# How to formally reason about strategic behaviour

Sasha Rubin

University of Naples “Federico II”, Italy

February 7, 2019, University of Sydney

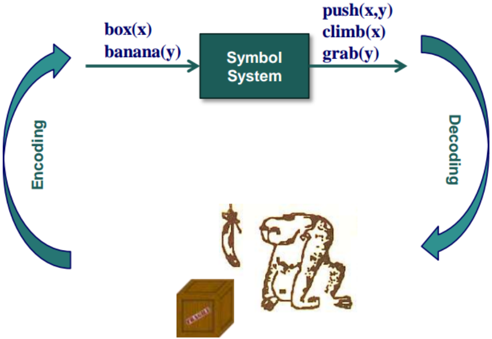
AI is about building agents to do tasks we think require intelligence

# What is an agent?

- An agent is something that *acts*, based on sensing the world.
- An agent is *rational* if it acts to achieve the best (expected) outcome. Russell, Norvig

# What is an agent?

- An agent is something that *acts*, based on sensing the world.
- An agent is *rational* if it acts to achieve the best (expected) outcome. Russell, Norvig



# …Multi-agent systems are everywhere!

Many real and imagined systems consist of multiple agents:

1. **distributed**, i.e., agents have their own view of the world.
2. **goal-directed**, i.e., agents strategise to achieve their own goals.

E.g.

- multiplayer games (winning tasks)
- robot assisted search-and-rescue (co-ordination tasks)
- autonomous transport vehicles (safety tasks)
- voting and auction protocols (fairness tasks)
- rational distributed computing (secret-sharing tasks)
- …

# Research Challenge

How to understand, predict and control agent behaviour?

# Lots of disciplines involved

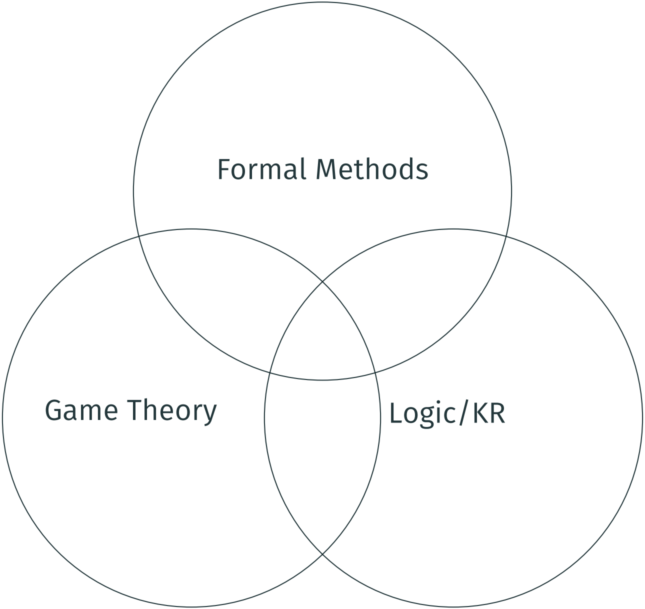
1. Humanities: How do we want agents to behave?
2. Computer engineering: How to engineer such agents?
3. Computer science: How to make sure agents behave as we want?
    3.1 Theoretical foundations
    3.2 Academic tools
    3.3 Industrial applications

# Research Challenge

How to understand, predict and control agent behaviour?

- **Methodology**
  Model agent tasks in logic (for understanding), and devise algorithms (for prediction/control).
- **Difficulty**
  Agents are distributed and goal directed.

# Methodology



# Formal Methods 101

$$\textbf{Model-checking paradigm}^{1} = \begin{cases} \text{model} & M \\ \text{specify} & \varphi \\ \text{algorithm for} & M \models \varphi \end{cases}$$

[1] If formula $\varphi$ talks about strategies, model-checking can usually be adapted to do control (not just prediction).

# Formal Methods 101

$$\textbf{Model-checking paradigm}^{1} = \begin{cases} \text{model} & M \\ \text{specify} & \varphi \\ \text{algorithm for} & M \models \varphi \end{cases}$$

Classical model-checking can't handle the complexities of multi-agent systems.

[1] If formula $\varphi$ talks about strategies, model-checking can usually be adapted to do control (not just prediction).

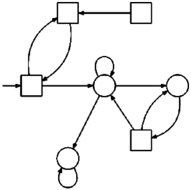
# My current research goals

$$\textbf{Model-checking paradigm} = \begin{cases} \text{model} & M \\ \text{specify} & \varphi \\ \text{algorithm for} & M \models \varphi \end{cases}$$

1. Devise models that capture essential aspects of multi-agent systems.
2. Devise logics that can specify complex agent/system goals.
3. Devise algorithms for model-checking, or prove that none exists.

Note. Steps 1 and 2 are useful in themselves! (Knowledge Representation philosophy)

# Models

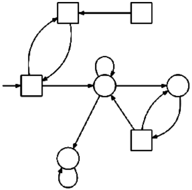


$M = \langle S, \mathrm{tr}, (Obs_a)_{a \in Ag} \rangle$ is a transition system modeling dynamics

| | |
|---|---|
| $S$ | set of states |
| $\mathrm{tr} : S \times Act \to S$ | transition function |
| $Obs_a : S \to \Omega$ | observation function |

Special case: $Obs_a(s) = s$ Full observation

# Models



$M = \langle S, \mathrm{tr}, (Obs_a)_{a \in Ag} \rangle$ is a transition system modeling dynamics

| | |
|---|---|
| $S$ | set of states |
| $\mathrm{tr} : S \times Act \to S$ | transition function |
| $Obs_a : S \to \Omega$ | observation function |

Special case: $Obs_a(s) = s$ Full observation

- Strategy tells an agent how to act based on its observations
- Strategy for each agent determines a path in $M$

# Logics

New logic: Strategy Logic for agents with partial observation

$$\overbrace{p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2}^{\text{Boolean}} \mid \overbrace{\mathrm{X}\,\varphi \mid \varphi_1 \,\mathrm{U}\, \varphi_2}^{\text{Temporal}} \mid \overbrace{(\exists s)\,\varphi \mid (a_i \mapsto s)\,\varphi}^{\text{Strategic}}$$

# Logics

New logic: Strategy Logic for agents with partial observation

$$\overbrace{p \mid \mathit{not}\,\varphi \mid \varphi_1\,\mathit{or}\,\varphi_2}^{\text{Boolean}} \mid \overbrace{\mathit{next}\,\varphi \mid \varphi_1\,\mathit{until}\,\varphi_2}^{\text{Temporal}} \mid \overbrace{(\mathit{exists}\,s)\,\varphi \mid (a\ \mathit{uses}\ s)\,\varphi}^{\text{Strategic}}$$

# Logics

$$\overbrace{p \mid \mathit{not}\,\varphi \mid \varphi_1\,\mathit{or}\,\varphi_2}^{\text{Boolean}} \mid \overbrace{\mathit{next}\,\varphi \mid \varphi_1\,\mathit{until}\,\varphi_2}^{\text{Temporal}} \mid \overbrace{(\mathit{exists}\,s)\,\varphi \mid (a\ \mathit{uses}\ s)\,\varphi}^{\text{Strategic}}$$

Boolean part expresses state properties:

- $\mathit{Crashed}(a_1)$, $\mathit{not}\,\mathit{Crashed}(a_1)$, $\mathit{InRoom}(a_1, r_2)$, $\ldots$

# Logics

$$\underbrace{p \mid \mathit{not}\,\varphi \mid \varphi_1\,\mathit{or}\,\varphi_2}^{\text{Boolean}} \mid \overbrace{\mathit{next}\,\varphi \mid \varphi_1\,\mathit{until}\,\varphi_2}^{\text{Temporal}} \mid \overbrace{(\mathit{exists}\,s)\,\varphi \mid (a\ \mathit{uses}\ s)\,\varphi}^{\text{Strategic}}$$

Temporal part expresses agent goals:

- *eventually* $\mathit{InRoom}(a_1, r_2)$, *never* $\mathit{Crashed}(a_1)$, $\ldots$

# Logics

$$\overbrace{p \mid \mathit{not}\,\varphi \mid \varphi_1\,\mathit{or}\,\varphi_2}^{\text{Boolean}} \mid \overbrace{\mathit{next}\,\varphi \mid \varphi_1\,\mathit{until}\,\varphi_2}^{\text{Temporal}} \mid \overbrace{(\mathit{exists}\,s)\,\varphi \mid (a\ \mathit{uses}\ s)\,\varphi}^{\text{Strategic}}$$

Strategic part expresses game-theoretic properties:

- $(\mathit{exists}\,s_1)\,(a_1\,\mathit{uses}\,s_1)\,\mathit{goal}_1$

  "I have a strategy to win" (solitaire)

- $(\mathit{exists}\,s_i)_i\,(a_i\,\mathit{uses}\,s_i)_i\,\mathit{goal}$

  "We have strategies that win" (coordination)

- $(\mathit{exists}\,s_1)\,(\mathit{forall}\,s_2)(a_i\,\mathit{uses}\,s_i)_i\,\mathit{goal}_1$

  "I have a strategy that beats any of yours" (tic-tac-toe)

- $(\exists s_i)_i\,(a_i \mapsto s_i)_i\,\&_i \neg[\neg \mathit{goal}_i \wedge (\exists s')(a_i \mapsto s')\mathit{goal}_i]$

  "We can play NE" (rational secret sharing, fair division, poker)

# Logics

$$\overbrace{p \mid \mathit{not}\,\varphi \mid \varphi_1\,\mathit{or}\,\varphi_2}^{\text{Boolean}} \mid \overbrace{\mathit{next}\,\varphi \mid \varphi_1\,\mathit{until}\,\varphi_2}^{\text{Temporal}} \mid \overbrace{(\mathit{exists}\,s)\,\varphi \mid (a\ \mathit{uses}\ s)\,\varphi}^{\text{Strategic}}$$

Natural, and very expressive

It can express

- state properties,
- (Boolean) agent goals,
- game-theoretic properties (NE, SPE, ESS, Pareto optimality, ...).

# Algorithms for $M \models \varphi$

# Distributed agents make things hard...

# Distributed agents make things hard...

- Multi-player games in TCS (70s) undecidable
- Distributed synthesis in FM (90s) undecidable
- Finite horizon DEC-POMDPs in AI (00s) NEXP-complete

Insight. The source of these difficulties is the ability of agents, each with their own observations of the world, to communicate privately with each other.

# Distributed agents make things hard...

What can be done?

1. Restrict observability
2. Restrict private communication

# Distributed agents make things hard...

What can be done? Restrict observability

How? Hierarchical observation/information

# Distributed agents make things hard...

What can be done? Restrict observability

How? Hierarchical observation/information

- Peterson, Reif ('79)
- Pnueli, Rosner ('90)
- Kupferman, Vardi ('01)
- Finkbeiner, Schewe ('05)
- vdMeyden, Wilke ('05)
- Berwanger, Mathew, vdBogaard ('16)
- Berthon, Maubert, Murano, R., Vardi (LICS '17)

Observation. Mathematically elegant, but none of the examples mentioned have hierarchical observation

# Distributed agents make things hard...

What can be done? Restrict private communication

How? Broadcast-communication/public-actions

# Distributed agents make things hard...

What can be done? Restrict private communication
How? Broadcast-communication/public-actions

- Lomuscio, vdMeyden, Ryan, Wilke ('00, '05) Broadcasting
- vDitmarsch, vdHoek ... Public Announcements Logic (DEL)
- Kominis, Geffner ('15,'17) Public Actions (Planning)
- Belardinelli, Lomuscio, Murano, R. (AAMAS '17, IJCAI '17)

# Models

Model: Public action transition system $M$ — the last action of each agent is observable to all agents.

E.g.,

- community-card games (bridge, poker)
- broadcasting distributed systems (secret-sharing protocols)
- epistemic puzzles (muddy children)
- auctions (open-outcry)

Many examples

# Algorithms for $M \models \varphi$

Theorem. $M \models \varphi$ is decidable if $M$ is a public-action transition system.

Complexity?

- In $(k+2)$-exptime in the model and the formula, and $(k-1)$-expspace-hard in the formula, where $k =$ quantifier-block depth of formula.

In fact. Computational complexity is similar to fully-observable transition systems!

Middle ground.

Fully observable TS $<$ Public Action TS $<$ Unrestricted TS

# Proofs

- Q. How did you come up with the algorithms?
- A. We used the automata-theoretic approach to model-checking.[2]
- Q. But what do the algorithms actually look like?
- A. Operations on automata that run on trees.
- Q. ??
- A. Think of it like manipulating **regular expressions** that match trees instead of strings!

[2] for which Vardi and Wolper won the Gödel prize for outstanding papers in TCS.

# Summary

- Current research goal

  Devise models, logics and algorithms for understanding, predicting, and controlling behaviour of agents in multi-agent systems.
- Challenge

  Combination of partial observability and private communication.
- Solutions

  Impose hierarchical observation, or restrict private communication, and use automata.

# Future research goals

Capture quantitative aspects

# Future research goals

## Capture quantitative aspects

1. Agents: Probabilistic (instead of just deterministic)
2. Tasks: Optimisation (instead of just Boolean)

# Future research goals

Context. Many protocols rely on coin-flips.

Project. Develop *Probabilistic Strategy Logic* for reasoning about stochastic multi-agent transition systems.

Methodology. Study restrictions on memory of probabilistic agents and reduce to reasoning about polynomial inequalities.[3]

[3] Formally, reduce $M \models \varphi$ to the first-order theory of real arithmetic, which is solvable in EXPSPACE by work of Tarski/Seidenberg.

# Future research goals (speculative)

Context. MDPs, POMDPs, DEC-POMDPs are important models.

Challenge. Agent goal is to optimise expected-reward, which is a very “low-level” specification.

Project. Unify reward- and declarative-specifications.

Methodology. ???

# Overview of recent research lines

Formal methods …

…for strategising agents with qualitative temporal goals

AAMAS'16/'17/'18, IJCAI'17, LICS'17

# Overview of recent research lines

Formal methods …

…for strategising agents with quantitative temporal goals

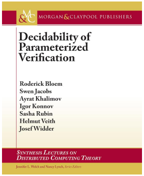
IJCAI'17, CSL'18

# Overview of recent research lines

Formal methods ...

...when the number of agents is not known apriori

CONCUR'14,VMCAI'14,ICALP'15,M&C'15 (book), IJCAR'16, VMCAI'18

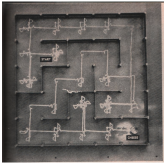


# Overview of recent research lines

Formal methods . . .

. . . when the environment is partially-known

PRIMA’15 (best-paper), AAMAS’15/’16, IJCAI’16/’17




European Commission

# What sort of researcher am I?

- I'm part of a growing group bringing insights from formal methods to bear on problems in AI.
- I'm rooted in Formal Methods and Knowledge Representation philosophies.
- My focus has broadened over time:
    1. Automata for reasoning about mathematical structures
    2. Logic/automata in Formal Methods
    3. Formal Methods and Artificial Intelligence

# What sort of researcher am I?

- I'm part of a growing group bringing insights from formal methods to bear on problems in AI.
- I'm rooted in Formal Methods and Knowledge Representation philosophies.
- My focus has broadened over time:
    1. Automata for reasoning about mathematical structures
    2. Logic/automata in Formal Methods
    3. Formal Methods and Artificial Intelligence

# Future research goals

Academic Tools

# Future research goals

Context. Existing tools for multi-agent systems (*MCK* vdMeyden et. al.; *Eve* Wooldridge et. al.; *MCMAS* Lomuscio et. al.) can't handle logics and models above.

Challenge. High computational complexity, complex constructions.

Project. Engineer practical algorithms by translating to optimised tools, such as planners.[4]

Methodology. Give new semantics for logics based on finite paths of $M$.

De Giacomo, Vardi IJCAI '13/'15/'16

McIlraith,... AAMAS/IJCAI/KR '18

[4] Planning in AI is a form of synthesis.