Multi-valued Logics and Abstractions for the Verification of Strategic Properties in MAS with Imperfect Information

Francesco Belardinelli

Imperial College London, UK & Université d'Evry, France

SR 2020

joint work with V. Malvone (Telecom Paris) & A. Lomuscio (Imperial College London)

Verification of (Multi-agent) Systems

The Verification Problem

Given a system S and specification P, does S satisfy P?

- Safety: errors cost lives (e.g., Therac-25).
- Mission: errors cost in terms of objectives (e.g., Arianne 5).
- Business: errors cost money (e.g., Pentium 5, Denver airport).

In safety-critical systems failure is not an option!

Verification of (Multi-agent) Systems

The Verification Problem

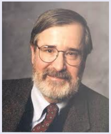
Given a system S and specification P, does S satisfy P?

- Safety: errors cost lives (e.g., Therac-25).
- Mission: errors cost in terms of objectives (e.g., Arianne 5).
- Business: errors cost money (e.g., Pentium 5, Denver airport).

In safety-critical systems failure is not an option!

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 $M_S \models \phi_P$



Properties to Check

80's-90's: single-component, stand-alone systems: temporal logics LTL, CTL [Pnu77].

Temporal Properties

The robot ...

- ...will **always** avoid obstacles. $G\ avoid\_obstacles$
- ...will **finally** reach its target. $F\ target$
- ...will **always makes progress** towards its goal. $G\ F\ move$
- ...will **eventually** be in the safe zone **forever**. $F\ G\ safe$

From System to Game Verification

Since 2000: systems with several components, interacting agents, game structures:

- **ATL** [AHK02]
- Coalition Logic [Pau02]
- Strategy Logic [CHP07, MMPV14]

From System to Game Verification

Since 2000: systems with several components, interacting agents, game structures:

- **ATL** [AHK02]
- Coalition Logic [Pau02]
- Strategy Logic [CHP07, MMPV14]

Strategic Properties

- **Coercion Resistance:** the attacker has a strategy whereby he will know how agent i has voted. $\langle\langle att \rangle\rangle F \bigvee_{1 \leq j \leq c} K_{att}(ch_i = j)$
- There is a [Nash, subgame-perfect, k-robust, ...] **equilibrium** such that ...

From System to Game Verification

Since 2000: systems with several components, interacting agents, game structures:

- **ATL** [AHK02]
- Coalition Logic [Pau02]
- Strategy Logic [CHP07, MMPV14]

Strategic Properties

- **Coercion Resistance:** the attacker has a strategy whereby he will know how agent i has voted. $\langle\langle att \rangle\rangle F \bigvee_{1 \leq j \leq c} K_{att}(ch_i = j)$
- There is a [Nash, subgame-perfect, k-robust, ...] **equilibrium** such that ...

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** [AHK02]
- Coalition Logic [Pau02]
- Strategy Logic [CHP07, MMPV14]

Strategic Properties

- **Coercion Resistance:** the attacker has a strategy whereby he will know how agent i has voted. $\langle\langle att \rangle\rangle F \bigvee_{1 \leq j \leq c} K_{att}(ch_i = j)$
- There is a [Nash, subgame-perfect, k-robust, ...] **equilibrium** such that ...

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 Problem with MAS Verification

MAS exhibit imperfect information:

- Agents have partial observability/imperfect information about the system.
- Perfect information unachievable or computationally costly.
- Imperfect information makes things hard(er).

The Problem with MAS Verification

MAS exhibit imperfect information:

- Agents have partial observability/imperfect information about the system.
- Perfect information unachievable or computationally costly.
- Imperfect information makes things hard(er).

Perfect Information: decidability results

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

Imperfect Information: undecidability results

Synthesis for reachability goals **(Peterson, Reif, 1979)**

How to tame Imperfect Information?

Semantic Restrictions:

- Hierarchical MAS **(Peterson, Reif, 1979), (Pnueli, Rosner, 1990), (Kupferman, Vardi, 2001), (Schewe, Finkbeiner, 2007), (Berwanger, Mathew, vdBogaard, 2015), (Berthon, Maubert, Murano, 2017)**
- MAS with public actions only. [BLMR17a, BLMR17b, BLMR18]
- ...

How to tame Imperfect Information?

Semantic Restrictions:

- Hierarchical MAS **(Peterson, Reif, 1979), (Pnueli, Rosner, 1990), (Kupferman, Vardi, 2001), (Schewe, Finkbeiner, 2007), (Berwanger, Mathew, vdBogaard, 2015), (Berthon, Maubert, Murano, 2017)**
- MAS with public actions only. [BLMR17a, BLMR17b, BLMR18]
- ...

This talk:

1. Bounded memory and 3-valued logic to approximate perfect recall [BLM18]
2. Perfect Information and 3-valued logic to approximate imperfect information [BLM19, BM20]

Concurrent Game Structures with Imperfect Information

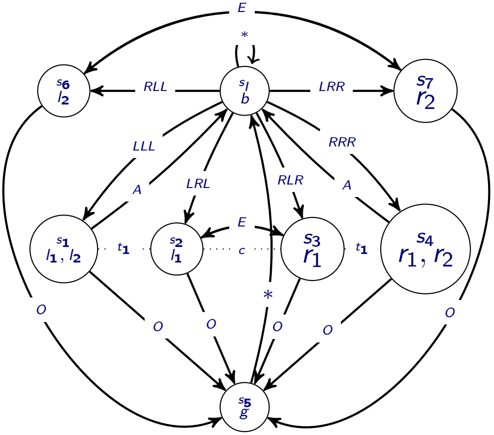
Definition (iCGS)

An iCGS is a tuple $M = \langle Ag, AP, S, s_0, \{Act_i\}_{i \in Ag}, \{\sim_i\}_{i \in Ag}, d, \delta, V \rangle$ where

- Ag is a set of **agents**.
- AP is a set of **atomic propositions**.
- S is a set of **states**, with **initial state** $s_0 \in S$.
- Each Act_i is a set of **actions**.
  $Act = \bigcup_{i \in Ag} Act_i$ is the set of all actions, and
  $ACT = \prod_{i \in Ag} Act_i$ is the set of all joint actions.
- Each $\sim_i$ is a relation of **indistinguishability** (equivalence) between states.
- The **protocol function** $d : Ag \times S \to (2^{Act} \setminus \emptyset)$ defines the availability of actions. The same actions are available in indistinguishable states.
- The **transition function** $\delta : S \times ACT \to S$ assigns a successor state $s' = \delta(s, \vec{a})$ to each state $s \in S$, for every joint action $\vec{a} \in ACT$.
- $V : S \times AP \to \{\top, \bot\}$ is the **two-valued labelling function**.

- **Perfect information**: for every $i \in Ag$, $\sim_i$ is the identity relation.

Variant of the TGC Scenario with Imperfect Information



- Three agents: t_1, t_2, and c.
- t_1 and t_2 need to coordinate L or R.
- c's actions: L, R, E, A, and O.
- t_1 can't observe t_2's choice.
- c can't observe t_1's and t_2's choices.
- Spec: t_1 and c have a strategy to coordinate to go left, but then an agreement has to be reached before visiting the initial state again.

Alternating-time Temporal Logic

To express specifications as above we consider ATL.

Specification Language: Alternating-time temporal logic

State (φ) and path (ψ) formulas in ATL^* are defined as:

$$\begin{aligned} \varphi &::= q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle \Gamma \rangle\rangle \psi \\ \psi &::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid (\psi U \psi) \end{aligned}$$

where $q \in AP$ and $\Gamma \subseteq Ag$.

$\langle\langle \Gamma \rangle\rangle \psi$ ::= "the agents in coalition Γ have a joint strategy to achieve goal ψ".

Alternating-time Temporal Logic

To express specifications as above we consider ATL.

Specification Language: Alternating-time temporal logic

State (φ) and path (ψ) formulas in ATL^* are defined as:

$$\begin{aligned} \varphi &::= q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle\Gamma\rangle\rangle\psi \\ \psi &::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid (\psi U \psi) \end{aligned}$$

where $q \in AP$ and $\Gamma \subseteq Ag$.

$\langle\langle\Gamma\rangle\rangle\psi$ ::= "the agents in coalition Γ have a joint strategy to achieve goal ψ".

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

$$\psi ::= X\varphi \mid (\varphi U \varphi) \mid (\varphi R \varphi)$$

Alternating-time Temporal Logic

To express specifications as above we consider ATL.

Specification Language: Alternating-time temporal logic

State (φ) and path (ψ) formulas in ATL^* are defined as:

$$\begin{aligned} \varphi &::= q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle \Gamma \rangle\rangle \psi \\ \psi &::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid (\psi U \psi) \end{aligned}$$

where $q \in AP$ and $\Gamma \subseteq Ag$.

$\langle\langle \Gamma \rangle\rangle \psi$ ::= "the agents in coalition Γ have a joint strategy to achieve goal ψ".

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

$$\psi ::= X\varphi \mid (\varphi U \varphi) \mid (\varphi R \varphi)$$

$\langle\langle t_1, c \rangle\rangle F(l_1 \wedge \neg b U g)$: t_1 and c have a strategy to coordinate to go left (l_1), but then an agreement has to be reached (g) before visiting the initial state again (b).

Strategies

Strategies

- Each agent can play some *actions* according to the *protocol function*.
- A **strategy** is a conditional plan that prescribes an action at each state.
- The composition of individual strategie induces a unique **outcome**.

Strategies

Strategies

- Each agent can play some *actions* according to the *protocol function*.
- A **strategy** is a conditional plan that prescribes an action at each state.
- The composition of individual strategie induces a unique **outcome**.

Memory in strategies

- Depending on the memory, we distinguish between:
  - *perfect recall (memoryful) strategies (R)* $\implies f : S^+ \to Act$
  - *imperfect recall (positional) strategies (r)* $\implies f : S \to Act$

(R) the players take a decision by considering the history of the game.

(r) the players take a decision by considering the current state of the game.

Strategies

Strategies

- Each agent can play some *actions* according to the *protocol function*.
- A **strategy** is a conditional plan that prescribes an action at each state.
- The composition of individual strategie induces a unique **outcome**.

Memory in strategies

- Depending on the memory, we distinguish between:
  - *perfect recall (memoryful) strategies (R)* $\implies f : S^+ \rightarrow Act$
  - *imperfect recall (positional) strategies (r)* $\implies f : S \rightarrow Act$

(R) the players take a decision by considering the history of the game.

(r) the players take a decision by considering the current state of the game.

Agents' information

- Depending on the players' information, we distinguish between:
  - *perfect information systems (I)*
  - *imperfect information systems (i)*

(I) the players have full knowledge of the state of the game, at every moment.

(i) the players come to decisions without having all relevant information at hand.

Model Checking ATL

Interpretation of ATL^* formulas on iCGS

The 2-valued (2V) satisfaction relation $\models^2$ for an iCGS M, state s, and ATL^* formula $\phi = \langle\langle\Gamma\rangle\rangle\psi$ is defined as

$(M, s) \models^2 \langle\langle\Gamma\rangle\rangle\psi$ iff for some joint strategy F_Γ, for all outcomes $p \in out(s, F_\Gamma), (M, p) \models^2 \psi$

where $out(s, F_\Gamma)$ is the set of all paths p starting from state s and compatible with F_Γ.

Model Checking ATL

Interpretation of ATL^* formulas on iCGS

The 2-valued (2V) satisfaction relation $\models^2$ for an iCGS M, state s, and ATL^* formula $\phi = \langle\langle\Gamma\rangle\rangle\psi$ is defined as

$(M, s) \models^2 \langle\langle\Gamma\rangle\rangle\psi$ iff for some joint strategy F_Γ, for all outcomes $p \in out(s, F_\Gamma), (M, p) \models^2 \psi$

where $out(s, F_\Gamma)$ is the set of all paths p starting from state s and compatible with F_Γ.

Model checking results for ATL:

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

Model Checking ATL

Interpretation of ATL^* formulas on iCGS

The 2-valued (2V) satisfaction relation $\models^2$ for an iCGS M, state s, and ATL^* formula $\phi = \langle\langle\Gamma\rangle\rangle\psi$ is defined as

$(M, s) \models^2 \langle\langle\Gamma\rangle\rangle\psi$ iff for some joint strategy F_Γ, for all outcomes $p \in out(s, F_\Gamma), (M, p) \models^2 \psi$

where $out(s, F_\Gamma)$ is the set of all paths p starting from state s and compatible with F_Γ.

Model checking results for ATL:

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

Idea of [BLM18]: can we approximate perfect recall with bounded recall?

Strategies with Bounded Recall

Hereafter we consider a bound $n \in \mathbb{N}^+ \cup \{\omega\}$.

Uniform Strategies with Bounded Recall

A **uniform strategy with** n**-bounded recall** for agent $i \in Ag$ is a function $f_i^n : S^{\leq n} \to Act_i$ such that for all n-histories h, h':

1. action $f_i^n(h)$ is enabled at h: $f_i^n(h) \in d(i, last(h))$
2. $h \sim_i h'$ implies $f_i^n(h) = f_i^n(h')$
   where, $h \sim_i h'$ iff $|h| = |h'|$ and for every $j \leq |h|$, $h_j \sim_i h'_j$.

Strategies with Bounded Recall

Hereafter we consider a bound $n \in \mathbb{N}^+ \cup \{\omega\}$.

Uniform Strategies with Bounded Recall

A **uniform strategy with** n**-bounded recall** for agent $i \in Ag$ is a function $f_i^n : S^{\leq n} \to Act_i$ such that for all n-histories h, h':

1. action $f_i^n(h)$ is enabled at h: $f_i^n(h) \in d(i, last(h))$
2. $h \sim_i h'$ implies $f_i^n(h) = f_i^n(h')$
   where, $h \sim_i h'$ iff $|h| = |h'|$ and for every $j \leq |h|$, $h_j \sim_i h'_j$.

About the bound

- For $n = 1 \Rightarrow$ imperfect recall (positional) strategies
- For $n = \omega \Rightarrow$ perfect recall (memoryful) strategies

Bounded recall v. bounded memory (strategies as trasducers of bounded size [Ves15]): related but orthogonal issues.

Semantics of ATL with bounded recall

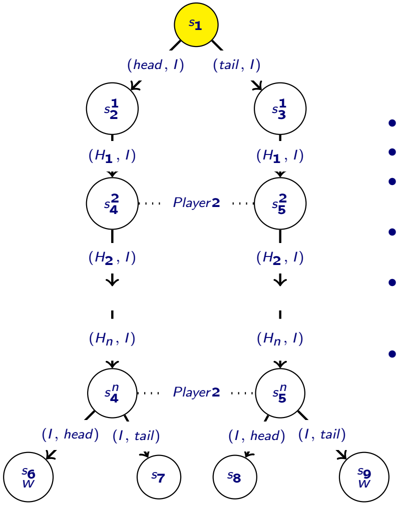
Interpretation of ATL^* formulas on iCGS

The two-valued (2V), n-bounded satisfaction relation $\models^2_n$ for an iCGS M, state s, and ATL^* formula $\phi = \langle\langle\Gamma\rangle\rangle\psi$ is defined as

$(M, s) \models^2_n \langle\langle\Gamma\rangle\rangle\psi$ iff for some n-bounded joint strategy F^n_Γ, for all outcomes $p \in out(s, F^n_\Gamma), (M, p) \models^2_n \psi$

where $out(s, F^n_\Gamma)$ is the set of all paths p starting from state s and compatible with F^n_Γ.

Example: matching pennies with recall



- Player 1 chooses first head or tail.
- Player 2 can see her choice.
- Then, there are $n-1$ steps in which the coin is hidden from Player 2.
- Consider $\langle\langle 2 \rangle\rangle Fwin_2$ and $m, n \in \mathbb{N}^+ \cup \{\omega\}$ with $m < n$.
- Player 2 has no strategy with m-bounded recall to win the game, but she has a n-bounded recall strategy.
- Hence, $s_1 \not\models_m^2 \langle\langle 2 \rangle\rangle Fwin_2$, but $s_1 \models_n^2 \langle\langle 2 \rangle\rangle Fwin_2$.

Model Checking Bounded Recall

Algorithm $MC(M, \varphi, n)$:

$M' = Inflate(M, n)$;
return $MC\_ATL(M', \varphi)$;

- Each state in M' represents a sequence of states in M of length at most n.
- There are exponentially many such sequences.

Model Checking Bounded Recall

Algorithm $MC(M, \varphi, n)$:

$M' = Inflate(M, n)$;
return $MC\_ATL(M', \varphi)$;

- Each state in M' represents a sequence of states in M of length at most n.
- There are exponentially many such sequences.

Complexity Results

ATL^* with:

- $n = \omega$ (perfect recall) is undecidable.
- $n \in \mathbb{N}^+$ is in *EXPTIME*.
- $n \in \mathbb{N}^+$ and fixed is *PSPACE*-complete (the same as imperfect recall).

ATL with:

- $n = \omega$ (perfect recall) is undecidable.
- $n \in \mathbb{N}^+$ is in *EXPTIME*.
- $n \in \mathbb{N}^+$ and fixed is Δ_2^P-complete (the same as imperfect recall).

Key Aspects on 2V Semantics

Naive idea: approximate perfect recall via bounded recall with an increasing bound.

Key Aspects on 2V Semantics

Naive idea: approximate perfect recall via bounded recall with an increasing bound.

Let $m, n \in \mathbb{N}^+ \cup \{\omega\}$ with $m < n$. There exists formulas φ and $\varphi' = \neg\varphi$ in *ATL* such that:

1. $(M, p) \not\models^2_m \varphi$ and $(M, p) \models^2_n \varphi$
2. $(M, p) \models^2_m \varphi'$ and $(M, p) \not\models^2_n \varphi'$

Just take $\varphi = \langle\langle 2 \rangle\rangle F\ win_2$ in the matching penny scenario above.

Key Aspects on 2V Semantics

Naive idea: approximate perfect recall via bounded recall with an increasing bound.

Let $m, n \in \mathbb{N}^+ \cup \{\omega\}$ with $m < n$. There exists formulas φ and $\varphi' = \neg\varphi$ in *ATL* such that:

1. $(M, p) \not\models^2_m \varphi$ and $(M, p) \models^2_n \varphi$
2. $(M, p) \models^2_m \varphi'$ and $(M, p) \not\models^2_n \varphi'$

Just take $\varphi = \langle\langle 2\rangle\rangle F\ win_2$ in the matching penny scenario above.

Consequences

- Any naive attempt to approximate PR by increasing the bound n will not succeed.
- The issue is with models, not just formulas.

$\Rightarrow$ To overcome this problem, we consider a 3-valued semantics.

3-valued Semantics for Bounded ATL

- $V : S \times AP \to \{\top, \bot, \mathrm{uu}\}$ is now a **three-valued labelling function**.
- $\bar{\Gamma} = Ag \setminus \Gamma$

The 3-valued (3V) (n-bounded) satisfaction relation $\models^3_n$ for an iCGS M, state s, and ATL^* formula $\varphi = \langle\langle\Gamma\rangle\rangle\psi$ is defined as

$((M,s) \models^3_n \langle\langle\Gamma\rangle\rangle\psi) = \top$ iff for some joint n-bounded strategy F^n_Γ, for all outcomes $p \in out(s, F^n_\Gamma), ((M,p) \models^3_n \psi) = \top$

$((M,s) \models^3_n \langle\langle\Gamma\rangle\rangle\psi) = \bot$ iff for some joint n-bounded strategy $F^n_{\bar{\Gamma}}$, for all outcomes $p \in out(s, F^n_{\bar{\Gamma}}), ((M,p) \models^3_n \psi) = \bot$

In all other cases the value of ϕ is undefined (uu).

3-valued Semantics for Bounded ATL

- $V : S \times AP \rightarrow \{\top, \bot, \mathrm{uu}\}$ is now a **three-valued labelling function**.
- $\bar{\Gamma} = Ag \setminus \Gamma$

The 3-valued (3V) (n-bounded) satisfaction relation $\models^3_n$ for an iCGS M, state s, and ATL^* formula $\varphi = \langle\langle\Gamma\rangle\rangle\psi$ is defined as

$((M, s) \models^3_n \langle\langle\Gamma\rangle\rangle\psi) = \top$ iff for some joint n-bounded strategy F^n_Γ, for all outcomes $p \in out(s, F^n_\Gamma), ((M, p) \models^3_n \psi) = \top$

$((M, s) \models^3_n \langle\langle\Gamma\rangle\rangle\psi) = \bot$ iff for some joint n-bounded strategy $F^n_{\bar{\Gamma}}$, for all outcomes $p \in out(s, F^n_{\bar{\Gamma}}), ((M, p) \models^3_n \psi) = \bot$

In all other cases the value of ϕ is undefined (uu).

In the matching penny scenario $\langle\langle 2\rangle\rangle F\ win_2$ is undefined for $m < n$.

- both $\langle\langle 2\rangle\rangle F\ win_2$ and $\langle\langle 1\rangle\rangle G\neg win_2$ are false (in the 2V m-bounded semantics).

Model Checking 3-valued ATL

We reduce 3V model checking to 2V model checking.

Given a model checking instance $((M, s) \models \varphi) = v$, for $v \in \{\top, \bot\}$:

1. For every atom $q \in AP$, introduce two new atoms $q_\top$ and $q_\bot$.
2. Define a 2V-model M' s.t. $q_\top$ (resp. $q_\bot$) is true whenever q is true (resp. false).
3. Model check translation $Transl(\varphi, v)$ on M'.
4. Transfer the result to the original 3V-model M.

Model Checking 3-valued ATL

We reduce 3V model checking to 2V model checking.

Given a model checking instance $((M, s) \models \varphi) = v$, for $v \in \{\top, \bot\}$:

1. For every atom $q \in AP$, introduce two new atoms $q_\top$ and $q_\bot$.
2. Define a 2V-model M' s.t. $q_\top$ (resp. $q_\bot$) is true whenever q is true (resp. false).
3. Model check translation $Transl(\varphi, v)$ on M'.
4. Transfer the result to the original 3V-model M.

Algorithm $Transl(\varphi, v)$

switch(φ)
  case $\varphi = q$:
  switch(v)
    case $v = \top$: return $q_\top$;
    case $v = \bot$: return $q_\bot$;
  case $\varphi = \neg\varphi'$:
  switch(v)
    case $v = \top$: return $Transl(\varphi', \bot)$;
    case $v = \bot$: return $Transl(\varphi', \top)$;

Model Checking 3-valued ATL

Algorithm *Transl*(φ, v) (cont.)

case $\varphi = \varphi' \wedge \varphi''$:
switch(v)
  case $v = \top$: return $Transl(\varphi', \top) \wedge Transl(\varphi'', \top)$;
  case $v = \bot$: return $Transl(\varphi', \bot) \vee Transl(\varphi'', \bot)$;
case $\varphi = \langle\langle\Gamma\rangle\rangle\psi$:
switch(v)
  case $v = \top$: return $\langle\langle\Gamma\rangle\rangle\ Transl(\psi, \top)$;
  case $v = \bot$: return $\langle\langle\bar{\Gamma}\rangle\rangle\ Transl(\psi, \bot)$;
case $\varphi = X\psi$:
switch(v)
  case $v = \top$: return $X\ Transl(\psi, \top)$;
  case $v = \bot$: return $X\ Transl(\psi, \bot)$;
case $\varphi = \psi U \psi'$:
switch(v)
  case $v = \top$: return $Transl(\psi, \top)\ U\ Transl(\psi', \top)$;
  case $v = \bot$: return $Transl(\psi, \bot)\ R\ Transl(\psi', \bot)$
case $\varphi = \psi R \psi'$:
switch(v)
  case $v = \top$: return $Transl(\psi, \top)\ R\ Transl(\psi', \top)$;
  case $v = \bot$: return $Transl(\psi, \bot)\ U\ Transl(\psi', \bot)$

Model Checking 3-valued ATL: Soundness

Lemma

For every iCGS M and ATL* formula φ, given $M' = Duplicate\_atoms(M)$,

$$(M', s) \models^2 Transl(\varphi, \top) \quad \Leftrightarrow \quad ((M, s) \models^3 \varphi) = \top \tag{1}$$
$$(M', s) \models^2 Transl(\varphi, \bot) \quad \Leftrightarrow \quad ((M, s) \models^3 \varphi) = \bot \tag{2}$$
$$(M', s) \models^2 \neg(\varphi_\top \vee \varphi_\bot) \quad \Leftrightarrow \quad ((M, s) \models^3 \varphi) = \mathrm{uu} \tag{3}$$

Complexity Results

The complexity of 3V model checking is the same as 2V.

$\Rightarrow$ Translation *Transl*() is polynomial,

Iterative Model Checking (1)

For every $m, n \in \mathbb{N}^+ \cup \{\omega\}$, formula ϕ in ATL^*, and $m \leq n$:

$$((M,s) \models^3_m \phi) = \top \quad \Rightarrow \quad ((M,s) \models^3_n \phi) = \top \tag{4}$$

$$((M,s) \models^3_m \phi) = \bot \quad \Rightarrow \quad ((M,s) \models^3_n \phi) = \bot \tag{5}$$

Iterative Model Checking (1)

For every $m, n \in \mathbb{N}^+ \cup \{\omega\}$, formula ϕ in ATL^*, and $m \leq n$:

$$((M,s) \models^3_m \phi) = \top \quad \Rightarrow \quad ((M,s) \models^3_n \phi) = \top \tag{4}$$
$$((M,s) \models^3_m \phi) = \bot \quad \Rightarrow \quad ((M,s) \models^3_n \phi) = \bot \tag{5}$$

$$((M,s) \models^3_n \phi) = \top \quad \Rightarrow \quad (M,s) \models^2_n \phi \tag{6}$$
$$((M,s) \models^3_n \phi) = \bot \quad \Rightarrow \quad (M,s) \not\models^2_n \phi \tag{7}$$

Iterative Model Checking (1)

For every $m, n \in \mathbb{N}^+ \cup \{\omega\}$, formula ϕ in ATL^*, and $m \leq n$:

$$((M,s) \models^3_m \phi) = \top \quad \Rightarrow \quad ((M,s) \models^3_n \phi) = \top \tag{4}$$
$$((M,s) \models^3_m \phi) = \bot \quad \Rightarrow \quad ((M,s) \models^3_n \phi) = \bot \tag{5}$$

$$((M,s) \models^3_n \phi) = \top \quad \Rightarrow \quad (M,s) \models^2_n \phi \tag{6}$$
$$((M,s) \models^3_n \phi) = \bot \quad \Rightarrow \quad (M,s) \not\models^2_n \phi \tag{7}$$

By (4) and (6): $((M,s) \models^3_m \phi) = \top \quad \Rightarrow \quad (M,s) \models^2_n \phi$ (8)

By (5) and (7): $((M,s) \models^3_m \phi) = \bot \quad \Rightarrow \quad (M,s) \not\models^2_n \phi$ (9)

Consequences

- By (8) and (9) we can design a procedure for PR, whereby ATL^* formulas are checked in the 3V semantics for increasingly larger bounds.
- If either $\top$ or $\bot$ is returned, by (8) and (9) this is also the truth value for the 2V semantics under perfect recall.

Iterative Model Checking (2)

Algorithm *Iterative_MC*(M, ψ, n):

$j = 1$, $k = \mathrm{uu}$;
while $j \leq n$ and $k = \mathrm{uu}$
    if $MC3(M, \psi, j, \top)$ then $k = \top$;
    else if $MC3(M, \psi, j, \bot)$ then $k = \bot$;
    $j = j + 1$;
end while;
if $k \neq \mathrm{uu}$ then return $(j - 1, k)$;
else return -1;

Iterative Model Checking (2)

Algorithm *Iterative_MC*(M, ψ, n):

$j = 1$, $k = \mathrm{uu}$;
while $j \leq n$ and $k = \mathrm{uu}$
    if $MC3(M, \psi, j, \top)$ then $k = \top$;
    else if $MC3(M, \psi, j, \bot)$ then $k = \bot$;
    $j = j + 1$;
end while;
if $k \neq \mathrm{uu}$ then return $(j - 1, k)$;
else return -1;

Soundness

- *Iterative_MC*() is sound for all bounds $n \in \mathbb{N}^+ \cup \{\omega\}$.
- I.e., if the value returned is different from -1, then $M \models_n^2 \phi$ iff $k = \top$.

Termination

- For $n \in \mathbb{N}^+ \Rightarrow$ *Iterative_MC*() terminates in *EXPTIME*.
- For $n = \omega \Rightarrow$ *Iterative_MC*() does not necessarily terminate.

Conclusions

- We introduced BR and 3V semantics on iCGS to tackle undecidability under PR and II.

Conclusions

- We introduced BR and 3V semantics on iCGS to tackle undecidability under PR and II.
- We proved preservation results for defined truth values from the BR to the PR case and from 3V to 2V semantics for **all** *ATL** specifications.

Conclusions

- We introduced BR and 3V semantics on iCGS to tackle undecidability under PR and II.
- We proved preservation results for defined truth values from the BR to the PR case and from 3V to 2V semantics for **all** ATL^* specifications.
- We introduced an iterative procedure that, in some cases, solves the MC problem under PR by taking a bounded amount of memory.

Conclusions

- We introduced BR and 3V semantics on iCGS to tackle undecidability under PR and II.
- We proved preservation results for defined truth values from the BR to the PR case and from 3V to 2V semantics for **all** ATL^* specifications.
- We introduced an iterative procedure that, in some cases, solves the MC problem under PR by taking a bounded amount of memory.
- Since model checking PR (under II) is undecidable in general, the procedure discussed is naturally partial.

Conclusions

- We introduced BR and 3V semantics on iCGS to tackle undecidability under PR and II.
- We proved preservation results for defined truth values from the BR to the PR case and from 3V to 2V semantics for **all** ATL^* specifications.
- We introduced an iterative procedure that, in some cases, solves the MC problem under PR by taking a bounded amount of memory.
- Since model checking PR (under II) is undecidable in general, the procedure discussed is naturally partial.
- We are currently working on implementing the described procedure on a symbolic MC for ATL with II.

PI Abstractions and 3-valued Logic to approximate II

Idea: abstract imperfect information away!

PI Abstractions and 3-valued Logic to approximate II

Idea: cluster indistinguishable states together to remove imperfect information from the model [BLM19, BM20].

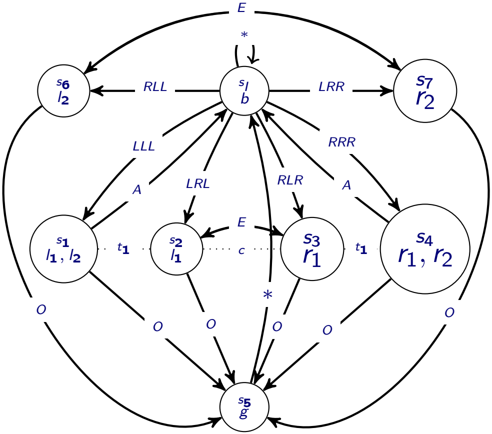
PI Abstractions and 3-valued Logic to approximate II

Idea: cluster indistinguishable states together to remove imperfect information from the model [BLM19, BM20].

- This yields clusters with possibly undefined truth values (when atoms are true in some states and false in others).

$\Rightarrow$ Abstraction and 3-valued Logic.

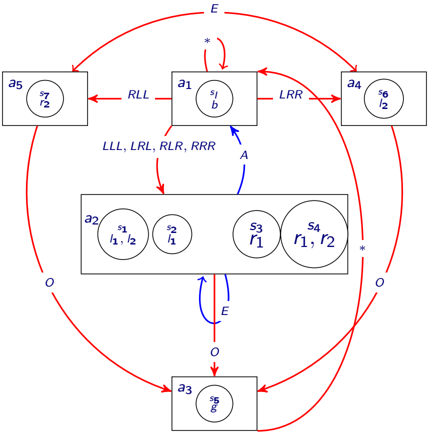
Variant of the TGC Scenario with Imperfect Information



- Three agents: t_1, t_2, and c.
- t_1 and t_2 need to coordinate L or R.
- c's actions: L, R, E, A, and O.
- t_1 can't observe t_2's choice.
- c can't observe t_1's and t_2's choices.
- Spec: $\langle\langle t_1, c\rangle\rangle F(l_1 \wedge \neg b U g)$.

Abstract TGC with PI

We cluster together states indistinguishable for t_1 and c.



- M_a is constructed over the common knowledge set of t_1 and c.
- 2 kinds of transitions:
  may: $\exists s \in t\ \exists s' \in t' : edge(s, s')$
  must: $\forall s \in t\ \exists s' \in t' : edge(s, s')$
- Intuitively,
  may: under-approximations
  must: over-approximations
- The spec is undefined.

Abstract CGS

- **Common knowledge** for coalition $\Gamma \subseteq Ag$: $\sim_{\Gamma}^{C} = (\bigcup_{i \in \Gamma} \sim_i)^*$
- **CK set:** $[s]_{\Gamma} = \{s' \in S \mid s' \sim_{\Gamma}^{C} s\}$

Abstract CGS

- **Common knowledge** for coalition $\Gamma \subseteq Ag$: $\sim_\Gamma^C = (\bigcup_{i \in \Gamma} \sim_i)^*$
- **CK set:** $[s]_\Gamma = \{s' \in S \mid s' \sim_\Gamma^C s\}$

Definition (Abstract CGS)

Given an iCGS M and a coalition $\Gamma \subseteq Ag$, the **abstract CGS**

$$M_\Gamma = \langle Ag, AP, S_\Gamma, [s_0]_\Gamma, \{Act_i\}_{i \in Ag}, d_\Gamma^{may}, d_\Gamma^{must}, \delta_\Gamma^{may}, \delta_\Gamma^{must}, V_\Gamma \rangle$$

is defined as:

1. $S_\Gamma = \{[s]_\Gamma \mid s \in S\}$ is the set of equivalence classes for all states $s \in S$, with initial state $[s_0]_\Gamma$;
2. for $t, t' \in S_\Gamma$ and joint action $\vec{a}$, $t' \in \delta_\Gamma^{may}(t, \vec{a})$ iff for some $s \in t$ and $s' \in t'$, $\delta(s, \vec{a}) = s'$;
3. for $t, t' \in S_\Gamma$ and joint action $\vec{a}$, $t' \in \delta_\Gamma^{must}(t, \vec{a})$ iff for all $s \in t$ there is $s' \in t'$ such that $\delta(s, \vec{a}) = s'$;
4. for $v \in \{\top, \bot\}$, $p \in AP$, and $t \in S_\Gamma$, $V_\Gamma(t, p) = v$ iff $V(s, p) = v$ for all $s \in t$; otherwise, $V_\Gamma(t, p) = \mathrm{uu}$.

Key remark: the abstract CGS has perfect information!

3-valued Semantics for Abstract CGS

Definition (x-Strategy (with perfect recall))

For $x \in \{may, must\}$, a x-**strategy with perfect recall** for agent $i \in Ag$ is a function $f_i^x : S^+ \to Act_i$ such that for every history $h \in S^+$, $f_i^x(h) \in d_\Gamma^x(i, last(h))$.

3-valued Semantics for Abstract CGS

Definition (x-Strategy (with perfect recall))

For $x \in \{may, must\}$, a x-**strategy with perfect recall** for agent $i \in Ag$ is a function $f_i^x : S^+ \rightarrow Act_i$ such that for every history $h \in S^+$, $f_i^x(h) \in d_\Gamma^x(i, last(h))$.

$p \in out(s, F_\Gamma^{must})$ iff for all $j \geq 0$, $p_{j+1} \in \delta^{may}(p_j, (F_\Gamma^{must}(p_{\leq j}), \vec{a}_{\overline{\Gamma}}))$ and for all $i \in \overline{\Gamma}$, $a_i \in d^{may}(i, p_j)$

$p \in out(s, F_\Gamma^{may})$ iff for all $j \geq 0$, $p_{j+1} \in \delta^{must}(p_j, (F_\Gamma^{may}(p_{\leq j}), \vec{a}_{\overline{\Gamma}}))$ and for all $i \in \overline{\Gamma}$, $a_i \in d^{must}(i, p_j)$

The 3-valued (3V) satisfaction relation $\models^3$ for an abstract CGS M_Γ, state s, and ATL^* formula $\varphi = \langle\langle\Gamma\rangle\rangle\psi$ is defined as

$((M_\Gamma, s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \top$ iff for some joint strategy F_Γ^{must}, for all outcomes $p \in out(s, F_\Gamma^{must}), ((M_\Gamma, p) \models^3 \psi) = \top$

$((M_\Gamma, s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \bot$ iff for every joint strategy F_Γ^{may}, for some outcome $p \in out(s, F_\Gamma^{may}), ((M_\Gamma, p) \models^3 \psi) = \bot$

In all other cases, φ is undefined (uu).

3-valued Semantics for Abstract CGS

Definition (x-Strategy (with perfect recall))

For $x \in \{may, must\}$, a x-**strategy with perfect recall** for agent $i \in Ag$ is a function $f_i^x : S^+ \to Act_i$ such that for every history $h \in S^+$, $f_i^x(h) \in d_\Gamma^x(i, last(h))$.

$p \in out(s, F_\Gamma^{must})$ iff for all $j \geq 0$, $p_{j+1} \in \delta^{may}(p_j, (F_\Gamma^{must}(p_{\leq j}), \vec{a}_{\overline{\Gamma}}))$ and for all $i \in \overline{\Gamma}$, $a_i \in d^{may}(i, p_j)$

$p \in out(s, F_\Gamma^{may})$ iff for all $j \geq 0$, $p_{j+1} \in \delta^{must}(p_j, (F_\Gamma^{may}(p_{\leq j}), \vec{a}_{\overline{\Gamma}}))$ and for all $i \in \overline{\Gamma}$, $a_i \in d^{must}(i, p_j)$

The 3-valued (3V) satisfaction relation $\models^3$ for an abstract CGS M_Γ, state s, and ATL^* formula $\varphi = \langle\langle\Gamma\rangle\rangle\psi$ is defined as

$((M_\Gamma, s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \top$ iff for some joint strategy F_Γ^{must}, for all outcomes $p \in out(s, F_\Gamma^{must}), ((M_\Gamma, p) \models^3 \psi) = \top$

$((M_\Gamma, s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \bot$ iff for every joint strategy F_Γ^{may}, for some outcome $p \in out(s, F_\Gamma^{may}), ((M_\Gamma, p) \models^3 \psi) = \bot$

In all other cases, φ is undefined (uu).

- **may**-components: under-approximations
- **must**-components: over-approximations

Properties

Conservativeness

- The 3V semantics for ATL^* is a conservative extension of its 2V semantics.

That is, in standard iCGS the 2V and 3V semantics coincide.

Properties

Conservativeness

- The 3V semantics for ATL^* is a conservative extension of its 2V semantics.

That is, in standard iCGS the 2V and 3V semantics coincide.

3-valued Model Checking

- ATL^*: 2EXPTIME-complete.
- ATL: PTIME-complete.

The same as for the 2V, perfect information case.

Main Property

Lemma

Given an iCGS M, state s, and coalition $\Gamma \subseteq Ag$, for every Γ-formula ϕ in ATL^*,

$$((M_\Gamma, [s]_\Gamma) \models_I^3 \phi) = \top \quad \Rightarrow \quad (M, s) \models_i^2 \phi$$
$$((M_\Gamma, [s]_\Gamma) \models_I^3 \phi) = \bot \quad \Rightarrow \quad (M, s) \not\models_i^2 \phi$$

$\Rightarrow$ We can verify imperfect information by checking 3V perfect information.

Limitation: results are restricted to Γ-formulas.

Main Property

Lemma

Given an iCGS M, state s, and coalition $\Gamma \subseteq Ag$, for every Γ-formula ϕ in ATL^*,

$$((M_\Gamma, [s]_\Gamma) \models^3_I \phi) = \top \quad \Rightarrow \quad (M, s) \models^2_i \phi$$
$$((M_\Gamma, [s]_\Gamma) \models^3_I \phi) = \bot \quad \Rightarrow \quad (M, s) \not\models^2_i \phi$$

$\Rightarrow$ We can verify imperfect information by checking 3V perfect information.

Limitation: results are restricted to Γ-formulas.

Question: what if undefined uu is returned?

Main Property

Lemma

Given an iCGS M, state s, and coalition $\Gamma \subseteq Ag$, for every Γ-formula ϕ in ATL^*,

$$((M_\Gamma, [s]_\Gamma) \models_I^3 \phi) = \top \quad \Rightarrow \quad (M, s) \models_i^2 \phi$$
$$((M_\Gamma, [s]_\Gamma) \models_I^3 \phi) = \bot \quad \Rightarrow \quad (M, s) \not\models_i^2 \phi$$

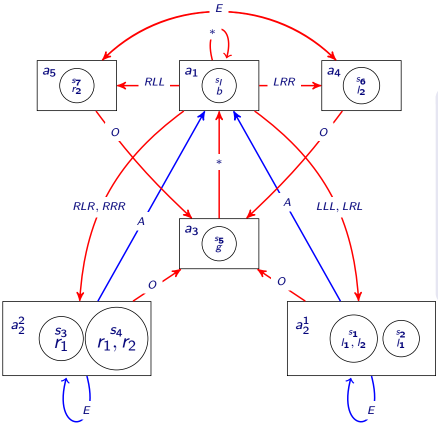
$\Rightarrow$ We can verify imperfect information by checking 3V perfect information.

Limitation: results are restricted to Γ-formulas.

Question: what if undefined uu is returned? Let's refine!

Refined Train Gate Controller

Failure couple (s_f, ϕ): ϕ is not defined at s_f but all its subformulas are.



- A state s_f can be split if:
  (i) for all the ingoing edges $\rightarrow$ the actions of t_1 and c are different;
  (ii) it is possible to construct two new states in accordance with II.
- The refinement procedure splits a_2.
- The spec is defined and it is true!

Refinement Procedure

Algorithm *Refinement*(M_Γ, M, s_f):

for $s, s' \in s_f$, do $m[s, s'] = true$;
  $Check_1(M_\Gamma, M, s_f, m)$;   check for "indistinguishable" incoming transitions
  $update = true$;
  while $update = true$
    $Check_2(M_\Gamma, s_f, m, update)$;
  $split = false$;
  while $s, s' \in s_f$ and $split = false$
    if $m[s, s'] = true$ then
    $remove(s_f, S_\Gamma)$;
    $add(v, S_\Gamma)$; $add(w, S_\Gamma)$; $add(s, v)$; $add(s', w)$;
    $split = true$;
    for $t \in s_f$
      if $m[s, t] = true$ then $add(t, w)$;
      else $add(t, v)$;

Refined CGS

Definition (Refinement)

Given an abstract CGS M_Γ, its refinement

$$M_\Gamma^r = \langle Ag, AP, S_\Gamma^r, s_0^r, \{Act_i\}_{i\in Ag}, d_\Gamma^{may}, d_\Gamma^{must}, \delta_\Gamma^{may}, \delta_\Gamma^{must}, V_\Gamma^r\rangle$$

obtained by an application of algorithm $Refinement(M_\Gamma, M, s_f)$ is defined as

1. S_Γ^r is the set S_Γ of states in M_Γ, possibly without the "failure" state s_f, but with the new states added by $Refinement()$.
   Then, s_0^r is the state in S_Γ^r such that $s_0 \in s_0^r$, for $s_0 \in M$.
2. For $x \in \{may, must\}$, the transitions relations δ_Γ^x and the protocol functions d_Γ^x are defined as for the abstract CGS.
3. For $v \in \{\top, \bot\}$, $p \in AP$, and $t \in S_\Gamma^r$, $V_\Gamma^r(t,p) = v$ iff $V(s,p) = v$ for all $s \in t$; otherwise, $V_\Gamma(s,p) = \mathrm{uu}$.

Main Preservation Result

Lemma

Given an iCGS M, state s, coalition Γ, its abstract CGS M_Γ with refinement M_Γ^r, and state $s_\Gamma^r \ni s$, for every Γ-formula ϕ in ATL^*,

$$((M_\Gamma^r, s_\Gamma^r) \models_I^3 \phi) = \top \quad \Rightarrow \quad (M, s) \models_i^2 \phi \tag{10}$$

$$((M_\Gamma^r, s_\Gamma^r) \models_I^3 \phi) = \bot \quad \Rightarrow \quad (M, s) \not\models_i^2 \phi \tag{11}$$

Main Preservation Result

Lemma

Given an iCGS M, state s, coalition Γ, its abstract CGS M_Γ with refinement M_Γ^r, and state $s_\Gamma^r \ni s$, for every Γ-formula ϕ in ATL^*,

$$((M_\Gamma^r, s_\Gamma^r) \models_I^3 \phi) = \top \quad \Rightarrow \quad (M, s) \models_i^2 \phi \tag{10}$$

$$((M_\Gamma^r, s_\Gamma^r) \models_I^3 \phi) = \bot \quad \Rightarrow \quad (M, s) \not\models_i^2 \phi \tag{11}$$

- By leveraging on this lemma we can refine iteratively to obtain a defined result.
- Since the problem is undecidable in general, this procedure is not guaranteed to produce a defined answer.
- Again, these results are limited to Γ-formulas.

Conclusions

- **Motivation:** undecidable model checking under PR and II.

Conclusions

- **Motivation:** undecidable model checking under PR and II.
- Finding (good) approximations/abstractions remains an open problem at present.

Conclusions

- **Motivation:** undecidable model checking under PR and II.
- Finding (good) approximations/abstractions remains an open problem at present.
- We presented a notion of abstraction between different classes of CGS to overcome this difficulty.

Conclusions

- **Motivation:** undecidable model checking under PR and II.
- Finding (good) approximations/abstractions remains an open problem at present.
- We presented a notion of abstraction between different classes of CGS to overcome this difficulty.
- We showed that iCGSs admit a 3V PI abstraction which preserves satisfaction back to the original 2V-model.

Conclusions

- **Motivation:** undecidable model checking under PR and II.
- Finding (good) approximations/abstractions remains an open problem at present.
- We presented a notion of abstraction between different classes of CGS to overcome this difficulty.
- We showed that iCGSs admit a 3V PI abstraction which preserves satisfaction back to the original 2V-model.
- We can given an incomplete but sound procedure for the original MC problem, which is undecidable in general.

Conclusions

- **Motivation:** undecidable model checking under PR and II.
- Finding (good) approximations/abstractions remains an open problem at present.
- We presented a notion of abstraction between different classes of CGS to overcome this difficulty.
- We showed that iCGSs admit a 3V PI abstraction which preserves satisfaction back to the original 2V-model.
- We can given an incomplete but sound procedure for the original MC problem, which is undecidable in general.
- **Key issue:** how to find failure states (efficiently)?

Conclusions

- **Motivation:** undecidable model checking under PR and II.
- Finding (good) approximations/abstractions remains an open problem at present.
- We presented a notion of abstraction between different classes of CGS to overcome this difficulty.
- We showed that iCGSs admit a 3V PI abstraction which preserves satisfaction back to the original 2V-model.
- We can given an incomplete but sound procedure for the original MC problem, which is undecidable in general.
- **Key issue:** how to find failure states (efficiently)?

Conclusions

- **Motivation:** undecidable model checking under PR and II.
- Finding (good) approximations/abstractions remains an open problem at present.
- We presented a notion of abstraction between different classes of CGS to overcome this difficulty.
- We showed that iCGSs admit a 3V PI abstraction which preserves satisfaction back to the original 2V-model.
- We can given an incomplete but sound procedure for the original MC problem, which is undecidable in general.
- **Key issue:** how to find failure states (efficiently)? Check our talk@KR2020!

Conclusions

- **Motivation:** undecidable model checking under PR and II.
- Finding (good) approximations/abstractions remains an open problem at present.
- We presented a notion of abstraction between different classes of CGS to overcome this difficulty.
- We showed that iCGSs admit a 3V PI abstraction which preserves satisfaction back to the original 2V-model.
- We can given an incomplete but sound procedure for the original MC problem, which is undecidable in general.
- **Key issue:** how to find failure states (efficiently)? Check our talk@KR2020!

Questions?

References

R. Alur, Th. Henzinger, and O. Kupferman.
Alternating-time temporal logic.
Journal of the ACM, 49(5):672–713, 2002.

F. Belardinelli, A. Lomuscio, and V. Malvone.
Approximating perfect recall when model checking strategic abilities.
In *KR2018*, 2018.

F. Belardinelli, A. Lomuscio, and V. Malvone.
An abstraction-based method for verifying strategic properties in multi-agent systems with imperfect information.
In *Proceedings of AAAI*, 2019.

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.

Francesco Belardinelli and Vadim Malvone.
A three-valued approach to strategic abilities under imperfect information.
In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*, 2020.