Tractable Verification of Multi-agent Systems

Francesco Belardinelli [1]

[1] Imperial College London, UK and Université d'Evry, France

joint work with S. Demri, A. Lomuscio, N. Murano, and S. Rubin

University of Surrey – 29 May 2019

Verification of (Multi-agent) Systems

The Verification Problem: given a system S and specification P, does S satisfy P?

- safety-critical systems (avionics, AUVs), security and communication protocols, etc.

Verification of (Multi-agent) Systems

The Verification Problem: given a system S and specification P, does S satisfy P?

- safety-critical systems (avionics, AUVs), security and communication protocols, etc.

Model checking in a nutshell [Clarke, Emerson, Sifakis]

1. Model S as some transition system M_S
2. Represent specification P as a formula ϕ_P in some logic-based language
3. Check whether $M_S \models \phi_P$

Verification of (Multi-agent) Systems

The Verification Problem: given a system S and specification P, does S satisfy P?

- safety-critical systems (avionics, AUVs), security and communication protocols, etc.

Model checking in a nutshell [Clarke, Emerson, Sifakis]

1. Model S as some transition system M_S
2. Represent specification P as a formula ϕ_P in some logic-based language
3. Check whether $M_S \models \phi_P$

Background assumptions:

- Discrete graphs/games are a good model for MAS.
- Logic is a good tool for representing properties.

Properties to check

80's-90's: monolithic systems, systems in isolation: LTL, CTL.

Temporal Properties

- the robot will **always** stay in the safe zone. $G\ safe$
- the robot will **finally** reach its target. $F\ target$
- the robot will **always makes progress** towards its goal. $GF\ move$

From System to Game Verification

Since 2000: systems with several components, interacting agents, game structures: ATL, Coalition Logic, Strategy Logic.

Epistemic properties

- **Anonimity:** the attacker does not know how agent i has voted. $\bigwedge_{1 \le j \le c} \neg K_{att}(ch_i = j)$

Strategic properties

- **Coercion Resistance:** the attacker has no strategy whereby he will know how agent i has voted. $\neg \langle\langle att \rangle\rangle F \bigvee_{1 \le j \le c} K_{att}(ch_i = j)$
- There is a [Nash, subgame-perfect, k-robust, ...] **equilibrium**.

Notions of strategies, equilibria from Game Theory → Rational Synthesis [KPV16]

⇒ Automated verification of strategic abilities of autonomous agents (MoChA, Verics, MCMAS)

From System to Game Verification

Since 2000: systems with several components, interacting agents, game structures: ATL, Coalition Logic, Strategy Logic.

Epistemic properties

- **Anonimity:** the attacker does not know how agent i has voted. $\bigwedge_{1 \leq j \leq c} \neg K_{att}(ch_i = j)$

Strategic properties

- **Coercion Resistance:** the attacker has no strategy whereby he will know how agent i has voted. $\neg\langle\langle att \rangle\rangle F \bigvee_{1 \leq j \leq c} K_{att}(ch_i = j)$
- There is a [Nash, subgame-perfect, k-robust, ...] **equilibrium**.

Notions of strategies, equilibria from Game Theory $\rightarrow$ Rational Synthesis [KPV16]

$\Rightarrow$ Automated verification of strategic abilities of autonomous agents (MoChA, Verics, MCMAS)

So far, so good ...

The Problems with MAS Verification

1. MAS require imperfect information:
   - Agents have partial observability.
   - Perfect information unachievable or computationally costly.
   - Imperfect information makes things hard(er).

2. Actions have costs:
   - Costs are not normally modelled in these specification languages.
   - Extension of logic for strategies with production/consumption of resources.

The Problems with MAS Verification

1. MAS require imperfect information:
   - Agents have partial observability.
   - Perfect information unachievable or computationally costly.
   - Imperfect information makes things hard(er).

2. Actions have costs:
   - Costs are not normally modelled in these specification languages.
   - Extension of logic for strategies with production/consumption of resources.

This talk:

1. MAS *with public actions only* $\Rightarrow$ Tractable model checking even with imperfect information. [BLMR17a, BLMR17b, BLMR18]

2. Tractable reasoning about resources in MAS. [BD19]

The Impact of Imperfect Information on Verification

The Information Problem: agents have imperfect/incomplete information about the overall state of the system.

The Impact of Imperfect Information on Verification

The Information Problem: agents have imperfect/incomplete information about the overall state of the system.

- Model checking ATL:

	perfect	**imperfect**
memoryless	PTIME-c. (A. H. K., 2002)	Δ_2^P-c. (Jamroga, Dix, 2006)
perfect recall		undec. (Dima, Tiplea, 2011)

The Impact of Imperfect Information on Verification

The Information Problem: agents have imperfect/incomplete information about the overall state of the system.

- Model checking ATL:

	perfect	**imperfect**
memoryless	PTIME-c. (A. H. K., 2002)	Δ_2^P-c. (Jamroga, Dix, 2006)
perfect recall		undec. (Dima, Tiplea, 2011)

- Long known and not limited to ATL.

Perfect Information: decidable

- Synthesis for LTL goals (Büchi, Landweber, 1969), (Rabin, 1972), (Pnueli, Rosner, 1989)
- Nash equilibria for LTL goals (Mogavero, Murano, Vardi, 2010)

Imperfect Information: undecidable

Synthesis for reachability goals (Peterson, Reif, 1979)

How to tame Imperfect Information

- Abstractions, Approximations: bisimulations, 3-valued logics.

  [BCD$^+$17]: bisimulations for the verification of anonymity and coercion-resistance in the ThreeBallot voting protocol.

  $\rightsquigarrow$ check Catalin Dima's 2018 talk @Surrey

How to tame Imperfect Information

- Abstractions, Approximations: bisimulations, 3-valued logics.

  [BCD$^+$17]: bisimulations for the verification of anonymity and coercion-resistance in the ThreeBallot voting protocol.

  $\rightsquigarrow$ check Catalin Dima's 2018 talk @Surrey

- **In this talk**
  - Semantic restrictions: MAS with only public actions. [BLMR17a, BLMR17b, BLMR18]

How to tame Imperfect Information

- Abstractions, Approximations: bisimulations, 3-valued logics.

  [BCD$^+$17]: bisimulations for the verification of anonymity and coercion-resistance in the ThreeBallot voting protocol.

  $\rightsquigarrow$ check Catalin Dima's 2018 talk @Surrey

- **In this talk**
  - Semantic restrictions: MAS with only public actions. [BLMR17a, BLMR17b, BLMR18]

The source of undecidability in [DT11] is the interplay between...

1. agents having incomparable observations
2. agents using private communication

What happens if we drop 1 or 2?

Drop incomparable observations

All following approaches preserve decidability.

Hierarchies of observations

- Hierarchical observations: chains of visibility

  (Peterson, Reif, 1979), (Pnueli, Rosner, 1990), (Kupferman, Vardi, 2001), (Schewe, Finkbeiner, 2007)

- Hierarchical information: information sets form a chain

  (Berwanger, Mathew, vdBogaard, 2015)

- Hierarchical instances: instance = formula + arena + hierarchy

  (Berthon, Maubert, Murano, 2017)

Drop incomparable observations

All following approaches preserve decidability.

Hierarchies of observations

- Hierarchical observations: chains of visibility
  (Peterson, Reif, 1979), (Pnueli, Rosner, 1990), (Kupferman, Vardi, 2001), (Schewe, Finkbeiner, 2007)
- Hierarchical information: information sets form a chain
  (Berwanger, Mathew, vdBogaard, 2015)
- Hierarchical instances: instance = formula + arena + hierarchy
  (Berthon, Maubert, Murano, 2017)

Here we focus on dropping 2.

Idea: drop private communication

- Public Announcement Logic is decidable (Gerbrandy & Groeneveld, 1997)
- Epistemic planning is easier (Pinchinat et al., 2015)
- LTLK synthesis is decidable (vdMeyden & Wilke, 2005)

Research question: is there a meaningful set up with imperfect information and public actions enjoying a tractable model checking problem?

Concurrent Game Structures with Imperfect Information

iCGS

An iCGS $M = \langle Ag, AP, S, S_0, \{Act_a\}_{a \in Ag}, \delta, \lambda, \{\sim_a\}_{a \in Ag} \rangle$ includes

- agents Ag
- atomic propositions AP
- actions Act_a and joint actions $ACT = \prod_{a \in Ag} Act_a$
- states S with initial states $S_0 \subseteq S$
- transition function $\delta : S \times ACT \to S$
- labelling function $\lambda : AP \to 2^S$
- **indistinguishability relation** $\sim_a \subseteq S^2$.

- **Perfect Information**: for each $a \in Ag$, $\sim_a$ is the identity relation.

Public Actions iCGS

PA-iCGS

An iCGS S **has only public actions** if for every agent $a \in Ag$, states $s, s' \in S$, and joint actions $J, J' \in ACT$,

$$s \sim_a s' \text{ and } J \neq J' \text{ imply } \delta(s, J) \not\sim_a \delta(s', J')$$

Intuition: no private communication can take place.

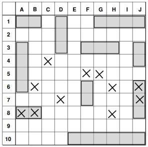
Public Actions iCGS

PA-iCGS

An iCGS S **has only public actions** if for every agent $a \in Ag$, states $s, s' \in S$, and joint actions $J, J' \in ACT$,

$$s \sim_a s' \text{ and } J \neq J' \text{ imply } \delta(s, J) \not\sim_a \delta(s', J')$$

Intuition: no private communication can take place.



Captures many scenarios of interest in Computer Science

- card/board games
- open-outcry auctions
- tweeting
- recording contexts (FHMV, 1995)
- broadcasting systems (Lomuscio, Meyden & Ryan, 2000)
- planning via public actions (Kominis & Geffner, 2015)

Alternating-time Temporal Logic

Definition (ATL*)

State (φ) and **path** (ψ) **formulas** are defined for $p \in AP$ and $A \subseteq Ag$:

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle \psi \\ \psi &::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid G\psi \mid \psi U \psi\end{aligned}$$

- ATL is the fragment of ATL* where path formulas are restricted as

$$\psi ::= X\varphi \mid G\varphi \mid \varphi U \varphi$$

Alternating-time Temporal Logic

Definition (ATL*)

State (φ) and **path** (ψ) **formulas** are defined for $p \in AP$ and $A \subseteq Ag$:

$$\begin{aligned} \varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle \psi \\ \psi &::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid G\psi \mid \psi U \psi \end{aligned}$$

- ATL is the fragment of ATL* where path formulas are restricted as

$$\psi \quad ::= \quad X\varphi \mid G\varphi \mid \varphi U \varphi$$

Strategies

- deterministic with perfect recall: $\sigma : S^+ \to \cup_{a \in Ag} Act_a$
- coherent for agent a: $\sigma(h) \in Act_a$
- uniform for agent a: $h \sim_a h'$ implies $\sigma(h) = \sigma(h')$

Alternating-time Temporal Logic

Interpretation given on **perfect recall, synchronous** iCGS.

Definition (Semantics)

Consider an iCGS M, history $h \in S^+$, computation $\pi \in S^\omega$, and $i \in \mathbb{N}$.

$$
\begin{array}{lll}
(M,h) \models p & \text{iff} & last(h) \in \lambda(p) \\
(M,h) \models \neg\varphi & \text{iff} & (M,h) \not\models \varphi \\
(M,h) \models \varphi_1 \wedge \varphi_2 & \text{iff} & (M,h) \models \varphi_1 \text{ and } (M,h) \models \varphi_2 \\
(M,h) \models \langle\langle A \rangle\rangle \psi & \text{iff} & \text{for some joint strategy } \sigma_A, \\
 & & \text{for all computations } \pi \text{ consistent with } h \text{ and } \sigma_A,\ (M,\pi,|h|) \models \psi \\
(M,\pi,i) \models \varphi & \text{iff} & (M,\pi_{\leq i}) \models \varphi \\
(M,\pi,i) \models \neg\psi & \text{iff} & (M,\pi,i) \not\models \psi \\
(M,\pi,i) \models \psi_1 \wedge \psi_2 & \text{iff} & (M,\pi,i) \models \psi_1 \text{ and } (M,\pi,i) \models \psi_2 \\
(M,\pi,i) \models X\psi & \text{iff} & (M,\pi,i+1) \models \psi \\
(M,\pi,i) \models \psi_1 U \psi_2 & \text{iff} & \text{for some } j \geq i,\ (M,\pi,j) \models \psi_2, \\
 & & \text{for all } k,\ i \leq k < j \text{ implies } (M,\pi,k) \models \psi_1
\end{array}
$$

Alternating-time Temporal Logic

Interpretation given on **perfect recall, synchronous** iCGS.

Definition (Semantics)

Consider an iCGS M, history $h \in S^+$, computation $\pi \in S^\omega$, and $i \in \mathbb{N}$.

$$
\begin{array}{lll}
(M,h) \models p & \text{iff} & \mathit{last}(h) \in \lambda(p) \\
(M,h) \models \neg\varphi & \text{iff} & (M,h) \not\models \varphi \\
(M,h) \models \varphi_1 \wedge \varphi_2 & \text{iff} & (M,h) \models \varphi_1 \text{ and } (M,h) \models \varphi_2 \\
(M,h) \models \langle\!\langle A \rangle\!\rangle \psi & \text{iff} & \text{for some joint strategy } \sigma_A, \\
 & & \text{for all computations } \pi \text{ consistent with } h \text{ and } \sigma_A,\ (M,\pi,|h|) \models \psi \\
(M,\pi,i) \models \varphi & \text{iff} & (M,\pi_{\leq i}) \models \varphi \\
(M,\pi,i) \models \neg\psi & \text{iff} & (M,\pi,i) \not\models \psi \\
(M,\pi,i) \models \psi_1 \wedge \psi_2 & \text{iff} & (M,\pi,i) \models \psi_1 \text{ and } (M,\pi,i) \models \psi_2 \\
(M,\pi,i) \models X\psi & \text{iff} & (M,\pi,i+1) \models \psi \\
(M,\pi,i) \models \psi_1 U \psi_2 & \text{iff} & \text{for some } j \geq i,\ (M,\pi,j) \models \psi_2, \\
 & & \text{for all } k,\ i \leq k < j \text{ implies } (M,\pi,k) \models \psi_1
\end{array}
$$

[DT11]: model checking ATL on iCGS with perfect recall is undecidable.

Decidable Model Checking

Theorem ([BLMR17a])

Model checking ATL* *on PA-iCGS is decidable. Specifically, it is* $2\mathrm{EXPTIME}$*-complete.*

- **Lower bound**: model checking ATL* is $2\mathrm{EXPTIME}$-hard already for perfect information (and perfect recall).
- **Upper bound**: the set of strategies making a formula true is recognised by a tree automaton (there exists a bijective encoding $\mu : S_0 \times ACT^* \to S^+$).

Decidable Model Checking

Theorem ([BLMR17a])

Model checking ATL* *on PA-iCGS is decidable. Specifically, it is* $2\mathrm{EXPTIME}$*-complete.*

- **Lower bound**: model checking ATL* is $2\mathrm{EXPTIME}$-hard already for perfect information (and perfect recall).
- **Upper bound**: the set of strategies making a formula true is recognised by a tree automaton (there exists a bijective encoding $\mu : S_0 \times ACT^* \to S^+$).

[BLMR17b]: decidability extends to Strategy Logic

- SL extends ATL* with explicit quantification on strategies as well as strategy binding.
- Model checking SL on PA-iCGS is decidable (TOWER-complete).

$\Rightarrow$ Complex specifications can *in principle* be checked on synchronous, perfect recall MAS as long as evolution is via public actions.

Application: Rational Synthesis

A game $G = \langle M, \{\gamma_a\}_{a \in Ag} \rangle$ is such that

- M is an iCGS
- LTL-formula γ_a is an individual objective for agent $a \in Ag$.

E-NASH (Kupferman et al., 2016)

Consider game G and (LTL) specification φ.
Is there some strategy profile $\vec{\sigma}$ such that

1. $\vec{\sigma}$ is a Nash equilibrium for G
2. the path induced by $\vec{\sigma}$ satisfies φ?

Strong rational synthesis (or A-NASH) amounts to decide whether **all** NE $\vec{\sigma}$ induce φ-satisfying paths.

Application: Rational Synthesis

G is a game on some PA-iCGS.

E-Nash Reduction

E-NASH for (G, φ) can be solved by model checking the SL specification:

$$M \models \exists x_1 \ldots \exists x_n (x_1, a_1) \ldots (x_n, a_n) \left[\bigwedge_{a \in Ag} (\exists y (y, a) \gamma_a \to \gamma_a) \land \varphi \right]$$

A-NASH can similarly be established.

$\Rightarrow$ **E-NASH (resp. A-NASH) on PA-iCGS is decidable, can be solved via model checking SL**.

Summary

Results:

- Imperfect information makes MAS verification hard(er): with perfect recall, it leads to undecidability
- PA-iCGS: a significant class of MAS for which model checking is decidable under the same assumptions.
- Verification of games with public actions only (incl. broadcasting protocols), where no private moves are possible.
- Extension to Strategy Logic and application to rational synthesis (E-NASH, A-NASH).

Future Work:

- Weakening public actions: allowing a "finite amount" of private information.
- Analysis of fragments of SL with lower complexity.

The Cost of Actions

Background

- ATL: logic to reason about the strategic abilities of agents in MAS.

The Cost of Actions

Background

- ATL: logic to reason about the strategic abilities of agents in MAS.

  actions have no cost (?!)

The Cost of Actions

Background

- ATL: logic to reason about the strategic abilities of agents in MAS.

  actions have no cost (?!)

- RB$\pm$ATL: resource-bounded extension of ATL. [ALNR14]

The Cost of Actions

Background

- ATL: logic to reason about the strategic abilities of agents in MAS.

  actions have no cost (?!)

- RB±ATL: resource-bounded extension of ATL. [ALNR14]

  normally harder model checking problem.

The Cost of Actions

Background

- ATL: logic to reason about the strategic abilities of agents in MAS.

  actions have no cost (?!)

- RB$\pm$ATL: resource-bounded extension of ATL. [ALNR14]

  normally harder model checking problem.

Research Question

- Can we reason about resources *efficiently*?

The Cost of Actions

Background

- ATL: logic to reason about the strategic abilities of agents in MAS.

  actions have no cost (?!)

- RB±ATL: resource-bounded extension of ATL. [ALNR14]

  normally harder model checking problem.

Research Question

- Can we reason about resources *efficiently*?

Main Contribution

- Model checking RB±ATL($\{1\}, 1$) is PTIME-complete. [BD19]

The Cost of Actions

Background

- ATL: logic to reason about the strategic abilities of agents in MAS.

  actions have no cost (?!)

- RB$\pm$ATL: resource-bounded extension of ATL. [ALNR14]

  normally harder model checking problem.

Research Question

- Can we reason about resources *efficiently*?

Main Contribution

- Reasoning about a single resource in CTL comes at no extra computational complexity.

The Cost of Actions

Background

- ATL: logic to reason about the strategic abilities of agents in MAS.

  actions have no cost (?!)

- RB±ATL: resource-bounded extension of ATL. [ALNR14]

  normally harder model checking problem.

Research Question

- Can we reason about resources *efficiently*?

Main Contribution

- Reasoning about a single resource in CTL comes at no extra computational complexity.

  Proof Strategy: we show that the control state reachability and non-termination problems for 1-VASS are in PTIME.

The Cost of Actions

Background

- ATL: logic to reason about the strategic abilities of agents in MAS.

  actions have no cost (?!)

- RB±ATL: resource-bounded extension of ATL. [ALNR14]

  normally harder model checking problem.

Research Question

- Can we reason about resources *efficiently*?

Main Contribution

- Reasoning about a single resource in CTL comes at no extra computational complexity.

  Proof Strategy: we show that the control state reachability and non-termination problems for 1-VASS are in PTIME.

Hereafter we assume perfect information!

Motivating Scenario

- A rover is exploring an unknown area.
- At any time the rover can move around or recharge its battery, but not at the same time.
- Moving around consumes one energy unit at every time step, whereas the rover can recharge of one energy unit at a time.
- Switching between modes also requires one energy unit.

Motivating Scenario

- A rover is exploring an unknown area.
- At any time the rover can move around or recharge its battery, but not at the same time.
- Moving around consumes one energy unit at every time step, whereas the rover can recharge of one energy unit at a time.
- Switching between modes also requires one energy unit.

Specification:

- Is it always the case that, given an energy budget of b units, the rover will be able to move?

Resource-bounded Concurrent Game Structures

Intuition: extension of CGS where actions consume as well as produce resources.

Definition (RB-CGS)

A **resource-bound CGS** is a tuple $S = \langle Ag, AP, S, S_0, \{Act_a\}_{a \in Ag}, \delta, \lambda, \mathbf{r}, \mathbf{cost} \rangle$ such that

- $\langle Ag, AP, S, S_0, \{Act_a\}_{a \in Ag}, \delta, \lambda \rangle$ is a CGS (with perfect information)
- $\mathbf{r} \geq 1$ is the number of *resources*
- $\mathbf{cost} : S \times Ag \times Act \to \mathbb{Z}^r$ is the *cost function*.

Resource-bounded Concurrent Game Structures

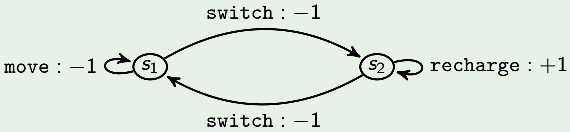
Intuition: extension of CGS where actions consume as well as produce resources.

Definition (RB-CGS)

A **resource-bound CGS** is a tuple $S = \langle Ag, AP, S, S_0, \{Act_a\}_{a \in Ag}, \delta, \lambda, \mathbf{r}, \mathbf{cost}\rangle$ such that

- $\langle Ag, AP, S, S_0, \{Act_a\}_{a \in Ag}, \delta, \lambda\rangle$ is a CGS (with perfect information)
- $\mathbf{r} \geq 1$ is the number of *resources*
- $\mathbf{cost} : S \times Ag \times Act \to \mathbb{Z}^r$ is the *cost function*.

Example (the rover)



Resource-bounded Alternating-time Temporal Logic

RB±ATL: extension of ATL to reason about resources. [ALNR14]

Definition (Satisfaction)

$(M, s) \models \langle\langle A \rangle\rangle^{\vec{b}} \psi$ iff for some joint $\vec{b}$-strategy σ_A,
for all computations $\pi \in Comp(s, \sigma_A)$, $(M, \pi) \models \psi$

- For a $\vec{b}$-strategy $\sigma_A : S^+ \rightarrow Act_A$ all computation are consistent with budget $\vec{b}$.
  - the actions of opponent coalition $Ag \setminus A$ are unrestricted.

Resource-bounded Alternating-time Temporal Logic

RB±ATL: extension of ATL to reason about resources. [ALNR14]

Definition (Satisfaction)

$(M, s) \models \langle\langle A \rangle\rangle^{\vec{b}} \psi$ iff for some joint $\vec{b}$-strategy σ_A,
for all computations $\pi \in Comp(s, \sigma_A)$, $(M, \pi) \models \psi$

- For a $\vec{b}$-strategy $\sigma_A : S^+ \to Act_A$ all computation are consistent with budget $\vec{b}$.
  - the actions of opponent coalition $Ag \setminus A$ are unrestricted.
- For $|Ag| = 1$, we obtain a resource-bounded version of CTL:

$$E^{\vec{b}}\psi ::= \langle\langle \{1\} \rangle\rangle^{\vec{b}} \psi \quad \text{and} \quad A^{\vec{b}}\psi ::= \neg E^{\vec{b}} \neg\psi = [\{1\}]^{\vec{b}} \psi$$

Resource-bounded Alternating-time Temporal Logic

RB±ATL: extension of ATL to reason about resources. [ALNR14]

Definition (Satisfaction)

$(M, s) \models \langle\!\langle A \rangle\!\rangle^{\vec{b}} \psi$ iff for some joint $\vec{b}$-strategy σ_A,
for all computations $\pi \in Comp(s, \sigma_A)$, $(M, \pi) \models \psi$

- For a $\vec{b}$-strategy $\sigma_A : S^+ \to Act_A$ all computation are consistent with budget $\vec{b}$.
  - the actions of opponent coalition $Ag \setminus A$ are unrestricted.
- For $|Ag| = 1$, we obtain a resource-bounded version of CTL:

$$E^{\vec{b}}\psi ::= \langle\!\langle \{1\} \rangle\!\rangle^{\vec{b}} \psi \quad \text{and} \quad A^{\vec{b}}\psi ::= \neg E^{\vec{b}} \neg \psi = [\![\{1\}]\!]^{\vec{b}} \psi$$

Example

It is always the case that, given an energy budget of b units, the rover will be able to move:

$$A^{\omega} G \ E^{b} F \ move$$

Model Checking RB±ATL*: Complexity

$r \backslash \lvert Ag \rvert$	∞	$\geq \mathbf{2}$	$\mathbf{1}$
∞	2EXPTIME-c [ABDL18]		EXPSPACE-c. [ABDL18]
$\geq \mathbf{1}$	(same as ATL*)		PSPACE-c [ABDL18] (same as CTL*)

- Tight complexity bounds for all flavours of RB±ATL*.
- In several cases the same complexity as resource-free logics.
- Still, very much intractable.

Model Checking RB±ATL: Complexity

<table>
<tr><th>$r \backslash |Ag|$</th><th>∞</th><th>≥ 2</th><th>1</th></tr>
<tr><td>∞</td><td colspan="2">2EXPTIME-c. [ABDL18]</td><td>EXPSPACE-c. [ABDL18]</td></tr>
<tr><td>≥ 4</td><td colspan="2">EXPTIME-c. [ABDL18]</td><td rowspan="3">in PSPACE [ABDL18]
PSPACE-h. [BFG+15]</td></tr>
<tr><td>3</td><td colspan="2" rowspan="2">in EXPTIME [ABDL18]
PSPACE-h. [BFG+15]</td></tr>
<tr><td>2</td></tr>
<tr><td>1</td><td colspan="2">in PSPACE [ALNR17]
PTIME-h. (from ATL)</td><td>ptime-c. [BD19]</td></tr>
</table>

Limitations:

- The model checking problem is normally harder (from PTIME-c. up to 2EXPTIME-c.).
- Loose complexity bounds in several cases (e.g., $r = 2, 3$ and $|Ag| \geq 2$).

Positive Results:

- Model checking RB±ATL($\{1\}, 1$) is PTIME-complete.

  $\Rightarrow$ as hard as CTL: reasoning about resources comes at no extra computational complexity!

Decision problems for VASS

We prove the PTIME-upper bound by solving decision problems for 1-VASS.

Definition (VASS)

A **Vector Addition System with States** is a tuple $V = \langle Q, r, R \rangle$ such that

1. Q is a set of **control states**
2. $r \geq 1$ is the number of **counters**
3. the **transition relation** R is a finite subset of $Q \times \mathbb{Z}^r \times Q$.

A **1-VASS** is a VASS with a single counter ($r = 1$).

Decision problems for VASS

We prove the PTIME-upper bound by solving decision problems for 1-VASS.

Definition (VASS)

A **Vector Addition System with States** is a tuple $V = \langle Q, r, R \rangle$ such that

1. Q is a set of **control states**
2. $r \geq 1$ is the number of **counters**
3. the **transition relation** R is a finite subset of $Q \times \mathbb{Z}^r \times Q$.

A **1-VASS** is a VASS with a single counter ($r = 1$).

Control state reachability problem CREACH(VASS):

Input: a VASS V, a configuration $(q_0, \vec{x}_0)$, and a control state q_f.

Question: Is there a finite run from $(q_0, \vec{x}_0)$ to a (final) configuration with state q_f?

Non-termination problem NONTER(VASS):

Input: a VASS V and a configuration $(q_0, \vec{x}_0)$.

Question: Is there an infinite run with initial configuration $(q_0, \vec{x}_0)$?

Decision problems for VASS

We prove the PTIME-upper bound by solving decision problems for 1-VASS.

Definition (VASS)

A **Vector Addition System with States** is a tuple $V = \langle Q, r, R \rangle$ such that

1. Q is a set of **control states**
2. $r \geq 1$ is the number of **counters**
3. the **transition relation** R is a finite subset of $Q \times \mathbb{Z}^r \times Q$.

A **1-VASS** is a VASS with a single counter ($r = 1$).

Control state reachability problem CREACH(VASS):

Input: a VASS V, a configuration $(q_0, \vec{x}_0)$, and a control state q_f.

Question: Is there a finite run from $(q_0, \vec{x}_0)$ to a (final) configuration with state q_f?

Non-termination problem NONTER(VASS):

Input: a VASS V and a configuration $(q_0, \vec{x}_0)$.

Question: Is there an infinite run with initial configuration $(q_0, \vec{x}_0)$?

Theorem

Both CREACH(1-VASS) and NONTER(1-VASS) are decidable in PTIME.

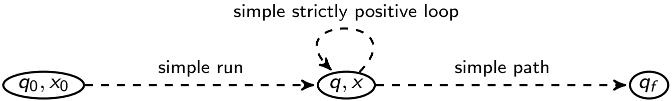
Decidability Results for 1-VASS

Theorem

CREACH(1-VASS) is decidable in PTIME.

Proof Idea: configuration (q_f, x_f) is reachable from (q_0, x_0) iff there is a finite run with

1. an initial simple run (*no repetitions*)
2. a simple *strictly positive* loop
3. a final simple path.



Same proof idea as [RY86], but actually we fixed that proof.

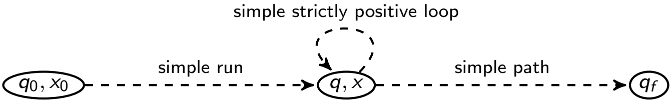
Decidability Results for 1-VASS

Theorem

CREACH(1-VASS) is decidable in PTIME.

Proof Idea: configuration (q_f, x_f) is reachable from (q_0, x_0) iff there is a finite run with

1. an initial simple run (*no repetitions*)
2. a simple *strictly positive* loop
3. a final simple path.



Same proof idea as [RY86], but actually we fixed that proof.

Theorem

NONTERM(1-VASS) is decidable in PTIME.

Proof Idea: there exists a non-terminating run from (q_0, x_0) iff there is a finite run that satisfies (1) and (2) above.

$\textsc{ptime}$-Upper Bound for RB±ATL$(\{1\}, 1)$

To decide whether $M \models \varphi$, we introduce a labelling algorithm that works bottom-up on the structure of formula φ.

- Subformulas $\phi = E^b(\phi_1 U \phi_2)$ are dealt with by solving CREACH(V^M).
- Subformulas $\phi = E^b G \phi'$ are dealt with by solving NONTERM(V^M).

The whole procedure is in $\textsc{ptime}$.

Summary

Main Result

- Reasoning about a single resource in CTL comes at no extra computational complexity!

Summary

Main Result

- Reasoning about a single resource in CTL comes at no extra computational complexity!

Future Work

- Budget Synthesis: find a (minimal) budget b such that $M \models \langle\!\langle A \rangle\!\rangle^b \psi$.
- Implementation in a model checking tool.
- Open problems: model checking complexity of RB$\pm$ATL$(\{1, 2\}, 1)$?

Conclusion

- Verification is a key issue for the deployment of Multi-agent Systems.
- We presented tractable instances of MAS model checking, mainly by restricting *meaningfully* the class of systems.
- Still lots to do . . .

References

N. Alechina, N. Bulling, S. Demri, and B. Logan.
On the complexity of resource-bounded logics.
Theoretical Computer Science, 750:69–100, 2018.

N. Alechina, B. Logan, H.N. Nguyen, and F. Raimondi.
Decidable model-checking for a resource logic with production of resources.
In *ECAI'14*, pages 9–14, 2014.

N. Alechina, B. Logan, H.N. Nguyen, and F. Raimondi.
Model-checking for resource-bounded ATL with production and consumption of resources.
Journal of Computer and System Sciences, 88:126–144, 2017.

F. Belardinelli, R. Condurache, C. Dima, W. Jamroga, and A. V. Jones.
Bisimulations for verifying strategic abilities with an application to threeballot.
In *AAMAS17*, pages 1286–1295, 2017.

F. Belardinelli and S. Demri.
Resource-bounded atl: the quest for tractable fragments.
In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS19)*, 2019.

M. Blondin, A. Finkel, S. Göller, C. Haase, and P. McKenzie.
Reachability in two-dimensional vector addition systems with states is PSPACE-complete.
In *LICS'15*, pages 32–43. ACM Press, 2015.

F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin.
Verification of multi-agent systems with imperfect information and public actions.
In *17*, pages 1268–1276, 2017.

Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin.
Verification of broadcasting multi-agent systems against an epistemic strategy logic.
In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 91–97. ijcai.org, 2017.

F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin.
Decidable verification of multi-agent systems with bounded private actions.
In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS18)*, 2018.

C. Dima and F. Tiplea.