi & \text{if} & G,\chi,\pi \not\models_{\mathsf{BSL}} \psi \\
G,\chi,\pi \models_{\mathsf{BSL}} \psi \vee \psi' & \text{if} & G,\chi,\pi \models_{\mathsf{BSL}} \psi \text{ or } G,\chi,\pi \models_{\mathsf{BSL}} \psi' \\
G,\chi,\pi \models_{\mathsf{BSL}} \mathbf{X}\psi & \text{if} & G,\chi^{\pi_{\leq 1}},\pi_{\geq 1} \models_{\mathsf{BSL}} \psi \\
G,\chi,\pi \models_{\mathsf{BSL}} \psi\mathbf{U}\psi' & \text{if} & \text{there is } i \geq 0 \text{ such that } G,\chi^{\pi_{\leq i}},\pi_{\geq i} \models_{\mathsf{BSL}} \psi', \text{ and} \\
 & & \text{for all } 0 \leq j < i,\ G,\chi^{\pi_{\leq j}},\pi_{\geq j} \models_{\mathsf{BSL}} \psi
\end{array}$$

The semantics of the unbinding operator comes without surprise: $(a,?)\varphi$ holds in an assignment if φ holds after we have removed a from the domain of this assignment. For the path quantifier, $\mathbf{E}\psi$ holds if there is an outcome of the current assignment in the current state that verifies ψ.

For a BSL^+-formula φ, we write $G,\chi \models_{\mathsf{BSL}} \varphi$ if $Paths_\omega(s_\iota),\chi,s_\iota \models_{\mathsf{BSL}} \varphi$. Classically, a *$BSL^+$-sentence* is a BSL^+-formula without free variables, and similarly for BSL-sentences. For a BSL^+-sentence φ, we write $G \models_{\mathsf{BSL}} \varphi$ if $G,\chi \models_{\mathsf{BSL}} \varphi$ for any assignment χ.

3.3 Expressivity of BSL

We establish that BSL and SL have the same expressivity, and postulate that BSL^+ is strictly more expressive than both logics. First, given two logics $\mathscr{L}$ and $\mathscr{L}'$ whose sentences are evaluated on CGS's, we say that *$\mathscr{L}'$ subsumes $\mathscr{L}$*, written $\mathscr{L} \preccurlyeq \mathscr{L}'$, if for every $\mathscr{L}$-sentence φ there is an $\mathscr{L}'$-sentence φ' such that, for every CGS G, $G \models \varphi$ if, and only if, $G \models \varphi'$. We say that $\mathscr{L}$ and $\mathscr{L}'$ are *equiexpressive* if $\mathscr{L} \preccurlyeq \mathscr{L}'$ and $\mathscr{L}' \preccurlyeq \mathscr{L}$. We say that *$\mathscr{L}'$ strictly subsumes $\mathscr{L}$*, written $\mathscr{L} \prec \mathscr{L}'$, if $\mathscr{L} \preccurlyeq \mathscr{L}'$ and $\mathscr{L}' \not\preccurlyeq \mathscr{L}$.

We start with the easy direction, showing that BSL subsumes SL.

Definition 1 *The translation* $\mathrm{tr} : SL \to BSL$ *is defined by induction as follows:*

$$\begin{array}{lcllcllcl}
\mathrm{tr}(p) & = & p & \mathrm{tr}(\neg\varphi) & = & \neg\mathrm{tr}(\varphi) & \mathrm{tr}(\varphi \vee \varphi') & = & \mathrm{tr}(\varphi) \vee \mathrm{tr}(\varphi') \\
\mathrm{tr}(\mathbf{X}\varphi) & = & \mathbf{EX}\mathrm{tr}(\varphi) & & & & \mathrm{tr}(\varphi\mathbf{U}\varphi') & = & \mathbf{E}\mathrm{tr}(\varphi)\mathbf{U}\mathrm{tr}(\varphi') \\
\mathrm{tr}(\langle\langle x\rangle\rangle\varphi) & = & \langle\langle x\rangle\rangle\mathrm{tr}(\varphi) & & & & \mathrm{tr}((a,x)\varphi) & = & (a,x)\mathrm{tr}(\varphi)
\end{array}$$

The following proposition easily follows from the fact that a complete assignment defines a unique path from any state.

Proposition 1 *For every CGS G, for every formula $\varphi \in SL$, assignment χ for φ state $s \in G$ such that $G,\chi,s \models_{SL} \varphi$ is defined, it holds that $G,\chi,s \models_{SL} \varphi$ if, and only if, $G,\chi,s \models_{BSL} \mathrm{tr}(\varphi)$.*

Proof sketch We only treat the case of the "next" operator, the one for "until" is similar and all the others are trivial. Assume that $G,\chi,s \models_{\mathsf{SL}} \mathbf{X}\varphi$ is defined. This means that χ is a complete assignment, hence $\mathrm{Out}(s,\chi)$ is a singleton, and the result follows from the semantics of SL and BSL. □

We now show that SL also subsumes BSL. Indeed, the path quantifier can be simulated by a series of existential strategy quantifications and corresponding bindings for the agents whose strategies are undefined in the current assignment. Concerning the unbinding operator, the idea is to remember, along the translation, which agents have been unbound, and use this information to correctly translate path quantifiers, as described above. Formally, we define a translation from BSL to SL, parameterized by the set of agents who are "currently" not bound to a strategy.

Definition 2 *Let $A \subseteq Ag$. The translations $\mathrm{tr}'_A : BSL \to SL$ and $\mathrm{tr}'^{\psi}_A : BSL_\psi \to SL$ are defined by mutual induction as follows:*

$$\begin{array}{lcllcl}
\mathrm{tr}'_A(p) & = & p & \mathrm{tr}'^{\psi}_A(\psi) & = & \mathrm{tr}'_A(\varphi) \\
\mathrm{tr}'_A(\neg\varphi) & = & \neg\mathrm{tr}'_A(\varphi) & \mathrm{tr}'^{\psi}_A(\neg\psi) & = & \neg\mathrm{tr}'^{\psi}_A(\psi) \\
\mathrm{tr}'_A(\varphi \vee \varphi') & = & \mathrm{tr}'_A(\varphi) \vee \mathrm{tr}'_A(\varphi') & \mathrm{tr}'^{\psi}_A(\psi \vee \psi') & = & \mathrm{tr}'^{\psi}_A(\psi) \vee \mathrm{tr}'^{\psi}_A(\psi') \\
\mathrm{tr}'_A(\langle\langle x\rangle\rangle\varphi) & = & \langle\langle x\rangle\rangle\mathrm{tr}'_A(\varphi) & \mathrm{tr}'^{\psi}_A(\mathbf{X}\psi) & = & \mathbf{X}\mathrm{tr}'^{\psi}_A(\psi) \\
\mathrm{tr}'_A((a,x)\varphi) & = & (a,x)\mathrm{tr}'_{A\setminus\{a\}}(\varphi) & \mathrm{tr}'^{\psi}_A(\psi\mathbf{U}\psi') & = & \mathrm{tr}'^{\psi}_A(\psi)\mathbf{U}\mathrm{tr}'^{\psi}_A(\psi') \\
\mathrm{tr}'_A((a,?)\varphi) & = & \mathrm{tr}'_{A\cup\{a\}}(\varphi) & & & \\
\mathrm{tr}'_A(\mathbf{E}\psi) & = & \langle\langle x_1\rangle\rangle\ldots\langle\langle x_k\rangle\rangle(a_{i_1},x_1)\ldots(a_{i_k},x_k)\mathrm{tr}'^{\psi}_A(\psi), & & & \\
& & \textit{where } x_1,\ldots,x_k \textit{ are fresh variables and } \{a_{i_1},\ldots,a_{i_k}\} = A. & & &
\end{array}$$

First, observe that if p is a BSL-formula, then it is in AP and not in AcP, so that p is indeed an SL formula. Before establishing the correctness of the translation, we need the following lemma. It essentially says that the evaluation of a formula $\mathrm{tr}'_A(\varphi)$ in an assignment χ is independent of how χ is defined on A: for an agent $a \in A$, whether χ is defined on a or not, and in the former case how it is defined, is of no consequence as the translation tr'_A remembers that a is not supposed to be bound to a strategy.

Lemma 1 *Let G be a CGS, $s \in G$ a state, $\varphi \in BSL$ a state formula and χ an s-total assignment for φ. For all $A \subseteq Ag$, $\{a_{i_1},\ldots,a_{i_k}\} \subseteq A$ and for all $\sigma_1,\ldots,\sigma_k \in \mathrm{Str}(s)$, letting $\chi_1 = \chi[a_{i_1} \mapsto \sigma_1,\ldots,a_{i_k} \mapsto \sigma_k]$ and $\chi_2 = \chi[a_{i_1} \mapsto ?,\ldots,a_{i_k} \mapsto ?]$, it holds that:*

P1: *$G,\chi,s \models_{SL} \mathrm{tr}'_A(\varphi)$ if, and only if, $G,\chi_1,s \models_{SL} \mathrm{tr}'_A(\varphi)$, and*

P2: *$G,\chi,s \models_{SL} \mathrm{tr}'_A(\varphi)$ if, and only if, $G,\chi_2,s \models_{SL} \mathrm{tr}'_A(\varphi)$.*

Proposition 2 *Let G be a CGS. For every state formula $\varphi \in BSL$, assignment χ for φ and state $s \in G$, it holds that $G,\chi,s \models_{BSL} \varphi$ if, and only if, $G,\chi,s \models_{SL} \mathrm{tr}'_{Ag\setminus dom(\chi)}(\varphi)$.*

We can now prove that SL and BSL have the same expressivity on the level of sentences.

Theorem 1 *SL and BSL are equiexpressive, with linear translations in both directions.*

Proof We first prove that $\mathsf{SL} \preccurlyeq \mathsf{BSL}$. Let φ be an SL-sentence. Clearly, $\mathrm{tr}(\varphi)$ is a BSL-sentence. Let G be a CGS with initial state s_ι, and let χ be any assignment. By definition, $G \models_{\mathsf{SL}} \varphi$ iff $G, \chi, s_\iota \models_{\mathsf{SL}} \varphi$. By Proposition 1, $G, \chi, s_\iota \models_{\mathsf{SL}} \varphi$ iff $G, \chi, s_\iota \models_{\mathsf{BSL}} \mathrm{tr}(\varphi)$, and by definition, the latter is equivalent to $G \models_{\mathsf{BSL}} \mathrm{tr}(\varphi)$.

Now, to prove that $\mathsf{BSL} \preccurlyeq \mathsf{SL}$, let φ be a BSL-sentence, and let $\varphi' = \mathrm{tr}'_{Ag}(\varphi)$. Observe that φ' is an SL-sentence: indeed, every temporal operator in φ is under the scope of some path quantifier, and by definition of tr'_{Ag}, every temporal operator in φ' is thus under the scope of a binding for each agent. Now, let G be a CGS and χ an assignment such that $Ag \setminus dom(\chi) = Ag$. By definition, $G \models_{\mathsf{BSL}} \varphi$ iff $G, \chi, s_\iota \models_{\mathsf{BSL}} \varphi$ (recall that since φ is a sentence, the choice of χ does not matter for the evaluation of φ). By Proposition 2, the latter is equivalent to $G, \chi, s_\iota \models_{\mathsf{SL}} \varphi'$, which by definition is equivalent to $G \models_{\mathsf{SL}} \varphi'$.

Concerning the size of the translations, the one of Definition 1 is clearly linear, and the one in Definition 2 is in $O(2|Ag||\varphi|)$, where $|Ag|$ is the number of agents and $|\varphi|$ the number of symbols in φ. The translation is thus linear in the size of the formula. $\square$

We can therefore transfer to BSL the following results known about SL [18]:

Corollary 1 *The model-checking problem for BSL is nonelementary decidable.*

Corollary 2 *The satisfiability problem for BSL is Σ_1^1-hard.*

On the other hand, because BSL^+ can express properties about the actions taken by agents, it should clearly be strictly more expressive than BSL and thus also SL, but we have not yet proved this.

Conjecture 1 *BSL^+ strictly subsumes BSL and SL.*

4 Epistemic Strategy Logic

In this section, we further extend the framework to account for imperfect information. For the logic to be expressive enough to express uniformity of strategies, we need to talk about actions played by the agents, and we therefore allow the use of atomic propositions in AcP.

4.1 Syntax

We add distributed knowledge operators to the language, one for each group of agents. The syntax of ESL is therefore described by the following grammar:

$$\begin{array}{ll} \text{State formulas:} & \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{E}\varphi \mid \langle\langle x\rangle\rangle\varphi \mid (a,x)\varphi \mid (a,?)\varphi \mid D_A\varphi \\ \text{Path formulas:} & \psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi, \end{array}$$

where $p \in AP^+$, $x \in Var$ and $A \subseteq Ag$.

We define, for each $a \in Ag$, $K_a\varphi := D_{\{a\}}\varphi$, and as for BSL and BSL^+, we write ESL_ψ for the set of ESL-path formulas.

4.2 Semantics

To represent the agents' imperfect information about the current situation in the game, we add binary indistinguishability relations in CGSs. Most works consider equivalence relations on states, which are extended to initial paths according to how much memory agents are supposed to have. Because in this work we do not want to make any such assumptions, we adopt a more general approach and directly take equivalence relations on initial paths.

We call *imperfect information concurrent game structure* (ICGS) a tuple $G_i = (G, \{\sim_a\}_{a\in Ag})$, where G is a CGS and for each $a \in Ag$, $\sim_a \subseteq (2^{AP^+})^* \times (2^{AP^+})^*$ is an indistinguishability equivalence relation for Agent a. For $A \subseteq Ag$, we let $\sim_A := \cap_{a\in A} \sim_a$: it is the *distributed knowledge relation* of agents in A. Given two initial paths $\rho = s_0 d_1 s_1 \dots d_n s_n$ and $\rho' = s'_0 d_1 s'_1 \dots d_m s'_m$ and a set of agents $A \subseteq Ag$, we shall write $\rho \sim_A \rho'$ whenever $\mu^+(s_0)\dots\mu^+(s_n) \sim_A \mu^+(s'_0)\dots\mu^+(s'_m)$, *i.e.* when the sequences of extended valuations along the plays are related by $\sim_A$. As usual in epistemic logic, the intended meaning of $\rho \sim_a \rho'$ is that in initial path ρ, Agent a considers it possible that ρ' is the actual initial path.

Because agents may infer knowledge from what they recall of the past of an initial path, we cannot evaluate state formulas merely in states of the game as we do for BSL^+, but we evaluate them in initial paths instead. Also, in order not to forget the past when we consider outcomes of an assignment, we define for every initial path ρ and assignment χ, $\mathrm{Out}(\rho,\chi) := \{\rho\cdot\rho' \mid \rho' \in \mathrm{Out}(\mathrm{last}(\rho),\chi)\}$.

Let $\varphi \in \mathsf{ESL}$ be a state formula (resp. let $\psi \in \mathsf{ESL}_\psi$ be a path formula), and let $G = (Q, \delta, q_\iota, \mu)$ be a CGS. Let χ be an assignment for φ (resp. for ψ), let $\rho \in Paths_*$ be an initial path, $\pi \in Paths_\omega$ a path, and $i \geq 0$. The semantics of ESL is defined inductively as follows:

$G_i,\chi,\rho \models_{\mathsf{ESL}} p$	if	$p \in \mu^+(\mathrm{last}(\rho))$
$G_i,\chi,\rho \models_{\mathsf{ESL}} \neg\varphi$	if	$G_i,\chi,\rho \not\models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} \varphi\vee\varphi'$	if	$G_i,\chi,\rho \models_{\mathsf{ESL}} \varphi$ or $G_i,\chi,\rho \models_{\mathsf{ESL}} \varphi'$
$G_i,\chi,\rho \models_{\mathsf{ESL}} \langle\langle x\rangle\rangle\varphi$	if	there exists $\sigma \in Str$ such that $G_i,\chi[x\mapsto\sigma],\rho \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} (a,x)\varphi$	if	$G_i,\chi[a\mapsto\chi(x)],\rho \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} (a,?)\varphi$	if	$G_i,\chi[a\mapsto ?],\rho \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} \mathbf{E}\psi$	if	there exists $\pi \in \mathrm{Out}(\rho,\chi)$ such that $G_i,\chi,\pi,\lvert\rho\rvert \models_{\mathsf{ESL}} \psi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} D_A\varphi$	if	for every initial path $\rho' \in Paths_*$ such that $\rho \sim_A \rho'$, $G_i,\chi,\rho' \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \varphi$	if	$G_i,\chi,\pi_{\leq i} \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \neg\psi$	if	$G_i,\chi,\pi,i \not\models_{\mathsf{ESL}} \psi$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \psi\vee\psi'$	if	$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \psi$ or $G_i,\chi,\pi,i \models_{\mathsf{ESL}} \psi'$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \mathbf{X}\psi$	if	$G_i,\chi^{\pi[i,i+1]},\pi,i+1 \models_{\mathsf{ESL}} \psi$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \psi\mathbf{U}\psi'$	if	there is $j \geq i$ such that $G_i,\chi^{\pi[i,j]},\pi,j \models_{\mathsf{ESL}} \psi'$, and for all $i \leq k < j$, $G_i,\chi^{\pi[i,k]},\pi,k \models_{\mathsf{ESL}} \psi$

We now give an example of a property that can be expressed in ESL but not in SL, BSL or BSL^+. The property we consider is the uniformity property of strategies, which is central in the paradigm of imperfect information.

4.3 Properties of strategies

A *uniform strategy*, in the context of games with imperfect information, usually means a strategy that respects the player's information, *i.e.* a strategy that assigns the same action in situations that are indistinguishable to the player [4, 14]. In SL, temporal formulas being only evaluated in complete assignments, it is clear that one cannot compare several outcomes of a given strategy for a player, so that it is hopeless to express such uniformity properties. In BSL, one can consider all the possible outcomes of a strategy, but one cannot talk about the actions taken by agents, so that expressing that a strategy assigns the same action in different situations is not possible either. In BSL^+, we can refer to the precise actions taken by the agents, but we have no way of relating situations that are indistinguishable to an agent. However, as we show below, ESL is expressive enough for this sort of properties.

We define a notion of uniformity, that we call *weak uniformity*, and that asks for a strategy to be uniform on all its outcomes from the current situation.

Definition 3 *Let $G_i = (G, \{\sim_a\}_{a\in Ag})$ be an ICGS, let $\rho \in Paths_*$ be an initial path and $a \in Ag$ an agent. A strategy σ is* weakly uniform for a in ρ *if, for all initial paths $\rho' \in Out(\rho, [a \mapsto \sigma])$ and $\rho'' \in Paths_*$ such that $\rho' \sim_a \rho''$, $\sigma(\rho') = \sigma(\rho'')$.*

Now let us define the following ESL-formula.

Definition 4 *For each $a \in Ag$, we define the formula*

$$a\texttt{-wUniform-aux} := \mathbf{AG}(\bigvee_{c\in Ac} K_a \mathbf{AX} p^a_c).$$

To understand the meaning of this formula, first observe that if an assignment χ binds an agent a to a strategy σ, *i.e.* $\chi(a) = \sigma$, then for every initial path $\rho \in Paths_*$, there is an action $c \in Ac$ such that p^a_c holds in all continuations of ρ of the form $\rho' = \rho \cdot ds$ that follow χ: this action is $\sigma(\rho) = d(a)$, the action played by Agent a in initial path ρ according to σ. Therefore, $G_i, \chi, \rho \models \mathbf{AX} p^a_{\sigma(\rho)}$. It follows that, when evaluated in an initial path ρ and assignment $[a \mapsto \sigma]$, where σ is a strategy, formula $a\texttt{-wUniform}$ says that at every point of every outcome in $\mathrm{Out}(\rho, \chi)$, there is an action that Agent a plays in *all $\sim_a$-related nodes*. Let us fix an ICGS $G_i = (G, \{\sim_a\}_{a\in Ag})$.

Proposition 3 *For every initial path $\rho \in Paths_*$ and agent $a \in Ag$, a strategy σ is weakly uniform for Agent a in ρ if, and only if, $G_i, [a \mapsto \sigma], \rho \models a\texttt{-wUniform-aux}$.*

However, $a\texttt{-wUniform}$ only has the intended meaning in an assignment that does not bind any other agent: indeed, otherwise we would only have that the strategy considered is uniform on the subset of its outcomes that follow the strategies assigned to the other agents. Consider now the following formula:

Definition 5 *For each $a \in Ag$, noting $\{a_1, \ldots, a_k\} = Ag \setminus \{a\}$, we define the formula*

$$a\texttt{-wUniform} := (a_1, ?)\ldots(a_k, ?)a\texttt{-wUniform-aux}.$$

The following proposition holds:

Proposition 4 *For every initial path $\rho \in Paths_*$, assignment χ and agent $a \in Ag$, a strategy σ is weakly uniform for Agent a in ρ if, and only if, $G_i, \chi[a \mapsto \sigma], \rho \models a\texttt{-wUniform}$.*

We now illustrate how various semantics of ATL with imperfect information can be expressed in ESL. We take the example of the ATL formula $\langle\langle A\rangle\rangle \mathbf{F} p$, where $A \subseteq Ag$. Assume that $A = \{a_1, \ldots, a_k\}$ and $Ag \setminus A = \{a_{k+1}, \ldots, a_n\}$. We consider three semantics: the basic one from [15], in which strategies are just required to be uniform, the *de dicto* semantics, where in addition the players must know that there is a strategy to achieve their goal, but may ignore what that strategy is, and the *de re* semantics, in which there must exist a strategy that the players know it ensures their goal (see [14], Sec. 3.2). With the first semantics, $\langle\langle A\rangle\rangle \mathbf{F} p$ would be translated in ESL as:

$$\langle\langle x_1\rangle\rangle \ldots \langle\langle x_k\rangle\rangle (a_1, x_1)\ldots(a_k, x_k)(\bigwedge_{1\le i\le k} a_i\texttt{-wUniform} \wedge \mathbf{AF} p).$$

For the *de dicto* semantics, one would write instead:

$$D_A\langle\langle x_1\rangle\rangle \dots \langle\langle x_k\rangle\rangle(a_1,x_1)\dots(a_k,x_k)(\bigwedge_{1\leq i\leq k} a_i\text{-}\mathtt{wUniform}\wedge \mathbf{AF}p),$$

while for the *de re* semantics, one would write:

$$\langle\langle x_1\rangle\rangle \dots \langle\langle x_k\rangle\rangle D_A(a_1,x_1)\dots(a_k,x_k)(\bigwedge_{1\leq i\leq k} a_i\text{-}\mathtt{wUniform}\wedge \mathbf{AF}p).$$

One may object that the notion of weak uniformity we consider is too weak compared to the usual one, which is that a strategy should be equal on all pairs of related initial paths. We argue that it is enough for a strategy to be uniform on all the initial paths it may be involved in while evaluating the formula.

For instance, in the example above, the objective is $\mathbf{AF}p$, so that it is enough to ensure that strategies for the agents are uniform on their outcome: if a satisfying set of strategies contains one σ_i that is not defined uniformly on some initial paths that are outside its outcome, this σ_i can easily be turned into a uniform strategy in the usual sense, it will still satisfy the formula.

Should we consider a more complex objective, in particular involving knowledge, weak uniformity may not be sufficient though. Consider the ESL formula $\langle\langle x\rangle\rangle(a,x)\mathbf{AG}K_a\mathbf{AF}p$, where $a\in Ag$, which means that Agent a wants a strategy such that she always knows that p will eventually be reached. This objective not only considers outcomes of the strategy from the current situation, but also outcomes from initial paths equivalent to the latter outcomes. In this case, we could strengthen the requirement on Agent a's strategy by repeating the weak-uniformity requirement after each knowledge operator. In the example:

$$\langle\langle x\rangle\rangle(a,x)(a\text{-}\mathtt{wUniform}\wedge\mathbf{AG}K_a(a\text{-}\mathtt{wUniform}\wedge\mathbf{AF}p)).$$

Finally, observe that if we introduced an artificial agent a_{mem} associated to the relation that relates two initial paths if they end up in the same state, then the formula $a_{\text{mem}}\text{-}\mathtt{wUniform}$ would characterize strategies that are memoryless on their outcomes from the current initial path, in the sense that their definition only depends on the last state of each initial path.

5 Conclusion

We have enriched SL with two operators, the path quantifier and the unbinding operator, which are convenient but do not add expressivity in the perfect information case; interestingly though, they do not increase complexity either. In the context of imperfect information however, these operators together with knowledge operators and the ability to talk about actions, allowed us to express properties of strategies which are usually fixed in the semantics of the logics, such as being uniform, *de re*, *de dicto*, memoryless. . . This feature makes our Epistemic Strategy Logic able to deal with a vast class of agents without having to change the semantics, and thus unifies many of the previous proposals in the area.

Of course this comes at a price, and the model-checking problem for this logic is certainly undecidable with perfect-recall relations and several agents. We believe that the next steps are, first, to see whether the syntactical fragments studied for SL with perfect information, such as One-Goal or Boolean-Goal Strategy Logic, can be transferred to BSL and then to ESL, and see whether they enjoy better complexity properties. The second natural move would be to look at structures which are known to work well with multiple agents with imperfect information: hierarchical knowledge [7, 20], recurring common knowledge of the state [6]. . .

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. Available at `http://doi.acm.org/10.1145/585265.585270`.

[2] Francesco Belardinelli (2014): *Reasoning about Knowledge and Strategies: Epistemic Strategy Logic*. In: *SR*, pp. 27–33. Available at `http://dx.doi.org/10.4204/EPTCS.146.4`.

[3] Francesco Belardinelli (2015): *A Logic of Knowledge and Strategies with Imperfect Information*. In: *Private communication*.

[4] Johan van Benthem (2001): *Games in Dynamic-Epistemic Logic*. *Bulletin of Economic Research* 53(4), pp. 219–248, doi:10.1111/1467-8586.00133.

[5] Johan van Benthem (2011): *Logical dynamics of information and interaction*. Cambridge University Press.

[6] Dietmar Berwanger & Anup Basil Mathew (2014): *Games with recurring certainty*. In: *Proceedings of SR 2014*, pp. 91–96, doi:10.4204/EPTCS.146.12.

[7] R. Bozianu, C. Dima & C. Enea (2013): *Model Checking an Epistemic mu-calculus with Synchronous and Perfect Recall Semantics*. In: *TARK'2013*. Available at `http://arxiv.org/abs/1310.6434`.

[8] Petr Cermák, Alessio Lomuscio, Fabio Mogavero & Aniello Murano (2014): *MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications*. In: *CAV*, pp. 525–532. Available at `http://dx.doi.org/10.1007/978-3-319-08867-9_34`.

[9] 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. Available at `http://dx.doi.org/10.1016/j.ic.2009.07.004`.

[10] Ronald Fagin, Joseph Y. Halpern, Yoram Moses & Moshe Y. Vardi (1995): *Reasoning about knowledge*. 4, MIT press Cambridge.

[11] Joseph Y. Halpern, Ron van der Meyden & Moshe Y. Vardi (2004): *Complete Axiomatizations for Reasoning about Knowledge and Time*. *SIAM J. Comput.* 33(3), pp. 674–703. Available at `http://dx.doi.org/10.1137/S0097539797320906`.

[12] W. van der Hoek & M. 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.

[13] W. Jamroga & T. Ågotnes (2006): *What agents can achieve under incomplete information*. In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, ACM, pp. 232–234.

[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] Wojtek Jamroga (2003): *Some remarks on alternating temporal epistemic logic*. *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pp. 133–140.

[16] François Laroussinie & Nicolas Markey (2013): *Satisfiability of ATL with strategy contexts*. In: *Proceedings Fourth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2013, Borca di Cadore, Dolomites, Italy, 29-31th August 2013.*, pp. 208–223, doi:10.4204/EPTCS.119.18. Available at `http://dx.doi.org/10.4204/EPTCS.119.18`.

[17] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2012): *What Makes Atl* Decidable? A Decidable Fragment of Strategy Logic*. In: *CONCUR 2012 - Concurrency Theory - 23rd International Conference, CONCUR 2012, Newcastle upon Tyne, UK, September 4-7, 2012. Proceedings*, pp. 193–208. Available at `http://dx.doi.org/10.1007/978-3-642-32940-1_15`.

[18] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2014): *Reasoning About Strategies: On the Model-Checking Problem*. *ACM Trans. Comput. Log.* 15(4), pp. 34:1–34:47, doi:10.1145/2631917. Available at `http://doi.acm.org/10.1145/2631917`.

[19] 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, December 15-18, 2010, Chennai, India*, pp. 133–144, doi:10.4230/LIPIcs.FSTTCS.2010.133. Available at `http://dx.doi.org/10.4230/LIPIcs.FSTTCS.2010.133`.

[20] A. Pnueli & R. Rosner (1989): *On the Synthesis of a Reactive Module*. In: *POPL'89*, pp. 179–190, doi:10.1145/75277.75293. Available at `http://doi.acm.org/10.1145/75277.75293`.

[21] John H. Reif (1984): *The complexity of two-player games of incomplete information*. *Journal of computer and system sciences* 29(2), pp. 274–301, doi:10.1016/0022-0000(84)90034-5.

[22] Pierre-Yves Schobbens (2004): *Alternating-time logic with imperfect recall*. *Electronic Notes in Theoretical Computer Science* 85(2), pp. 82–93.

13

Regular Paper

An Arrow-based Dynamic Logic of Norms

Louwe B. Kuijer

We introduce a Normative Arrow Update Logic (NAUL), which combines the formalisms of Arrow Update Logic (a variant of Dynamic Epistemic Logic) with concepts from Normative Temporal Logic (a type of Normative System). Using NAUL, we can draw distinctions between dynamic and static applications of norms, and between additive, multiplicative and sequential combination of norms. We show that the model checking problem for NAUL can be solved in polynomial time, and that NAUL is strictly more expressive than both CTL and Arrow Update Logic.

1 Introduction

In many situations an agent will be able to choose between a number of different actions. Sometimes each choice is as good as another, but usually some constraints will apply. For example, some of the actions available to the agent may be irrational, illegal, immoral, impolite or in breach of some code of conduct the agent has agreed to. We refer to such constraints as norms. A norm guides the behavior of an agent by dividing the available actions into those that are allowed (by the norm) and those that are disallowed. For example, "don't commit murder", "don't make a losing move if a winning move is available" and "use knife and fork when eating" are norms (of law, rationality and etiquette respectively).

In order to choose a course of action, we need to be able to make two kinds of decisions. Firstly, we need to decide whether we will adopt a norm. Secondly, we need to determine whether a given action is allowed by a norm.[1] In order to assist with both decision making procedures we need a logic that allows us to (a) formally represent norms and (b) determine the consequences of adopting a norm. In this paper, we introduce Normative Arrow Update Logic (NAUL) for this purpose. In the language of NAUL, we have two kinds of objects: norms and formulas. A norm N specifies the actions that are allowed by N. Formulas can contain norm-operators, which allows us to determine the consequences of a norm: $[N]\varphi$ holds if φ is guaranteed to be true under the assumption that all agents obey the norm N.[2] For example, if we want to prevent deadlock, then a norm N will satisfy our goal if and only if $[N]G\neg \mathit{deadlock}$ holds.

In addition to allowing agents to determine whether an action is allowed, the explicit representation of norms in NAUL also allows us to combine norms in three different ways:

Additive an action is allowed by the norm $N_1 + N_2$ if it is allowed by either N_1 or N_2. Example: suppose N_1 requires agents to use knife and fork when they eat, and that N_2 requires agents to use chopsticks. Then $N_1 + N_2$ allows agents the choice between knife and fork or chopsticks.

Multiplicative an action is allowed by the norm $N_1 \times N_2$ if it is allowed by both N_1 and N_2. Example: suppose N_1 requires agents to drive on the right side of the road, and that N_2 requires agents to yield to traffic that comes from the right. Then $N_1 \times N_2$ requires agents to do both.

Sequential an action is allowed by the norm $N_1 \circ N_2$ if it is among the actions that are allowed by N_2 after all actions that are disallowed by N_1 are removed. Example: suppose N_1 requires agents to

[1] Note that the agents who decide whether a norm gets accepted are not necessarily the same agents that need to follow the norm. For example, laws generally do not apply only to the people who wrote them.

[2] In addition to the "dynamic" norm operator $[N]$, NAUL also uses "static" norm operators $\Box_N, G_N$ and F_N. See Section 4.1 for an explanation of the difference between these normative operators.

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

obey the law, and that N_2 requires agents to pursue their self-interest. Then $N_1 \circ N_2$ is a norm that tells agents to pursue their self-interest, but only within the limits of the law.[3]

1.1 Overview

The remainder of this paper is organized as follows. First, in Section 2, we compare NAUL to a few other normative systems. Then, in Section 4, we use an example to illustrate the three different ways of combining norms, as well as the difference between the static and dynamic normative operators. Finally, in Section 5, we show that NAUL is strictly more expressive than CTL and AUL*.

2 Comparison to Other Logics

For reasons of brevity we cannot give a full overview of the history of normative systems here, nor can we compare NAUL to every other logic of norms. We do compare NAUL to four existing logics that are especially relevant to this paper, because they can be seen as direct predecessors to NAUL. Specifically, we compare it to AUL* [7], NTL [1], CTL [4] and the social laws from [9].

NAUL combines the technical methods of Arrow Update Logic (AUL*) [7] with ideas from Normative Temporal Logic (NTL) [1]. There are two differences between NAUL and AUL*. The first difference is technical in nature: NAUL has G and F temporal operators, while AUL* only has G. The second difference is non-technical: in AUL* we would interpret an update $[N]$ as an epistemic event, while in NAUL we interpret $[N]$ as the application of a norm.

The main difference between NAUL and NTL is that, unlike in NAUL, norms in NTL are (to some extent) meta-logical objects. Specifically, a norm η in NTL is simply a subset of all possible actions. In NAUL a norm N is not identical to a subset of actions; instead it consists of a number of formulas that determine a subset of actions that are allowed. Defining norms in this way has three advantages.

Firstly, in NAUL there can be interaction between norms and formulas. As a very simple example, consider the triple $(\top, \mathscr{A}, p)$. This triple defines a norm that, roughly speaking, means "do not take any action that would cause $\neg p$ to hold." The formula $p \rightarrow [(\top, \mathscr{A}, p)]Gp$ is valid: if everyone obeys the norm $(\top, \mathscr{A}, p)$ then p will hold forever. For this validity it is critical that p occurs in both $(\top, \mathscr{A}, p)$ and in Gp, so there is interaction between the norm and Gp. Such interaction is impossible if a norm is simple a set of actions. Secondly, in NTL a norm η could allow an action in one situation while disallowing it in another, even if the two situations are indistinguishable. If such a norm η were to be adopted, an agent would be incapable of determining whether the action is allowed. The norm therefore fails to be action-guiding. In NAUL every norm must be defined using NAUL formulas, so a norm N automatically treats two situations the same if they are indistinguishable. As such, it is always possible to determine whether an action is allowed by N.[4] Thirdly, because norms are defined we can combine them sequentially. Consider the sequential example given above, so N_1 requires an agent to obey the law whereas N_2 requires an agent to pursue its own self-interest. The effect of N_2 changes, depending on whether we apply it by itself or after first applying N_1. Since NTL norms η_1, η_2 are simply sets of actions there is no systematic way to change the effect of η_2 depending on the application of η_1.

NAUL uses a number of temporal operators to describe the consequences of a norm. Specifically, the operators $\Box$, G and F. These correspond to the CTL [4] operators AX, AG and AF. Normally, EU cannot

[3]Note that this is different from $N_1 \times N_2$, which would allow an action only if it is both legal and in the agent's interest. In cases where following the law is not in the agent's interest, $N_1 \times N_2$ would forbid every action.

[4]A side effect of the requirement that norms are defined by NAUL formulas is that NAUL, unlike NTL, is invariant under bisimulation.

be defined using only AX,AG and AF. In NAUL, however, it is possible to see EU as an abbreviation of other operators. As a result, NAUL is strictly more expressive than CTL, see Section 5.1.

Norms in NAUL are similar to the social laws from [9], except that where social laws in [9] define forbidden actions only be a precondition, NAUL defines them by both a precondition and a postcondition. This allows NAUL To model certain norms that cannot be represented as social laws in [9].

3 Normative Arrow Update Logic

3.1 The Setting

We want norms that guide our behavior by telling us whether a given action is allowed. In order to do this we first need a model of agency. We will use a relatively simple kind of transitions system. Let $\mathscr{A}$ be a finite set of agents and $\mathscr{P}$ a countably infinite set of propositional variables.

Definition 1. A *model* $\mathscr{M}$ is a triple $\mathscr{M} = (S,R,v)$ where S is a set of states, $R : \mathscr{A} \to S \times S$ maps each agent to an accessibility relation on S, and $v : \mathscr{P} \to 2^S$ is a valuation. A *pointed model* is a pair $\mathscr{M}, s$ where $\mathscr{M} = (S,R,v)$ is a model and $s \in S$.

Definition 2. Let $\mathscr{M} = (S,R,v)$ be a model. A *transition* in $\mathscr{M}$ is a triple (s_1,a,s_2) where $(s_1,s_2) \in R(a)$. The transition (s_1,a,s_2) *starts* in s_1 and *ends* in s_2, and is denoted by $s_1 \xmapsto{a} s_2$.

The intended meaning of a transition $s_1 \xmapsto{a} s_2$ is that, if the system is in state s_1, then agent a can take an action that changes the system's state to s_2. We use paths to represent sequences of actions.

Definition 3. A *path* in $\mathscr{M}$ is a (possibly finite) sequence $s_1 \xmapsto{a_1} s_2, s_2 \xmapsto{a_2} s_3, \cdots$ of transitions in $\mathscr{M}$ where each transitions begins in the state where the previous transition ends. A single state s is considered a degenerate path that contains no transitions. A path P' *extends* a path P if P is an initial segment of P'.

We abuse notation by writing $s \in P$ if s is one of the states that occur in P. We also omit the starting state of all but the first transition in a path, so we write $s_1 \xmapsto{a_1} s_2 \xmapsto{a_2} s_3 \cdots$ for $s_1 \xmapsto{a_1} s_2, s_2 \xmapsto{a_2} s_3, \cdots$. Note that we do not require paths to be infinite. By ending paths with $\cdots$ we do not imply that they are infinite, so $s_1 \xmapsto{a_1} s_2 \xmapsto{a_2} s_3 \cdots$ may or may not be finite. A path $s_1 \xmapsto{a_1} s_2 \cdots s_n \xmapsto{a_n} s_{n+1}$ is always finite, though.

Remark 1. The transition systems that we use here cannot model simultaneous actions. Simultaneity can be added to the framework in a relatively straightforward way, but doing so requires significantly more complicated notation so we will not do so here.

3.2 Defining NAUL

Having dealt with the necessary technical preliminaries we can define Normative Arrow Update Logic.

Definition 4. The formulas of $\mathscr{L}_{NAUL}$ are given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid [N]\varphi \mid \Box_N \varphi \mid G_N \varphi \mid F_N \varphi$$
$$N ::= (\varphi, a, \varphi) \mid N, (\varphi, a, \varphi)$$

where $p, \in \mathscr{P}$ and $a \in \mathscr{A}$.

Strictly speaking a norm N is a list of clauses, but we abuse notation by identifying it with the set of its clauses. Additionally, we use a number of abbreviations.

Definition 5. We use $\wedge, \rightarrow, \leftrightarrow, \bigwedge, \bigvee$ and $\Diamond_N$ in the usual way as abbreviations. Furthermore, we use $\hat{G}_N$ and $\hat{F}_N$ as abbreviations for $\neg G_N \neg$ and $\neg F_N \neg$. Given $B \subseteq \mathscr{A}$ we write (φ, B, ψ) for $\{(\varphi, a, \psi) \mid a \in B\}$, $\Box_B$ for $\Box_{(\top,B,\top)}$, G_B for $G_{(\top,B,\top)}$ and F_B for $F_{(\top,B,\top)}$. Finally, we use $\Box, G$ and F for $\Box_{\mathscr{A}}$, $G_{\mathscr{A}}$ and $F_{\mathscr{A}}$.

Remark 2. The operator $[N]$ does not add expressivity; for every formula φ containing $[N]$ there is an equivalent formula φ' that does not contain $[N]$. The way to define $[N]$ as an abbreviation for other operators is rather complicated, however, so it is convenient to define $[N]$ directly.

The semantics of $\mathscr{L}_{NAUL}$ are given by the following two interdependent definitions.

Definition 6. Let a model $\mathscr{M} = (S, R, v)$ and a norm N be given. A transition $s_1 \xmapsto{a} s_2$ *satisfies* N if there is a clause $(\varphi, a, \psi) \in N$ such that $\mathscr{M}, s_1 \models \varphi$ and $\mathscr{M}, s_2 \models \psi$. A path $s_1 \xmapsto{a_1} s_2 \xmapsto{a_2} s_3 \cdots$ is an *N-path* if every transition $s_i \xmapsto{a_i} s_{i+1}$ in the path satisfies N. An N-path is *full* if there is no N-path that extends it.

Definition 7. Let $\mathscr{M} = (S, R, v)$ be a transition system and $s \in S$. The relation $\models$ is given as follows.

$$\begin{array}{lcl}
\mathscr{M}, s \models p & \Leftrightarrow & s \in v(p) \text{ for } p \in \mathscr{P}, \\
\mathscr{M}, s \models \neg\varphi & \Leftrightarrow & \mathscr{M}, s \not\models \varphi, \\
\mathscr{M}, s \models \varphi_1 \vee \varphi_2 & \Leftrightarrow & \mathscr{M}, s \models \varphi_1 \text{ or } \mathscr{M}, s \models \varphi_2, \\
\mathscr{M}, s \models \Box_N \varphi & \Leftrightarrow & \text{for every } N\text{-path } s \xmapsto{a} s' \text{ we have } \mathscr{M}, s' \models \varphi, \\
\mathscr{M}, s \models G_N \varphi & \Leftrightarrow & \text{for every } N\text{-path } P \text{ starting in } s \text{ and} \\
 & & \text{every } s' \in P \text{ we have } \mathscr{M}, s' \models \varphi, \\
\mathscr{M}, s \models F_N \varphi & \Leftrightarrow & \text{for every full } N\text{-path starting in } s \text{ there is} \\
 & & \text{some } s' \in P \text{ such that } \mathscr{M}, s' \models \varphi, \\
\mathscr{M}, s \models [N]\varphi & \Leftrightarrow & \mathscr{M} * N, s \models \varphi
\end{array}$$

where $\mathscr{M} * N = (W, R * N, v)$ and, for every $a \in \mathscr{A}$,

$$R * N(a) = \{(s, s') \in R(a) \mid s \xmapsto{a} s' \text{ satisfies N}\}.$$

Note that the single state s is an N-path for every norm N, so $\mathscr{M}, s \models G_N \varphi$ implies $\mathscr{M}, s \models \varphi$. Each of the operators has an intended meaning. For the non-Boolean operators, this meaning is as follows:

$\Box_N \varphi$	"φ holds after any single action that is allowed by N"
$G_N \varphi$	"φ holds after every sequence of actions that is allowed by N"
$F_N \varphi$	"every full sequence of actions that is allowed by N contains at least one φ state"
$[N]\varphi$	"if all agents forever obey the norm N from now on, the formula φ will hold"

3.3 Combining Norms

As mentioned in the introduction, we want to be able to combine norms in three different ways: additively, multiplicatively and sequentially. An action is allowed by $N_1 + N_2$ if it is allowed by either N_1 or N_2, allowed by $N_1 \times N_2$ if it is allowed by both N_1 and N_2, and allowed by $N_1 \circ N_2$ if it is allowed by N_2 after all options that do not satisfy N_1 are discarded. Using NAUL we can define these norm operators.

Definition 8. Let N_1 and N_2 be norms. Then

$$\begin{aligned}
N_1 + N_2 &:= N_1 \cup N_2 \\
N_1 \times N_2 &:= \{(\varphi_1 \wedge \psi_1, a, \varphi_2 \wedge \psi_2) \mid (\varphi_1, a, \varphi_2) \in N_1, (\psi_1, a, \psi_2) \in N_2\} \\
N_1 \circ N_2 &:= N_1 \times \{([N_1]\psi_1, a, [N_1]\psi_2) \mid (\psi_1, a, \psi_2) \in N_2\}
\end{aligned}$$

Proposition 1. *Let $s_1 \xmapsto{a} s_2$ be a transition in $\mathscr{M}$. Then*

1. *$s_1 \xmapsto{a} s_2$ satisfies $N_1 + N_2$ if and only if it satisfies either N_1 or N_2,*
2. *$s_1 \xmapsto{a} s_2$ satisfies $N_1 \times N_2$ if and only if it satisfies both N_1 and N_2,*
3. *$s_1 \xmapsto{a} s_2$ satisfies $N_1 \circ N_2$ if and only if it satisfies N_1, and it satisfies N_2 in $\mathscr{M} * N_1$.*

3.4 Complexity

Let $\mathscr{M} = (S, R, v)$, $s \in S$ and ψ be given. We want to determine whether $\mathscr{M}, s \models \psi$. We can find a polynomial time algorithm for this problem by modifying the CTL algorithm from [4]. The modified algorithm works as follows. Start by making a list of all subformulas of ψ and all norms occurring in ψ. Order this list in the following way. Let ξ_1, ξ_2 be formulas or norms. Then ξ_1 comes before ξ_2 if ξ_1 is a part of ξ_2, or if neither ξ_1 nor ξ_2 is part of the other and ξ_1 appears to the left of ξ_2 in ψ. Now, label each norm and subformula by the sequence of norms inside the scope of which they appear. Consider, for example, $\psi = [N_1][N_2]\varphi_3$ with $N_2 = (\varphi_1, a, \varphi_2)$. Here we would label φ_3 as $\varphi_3^{N_1,N_2}$, N_2 as $N_2^{N_1}$ and φ_1 as $\varphi_1^{N_1}$. Now, go through the list one element at a time. If the element is a norm N^σ, label some of the transitions in $\mathscr{M}$ by σ, N in the following way.

```
for each a ∈ 𝒜
   for each (s1, s2) ∈ R(a)
      for each (φ1^σ, a, φ2^σ) ∈ N^σ
         if (s1, s2) is labeled σ and s1 is labeled φ1^σ and s2 is labeled φ2^σ
            then label (s1, s2) with σ, N
```

If the element is a formula χ^σ, then label the states by either χ^σ or $\overline{\chi^\sigma}$. The way to do this labeling depend on the main connective of χ. For the Boolean and temporal connectives, the labeling is very similar to that in the CTL algorithm, the only difference is that we ignore transitions that do not have the right label. We give pseudo-code for the case $\chi^\sigma = G_N\varphi^\sigma$, the other cases are as one would expect.

```
for each s ∈ S
   if s is labeled \overline{φ^σ}
      then label s with \overline{G_N φ^σ}
for i = 1 to |S|
   for each s ∈ S
      for each (s, s') ∈ ⋃_{a∈𝒜} R(a)
         if (s, s') is labeled σ, N and s' is labeled \overline{G_N φ^σ}
            then label s with \overline{G_N φ^σ}
for each s ∈ S
   if s is not labeled \overline{G_N φ^σ}
      then label s with G_N φ^σ
```

The only type of formula that is left is $\chi^\sigma = [N]\varphi^\sigma$.

```
for each s ∈ S
   if s is labeled φ^{N,σ}
      then label s with [N]φ^σ
   else
      label s with \overline{[N]φ^σ}
```

Note that the way we ordered the list of subformulas and norms guarantees that the labels are available when needed. The preliminary steps can be done in $O(|\psi|^2)$. Labeling transitions with N takes $O(|R| \cdot |N|) \leq O(|R| \cdot |\psi|)$. Labeling states takes $O(|S| \cdot |R|)$. There are $O(|\psi|)$ different formulas and norms with which transitions/states need to be labeled, so the entire algorithm takes $O(|S| \cdot |R| \cdot |\psi|^2)$.

4 Example: Self-driving Cars

Suppose we have a racetrack[5] where a number of self-driving cars operate. We want to equip the cars with norms that will guarantee that they will (a) avoid collisions with each other and with stationary objects and (b) avoid "deadlock" situations where no one can act.

First, we need to represent a number of relevant facts in NAUL. Let *coll*, *drive_right*, $approach\_right_a$ and $intersect_a$ be propositional variables that represent "a collision happens", "the car is driving on the right side of the road", "a car approaches from the right (from a's point of view)" and "a is on an intersection" respectively. Also note that situations where no one can act are represented by $\Box\bot$.

We will first create a norm N_c that is supposed to prevent collisions. The norm should prevent collisions for every point in the future, but of course it cannot do so if a collision has already occurred. N_c is therefore successful if we have $\neg coll \to [N_c]G\neg coll$. The simplest way to guarantee this property is to disallow every action; we have $\models \neg coll \to [(\bot, \mathscr{A}, \bot)]G\neg coll$. Forbidding every action is not a very suitable solutions, however. Even though we did not explicitly encode this in the goal formula, we would like to have a *reasonable* norm, in the sense that (whenever possible) the norm allows at least one action. We therefore take $N_c := (\top, \mathscr{A}, \neg F coll)$, so agents are not allowed to take any action that will inevitably lead to a collision at some point in the future. This norm is indeed successful, we have $\models \neg coll \to [N_c]G\neg coll$.

Next, let us construct a norm N_d that prevents deadlock. Here we have to be a bit careful. We want N_d to prevent situations where no one can act. We can interpret this either as "there must be some available action that is, in principle, possible" or as "there must be some available action that is not only possible but also allowed." The norm N_d satisfies the first requirement if $G_{N_d}\Diamond\top$ holds, and the second requirement if $[N_d]G\Diamond\top$ holds. We will assume that $[N_d]G\Diamond\top$ is a more faithful representation of the natural language requirement to avoid deadlock. This means $(\bot, \mathscr{A}, \bot)$ will not do. Instead, we should take $N_d := (\top, \mathscr{A}, \neg F\Box\bot)$. This gives us $\models \neg F\Box\bot \to [N_d]G\Diamond\top$. In other words, as long as there is an infinite path the norm N_d forces agents to follow such a path.

Now that we have two norms N_c and N_d that individually prevent collisions and deadlock we only need to combine the norms. As discussed above, there are three ways to do so. The additive way of combining the norms is clearly not what we are looking for: $N_c + N_d$ allows agents to collide as long as they do not cause deadlock at the same time. The multiplicative combination $N_c \times N_d$ is more suitable, it prevents collisions as well as certain kinds of deadlock. However, the norm $N_c \times N_d$ is not quite what we are looking for either. The problem is that $N_c \times N_d$ allows agents to perform actions that result in a situation where movement, while possible, is disallowed because it will lead to a collision. The compositional combination solves this problem: the norm $N_c \circ N_d$ allows exactly those actions that lead to neither collisions nor situations where agents cannot or are not allowed to act. In other words, we have $\models \neg F(coll \vee \Box\bot) \to [N_c \circ N_d]G(\neg coll \wedge \Diamond\top)$.

4.1 Static and Dynamic Operators

The self-driving cars example is also useful for illustrating the difference between the static operators $\Box_N$, G_N, and F_N on the one hand, and the dynamic operator $[N]$ on the other. We have $\mathscr{M}, s \models G_N\varphi$ if φ holds after every sequence of action that starts in s and is allowed by N. Importantly, during the evaluation of φ it is not assumed that everyone follows N. We have $\mathscr{M}, s \models [N]G\varphi$ if, under the assumption that all

[5] Setting our example on a closed racetrack allows us to ignore complications arising from interaction between robots and humans. NAUL can, of course, be used to model norms in human-machine interactions as well. That would require more complicated norms, however, and we want to provide a relatively simple example.

agents follow N permanently from now on, every sequence of actions leads to a φ state. In this case, during the evaluation of φ, we do assume that all agents follow N.

Recall that N_c prevents collisions, we have $\models \neg coll \rightarrow [N_c]G\neg coll$. So, under the assumption that all agents permanently follows N_c, there will be no collisions. But now suppose that we do not completely trust the agents to follow the norm. If the agents do not follow norms at all, no norm can prevent collisions. A more interesting situation is if the agents try to follow the norm, but occasionally make mistakes. Under these circumstances we cannot fully eliminate the possibility of collisions. But we may be able to make them highly unlikely, by requiring that N_c not only avoids collisions, but also situations where a single mistake could cause a collision. We cannot phrase this stronger success condition as $[N_c]\varphi$ for any φ. After all, the φ in $[N_c]\varphi$ is evaluated under the assumption that all agents follow the norm N_c—so no mistakes are made. This is where the static operator G_{N_c} is useful. Consider the formula $G_{N_c}(\neg coll \wedge \Box\neg coll)$. The $\Box$ in that formula is not evaluated under the assumption that the agents follow N_c, so $G_{N_c}(\neg coll \wedge \Box\neg coll)$ holds exactly if every sequence of actions allowed by N_c leads to a state where there is no collision and no single action can cause a collision.

4.2 Simpler norms

Usually, the decision whether to adopt a norm is made in advance, while the decision whether an action is allowed by a norm has to be made in the heat of the moment. As such, the second type of decisions is more time-sensitive than the first. Now, consider what we need to know in order to decide whether to adopt N. Typically, we will have some goal formula φ_g and we have to compute whether N guarantees this goal, so whether $[N]\varphi_g$ holds. If, on the other hand, we want to determine whether an action is allowed by N, the goal formula is irrelevant. All we need to do is determine in which worlds the formulas contained in N hold. While the model checking problem for NAUL can be solved in polynomial time, some operators still take more time than others. In particular, G_N and F_N are relatively expensive while $\Box_N$ and the Boolean connectives are relatively cheap. As such, it is a good idea to avoid G_N and F_N inside norms. That way the most time sensitive decision can be made quickly.

Recall that we chose $N_c = (\top, \mathscr{A}, \neg F\Box coll)$. The formula $\neg F\Box coll$ is relatively expensive, so ideally we would replace it by a simpler formula. It may be useful to compare our norm N_c to traffic regulations. The primary purpose of such regulations is to prevent collisions, but instead of the general rule "don't cause collisions" they tend to contain a lot of specific instructions such as "drive on the right" and "stop at a red traffic light."[6] The result is that while it is hard for the lawmakers to decide which rules should be adopted, it is easy for a driver to determine whether an action is allowed by the rules.

We should try to do something similar for our anti-collision norm. We could, for example, create two new norms $N_r := (\top, \mathscr{A}, drive\_right)$ and $N_y := \bigcup_{a\in\mathscr{A}}\{(\neg approach\_right_a, a, \top), (\top, a, \neg intersect_a)\}$, which state that agents should drive on the right and that they should not move on to an intersection if another agent is approaching from the right. Whether the combined norm $N_r \times N_y$ is effective (so whether $[N_r \times N_y]G\neg coll$ holds) depends on the details of the racetrack on which the agents operate. But if $N_r \times N_y$ is effective, it is a more suitable norm than N_c.

5 Expressivity

In order to better determine the place of NAUL in the landscape of different logics, we will compare its expressivity to that of two other salient logics: Computation Tree Logic (CTL) [4] and Arrow Update

[6]Many jurisdictions do have a few general rules in addition to the specific ones, like a ban on reckless driving.

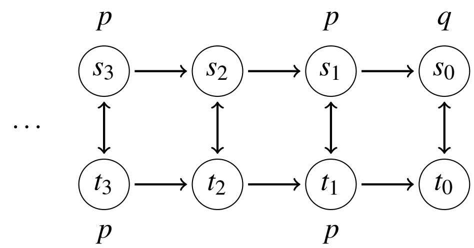


Figure 1: The model $\mathscr{M}_{CTL}$. The states s_i and t_i satisfy p iff i is odd.

Logic (AUL*) [7].[7] For reasons of brevity we do not provide definitions of the logics that we compare NAUL to, full definitions can be found in the cited publications. We show that NAUL is strictly more expressive than CTL and AUL*.

Remark 3. CTL is usually interpreted over different models than AUL* and NAUL. In particular, CTL tends to use single-agent serial models. Strictly speaking, this makes it impossible to compare the expressivity of NAUL to that of CTL. This problem can be solved by either extending CTL to multi-agent non-serial models—which can be done in a straightforward way—or by restricting AUL* and NAUL to single-agent serial models. The results presented here hold regardless of which of these solutions we use.

5.1 NAUL vs. CTL

First, we show that NAUL is at least as expressive as CTL. The subset $\neg, \vee, AX$, AF and EU of CTL operators is sufficient to define all of CTL. The operators $\neg, \vee, AX$ and AF are also NAUL operators, although AX and AF are denoted $\Box$ and F in NAUL. As such, it suffices to show that EU can be defined in NAUL.

Lemma 1. *We have* $\models E(\varphi U \psi) \leftrightarrow \neg G_{(\varphi,\mathscr{A},\top)}\neg\psi$.

Proof. We have $\mathscr{M}, s \models \neg G_{(\varphi,\mathscr{A},\top)}\neg\psi$ if and only if there is a $(\varphi,\mathscr{A},\top)$ path from s that contains a ψ state. Because such a path is a $(\varphi,\mathscr{A},\top)$ path, it contains only φ states before the ψ state. As such, $\mathscr{M}, s \models E(\varphi U \psi) \leftrightarrow \neg G_{(\varphi,\mathscr{A},\top)}\neg\psi$. This is true for any $\mathscr{M}, s$, so $\models E(\varphi U \psi) \leftrightarrow \neg G_{(\varphi,\mathscr{A},\top)}\neg\psi$. $\Box$

Left to show is that CTL is not at least as expressive as NAUL. Consider the model $\mathscr{M}_{CTL}$ shown in Figure 1, and note that the NAUL formula $[(p,\mathscr{A},\neg p),(\neg p,\mathscr{A},p)]G\neg q$ distinguishes between $\mathscr{M}_{CTL}, s_i$ and $\mathscr{M}_{CTL}, t_i$ for all $i \in \mathbb{N}$. We show that there is no CTL formula that similarly distinguishes s_i from t_i.

Lemma 2. *Let φ be any CTL formula, and let n be the modal depth of φ. Then φ does not distinguish between $\mathscr{M}_{CTL}, s_i$ and $\mathscr{M}_{CTL}, t_i$ for $i > n$.*

Proof. By induction. As base case, suppose $n = 0$. Then φ is a Boolean formula, so it cannot distinguish between s_i and t_i for $i \neq 0$. Assume as induction hypothesis that the lemma holds for all $n' < n$.

If a Boolean combination distinguishes between two states then so does at least one of the combined formulas, so we can assume without loss of generality that the main connective of φ is AX, AF or EU.

- Suppose $\varphi = AX\psi$. In order for φ to distinguish between s_i and t_i, ψ must distinguish between t_i and s_i or s_{i-1} and t_{i-1}. This contradicts the induction hypothesis, since ψ is of modal depth $n-1$.

[7] For technical reasons NAUL and NTL are trivially incomparable in expressivity.

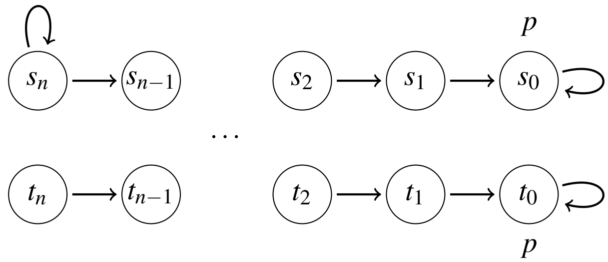


Figure 2: The model $\mathscr{M}^n_{AUL}$.

- Suppose $\varphi = AF\psi$. There are two possibilities. Firstly, ψ may hold on either s_i or t_i. Then, by the induction hypothesis it holds on both s_i and t_i. As such, φ holds on both s_i and t_i, and therefore does not distinguish between them.

  The second possibility is that ψ holds on neither s_i nor t_i. Suppose φ does not hold on s_i, so there is some path $s_i \longmapsto x_1 \longmapsto x_2 \longmapsto \cdots$ that does not contain a ψ state. Then the path $t_i \longmapsto s_i \longmapsto x_1 \longmapsto x_2 \longmapsto \cdots$ also does not contain a ψ state. So φ does not hold on t_i. Analogously, if φ does not hold on t_i then it does not hold on s_i. This shows that φ does not distinguish between s_i and t_i.

- Suppose $\varphi = E(\psi_1 U \psi_2)$. There are three possibilities. The first possibility is that ψ_2 holds on either of s_i and t_i and therefore—by the induction hypothesis—on both. Then φ holds on both states, and therefore does not distinguish between them.

  The second possibility is that both ψ_1 and ψ_2 hold on neither state. Then φ holds on neither state, and therefore does not distinguish between them.

  The final possibility is that ψ_1 holds on neither state, but ψ_1 holds on either and therefore—by the induction hypothesis—both states. Suppose φ holds on s_i. Then there is some path $s_i \longmapsto x_1 \longmapsto x_2 \longmapsto \cdots$ that satisfies ψ_1 until ψ_2. This implies that the path $t_i \longmapsto s_i \longmapsto x_1 \longmapsto x_2 \longmapsto \cdots$ also satisfies ψ_1 until ψ_2, so φ holds on t_i as well. Analogously, this reasoning shows that if φ holds on t_i then it also holds on s_i. This shows that φ does not distinguish between s_i and t_i.

In all cases, φ doesn't distinguish s_i from t_i. This completes the induction step and thereby the proof. □

Theorem 1. *NAUL is strictly more expressive than CTL.*

Proof. Lemma 1 shows that NAUL is at least as expressive as CTL. Lemma 2 shows that there is no CTL formula equivalent to $[(p,\mathscr{A},\neg p),(\neg p,\mathscr{A},p)]G\neg q$, so CTL is not at least as expressive as NAUL. □

5.2 NAUL vs. AUL*

The only difference between NAUL and AUL* is that NAUL has an F_N operator while AUL* does not. NAUL is therefore trivially at least as expressive as AUL*. Left to show is that AUL* is not at least as expressive as NAUL. In order to do so, we will use a sequence of models $\mathscr{M}^n_{AUL}$, which is shown in Figure 2. For reasons of brevity we will assume that AUL* does not contain the $[N]$ operator; we can safely do this because $[N]$ can be seen as an abbreviation in both NAUL and AUL*.

Lemma 3. *Let φ be any AUL* formula, and let m be the modal depth of φ. Then, for every $n > m$ and every $n \geq i, j > m$, φ does not distinguish between $\mathscr{M}^n_{AUL}, s_i$ and $\mathscr{M}^n_{AUL}, t_j$. Furthermore, for every $n > i \geq 0$, φ does not distinguish between $\mathscr{M}^n_{AUL}, s_i$ and $\mathscr{M}^n_{AUL}, t_i$.*

Proof. The second claim in the lemma is trivial: for every $i < n$, the states s_i and t_i are bisimilar and these logics respect bisimilarity. It remains to show that φ cannot distinguish between s_i and t_j for $i, j > m$. We do this by induction. As base case, suppose $m = 0$. For every $i, j > 0$, the states s_i and t_j agree on all propositional variables, so φ does not distinguish between them.

Suppose then as induction hypothesis that $m > 0$ and that the lemma holds for all $m' < m$. If a Boolean combination of formulas distinguishes between two states then so does at least one of the combined formulas, so we can assume without loss of generality that the main connective of φ is $\Box_N$ or G_N.

- Suppose $\varphi = \Box_N \psi$. In order for φ to distinguish between s_i and t_j it is necessary for either ψ or one of the formulas in N to distinguish between s_i and t_j, or between s_{i-1} and t_{j-1}. Each of the formulas in N as well as ψ are of modal depth $\leq m-1$, so by the induction hypothesis they cannot distinguish between these states. This implies that φ does not distinguish between s_i and t_j.
- Suppose $\varphi = G_N \psi$. In order to distinguish between s_i and t_j, exactly one of the states must have a path containing a $\neg\psi$ state. There are two ways this could happen: either there is some k such that exactly one of s_k and t_k satisfies ψ, or there is a k such that s_k and t_k both satisfy $\neg\psi$, but only one of them is reachable from s_i or t_j by an N-path.

  The first option cannot occur; the induction hypothesis implies that ψ cannot distinguish between s_k and t_k for any k. The second option also cannot occur. Such a reachability difference would require some formula in N to distinguish between s_k and t_k with $k < n$ or between s_k and t_l with $k, l > m-1$. The induction hypothesis implies that neither distinction is possible.

In both cases, φ doesn't distinguish s_i from t_j. This completes the induction step and thereby the proof. □

Theorem 2. *NAUL is strictly more expressive than AUL*.*

Proof. NAUL is trivially at least as expressive as AUL*. From Lemma 3 it follows that there is no AUL* formula equivalent to the NAUL formula Fp. □

6 Conclusion

We introduced Normative Arrow Update Logic (NAUL), a logic that used techniques from Arrow Update Logic (AUL*) and applies them to ideas from Normative Temporal Logic (NTL). Using NAUL, we can distinguish between additive, multiplicative and sequential combination of norms, as well as between dynamic and static ways to consider norms. We have shown that the model checking problem of NAUL can be solved in polynomial time. Furthermore, we have shown and that NAUL is strictly more expressive than AUL* and CTL. In particular, this means that the EU operator from CTL can be simulated in NAUL, and that the F_N operator from NAUL cannot be simulated in AUL*.

We will close by mentioning a few salient topics for further research. Firstly, we still need to find an axiomatization, as well as an algorithm that solves the satisfiability problem for NAUL. Secondly, in NAUL an action can only be completely allowed, or completely disallowed. In effect, this means NAUL makes use of an Andersonian sanction [2]. As a result, NAUL cannot model so-called contrary-to-duty obligations, see [3, 6, 8]. It may therefore be interesting to extend NAUL with operators that allow actions to be somewhere in between allowed and disallowed. Thirdly, the model checking algorithm presented in this paper runs in polynomial time but is still rather inefficient. Developing a more efficient algorithm might therefore be interesting. Finally, it may be interesting to develop a variant of Arbitrary Arrow Update Logic (AAUL) [5] that can be applied to NAUL. Such a variant of AAUL would provide us with formulas of the form $[\cdot]\varphi$, meaning "there is some norm that guarantees the truth of φ."

References

[1] Thomas Ågotnes, Wiebe van der Hoek, Juan A. Rodríguez-aguilar, Carles Sierra & Michael Wooldridge (2007): *On the logic of normative systems*. In: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI07)*, pp. 1175–1180.

[2] Alan Ross Anderson & Omar Khayyam Moore (1957): *The Formal Analysis of Normative Concepts*. *American Sociological Review* 22, pp. 9–17.

[3] Roderick M. Chisholm (1963): *Contrary-to-Duty Imperatives and Deontic Logic*. *Analysis* 24, pp. 33–36.

[4] Edmund M. Clarke & E. Allen Emerson (1982): *Design and synthesis of synchronization skeletons using branching time temporal logic*. In Dexter Kozen, editor: *Logics of Programs*, *LNCS* 131, pp. 52–71.

[5] Hans van Ditmarsch, Wiebe van der Hoek & Barteld Kooi: *Arbitrary Arrow Update Logic*. Presented at AIML 2014.

[6] James William Forrester (1984): *Gentle Murder, or the Adverbial Samaritan*. *The Journal of Philosophy* 81, pp. 193–197.

[7] Barteld Kooi & Bryan Renne (2011): *Arrow Update Logic*. *The Review of Symbolic Logic* 4, pp. 536–559.

[8] Louwe B. Kuijer (2012): *Sanction Semantics and Contrary-to-Duty Obligations*. In D. Elgesem T. Ågotnes, J. Broersen, editor: *Deontic Logic in Computer Science, Proceedings of the 11th International Conference on Deontic Logic in Computer Science (DEON 2012)*, pp. 76–90.

[9] Yoav Shoham & Moshe Tennenholtz (1992): *On the Synthesis of Useful Social Laws for Artificial Agent Societies*. In: *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI'92, pp. 276–281.

14

Regular Paper

The risk of divergence

Pierre Lescanne
University of Lyon, École normale supérieure de Lyon, CNRS (LIP),
46 allée d'Italie, 69364 Lyon, France

We present infinite extensive strategy profiles with perfect information and we show that replacing finite by infinite changes the notions and the reasoning tools. The presentation uses a formalism recently developed by logicians and computer science theoreticians, called coinduction. This builds a bridge between economic game theory and the most recent advance in theoretical computer science and logic. The key result is that rational agents may have strategy leading to *divergence*.

Keywords: divergence, decision, infinite game, sequential game, coinduction.

1 Introduction

Strategies are well described in the framework of sequential games, aka. games in extensive forms with perfect information. In this paper, we describe rational strategies leading to divergence.[1] Indeed divergence understands that the games, the strategies and the strategy profiles are infinite. We present the notion of infinite strategy profiles together with the logical framework to reason on those objects, namely coinduction.

2 Decisions in Finite Strategy Profiles

To present strategy reasoning, we use one of the most popular framework, namely *extensive games with perfect information* ([12] Chapter 5 or [3]) and we adopt its terminology. In particular we call *strategy profile* an *organized* set of strategies, merging the decisions of the agents. This organization mimics this of the game and has the same structure as the game itself. They form the set[2] $\mathsf{StratProf}$. By *"organized"*, we mean that the strategic decisions are associated with the nodes of a tree which correspond to positions where agents have to take decisions. In our approach strategy profiles are first class citizens and games are byproduct. In other words, strategy profiles are defined first and extensive games are no more than strategy profiles where all the decisions have been erased. Therefore we will only speak about strategy profiles, keeping in mind the underlying extensive game, but without giving them a formal definition[3]. For simplicity and without loss of generality, we consider only dyadic strategy profiles (i.e.; double choice strategy profiles) , that are strategy profiles with only two choices at each position. Indeed it is easy to figure out how multiple choice extensive

[1] In this paper we use "divergence" instead of "escalation" since it is somewhat dual convergence a concept which plays a key role in what follows.

[2] To be correct, we should say the *"they form the coalgebra"*.

[3] A direct definition of games is possible, but is not necessary in this paper.

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

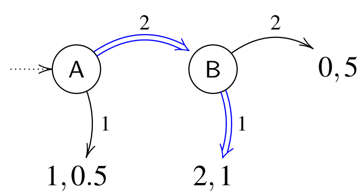
strategy profiles can be represented by double choice extensive strategy profiles. We let the reader imagine such an embedding. Therefore, we consider a set of choices: $\mathsf{Choice} = \{1,2\}$.

Along the paper, our examples need only a set of two agents: $\mathsf{Agent} = \{\mathsf{A},\mathsf{B}\}$. In this paper we use coinduction and corecursion as basic tools for reasoning correctly about and defining properly infinite objects. Readers who want to know more about those concepts are advised to read introductory papers [5, 14], while specific applications to infinite strategy profiles and games are introduced in [9].

Definition 1 *A* finite strategy profile *is defined* by induction *as follows:*

- *either given a* utility assignment u *(i.e., a function* $u : \mathsf{Agent} \to \mathbb{R}$*)* $\langle\langle u\rangle\rangle$ *is a* finite strategy profile*, which corresponds to an ending position.*
- *or given an* agent a*, a* choice c *and two* finite strategy profiles s_1 *and* s_2*,* $\langle\langle a,c,s_1,s_2\rangle\rangle$ *is a* finite strategy profile*.*

For instance, a strategy profile can be drawn easily with the convention that 1 is represented by going down and 2 is represented by going right. The chosen transition is represented by a double arrow . The other transition is represented by a simple arrow . For instance



is a graphic representation of the strategy profile

$$s_\alpha = \langle\langle \mathsf{A}, 2, \langle\langle \mathsf{A}\mapsto 1, \mathsf{B}\mapsto 0.5\rangle\rangle, \langle\langle \mathsf{B}, 1, \langle\langle \mathsf{A}\mapsto 2, \mathsf{B}\mapsto 1\rangle\rangle, \langle\langle \mathsf{A}\mapsto 0, \mathsf{B}\mapsto 5\rangle\rangle\rangle\rangle\rangle\rangle.$$

From a finite strategy profile, say s, we can define a *utility assignment*, which we write $\widehat{s}$ and which we define as follows:

- $\widehat{\langle\langle u\rangle\rangle} = u$
- $\widehat{\langle\langle a,c,s_1,s_2\rangle\rangle} = \textbf{case } c \textbf{ of } 1 \to \widehat{s_1} \mid 2 \to \widehat{s_2}$

For instance $\widehat{s_\alpha}(\mathsf{A}) = 2$ and $\widehat{s_\alpha}(\mathsf{B}) = 1$.

We define an equivalence $s =_g s'$ among finite strategy profiles, which we read as *"s and s'* have the same (underlying) game*"*.

Definition 2 *We say that two strategy profiles s and s'* have the same game *and we write $s =_g s'$ iff by induction*

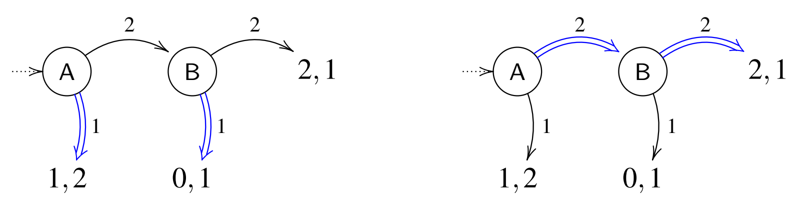
- *either* $s = \langle\langle u\rangle\rangle$ *and* $s' = \langle\langle u\rangle\rangle$
- *or* $s = \langle\langle a,c,s_1,s_2\rangle\rangle$ *and* $s' = \langle\langle a',c',s'_1,s'_2\rangle\rangle$ *and* $a = a'$, $s_1 =_g s_1$ *and* $s_2 =_g s'_2$.

We can define a family of finite strategy profiles that are of interest for decisions. First we start with *backward induction*. Following [20], we consider 'backward induction', not as a reasoning method, but as a predicate that specifies some strategy profiles.

Definition 3 (Backward induction) *A finite strategy profile s is* backward induction *if it satisfies the predicate* BI*, where* BI *is defined recursively as follows:*

- $\mathsf{BI}(\langle\!\langle u \rangle\!\rangle)$*, i.e., by definition an ending position is 'backward induction'.*
- $\mathsf{BI}(\langle\!\langle a, 1, s_1, s_2 \rangle\!\rangle) \;\Leftrightarrow\; \mathsf{BI}(s_1) \wedge \mathsf{BI}(s_2) \wedge \widehat{s_1} \geq \widehat{s_2}$.
- $\mathsf{BI}(\langle\!\langle a, 2, s_1, s_2 \rangle\!\rangle) \;\Leftrightarrow\; \mathsf{BI}(s_1) \wedge \mathsf{BI}(s_2) \wedge \widehat{s_2} \geq \widehat{s_1}$.

In other words, a strategy profile which is not an ending position is 'backward induction' if both its direct strategy subprofiles are and if the choice leads to a better utility, as shown by the comparison of the utility assignments to the direct strategy subprofiles. The two following strategy profiles are 'backward induction' [12](Example 158.1)



An agent is rational if she makes a choice dictated by backward induction and if she keeps being rational in the future. We write this predicate Rat_f where the index f insists on finiteness making it distinct from the predicate Rat_∞ on infinite strategy profiles.

Definition 4 (Rationality for finite strategy profiles) *The predicate* Rat_f *is defined recursively as follows:*

- $\mathsf{Rat}_f(\langle\!\langle u \rangle\!\rangle)$,
- $\mathsf{Rat}_f(\langle\!\langle a, c, s_1, s_2 \rangle\!\rangle) \Leftrightarrow \exists \langle\!\langle a, c, s'_1, s'_2 \rangle\!\rangle \in \mathsf{StratProf}$,
  - $\langle\!\langle a, c, s'_1, s'_2 \rangle\!\rangle =_g \langle\!\langle a, c, s_1, s_2 \rangle\!\rangle$
  - $\mathsf{BI}(\langle\!\langle a, c, s'_1, s'_2 \rangle\!\rangle)$
  - $\mathsf{Rat}_f(s_c)$

Then we can state a variant of Aumann theorem [1] saying that backward induction coincides with rationality.

Theorem 5 $\forall s \in \mathsf{StratProf}, \mathsf{Rat}_f(s) \Leftrightarrow \mathsf{BI}(s)$.

3 Decisions in Infinite Strategy Profiles

We extend the concept of backward induction and the concept of rationality to infinite strategy profiles. For that, we replace induction by coinduction.[4] Notice that we mix up recursive and corecursive definitions, and that we reason sometime by induction and sometime by coinduction. Therefore we advise the reader to be cautious and to pay attention to when we use one or the other. We write $\mathsf{InfStratProf}$ the set of finite or infinite strategy profiles.

[4] For readers not familiar with coinduction and not willing to read [5] or [14], we advise her to pretend just that corecursive definitions define infinite objects and coinduction allows reasoning specifically on their infinite aspects, whereas recursive definition define finite objects and induction allows reasoning on their finite aspects.

Definition 6 *The set* finite or infinite strategy profiles $\mathsf{InfStratProf}$ *is defined* corecursively *as follows:*

- *either given a* utility assignment u*, then* $\langle\langle u\rangle\rangle \in \mathsf{InfStratProf}$*, which corresponds to an ending position.*
- *or given an* agent a*, a* choice c *and two* strategy profiles $s_1 \in \mathsf{InfStratProf}$ *and* $s_2 \in \mathsf{InfStratProf}$*, then* $\langle\langle a,c,s_1,s_2\rangle\rangle \in \mathsf{InfStratProf}$.

We cannot define the utility assignments on all infinite strategy profiles, only on those on which the utility can be "computed". The strategy profiles on which utility assignments are defined are called *convergent*, since when one follows the path indicated by the choices one "converges", that is that one gets to an ending position, i.e., a position where utilities are actually attributed. The predicate *convergent* is defined by induction, meaning that, on s, after finitely many steps following the choices of s an ending position is reached. "Finitely many steps" is a finite aspect and this is why we use an inductive definition.

Definition 7 (Convergent) *Saying that* s *is* convergent *is written* $\downarrow s$*.* $\downarrow s$ *is defined by induction as follows:*

- $\downarrow \langle\langle u\rangle\rangle$ *or*
- *if* $\downarrow s_1$ *then* $\downarrow \langle\langle a,1,s_1,s_2\rangle\rangle$ *or*
- *if* $\downarrow s_2$ *then* $\downarrow \langle\langle a,2,s_1,s_2\rangle\rangle$ *or*

On convergent strategy profiles we can assign utilities. The resulting function is written $\hat{s}$ when applied to a strategy profile s.

Definition 8 (Utility assignment) $\hat{s}$ *is defined corecursively on every strategy profile.*

$$\begin{array}{lllll} when & s = \langle\langle u\rangle\rangle & \widehat{s} & = & f \\ when & s = \langle\langle a,1,s_1,s_2\rangle\rangle & \widehat{s} & = & \widehat{s_1} \\ when & s = \langle\langle a,2,s_1,s_2\rangle\rangle & \widehat{s} & = & \widehat{s_2} \end{array}$$

The function $\hat{\cdot}$ *has to be specified on an infinite object and this is why we use a corecursive definition.*

Proposition 9 *If* $\downarrow s$*, then* $\hat{s}$ *returns a value.*

Actually convergent strategy profiles are not enough as we need to know the utility assignment not only on the whole strategy profile but also on strategy subprofiles. For that, we need to insure that from any internal position we can reach an ending position, which yields that on any position we can assign a utility. We call *always-convergent* such a predicate[5] and we write it $\Box\!\downarrow$.

Definition 10 (Always-convergent)

- $\Box\!\downarrow \langle\langle u\rangle\rangle$ *that is that for whatever* u*,* $\langle\langle u\rangle\rangle$ *is always-convergent*
- $\Box\!\downarrow \langle\langle a,c,s_1,s_2\rangle\rangle$ *if*
  - $\langle\langle a,c,s_1,s_2\rangle\rangle$ *is convergent (i.e.,* $\downarrow \langle\langle a,c,s_1,s_2\rangle\rangle$*), and*
  - s_1 *is always-convergent (i.e.,* $\Box\!\downarrow s_1$*), and*
  - s_2 *is always-convergent (i.e.,* $\Box\!\downarrow s_2$*).*

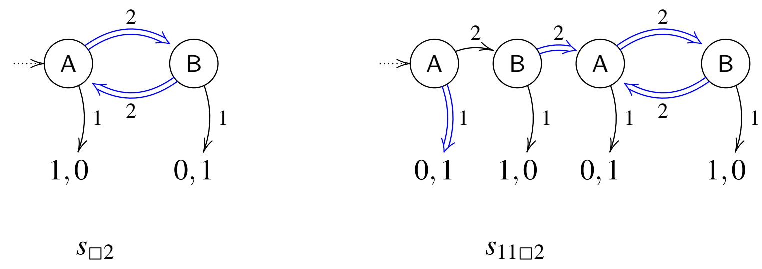


Figure 1: Two examples of strategy profiles

Proposition 11 $\Box{\downarrow} s \quad \Rightarrow \quad {\downarrow} s.$

$s_{\Box 2}$ in Figure 1 is a typically non convergent strategy profile, wherever $s_{12\Box 2}$ in the same figure is a typically convergent and not always-convergent strategy profile.

Using the concept of always-convergence we can generalize the notion of backward induction to this that the tradition calls *subgame perfect equilibrium* [15] and which we write here SPE. In short SPE is a corecursive generalization of BI. First we define an auxiliary predicate.

Definition 12 (PE)

$$\begin{array}{rcll} \mathsf{PE}(s) & \Leftrightarrow & \Box{\downarrow} s & \wedge\ s = \langle\langle a,1,s_1,s_2\rangle\rangle \Rightarrow \widehat{s_1}(a) \geq \widehat{s_2}(a) \\ & & & \wedge\ s = \langle\langle a,2,s_1,s_2\rangle\rangle \Rightarrow \widehat{s_2}(a) \geq \widehat{s_1}(a) \end{array}$$

We define *SPE* as *always*-PE. In other words, a strategy profile s is a subgame perfect equilibrium if $\Box\mathsf{PE}(s)$. $\Box$ applies to a predicate.

Definition 13 (Always) *Given a predicate P, the predicate $\Box P$ is defined corecursively as follows.*

- *if $P(\langle\langle u\rangle\rangle)$ then $\Box P(\langle\langle u\rangle\rangle)$ and*
- *if $\Box P(s_1)$, $\Box P(s_2)$ and $P(\langle\langle a,c,s_1,s_2\rangle\rangle)$ then $\Box P(\langle\langle a,c,s_1,s_2\rangle\rangle)$*

Formally *SPE* is $\Box$PE. Besides we may notice that the notation used for always-convergence (Definition 10) is consistent with Definition 13. Now thanks to SPE we can give a notion of rationality for infinite strategy profiles. Like for finite strategy profiles we define corecursively, this time, an equivalence $s =_g s'$ on infinite strategy profiles (read s and s' *have the same game*). Two strategy profiles have the same game if at each step, they have the same agent and their respective direct strategy subprofiles have the same game and only the choices differ.

Definition 14 *We say that two strategy profiles s and s'* have the same game *and we write $s =_g s'$ iff corecursively*

- *either $s = \langle\langle u\rangle\rangle$ and $s' = \langle\langle u\rangle\rangle$*
- *or $s = \langle\langle a,c,s_1,s_2\rangle\rangle$ and $s' = \langle\langle a',c',s'_1,s'_2\rangle\rangle$ and $a = a'$, $s_1 =_g s_1$ and $s_2 =_g s'_2$.*

Definition 15 (Rationality for finite or infinite strategy profiles) Rat_∞ *is defined* corecursively *as follows.*

[5] Traditionally $\Box$ is the notation for the modality (i.e., the predicate transformer) *always*.

- $\mathsf{Rat}_\infty(\langle\langle u\rangle\rangle)$,
- $\mathsf{Rat}_\infty(\langle\langle a,c,s_1,s_2\rangle\rangle) \Leftrightarrow \exists \langle\langle a,c,s'_1,s'_2\rangle\rangle \in \mathsf{InfStratProf}$,
$$\langle\langle a,c,s'_1,s'_2\rangle\rangle =_g \langle\langle a,c,s_1,s_2\rangle\rangle \wedge \mathsf{SPE}(\langle\langle a,c,s'_1,s'_2\rangle\rangle) \wedge \mathsf{Rat}_\infty(s_c)$$

The reader may notice the similarity with Definition 4 of rationality for finite games. The difference is twofold: the definition is corecursive instead of recursive and BI has been replaced by SPE. Let us now define a predicate that states the opposite of convergence [6]

Definition 16 (Divergence) *$\uparrow s$ is defined corecursively as follows:*

- *if $\uparrow s_1$ then $\uparrow \langle\langle a,1,s_1,s_2\rangle\rangle$,*
- *if $\uparrow s_2$ then $\uparrow \langle\langle a,2,s_1,s_2\rangle\rangle$.*

$s_{\Box 2}$ in Figure 1 is a typical divergent strategy profile. The main theorem of this paper can then be stated, saying that there exists a strategy profile that is both divergent and rational.

Theorem 17 (Risk of divergence) $\exists s \in \mathsf{InfStratProf}, \mathsf{Rat}_\infty(s) \wedge \ \uparrow s.$

4 Extrapolating the centipede

As an illustration of the above concepts, we show, in this section, two simple extensions to infinity of a folklore example. The centipede has been proposed by Rosenthal [13]. Starting from a wording suggested by Aumann [1] we study two infinite generalization[7]. Wikipedia [24] says:

> Consider two players: Alice and Bob. Alice moves first. At the start of the game, Alice has two piles of coins in front of her: one pile contains 4 coins and the other pile contains 1 coin. Each player has two moves available: either **”take”** the larger pile of coins and give the smaller pile to the other player or **”push”** both piles across the table to the other player. Each time the piles of coins pass across the table, the quantity of coins in each pile doubles.

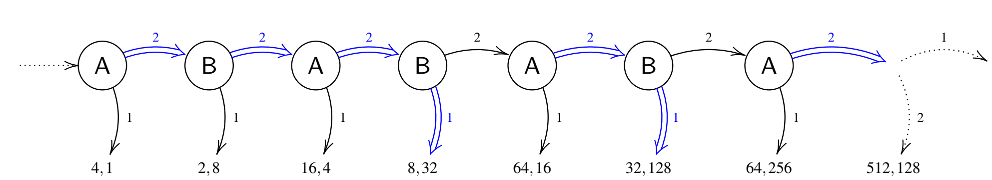


Figure 2: A sketch of a strategy profile of the ∞pede.

[6]People used to coinduction know why it is better to define *divergence* directly instead of defining it as the negation of convergence.

[7]The reason why we call them ∞*pede* and ω*pede*.

4.1 The ∞pede

$P_\infty(s)$ is a set of strategy profiles extending the strategy profiles of the centipede. Such an infinite strategy profile can only be is sketched on Figure 2 . Actually proposing an infinite extension of the centipede is quite natural for two reasons. First there is no natural way to make the game finite. Indeed in the definition of the game, nothing precise is said about its end, when no player decides to take a pile. For instance, Wikipedia [24] says:

> The game continues for a fixed number of rounds or until a player decides to end the game by pocketing a pile of coins.

We do no know what the utilities are in the end position described as "a fixed number of rounds". Since A started, we can assume that the end after a fixed number of rounds is B's turn and that there are outcomes like:

1. B receives 2^{n+1} coins and A receives 2^{n+3} coins like for the previous B rounds and that is all.
2. B chooses between
   (a) receiving 2^{n+1} coins whereas A receives 2^{n+3} or
   (b) sharing with A, each one receiving 2^{n+2}.
3. Both A and B receive nothing.

Moreover the statement *"Each player has two moves available: either **"take"** ... or **push**..."* is not true, in the ending position. We are not hair-splitting since the end positions are the initializations of the (backward) induction and must be defined as precisely as the induction step. Ending with 2.(b) does not produce the same backward induction as the others. Let us consider the strategy profiles

$$\begin{aligned} p_n &= \langle\langle \mathsf{A}, 1, \langle\langle \mathsf{A} \mapsto 2^{2n+2}, \mathsf{B} \mapsto 2^{2n} \rangle\rangle, \pi_n \rangle\rangle \\ \pi_n &= \langle\langle \mathsf{B}, 1, \langle\langle \mathsf{A} \mapsto 2^{2n+1}, \mathsf{B} \mapsto 2^{2n+3} \rangle\rangle, p_{n+1} \rangle\rangle \end{aligned}$$

In words, the p_n's and the π_n's are the strategy subprofiles of the ∞pede in which Alice and Bob stop always. Notice that

$$\begin{array}{ll} \widehat{p_n}(\mathsf{A}) = 2^{2n+2} & \widehat{p_n}(\mathsf{B}) = 2^{2n} \\ \widehat{\pi_n}(\mathsf{A}) = 2^{2n+1} & \widehat{\pi_n}(\mathsf{B}) = 2^{2n+3} \end{array}$$

Theorem 18

1. $\forall n \in \mathbb{N}, \mathsf{SPE}(p_n) \wedge \mathsf{SPE}(\pi_n)$,

2. $\forall s \in \mathsf{InfStratProf}, s =_g p_0 \wedge \mathsf{SPE}(s) \quad \Leftrightarrow s = p_0$.

In other words, all the p_n's and the π_n's are 'backward induction'. Moreover for the ∞pede, p_0 is the only 'backward induction'. strategy profile.

Proof: One can easily prove that for all n, $\Box\!\downarrow p_n$ and $\Box\!\downarrow \pi_n$.

Assuming $\mathsf{SPE}(\pi_n)$ and $\mathsf{SPE}(p_{n+1})$ (coinduction) and since

$$\langle\langle \mathsf{A} \mapsto 2^{2n+2}, \widehat{\mathsf{B} \mapsto 2^{2n}} \rangle\rangle(\mathsf{A}) \geq \widehat{\pi_n}(\mathsf{A})$$

we conclude that $\mathsf{SPE}(\langle\langle \mathsf{A}, 1, \langle\langle \mathsf{A} \mapsto 2^{2n+2}, \mathsf{B} \mapsto 2^{2n} \rangle\rangle, \pi_n \rangle\rangle)$ that is $\mathsf{SPE}(p_n)$. The proof of $\mathsf{SPE}(\pi_n)$ is similar.

For the proof of 2. we notice that in a strategy profile in SPE with the same game as p_0, there is no strategy subprofile such that the agent chooses 2 and the next agent chooses 1. Assume the strategy subprofile is $s_n = \langle\!\langle \mathsf{A}, 2, \langle\!\langle \mathsf{A} \mapsto 2^{2n+2}, \mathsf{B} \mapsto 2^{2n} \rangle\!\rangle, \langle\!\langle \mathsf{B}, 1, \langle\!\langle \mathsf{A} \mapsto 2^{2n+1}, \mathsf{B} \mapsto 2^{2n+3} \rangle\!\rangle, \sigma_n \rangle\!\rangle \rangle\!\rangle$. ans that $\mathsf{SPE}(s_n)$ and $\mathsf{SPE}(\sigma_n)$. If it would be the case and if we write $t = \langle\!\langle \mathsf{A} \mapsto 2^{2n+2}, \mathsf{B} \mapsto 2^{2n} \rangle\!\rangle$ and $t' = \langle\!\langle \mathsf{B}, 1, \langle\!\langle \mathsf{A} \mapsto 2^{2n+1}, \mathsf{B} \mapsto 2^{2n+3} \rangle\!\rangle, \sigma_n \rangle\!\rangle$, we notice that $\hat{t}(A) = 2^{2n+2} > \widehat{t'}(A) = 2^{2n+1}$. This is in contradiction with $\mathsf{SPE}(s_n)$. $\square$

We deduce that the strategy profile d_0, which diverges, is not in Rat_∞ and more generally there is no strategy profile in Rat_∞ for the ∞pede.

$$\begin{array}{rcl} d_n & = & \langle\!\langle \mathsf{A}, 2, \langle\!\langle \mathsf{A} \mapsto 2^{2n+2}, \mathsf{B} \mapsto 2^{2n} \rangle\!\rangle, \pi_n \rangle\!\rangle \\ \delta_n & = & \langle\!\langle \mathsf{B}, 2, \langle\!\langle \mathsf{A} \mapsto 2^{2n+1}, \mathsf{B} \mapsto 2^{2n+3} \rangle\!\rangle, p_{n+1} \rangle\!\rangle \end{array}$$

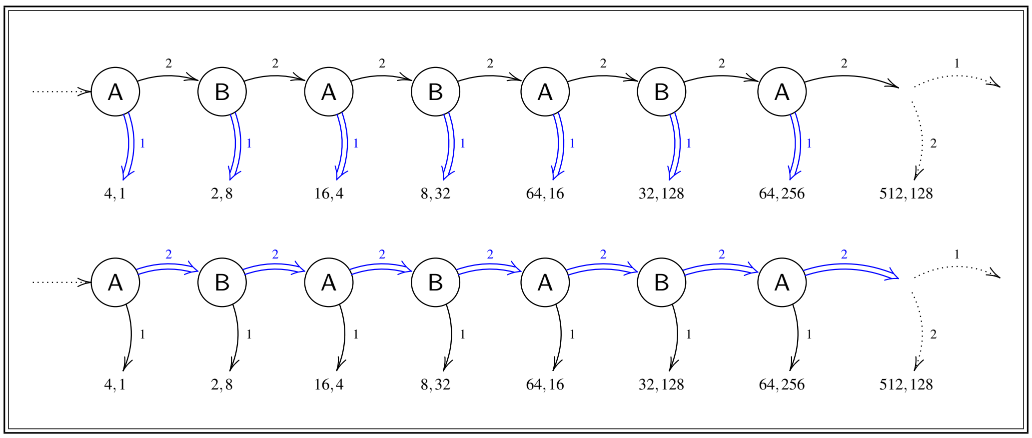


Figure 3: Strategy profiles p_0 and d_0 of the ∞pede.

4.2 The ωpede

We know[8] that *"trees don't grow to the sky"*. In our case this means that there is a natural number ω after which piles cannot be doubled.footnotePeople speak of *limited payroll*. In other words, after ω, the piles keep the same size 2^ω. An example of strategy profile is sketched on Figure 4. In this family of strategy profiles, which we write P_ω , the utilities stay stable after the ω^{th} positions. Every *always-convergent* strategy profile of P_ω, such that agents **push** until ω is in SPE. We conclude the existence of rational divergent strategy profiles in P_ω. In other words in the ω*pede* there is a risk of divergence.

Theorem 19 $\exists s \in P_\omega, \mathsf{Rat}_\infty(s) \wedge \uparrow s$.

One may imagine that divergence is when optimistic agents hope a reverse of the reverse of tendency.

[8] Usually agents do not believe this. See [9] for a discussion of the beliefs of the agents w.r.t. the infiniteness of the world.

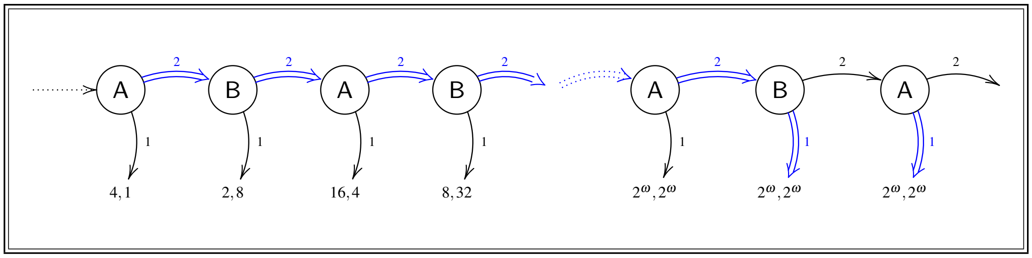


Figure 4: A 'backward induction' strategy profile for the ωpede.

Comments: The ωpede example is degenerated, but it is interesting in two respects. First, it shows a very simple and naive case of rational divergence. Second it shows that cutting the infinite game, case 2. (b) is the most natural way, with a equilibrium in which agents **take** until the end.

5 Two examples

$0,1$ **strategy profiles** $0,1$ strategy profiles are strategy profiles with the shape of an infinite "comb" in which the utilities are 0 for the agent who quits and 1 for the other agent. It can be shown [8] that strategy profiles where one agent continues always and the other quits infinitely often (in other words the other agent never continues always) are in SPE. For this reason, the strategy profile where both agents continue always is in Rat_∞, which shows that divergence is rational.

The dollar auction The *dollar auction* is a well known game [16, 6, 11]. Its strategy profiles have the same infinite comb shape as the $0,1$ strategy profiles, the ∞pede and the ωpede with the sequence of pairs of utilities:

$$(0,100)\ (95,0)\ (-5,95)\ (90,-5)\ (-10,90)\ (85,-10)\ \ldots\ (-5n,100-5n)\ (100-5(n+1),-5n)\ \ldots$$

and corresponds to an auction in which the bet of the looser is not returned to her. We have shown [10] that the dollar auction may diverge with rational agents. People speak of *escalation* in this case. The divergent strategy profile of the dollar auction is in Rat_∞.

6 Reflection

Examples like the dollar auction or the $0,1$ raise the following question: "How is it possible in an escalation that the agents do not see that they are entering a hopeless process?". The answer is "reflection". Indeed, when reasoning, betting and choosing, the agents should leave the world where they live and act in order to observe the divergence. If they are wise, they change their beliefs in an infinite world as soon as they realize that they go nowhere [17]. This ability is called reflection and is connected to observability, from the theoretical computer science point of view, which is itself connected to coalgebras and to coinduction [5]. In other words, agents should leave the environment in which they are enclosed and observe themselves.

7 Singularities and divergence

Divergence is called singularity, bubble, crash, escalation, or turbulence according to the context or the scientific field. In mechanics this is considered as a topics by itself. Leonardo da Vinci's drawings show that he considered early turbulence and vortices and only Reynolds during the XIX$^{\text{th}}$ century studied it from a scientific point of view. In many other domains, phenomena of this family are rejected from the core of the field, despite they have been observed experimentally. Scientists, among them mainstream economists [2], prefer smoothness, continuity and equilibria [23] and they often claim that departing from this leads to "paradoxes" [16]. In [7], we surveyed Zeno of Elea's paradox from the point of view of coinduction, as well as Weierstrass function [22], the first mathematical example showing discontinuity at the infinite. Here we would like to address two other cases. In 1935, that is one year before his famous article in the *Proceedings of the London Mathematical Society* [19], Alan Turing wrote a paper [18] presenting his result for a publication in the Proceedings of the French Academy of Science. In this paper he calls "nasty" a machine that terminates and "nice" a machine that does not terminate, showing his positive view of non terminating computations.[9] In 1795, Laplace published his book *Exposition du Système du Monde* and proposed the first clear vision of the notion of blackhole, but probably in order not to hurt his contemporaries, he found wiser to remove this presentation from the third edition of his book. Then we had to wait Schwartzschild in 1915, few months after the publication by Einstein of the general theory of relativity, for a second proposal of the concept of blackhole. But at that time the general relativity was not yet fully accepted as were not blackholes. Only recently, at the end of the last century, the general relativity has been considered as "the" theory of gravitation and there is no more doubt on the existence of blackholes. Since blackholes are singularities in gravitation, they are for the general theory of relativity the equivalent of divergent strategy profiles for game theory.

Contribution of this paper

Unlike previous presentations of similar results [10, 8, 21] here we focus on the concept of strategy profile which is central for the those of convergence, of divergence and of equilibrium and is more targeted for a workshop on *strategy reasoning*. Moreover we introduce the *ωpede* (a new infinite version of the centipede) and *"divergent" strategy profiles* are those that where called "escalation" in previous literature. This terminology seems better fitted for its duality with convergence.

8 Conclusion

We have shown that strategy profiles in which no fixed limit is set must be studied as infinite objects using coinduction and corecursion. In these infinite objects, the risk of divergence is real and should be considered seriously.

[9] Notice that he changed his terminology in [18] and calls "circular" the terminating machine and "circular-free" the non terminating machine.

References

[1] Robert J. Aumann (1995): *Backward Induction and Common Knowledge of Rationality*. Games and Economic Behavior 8, pp. 6–19.

[2] David Colander, Richard P. F. Holt & Jr Rosser, Barkley J. (2004): *The Changing Face of Mainstream Economics*. Review of Political Economy 16(4), p. 485499, doi:10.1080/0953825042000256702.

[3] Shaun Hargreaves-Heap & Yanis Varoufakis (2004): *Game Theory: A Critical Introduction*. Routledge.

[4] Furio Honsell, Marina Lenisa & Rekha Redamalla (2012): *Categories of Coalgebraic Games*. In Branislav Rovan, Vladimiro Sassone & Peter Widmayer, editors: Mathematical Foundations of Computer Science 2012 - 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27-31, 2012. Proceedings, Lecture Notes in Computer Science 7464, Springer, pp. 503–515, doi:10.1007/978-3-642-32589-2_45. Available at http://dx.doi.org/10.1007/978-3-642-32589-2_45.

[5] Bart Jacobs (2012): *Introduction to Coalgebra. Towards Mathematics of States and Observations*. Online book. Version 2.0.

[6] Wolfgang Leininger (1989): *Escalation and cooperation in conflict situations*. J. of Conflict Resolution 33, pp. 231–254.

[7] Pierre Lescanne (2011): *Rationality and Escalation in Infinite Extensive Games*. CoRR abs/1112.1185. Available at http://arxiv.org/abs/1112.1185.

[8] Pierre Lescanne (2013): *A Simple Case of Rationality of Escalation*. In Reiko Heckel & Stefan Milius, editors: CALCO, Lecture Notes in Computer Science 8089, Springer, pp. 191–204. Available at http://dx.doi.org/10.1007/978-3-642-40206-7_15.

[9] Pierre Lescanne (2014): *Intelligent escalation and the principle of relativity*. Technical Report ensl-01096264v4, LIP - ENS Lyon.

[10] Pierre Lescanne & Matthieu Perrinel (2012): *"Backward" coinduction, Nash equilibrium and the rationality of escalation*. Acta Inf. 49(3), pp. 117–137. Available at http://dx.doi.org/10.1007/s00236-012-0153-3.

[11] Barry O'Neill (1986): *International escalation and the dollar auction*. J. of Conflict Resolution 30(33-50).

[12] Martin J. Osborne (2004): *An Introduction to Game Theory*. Oxford.

[13] R. W. Rosenthal (1981): *Games of perfect information, predatory pricing and the chain-store paradox*. Journal of Economy Theory 25(1), pp. 92–100.

[14] Jan J. M. M. Rutten (2000): *Universal coalgebra: a theory of systems*. Theor. Comput. Sci. 249(1), pp. 3–80. Available at http://dx.doi.org/10.1016/S0304-3975(00)00056-6.

[15] Reinhard Selten (1965): *Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit*. Zeitschrift für gesamte Staatswissenschaft 121.

[16] Martin Shubik (1971): *The Dollar auction game: A paradox in noncooperative behavior and escalation*. Journal of Conflict Resolution 15(1), pp. 109–111.

[17] Keith E. Stanovich (2009): *Two minds: Dual processis and beyond, J. Evans and K. Frankish eds.*, chapter Distinguishing the reflective, algorithmic, and autonomous minds: Is it time for a tri-process theory? Oxford University Press.

[18] A. M. Turing (1935): *First draft of précis of 'Computable Numbers', made for Comptes Rendues.* http://www.turingarchive.org/browse.php/K/4.

[19] A. M. Turing (1936): *On Computable Numbers, with an Application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society. Second Series 42, pp. 230–265.

[20] René Vestergaard (2006): *A Constructive Approach to Sequential Nash Equilibria*. Inf. Process. Lett. 97, pp. 46–51.

[21] René Vestergaard (2007): *Cascaded Games* 4545, pp. 185–201. Available at http://dx.doi.org/10.1007/978-3-540-73433-8_14.

[22] Karl Weierstrass (1872): *Über continuirliche Funktionen eines reellen Arguments, die für keinen Werth des letzteren einen bestimmten Differentialquotienten besitzen*. in Karl Weiertrass Mathematische Werke, Abhandlungen II Gelesen in der Königl. Akademie der Wisseschaften, am 18 Juli 1872, pp. 71–74.

[23] E. Roy Weintraub (2002): *Neoclassical Economics*. Library of Economics and Liberty. In David R. Henderson (ed.). *Concise Encyclopedia of Economics* (1st ed.).

[24] Wikipedia (2015): *Centipede game — Wikipedia, The Free Encyclopedia*. Available at http://en.wikipedia.org/w/index.php?title=Centipede_game&oldid=663077006. [Online; accessed 25-May-2015].

15

Regular Paper

Preference Refinement in Normative Multi-agent System

Xin Sun
Faculty of Science, Technology and Communication
University of Luxembourg
Luxembourg
xin.sun@uni.lu

In this paper we start with augmenting weighted boolean game with norms. Using ideas from input/output logic, the normative status of strategies are discussed. The preference relation in boolean games are refined by the normative status of strategies. Normative boolean game and notions like normative Nash equilibrium are then introduced. After formally presenting the model, we use an example to show that non-optimal Nash equilibrium can be avoided by making use of norms. We study the complexity issues related to normative status and normative Nash equilibrium.

Key words: boolean game, norm, input/output logic

1 Introduction

Generally speaking, the study of the interplay of games and norms can be divided into two main branches: the first, mostly originating from economics and game theory [20], treats norms as mechanisms that enforce desirable properties of social interactions; the second, that has its roots in social sciences and evolutionary game theory [13] views norms as (Nash or correlated) equilibrium that result from the interaction of rational agents. This paper belongs to the first branch. Our research question is:

How to regulate agents' behaviors using norms in boolean games?

Boolean game is a class of games based on propositional logic. It was firstly introduced by Harrenstein et al. [19] and further developed by several researchers [18, 15, 8]. In a boolean game, each agent i is assumed to have a goal, represented by a propositional formula ϕ_i over some set of propositional variables $\mathbb{P}$. Each agent i is associated with some subset $\mathbb{P}_i$ of the variables, which are under the unique control of agent i. The choices, or strategies, available to i correspond to all the possible assignment of truth or falsity to the variables in $\mathbb{P}_i$. An agent will try to choose an assignment so as to satisfy his goal ϕ_i. Strategic concerns arise because whether i's goal is in fact satisfied will depend on the choices made by other agents.

Norms are social rules regulating agents' behavior by prescribing which actions are obligatory, forbidden or permitted. In the boolean game theoretical setting, norms classify strategies as moral, legal or illegal. Such classification transforms the game by updating the preference relation in the boolean game. By designing norms appropriately, non-optimal equilibrium in the original game might be avoided. To represent (conditional) norms in boolean games, we need a logic of norms, which has been extensively studied in the deontic logic community.

In the first volume of the handbook of deontic logic [16], input/output logic [23] appears as one of the new achievement in deontic logic in recent years. Input/output logic takes its origin in the study of conditional norms. The basic idea is: norms are conceived as a deductive machine, like a black box which produces normative statements as output, when we feed it factual statements as input.

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

In this paper, using ideas from input/output logic, the normative status of strategies are discussed. The preference relation in boolean games are refined by the normative status of strategies. We understand this as the mechanism of norms regulate agents' behaviors. Normative boolean game and notions like normative Nash equilibrium are then introduced. We show in this paper non-optimal Nash equilibrium can be avoided by making use of norms.

The structure of this paper is the following: We present some background knowledge, including boolean game, input/output logic and complexity theory in Section 2. Normative boolean game is introduced and its complexity issues are studied in Section 3. We conclude this paper in Section 4.

2 Background

2.1 Propositional logic

Let $\mathbb{P} = \{p_0, p_1, \ldots\}$ be a finite set of propositional variables and $L_{\mathbb{P}}$ be the propositional language built from $\mathbb{P}$ and boolean constants $\top$ (true) and $\bot$ (false) with the usual connectives $\neg, \vee, \wedge, \to$ and $\leftrightarrow$. Formulas of $L_{\mathbb{P}}$ are denoted by ϕ, ψ etc. A literal is a variable $p \in \mathbb{P}$ or its negation. $2^{\mathbb{P}}$ is the set of the valuations for $\mathbb{P}$, with the usual convention that for $V \in 2^{\mathbb{P}}$ and $p \in V$, V gives the value true to p if $p \in V$ and false otherwise. $\models$ denotes the classical logical consequence relation.

Let $X \subseteq \mathbb{P}$, 2^X is the set of X-valuations. A partial valuation (for $\mathbb{P}$) is an X-valuation for some $X \subseteq \mathbb{P}$. Partial valuations are denoted by listing all variables of X, with a "$+$" symbol when the variable is set to be true and a "$-$" symbol when the variable is set to be false: for instance, let $X = \{p, q, r\}$, then the X-valuation $V = \{p, r\}$ is denoted $\{+p, -q, +r\}$. If $\{\mathbb{P}_1, \ldots, \mathbb{P}_n\}$ is a partition of $\mathbb{P}$ and $V_1, \ldots, V_n$ are partial valuations, where $V_i \in 2^{\mathbb{P}_i}$, $(V_1, \ldots, V_n)$ denotes the valuation $V_1 \cup \ldots \cup V_n$.

2.2 Boolean game

Boolean games introduced by Harrenstein *et al* [19] are zero-sum games with two players, where the strategies available to each player consist in assigning a truth value to each variable in a given subset of $\mathbb{P}$. Bonzon *et al* [7] give a more general definition of a boolean game with any number of players and not necessarily zero-sum. Sun [36] further generalizes boolean games such that the utility of each agent is not necessarily in $\{0, 1\}$. Such generalization is reached by representing the goals of each agent as a set of weighted formulas. We call such boolean game weighted boolean game. The idea of using weighted formulas to define utility can also be found in many literature among which we mention satisfiability game [4] and weighted boolean formula game [27].

Definition 1 (weighted boolean game) *A weighted boolean game is a 4-tuple* $(Agent, \mathbb{P}, \pi, Goal)$*, where*

1. $Agent = \{1, \ldots, n\}$ *is a set of agents.*
2. $\mathbb{P}$ *is a finite set of propositional variables.*
3. $\pi : Agent \mapsto 2^{\mathbb{P}}$ *is a control assignment function such that* $\{\pi(1), \ldots, \pi(n)\}$ *forms a partition of* $\mathbb{P}$*. For each agent i,* $2^{\pi(i)}$ *is the strategy space of i.*
4. $Goal = \{Goal_1, \ldots, Goal_n\}$ *is a set of weighted formulas of* $L_{\mathbb{P}}$*. That is, each* $Goal_i$ *is a finite set* $\{\langle \phi_1, m_1 \rangle, \ldots, \langle \phi_k, m_k \rangle\}$ *where* $\phi_j \in L_{\mathbb{P}}$ *and* m_j *is a real number representing the weight of* ϕ_j*.*

A strategy for agent i is a partial valuation for all the variables i controls. Note that since $\{\pi(1), \ldots, \pi(n)\}$ forms a partition of $\mathbb{P}$, a strategy profile S is a valuation for $\mathbb{P}$. In the rest of the paper we make use of the following notation, which is standard in game theory. Let $G = (Agent, \mathbb{P}, \pi, Goal)$ be a weighted boolean game with $Agent = \{1, \dots, n\}$, $S = (s_1, \dots, s_n)$ be a strategy profile, we use S_{-i} to denote the projection of S on $Agent - \{i\}$: $S_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ and S_i to denote the projection of S on i's strategy.

Agents' utilities in weighted boolean games are induced by their goals. For every agent i and every strategy profiles S, $u_i(S) = \Sigma\{m_j : \langle \phi_j, m_j \rangle \in Goal_i, S \models \phi_j\}$. Agent's preference over strategy profile is induced by his utility function naturally: $S \leq_i S'$ iff $u_i(S) \leq u_i(S')$. Dominating strategies and pure-strategy Nash equilibria are defined as usual in game theory [28].

2.3 Input/output logic

In input/output logic, a norm is an ordered pair of formulas $(\phi, \psi) \in L_{\mathbb{P}} \times L_{\mathbb{P}}$. There are two types of norms which are used in input/output logic, obligatory norms and permissive norms. Let $N = O \cup P$ be a set of obligatory and permissive norms. A pair $(\phi, \psi) \in O$, call it an obligatory norm, is read as "given ϕ, it is obligatory to be ψ". A pair $(\phi, \psi) \in P$, call it a permissive norm, is read as "given ϕ, it is permitted to be ψ".

Obligatory norms O can be viewed as a function from $2^{L_{\mathbb{P}}}$ to $2^{L_{\mathbb{P}}}$ such that for a set Φ of formulas, $O(\Phi) = \{\psi \in L_{\mathbb{P}} : (\phi, \psi) \in O \text{ for some } \phi \in \Phi\}$.

Definition 2 (Semantics of input/output logic [23]) *Given a finite set of obligatory norms O and a finite set of formulas Φ, $out(O, \Phi) = Cn(O(Cn(\Phi)))$, where Cn is the consequence relation of propositional logic, i.e $Cn(\Phi) = \{\phi \in L_{\mathbb{P}} : \Phi \models \phi\}$.*[1]

Intuitively, the procedure of the semantics is as following: We first have in hand a set of formulas Φ (call it the input) as a description of the current state. We then close it by logical consequence $Cn(\Phi)$. The set of norms, like a deductive machine, accepts this logically closed set and produces a set of formulas $O(Cn(\Phi))$. We finally get the output $Cn(O(Cn(\Phi)))$ by applying the logical closure again. $\psi \in out(O, \phi)$ is understood as "ψ is obligatory given facts Φ and norms O".

2.3.1 Permission in input/output logic

Philosophically, it is common to distinguish between two kinds of permission: negative permission and positive permission. Negative permission is straightforward to describe: something is negatively permitted according to certain norms iff it is not prohibited by those norms. That is, iff there is no obligation to the contrary. Positive permission is more elusive. Makinson and van der Torre [25] distinguish two types of positive permission: static and dynamic permission. For the sake of simplicity, in this paper when discussing positive permission we only mean static permission.

Definition 3 (negative permission [25]) *Given a finite set of norms $N = O \cup P$ and a finite set of formulas Φ, $NegPerm(N, \Phi) = \{\psi \in L_{\mathbb{P}} : \neg\psi \notin out(O, \Phi)\}$.*

Intuitively, ϕ is negatively permitted iff ϕ is not forbidden. Since a formula is forbidden iff its negation is obligatory, ϕ is not forbidden is equivalent to $\neg\phi$ is not obligatory. Permissive norms plays no role in negative permission.

[1] In Makinson and van der Torre [23], this logic is called simple-minded input/output logic. Different input/output logics are developed in Makinson and van der Torre [23] as well. For example reusable input/output logic validates transitivity. In that logic we can derive r from $\Phi = \{p\}$ and $O = \{(p, q), (q, r)\}$. A technical introduction of input/output logic can be found in Sun [35].

Definition 4 (positive permission [25]) *Given a finite set of formulas Φ, a finite set of norms $N = O \cup P$ where O is a set of obligatory norms and P is a set of permissive norms.*

- *If $P \neq \emptyset$, then $PosPerm(N,\Phi) = \{\psi \in L_{\mathbb{P}} : \psi \in out(O \cup \{(\phi',\psi')\},\Phi),\ for\ some\ (\phi',\psi') \in P\}$.*
- *If $P = \emptyset$, then $PosPerm(N,\Phi) = out(O,\Phi)$.*

Intuitively, permissive norms are treated like weak obligatory norms, the basic difference is that while the latter may be used jointly, the former may only be applied one by one. As an illustration of such difference, image a situation in which a man is permitted to date one of several girls, but not all of them.

2.4 Complexity theory

Complexity theory is the theory to investigate the time, memory, or other resources required for solving computational problems. In this subsection we briefly review those concepts and results from complexity theory which will be used in this paper. More comprehensive introduction of complexity theory can be found in Arora and Barak [3].

We assume the readers are familiar with notions like Turing machine and the complexity class P, NP and coNP. The boolean hierarchy is the hierarchy of boolean combinations (intersection, union and complementation) of NP classes. BH_1 is the same as NP. BH_2 is the class of languages which are the intersection of a language in NP and a language in coNP. Wagner [40] shows that the following 2-parity SAT problem is complete for BH_2:

Given two propositional formulas ϕ_1 and ϕ_2 such that if ϕ_2 is satisfiable then ϕ_1 is satisfiable, is it true that ϕ_1 is satisfiable while ϕ_2 is not?

Oracle Turing machine and two complexity classes related to oracle Turing machine will be used in this paper.

Definition 5 (oracle Turing machine [3]) *An oracle for a language L is a device that is capable of reporting whether any string w is a member of L. An oracle Truing machine M^L is a modified Turing machine that has the additional capability of querying an oracle. Whenever M^L writes a string on a special oracle tape it is informed whether that string is a member of L, in a single computation step.*

P^{NP} is the class of problems solvable by a deterministic polynomial time Turing machine with an NP oracle. NP^{NP} is the class of problems solvable by a non-deterministic polynomial time Turing machine with an NP oracle.

3 Normative status

Definition 6 (normative multi-agent system) *A normative multi-agent system is a triple (G,N,E) where*

- *$G = (Agent,\mathbb{P},\pi,Goal)$ is a weighted boolean game.*
- *$N = O \cup P \subseteq L_{\mathbb{P}} \times L_{\mathbb{P}}$ is a finite set of obligatory and permissive norms.*
- *$E \subseteq L_{\mathbb{P}}$ is a finite set of formulas representing the environment.*

In a normative multi-agent system, strategies are classified as moral, legal or illegal. Such classification is sensitive to not only the normative system but also the environment.

	$+q$	$-q$
$+p$	$(1,1)$	$(0,1)$
$-p$	$(0,1)$	$(0,0)$

Definition 7 (moral, legal and illegal strategy) *Given a normative multi-agent system (G,N,E), for each agent i, a strategy $(+p_1,\ldots,+p_m,-q_1,\ldots,-q_n)$ is moral if*

$$p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_n \in out(O,E).$$

The strategy is positively legal if

$$p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_n \in PosPerm(N,E).$$

The strategy is negatively legal if

$$p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_n \in NegPerm(N,E).$$

The strategy is illegal if

$$\neg(p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_n) \in out(O,E).$$

Moral, positively legal, negatively legal and illegal are the four normative positions of strategies. We assume the normative position degrades from moral to positively legal, then further to negatively illegal, and finally to illegal. The **normative status** of a strategy is the highest normative position it has.

Example 1 *Let (G,N,E) be a normative multi-agent system as following:*

- $G = (Agent, \mathbb{P}, \pi, Goal)$ *is a weighted boolean game with*
  - $Agent = \{1,2\}$,
  - $\mathbb{P} = \{p,q\}$,
  - $\pi(1) = \{p\}$, $\pi(2) = \{q\}$,
  - $Goal_1 = \{\langle p \wedge q, 1\rangle\}$, $Goal_2 = \{\langle p \vee q, 1\rangle\}$.
- $N = O \cup P$ *where* $O = \{(\top, p)\}, P = \{(\top, q)\}$.
- $E = \emptyset$.

Then $out(O,E) = Cn(\{p\})$, $Perm(O,E) = Cn(\{p,q\})$. Therefore normative status of $+p, +q, -q, -p$ is respectively moral, positively legal, negatively legal and illegal. $\dashv$

Theorem 1 *Given a normative multi-agent system (G,N,E) and a strategy $(+p_1,\ldots,+p_m,-q_1,\ldots,-q_n)$, deciding whether this strategy is negatively legal is NP complete.*

Proof: Concerning the NP hardness, we prove by reducing the satisfiability problem of propositional logic to our problem: Let $\phi \in L_{\mathbb{P}}$ be a formula. Let $O = \{(\neg\phi, \neg p)\}$, $E = \emptyset$. Then $p \in NegPerm(N,E)$ iff $\neg p \notin out(O,E) = Cn(O(Cn(E))) = Cn(O(Cn(\top)))$ iff $\not\models \neg\phi$ iff ϕ is satisfiable.

Now we prove the NP membership. We provide the following non-deterministic Turing machine to solve our problem. Let $O = \{(\phi_1,\psi_1),\ldots,(\phi_n,\psi_n)\}$, E be a finite set of formulas and $p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k$ be a formula.

1. Guess a sequence of valuation $V_1,\ldots,V_n,V'$ on the propositional letters appears in $E \cup \{\phi_1,\ldots,\phi_n\} \cup \{\psi_1,\ldots,\psi_n\} \cup \{p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k\}$.

2. Let $N' \subseteq N$ be the set of obligatory norms which contains all (ϕ_i, ψ_i) such that $V_i(E) = 1$ and $V_i(\phi_i) = 0$.
3. Let $\Psi = \{\psi : (\phi, \psi) \in N - N'\}$.
4. If $V'(\Psi) = 1$ and $V'(\neg(p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k)) = 0$. Then return "accept" on this branch. Otherwise return "reject" on this branch.

The main intuition of the proof is: N' collects all norms which *cannot* be triggered by E. [2] In some branches we much have that N' contains exactly those norms which are not triggered by E. In those lucky branches Ψ is the same as $N(Cn(E))$. If there is a valuation V' such that $V'(\Psi) = 1$ and $V'(\neg(p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k)) = 0$, then we know $\neg(p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k) \notin Cn(\Psi) = Cn(N(Cn(E)))$.

It can be verified that $\neg(p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k) \notin Cn(O(Cn(E)))$ iff the algorithm returns "accept" on some branches and the time complexity of the Turing machine is polynomial. $\dashv$

Corollary 1 *Given a normative multi-agent system (G,N,E) and a strategy $(+p_1, \ldots, +p_m, -q_1, \ldots, -q_n)$, deciding whether this strategy is moral/illegal is co-NP complete.*

Corollary 2 *Given a normative multi-agent system (G,N,E) and a strategy $(+p_1, \ldots, +p_m, -q_1, \ldots, -q_n)$, deciding whether the normative status of this strategy is moral is co-NP complete.*

Theorem 2 *Given a normative multi-agent system (G,N,E) and a strategy $(+p_1, \ldots, +p_m, -q_1, \ldots, -q_n)$, deciding whether this strategy is positively legal is coNP complete.*

Proof: The coNP hardness can be proved by a reduction from the tautology problem of propositional logic. Here we omit the details.

Concerning the coNP membership, let $N = O \cup P$, $P = \{(\phi_1, \psi_1), \ldots, (\phi_m, \psi_m)\}$. Note that $PosPerm(N, E) = out(O \cup \{(\phi_1, \psi_1)\}, E) \cup \ldots \cup out(O \cup \{(\phi_m, \psi_m)\}, E)$. The NP membership follows from the fact that the NP class is closed under union. $\dashv$

Theorem 3 *Given a normative multi-agent system (G,N,E) and a strategy $(+p_1, \ldots, +p_m, -q_1, \ldots, -q_n)$, deciding whether the normative status of this strategy is positively legal is BH_2 complete.*

Proof: The BH_2 hardness can be proved by a reduction from the 2-Parity SAT problem. Given two propositional formulas ϕ_1 and ϕ_2 such that if ϕ_2 is satisfiable then ϕ_1 is satisfiable. Our aim is to decide if ϕ_1 is satisfiable and ϕ_2 is not satisfiable.

Let $N = O \cup P$, $O = (\neg\phi_1, p)$, $P = (\neg\phi_2, p)$, $E = \emptyset$. Then the normative status of $+p$ is positively legal iff $p \notin out(O, \emptyset)$ and $p \in out(O \cup P, \emptyset)$ iff $p \notin Cn(O(Cn(\emptyset))$ and $p \notin Cn(O \cup P(Cn(\emptyset))$ iff $\neg\phi_1$ is not a tautology and $\neg\phi_2$ is a tautology, which is equivalent to ϕ_1 is satisfiable and ϕ_2 is not satisfiable.

The BH_2 membership is proved by showing that deciding whether the normative status of a strategy is positively legal is in fact a intersection of a NP problem (the strategy is not moral) and a coNP problem (the strategy is positively legal). Here we omit the details. $\dashv$

Theorem 4 *Given a normative multi-agent system (G,N,E) and a strategy $(+p_1, \ldots, +p_m, -q_1, \ldots, -q_n)$, deciding whether the normative status of this strategy is negatively legal is NP complete.*

Proof: The NP hardness is easy to prove. Here we focus on the NP membership. Let $N = O \cup P$, $P = \{(\phi_1, \psi_1), \ldots, (\phi_k, \psi_k)\}$.

The normative status of this strategy is negatively legal iff the following are true:

[2] We say a norm (ϕ, ψ) is triggered by E if $\phi \in Cn(E)$).

- $p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \neg q_n \notin out(O,E)$.
- $p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \neg q_n \notin out(O \cup \{(\phi_1, \psi_1)\}, E)$.
- ......
- $p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \neg q_n \notin out(O \cup \{(\phi_k, \psi_k)\}, E)$.
- $\neg(p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \neg q_n) \notin out(O,E)$.

The NP membership follows from the fact that the NP class is closed under intersection. $\dashv$

Theorem 5 *Given a normative multi-agent system (G,N,E) and a strategy $(+p_1,\ldots,+p_m,-q_1,\ldots,-q_n)$, deciding whether the normative status of this strategy is illegal is BH_2 complete.*

Proof: Similar to the proof of Theorem 3. $\dashv$

3.1 Normative boolean game

In a normative multi-agent system, agent's preference over strategy profiles is changed by the normative status of strategies. The basic ideas is:

1. an agent prefers strategy profiles with higher utility.
2. for two strategy profiles of the same utility, the agent prefers the one which contains his strategy of higher normative status.

Definition 8 (normative boolean game) *Given a normative multi-agent system (G,N,E) where $G = (Agent, \mathbb{P}, \pi, Goal)$, it induces a normative boolean game $G^N = (Agent, \mathbb{P}, \pi, \prec_1, \ldots \prec_n)$ where $\prec_i$ is the preference of i over strategy profiles such that $S \prec_i S'$ if either*

$$u_i(S) < u_i(S')$$

or

$$u_i(S) = u_i(S') \text{ and the normative status of } S'_i \text{ is higher than that of } S_i.$$

Theorem 6 *Given a normative multi-agent system (G,N,E), an agent i and two strategy profiles S and S', deciding whether $S \prec_i S'$ is in P^{NP}.*

Proof: (sketch) This problem can be solved by a polynomial time deterministic Turing machine with an NP oracle. We only need to call the oracle to test if S is moral, positively legal, negative legal or illegal. And the same test for S'. The utility of S and S' can be calculated in polynomial time. $\dashv$

Definition 9 (normative Nash equilibrium) *Given a normative multi-agent system (G,N,E), a strategy profile S is a normative Nash equilibrium if it is a Nash equilibrium in the normative boolean game G^N.*

Normative Nash equilibrium, as a solution concept of normative boolean games, is a refined notion of Nash equilibrium. Every normative Nash equilibrium is a Nash equilibrium, but not vice verse.

Example 2 *Let (G,N,E) be a normative system as following:*

- $G = (Agent, \mathbb{P}, \pi, Goal)$ *is a boolean game with*
  - $Agent = \{1,2\}$,
  - $\mathbb{P} = \{p,q\}$,

	$+q$	$-p$
$+p$	$(2,1)$	$(1,1)$
$-p$	$(0,0)$	$(0,0)$

- $\pi(1) = \{p\}$, $\pi(2) = \{q\}$,
- $Goal_1 = \{\langle p \wedge q, 2\rangle, \langle p \wedge \neg q, 1\rangle\}$ $Goal_2 = \{\langle p, 1\rangle\}$.

- *$N = O \cup P$ where $O = \{(\top, q)\}$, $P = \emptyset$.*
- *$E = \emptyset$.*

There are two Nash equilibria: $(\{+p\},\{+q\})$ and $(\{+p\},\{-q\})$. There is only one normative Nash equilibrium: $(\{+p\},\{+q\})$. From the perspective of social welfare, $(\{+p\},\{-q\})$ is not an optimal equilibrium because its social welfare is $1+1=2$, while the social welfare of $(\{+p\},\{+q\})$ is $2+1=3$. Therefore this example shows that by designing norms appropriately, non-optimal equilibrium is avoided.

Theorem 7 *Given a normative multi-agent system (G,N,E) and a strategy profile S, deciding whether S is normative Nash equilibrium is in $coNP^{NP}$.*

Proof: (sketch) The complement of this problem can be solved by a polynomial time non-deterministic Turing machine with an NP oracle: if S is not a normative Nash equilibrium, then we can guess a strategy $S' = (s'_i, S_{-i})$ such that $S \prec_i S'$ for some agent i. With the help of an NP oracle, testing if $S \prec_i S'$ can be done in polynomial time.

$\dashv$

4 Conclusion

In the present paper we augment weighted boolean game with norms. Using ideas from input/output logic, the normative status of strategies are discussed. The preference relation in boolean games are refined by the normative status of strategies. Normative boolean game and notions like normative Nash equilibrium are then introduced. After formally presenting the model, we use an example to show that non-optimal Nash equilibrium can be avoided by making use of norms. We study the complexity issues related to normative status and normative Nash equilibrium. Some of our complexity results are not complete, which leaves rooms for future work.

References

[1] Thomas Ågotnes, Wiebe van der Hoek, Juan A. Rodríguez-Aguilar, Carles Sierra & Michael Wooldridge (2007): *On the Logic of Normative Systems*. In Veloso [39], pp. 1175–1180. Available at `http://dli.iiit.ac.in/ijcai/IJCAI-2007/PDF/IJCAI07-190.pdf`.

[2] Giulia Andrighetto, Guido Governatori, Pablo Noriega & Leendert W. N. van der Torre, editors (2013): *Normative Multi-Agent Systems*. *Dagstuhl Follow-Ups* 4, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik. Available at `http://drops.dagstuhl.de/opus/portals/dfu/index.php?semnr=13003`.

[3] Sanjeev Arora & Boaz Barak (2009): *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, USA.

[4] Vittorio Bilò (2007): *On Satisfiability Games and the Power of Congestion Games*. In Kao & Li [22], pp. 231–240. Available at `http://dx.doi.org/10.1007/978-3-540-72870-2_22`.

[5] Guido Boella, Leendert van der Torre & Harko Verhagen (2008): *Introduction to the special issue on normative multiagent systems*. Autonomous Agents and Multi-Agent Systems 17(1), pp. 1–10. Available at http://dx.doi.org/10.1007/s10458-008-9047-8.

[6] Elise Bonzon, Marie-Christine Lagasquie-Schiex & Jérôme Lang (2009): *Dependencies between players in Boolean games*. Int. J. Approx. Reasoning 50(6), pp. 899–914. Available at http://dx.doi.org/10.1016/j.ijar.2009.02.008.

[7] Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang & Bruno Zanuttini (2006): *Boolean Games Revisited*. In Brewka et al. [9], pp. 265–269. Available at http://www.booksonline.iospress.nl/Content/View.aspx?piid=1688.

[8] Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang & Bruno Zanuttini (2009): *Compact preference representation and Boolean games*. Autonomous Agents and Multi-Agent Systems 18(1), pp. 1–35. Available at http://dx.doi.org/10.1007/s10458-008-9040-2.

[9] Gerhard Brewka, Silvia Coradeschi, Anna Perini & Paolo Traverso, editors (2006): *ECAI 2006, 17th European Conference on Artificial Intelligence, August 29 - September 1, 2006, Riva del Garda, Italy, Including Prestigious Applications of Intelligent Systems (PAIS 2006), Proceedings*. Frontiers in Artificial Intelligence and Applications 141, IOS Press.

[10] Nils Bulling, Leendert W. N. van der Torre, Serena Villata, Wojtek Jamroga & Wamberto Weber Vasconcelos, editors (2014): *Computational Logic in Multi-Agent Systems - 15th International Workshop, CLIMA XV, Prague, Czech Republic, August 18-19, 2014. Proceedings*. Lecture Notes in Computer Science 8624, Springer. Available at http://dx.doi.org/10.1007/978-3-319-09764-0.

[11] Samuel R. Buss & Louise Hay (1991): *On Truth-Table Reducibility to SAT*. Inf. Comput. 91(1), pp. 86–102. Available at http://dx.doi.org/10.1016/0890-5401(91)90075-D.

[12] Ronald Coase (1960): *The problem of social cost.* Journal of Law and Economics, 1.

[13] James Coleman (1998): *Foundations of Social Theory*. Belnap Press.

[14] Xiaotie Deng & Fan Chung Graham, editors (2007): *Internet and Network Economics, Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007, Proceedings*. Lecture Notes in Computer Science 4858, Springer.

[15] Paul E. Dunne, Wiebe van der Hoek, Sarit Kraus & Michael Wooldridge (2008): *Cooperative Boolean games*. In Padgham et al. [29], pp. 1015–1022. Available at http://doi.acm.org/10.1145/1402298.1402363.

[16] Dov Gabbay, John Horty, Xavier Parent, Ron van der Meyden & Leendert van der Torre, editors (2014): *Handbook of Deontic Logic and Normative Systems*. College Publications, London.

[17] John Grant, Sarit Kraus, Michael Wooldridge & Inon Zuckerman (2011): *Manipulating Boolean Games through Communication*. In Walsh [41], pp. 210–215. Available at http://ijcai.org/papers11/Papers/IJCAI11-046.pdf.

[18] Paul Harrenstein (2004): *Logic in conflict*. Ph.D. thesis, Utrecht University.

[19] Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer & Cees Witteveen (2001): *Boolean Games*. In: Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge, TARK '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 287–298. Available at http://dl.acm.org/citation.cfm?id=1028128.1028159.

[20] Leonid Hurwicz (1996): *Institutions as families of game forms.* Japanese Economic Review 47(2), pp. 113–132.

[21] Leonid Hurwicz (2008): *But who will guard the guardians?* American Economic Review 98(3), pp. 577–585.

[22] Ming-Yang Kao & Xiang-Yang Li, editors (2007): *Algorithmic Aspects in Information and Management, Third International Conference, AAIM 2007, Portland, OR, USA, June 6-8, 2007, Proceedings*. Lecture Notes in Computer Science 4508, Springer.

[23] David Makinson & Leendert van der Torre (2000): *Input-output logics*. Journal of Philosophical Logic 29, pp. 383–408.

[24] David Makinson & Leendert van der Torre (2001): *Constraints for input/output logics*. Journal of Philosophical Logic 30(2), pp. 155–185.

[25] David Makinson & Leendert van der Torre (2003): *Permission from an input/output perspective*. Journal of Philosophical Logic 32, pp. 391–416.

[26] David Makinson & Leendert van der Torre (2003): *What is input/output logic?* In B. Lowe, W. Malzkorn & T. Rasch, editors: Foundations of the Formal Sciences II: Applications of Mathematical Logic in Philosophy and Linguistics, pp. 163–174.

[27] Marios Mavronicolas, Burkhard Monien & Klaus W. Wagner (2007): *Weighted Boolean Formula Games*. In Deng & Graham [14], pp. 469–481. Available at http://dx.doi.org/10.1007/978-3-540-77105-0_51.

[28] Martin Osborne & Ariel Rubinstein (1994): *A Course in Game Theory*. The MIT Press, Cambridge, Mass.

[29] Lin Padgham, David C. Parkes, Jörg P. Müller & Simon Parsons, editors (2008): *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 2*. IFAAMAS.

[30] Xavier Parent (2011): *Moral particularism in the light of deontic logic*. Artif. Intell. Law 19(2-3), pp. 75–98, doi:10.1007/s10506-011-9108-1. Available at http://dx.doi.org/10.1007/s10506-011-9108-1.

[31] Luigi Sauro & Serena Villata (2013): *Dependency in Cooperative Boolean Games*. J. Log. Comput. 23(2), pp. 425–444. Available at http://dx.doi.org/10.1093/logcom/exr030.

[32] Yoav Shoham & Kevin Leyton-Brown (2009): *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

[33] Yoav Shoham & Moshe Tennenholtz (1992): *On the Synthesis of Useful Social Laws for Artificial Agent Societies (Preliminary Report)*. In Swartout [37], pp. 276–281. Available at http://www.aaai.org/Library/AAAI/1992/aaai92-043.php.

[34] Yoav Shoham & Moshe Tennenholtz (1996): *On Social Laws for Artificial Agent Societies: Off-Line Design*. Artif. Intell. 73(1-2), pp. 231–252. Available at http://dx.doi.org/10.1016/0004-3702(94)00007-N.

[35] Xin Sun (2014): *How to Build Input/Output Logic*. In Bulling et al. [10], pp. 123–137. Available at http://dx.doi.org/10.1007/978-3-319-09764-0_8.

[36] Xin Sun (2015): *Boolean game with prioritized norms*. Accepted by The Fifth International Conference on Logic, Rationality and Interaction (LORI2015).

[37] William R. Swartout, editor (1992): *Proceedings of the 10th National Conference on Artificial Intelligence. San Jose, CA, July 12-16, 1992*. AAAI Press / The MIT Press. Available at http://www.aaai.org/Conferences/AAAI/aaai92.php.

[38] Edna Ulmann-Margalit (1977): *The Emergence of Norms.* Clarendon Press,, Oxford.

[39] Manuela M. Veloso, editor (2007): *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007.*

[40] Klaus Wagner (1986): *More complicated questions about maxima and minima, and some closures of NP*. In Laurent Kott, editor: Automata, Languages and Programming, Lecture Notes in Computer Science 226, Springer Berlin Heidelberg, pp. 434–443. Available at http://dx.doi.org/10.1007/3-540-16761-7_93.

[41] Toby Walsh, editor (2011): *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. IJCAI/AAAI.

[42] Gerhard Weiss, editor (2013): *Multiagent systems*, 2 edition. MIT press.

[43] Michael Wooldridge (2009): *An Introduction to MultiAgent Systems (2. ed.).* Wiley.

[44] Georg von Wright (1952): *Deontic logic*. *Mind* 60, pp. 1–15.