Sasha Rubin

Research plan: Formal Methods in Artificial Intelligence *August 2017*

This document describes my research plan for 2018 – 2020 whose objective is to develop **Formal Methods** to describe and reason about **Artificial Agents**.

Overview

Artificial Intelligence (AI) studies the principles behind thinking and reasoning, and it does so in a mathematical and algorithmic way. Systems built on the insights of AI are called *artificial agents (AA)*. As artificial agents are increasingly deployed in the world (e.g., software agents on the Internet, driverless cars, software that play and compete with humans at games, robots exploring new and dangerous environments, etc.) a clear challenge has materalised: there is a need for humans to be able to *trust* the decisions made by AA, the need for *meaningful interactions* between humans and AA, and the need for *transparent* AA [1]. One path to meeting these needs is to create *explainable AI (XAI)*, i.e., to enable humans to understand, trust and manage AA [21].

This is a grand challenge that involves many facets of computer science: psychology (to understand what is a good explanation), knowledge representation and logic (to formalise this in a way accessible to humans and machines), algorithms (to produce good explanations), NLP (to interface with human users), software engineering (to produce reliable and efficient programs), etc.

I take as an hypothesis that this challenge cannot be met without having some formal guarantees on the behaviour of AA. It is in this context that Formal Methods (FM) will play a role.

> FM is built on three pillars:
> "Modeling" problems at the correct levels of abstraction to be able to reason about them, "Verification" for deducing that systems do indeed have specified properties, "Synthesis" for automatically producing correct-by-design systems.

All three pillars are founded on techniques from mathematical logic. Logic is used to formally represent aspects of the system so that both expert humans and computers can manipulate them and reason about them.

Although FM were pioneered in the areas of hardware and software verification, there are many connections between FM and AI.

Connections between planning (in AI) and synthesis (in FM)

Planning is a branch of AI that addresses the problem of generating a course of action to achieve a desired goal, given a description of the domain of interest and its initial state. The area is central to the development of artificial agents. Besides theoretical insights, Planning provides practical tools based on heuristic search and symbolic methods [**?**].

Synthesis is a cornerstone of FM that addresses the problem of automatically producing systems that meet a given specification.

There are many similarities between these two areas, but also important differences.

Similarities

Both Planning and Synthesis are model-based controller design. That is, ...

Differences

goals are dynamic, that is, goals are produced continuously, as the agent operates. However, while in Planning a plan for the next goal must be built only after the current goal is achieved, mechanisms have the option to drop the current goal or combine it with the next one, thus adding a complication that cannot be

dealt with straightforwardly by standard Planning techniques.

planning in AI and *reactive synthesis* in formal methods

1. same basic idea... model-based controller design
2. central to both are succinct representations of the systems: STRIPS, PDDL, in planning and LTL/LTLf in synthesis. but for different reasons! planning typically deals with reachability goals on compact domains; synthesis with LTL goals on explicit domains.
3. both understood early on the computational complexity of the problem PSPACE-complete ... 2EXPTIME-complete
4. planning responded by finding ways to treat the "easy but large cases", e.g., heuristic search
5. reactive synthesis responded by studying the theory of the problem (e.g., fragments, extensions), and providing algorithms based on logic and automata theory (antichain).
6. some work (Sheilah...) has studied translations between these, i.e., LTL –> AFW/NFW –> planning domain

Formal methods (FM) is an umbrella-term that describes principles and techniques for reasoning about systems with some digital component, such as software, hardware, cyber-physical systems, etc.

> FM is built on three pillars: modeling, verification, and synthesis.

All three pillars are founded on techniques from mathematical logic. Logic is used to formally represent aspects of the system so that both expert humans and computers can manipulate them and reason about them.

Current Research — Formal methods for multi-agent systems

Multi-agent systems (MAS) involve multiple individual agents (these may be people, software, robots) each with their own goals. Such systems can be viewed as multi-player games, and thus notions from game-theory (e.g., strategies, knowledge, and equilibria) are used to reason about them. Agents in realistic MAS often lack information about other agents and the environment, and this is often categorised in one of two ways: a) *incomplete information* and b) *imperfect information*.

a) MAS with incomplete information

Incomplete-information refers to uncertainty about the environment (i.e., the structure of the game). I have considered two sources of incomplete information for MAS.

First, the *number of agents* may not be known, or may not be bounded a priori. In a series of papers, I have contributed to a generalisation of a cornerstone paper on verification of such systems ("Reasoning about Rings", E.A. Emerson, K.S. Namjoshi, POPL, 1995) from ring topologies to arbitrary topologies [4, 5, 11, 6]. Other work on this topic studied the relative power of standard communication-primitives assuming an unknown number of agents [3], as well as the complexity of model-checking timed systems assuming an unknown number of agents [2]. I also contributed to a book on this topic published by Morgan&Claypool in 2015 [16, 17].

Second, the agents may be operating in a *partially-known environment*. For instance, the agents may know they are in a ring, but may not know the size of the ring. I launched the application of automata theory for the verification of high-level properties of light-weight mobile agents in partially-known environments [32]. In follow-up work I explored this theme further, including finding ways to model agents on grids — the most common abstraction of 2D and 3D space [7, 8, 29].

b) MAS with imperfect information

Even if agents have certainty about the structure of the system, they may not know exactly which state the system is in. This is called imperfect information and the associated logic for reasoning about such cases are called *epistemic*. I have studied strategic-epistemic logics in a number of works, namely, with a prompt modality (thus allowing one to express that a property holds "promptly" rather than simply "eventually") [9],

and on systems with public-actions (such as certain card games, including a hand of Poker or a round of Bridge) [14, 13]. The importance of these last works is that they give the first decidability (and sometimes optimal complexity) results for strategic reasoning about games of imperfect information in which the agents may have arbitrary observations. In contrast, following classical restrictions on the observations or information of agents, I have also shown how to extend strategy logic by epistemic operators and identified a decidable fragment in which one can express equilibria concepts [15].

Foundations of Automated Planning

Planning in AI can be viewed as the problem of finding strategies in one- or two-player graph-games. In this model vertices represent states, edges represent transitions, and the players represent the agents. I have contributed foundational work to such games. Concretely, I recently extended the classic belief-space construction for games of imperfect-information from finite arenas to infinite-arenas [20] (infinite arenas often arise in the study of MAS with incomplete information, see above). I have also used these ideas to elucidate the role of observation-projections in generalised planning problems [19, 18]. I have generalised classic results about certain games with quantitative objectives (i.e., Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. International Journal of Game Theory, 8:109–113, 1979) to so-called first-cycle games, i.e., games in which play stops the moment a vertex is repeated [10].

Past Research — Algorithmic Model Theory

My early work contributed to a research program called “Algorithmic Model Theory" whose aim is to develop and extend the success of Finite Model Theory to infinite structures that can be reasoned about algorithmically.

Specifically, my PhD work pioneered the development of “automatic structures”: this is a generalisation of the regular languages from sets to mathematical objects with structure, such as graphs, arithmetics, algebras, etc. The fundamental property of automatic structures is that one can automatically answer logic-based queries about them (precisely, their first-order theory is decidable). I gave techniques for proving that structures are or are not automatic (similar to, but vastly more complicated than, pumping lemmas for regular languages), I studied the computational complexity of deciding when two automatic structures are the same (isomorphic), and I found extensions of the fundamental property, thus enriching the query language [12, 22, 23, 24, 25, 26, 27, 31]. I have also worked on extensions of automatic structures to include oracle computation [28, 30].

There are many connections between AI and FM. Som

Planning in AI and Synthesis in FM

planning in AI and *reactive synthesis* in formal methods

1. same basic idea... model-based controller design
2. central to both are succinct representations of the systems: STRIPS, PDDL, in planning and LTL/LTLf in synthesis. but for different reasons! planning typically deals with reachability goals on compact domains; synthesis with LTL goals on explicit domains.
3. both understood early on the computational complexity of the problem PSPACE-complete ... 2EXPTIME-complete
4. planning responded by finding ways to treat the "easy but large cases", e.g., heuristic search
5. reactive synthesis responded by studying the theory of the problem (e.g., fragments, extensions), and providing algorithms based on logic and automata theory (antichain).
6. some work (Sheilah...) has studied translations between these, i.e., LTL –> AFW/NFW –> planning domain

What's missing from within this picture?

1. formal connections between the two fields: e.g., reducing synthesis to planning (cf Sheilah's work).
2. formal connections within planning problems: e.g., reduce LTLf planning to reachability planning...
3. clear idea of how one can exploit modern planners and heuristic methods to solve problems in automata

in practice, e.g., do domain-independent heuristics work LTLf/LDLf/LTL...? perhaps "LTL-dependent heuristics" should be studied...

4. clear connections between DEL/epistemic programs/GDLIII (Thieschler) and synthesis framework/algorithms.

Relation with other work - In fact a first-person view of agents has long be advocated in reasoning about action in Knowledge Representation [McHa69, Reit01]. - many architectures used in robotics to capture mental states are not formal enough to admit to formal methods [**?**], although some recent work at UNSW aims to address this. - to deal with the social component I plan to use notions of "contexts" that defined expected behaviours to other agents that is used to reason about them.

- situation awareness, incomplete information, information classification and actions ontologies, reasoning about othersâĂŹ expected behaviors and violations, strategic action deliberation, and synthesis and refinement of execution plans.

Potential future directions - E. Asarin, R. Chane-Yack-Fa, and D. Varacca, âĂIJFair adversaries and randomization in two-player games,âĂİ in Proceedings of FoSSaCS 2010, ser. LNCS, vol. 6014. Springer, 2010, pp. 64âĂŞ78.

- How Good Is a Strategy in a Game With Nature? - Paying cost for resolving observation (Natasha,Blaise).

How can I complement research at UNSW. 1. Theoretical foundations of strategic epistemic reasoning in complex environments. MIT: Strategic Reasoning and Planning for General Game-Playing Robots (Australia-Germany Joint Research Cooperation Scheme 2016-2017) MIT: Universal Game-Playing Systems for Randomised and Imperfect-Information Games (ARC-DP 2012-2015)

2. Theory of distributed synthesis (information forks, automata theory for controller synthesis) could be used to analyse the cognitive meta-hierarchy of David Rajaratnam Bernhard Hengst Maurice Pagnucco Claude Sammut Michael Thielscher

How my work can be complemented by work at UNSW: 1. *unstructured* and incomplete environments are central to many robotic applications (e.g., rescue robots). Adapting definitions and results in reactive synthesis to such a setting is a clear and present challenge.

2. Insights from information flow in security of distributed system (CC Morgan) could yield insights into how strategies of different agents in the distributed synthesis problem signal private information to other agents. Note that the latter is in some sense the dual problem: how can one define strategies that *do* leak enough private information that the distributed players can co-ordinate and achieve a joint objective.

3. Toby Walsh, Haris Aziz?

Project Offers ==========

??? When faced with a dynamical sys- tem that you want to simulate, control, analyze, or otherwise investigate, first axiomatize it in a suitable logic. Through logical entailment, all else will follow, including system control, simulation, and analysis.

- games of incomplete information, imperfect information - epistemic planning

concretely: probabilistic DEL with public announcements PATL* on broadcast iCGS

Quantitative SL: add weights to the arena (e.g., to actions or to states), and add atomic formulas to the logic of the form "the mean-payoff for player i is at least c".

Question: is model-checking decidable if we add these to ATL? ATL*? SL?

– controller manages a collection of programmable mechanisms - monitors and responds to events (e.g., shifts in load, certain specifications fail, ...)

- reprogram mechanisms on the fly (e.g., change ???

controller is centralised (1 agent vs 1 environment)

FSMs (1) intuitively and concisely capture control dynamics in response to network events; and (2) their structure makes them amenable to verification.

what are the external events? timing,

what is the current way to solve the problems that whitemech would solve.

References

[1] ACM U.S. Public Policy Council and ACM Europe Policy Committee. Statement on algorithmic transparency and accountability. ACM, 2017.

[2] B. Aminof, S. Rubin, Francesco Spegni, and F. Zuleger. Liveness of parameterized timed

networks. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, pages 375–387, 2015.

[3] B. Aminof, S. Rubin, and F. Zuleger. On the expressive power of communication primitives in parameterised systems. In *Logic for Programming, Artificial Intelligence, and Reasoning - 20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings*, pages 313–328, 2015.

[4] Benjamin Aminof, Swen Jacobs, Ayrat Khalimov, and Sasha Rubin. Parameterized model checking of token-passing systems. In *Verification, Model Checking, and Abstract Interpretation - 15th International Conference, VMCAI 2014, San Diego, CA, USA, January 19-21, 2014, Proceedings*, pages 262–281, 2014.

[5] Benjamin Aminof, Tomer Kotek, Sasha Rubin, Francesco Spegni, and Helmut Veith. Parameterized model checking of rendezvous systems. In *CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings*, pages 109–124, 2014.

[6] Benjamin Aminof, Tomer Kotek, Sasha Rubin, Francesco Spegni, and Helmut Veith. Parameterized model checking of rendezvous systems. *Distributed Computing*, pages 1–36, 2017.

[7] Benjamin Aminof, Aniello Murano, Sasha Rubin, and Florian Zuleger. Verification of asynchronous mobile-robots in partially-known environments. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference, Bertinoro, Italy, October 26-30, 2015, Proceedings*, pages 185–200, 2015.

[8] Benjamin Aminof, Aniello Murano, Sasha Rubin, and Florian Zuleger. Automatic verification of multi-agent systems in parameterised grid-environments. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS 2016)*, pages 1190–1199, 2016.

[9] Benjamin Aminof, Aniello Murano, Sasha Rubin, and Florian Zuleger. Prompt alternating-time epistemic logics. In *Principles of Knowledge Representation and Reasoning: Proc. of the 15th International Conference, (KR 2016)*, pages 258–267, 2016.

[10] Benjamin Aminof and Sasha Rubin. First cycle games. *Information and Computation*, 2016.

[11] Benjamin Aminof and Sasha Rubin. Model checking parameterised multi-token systems via the composition method. In *Proc. 8th International Joint Conference on Automated Reasoning, (IJCAR 2016)*, pages 499–515, 2016.

[12] Vince Bárány, Erich Grädel, and Sasha Rubin. Automata-based presentations of infinite structures. In Javier Esparza, Christian Michaux, and Charles Steinhorn, editors, *Finite and Algorithmic Model Theory*, pages 1–76. Cambridge University Press, 2011. Cambridge Books Online.

[13] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. Verification of broadcasting multi-agent systems against an epistemic strategy logic. In *International Joint Conference on Artificial Intelligence (IJCAI 2017)*, 2017.

[14] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. Verification of multi-agent systems with imperfect information and public actions. In *Proceedings of the 2017 International Conference on Autonomous Agents & Multiagent Systems (AAMAS 2017)*, 2017.

[15] Raphael Berthon, Bastien Maubert, Aniello Murano, Sasha Rubin, and Moshe Vardi. Hierarchical strategic reasoning. In *IEEE Symposium on Logic in Computer Science (LICS 2017)*, 2017.

[16] Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder. *Decidability of Parameterized Verification*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2015.

[17] Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder. Decidability in parameterized verification. *SIGACT News*, 47(2):53–64, 2016.

[18] Blai Bonet, Giuseppe De Giacomo, Hector Geffner, and Sasha Rubin. Generalized planning: Non-deterministic abstractions and trajectory constraints. In *IJCAI*, 2017.

[19] Blai Bonet, Giuseppe De Giacomo, Hector Geffner, and Sasha Rubin. Generalized planning: Non-deterministic abstractions and trajectory constraints. In *ICAPS 2017 Workshop on Generalized Planning*, 2017.

[20] Giuseppe De Giacomo, Antonio Di Stasio, Aniello Murano, and Sasha Rubin. Imperfect information games and generalized planning. In *International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 2016.

[21] David Gunning. Explainable artificial intelligence (xai). Broad Agency Announcement. DARPA-BAA-16-53.

[22] Hajime Ishihara, Bakhadyr Khoussainov, and Sasha Rubin. Some results on automatic structures. In *LICS 2002, 17th IEEE Symposium on Logic in Computer Science, 22-25 July 2002, Copenhagen, Denmark, Proceedings*, page 235, 2002.

[23] Bakhadyr Khoussainov, André Nies, Sasha Rubin, and Frank Stephan. Automatic structures: Richness and limitations. In *LICS 2004, 19th IEEE Symposium on Logic in Computer Science, 14-17 July 2004, Turku, Finland, Proceedings*, pages 44–53, 2004.

[24] Bakhadyr Khoussainov, André Nies, Sasha Rubin, and Frank Stephan. Automatic structures: Richness and limitations. *Logical Methods in Computer Science*, 3(2), 2007.

[25] Bakhadyr Khoussainov, Sasha Rubin, and Frank Stephan. On automatic partial orders. In *LICS 2003, 18th IEEE Symposium on Logic in Computer Science, 22-25 June 2003, Ottawa, Canada, Proceedings*, pages 168–177, 2003.

[26] Bakhadyr Khoussainov, Sasha Rubin, and Frank Stephan. Definability and regularity in automatic structures. In *STACS 2004, 21st Annual Symposium on Theoretical Aspects of Computer Science, Montpellier, France, March 25-27, 2004, Proceedings*, pages 440–451, 2004.

[27] Bakhadyr Khoussainov, Sasha Rubin, and Frank Stephan. Automatic linear orders and trees. *ACM Transactions on Computational Logic*, 6(4):675–700, 2005.

[28] Alex Kruckman, Sasha Rubin, John Sheridan, and Ben Zax. A myhill-nerode theorem for automata with advice. In *Proceedings Third International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2012, Napoli, Italy, September 6-8, 2012.*, pages 238–246, 2012.

[29] A. Murano, Giuseppe Perelli, and S. Rubin. Multi-agent path planning in known dynamic environments. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference, Italy, 2015, Proceedings*, volume 9387 of *LNCS*, pages 218–231. Springer, 2015.

[30] Alexander Rabinovich and Sasha Rubin. Interpretations in trees with countably many branches. In *LICS 2012, Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, Dubrovnik, Croatia, June 25-28, 2012*, pages 551–560, 2012.

[31] Sasha Rubin. Automata presenting structures: A survey of the finite string case. *Bulletin of Symbolic Logic*, 14(2):169–209, 2008.

[32] Sasha Rubin. Parameterised verification of autonomous mobile-agents in static but unknown environments. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems, (AAMAS 2015)*, pages 199–208, 2015.