Formal Verification of Multi-Agent Systems

Vadim Malvone

Télécom Paris

March 9, 2022

Preface

System Correctness

- A very important problem in safety-critical systems.
- In such systems failure is not an option.

Preface

System Correctness

- A very important problem in safety-critical systems.
- In such systems failure is not an option.

Formal verification

- It studies in depth system correctness.
- It is based on mathematic/logical methods.
- One of the contributions is model checking.

Model checking (1)

$M \models \varphi$

There are three fundamental parts:

- M : modeling a system;
- φ : specifying a property;
- $\models$: verifying that the model M satisfies the property φ.

Model checking (2)

Model

A Kripke Structure is a tuple $K = \langle \mathrm{AP}, \mathrm{St}, S_0, R, L \rangle$, where:

- AP is a set of atomic propositions;
- St is a set of states;
- $S_0 \subseteq \mathrm{St}$ is a set of initial states;
- $R \subseteq \mathrm{St} \times \mathrm{St}$ is a transition relation;
- $L : \mathrm{St} \to 2^{\mathrm{AP}}$ is a labeling function.

Model checking (3)

Specification

- Temporal logics allow to describe the order of the events without define the time explicitly.
- The main temporal logics:
  - LTL (Linear Temporal Logic), in a computation the events are totally ordered.
  - CTL (Computation Tree Logic), in a computation the events are partially ordered.

Model checking (4)

Verification

Given a Kripke structure K and a specification φ, the problem of the model checking consists to verify if $K, s \models \varphi$, for each initial state s.

Multi-agent Systems (1)

Key aspects

- There are many agents interacting among them.
- Each agent has a set of *strategies*.
- A *strategy* is a conditional plan that at each step of the game prescribes an action.
- The composition of strategies induces a computation.

Multi-agent Systems (2)

Model

- A game structure is a tuple $G =< \mathrm{AP}, St, s_I, Ac, Ag, tr >$, where:
  - AP is a set of atomic propositions;
  - St is a set of states;
  - $s_I \in S$ is a designated initial state;
  - Ac is a set of actions;
  - Ag is a set of agents;
  - tr is a transition function.

Multi-agent Systems (2)

Model

- A game structure is a tuple $G =< \text{AP}, St, s_I, Ac, Ag, tr >$, where:
  - AP is a set of atomic propositions;
  - St is a set of states;
  - $s_I \in S$ is a designated initial state;
  - Ac is a set of actions;
  - Ag is a set of agents;
  - tr is a transition function.
- It depends on the interaction between the agents:
  - *Turn-based* $\Rightarrow$ The states of the game are partitioned between the agents, then the owner of a state determines the move to take and thus the next state of the game.
  - *Concurrent* $\Rightarrow$ The agents choose a move (i.e, actions) simultaneously and independently, and the choices together determine the next state of the game.

Multi-agent Systems (3)

Strategies

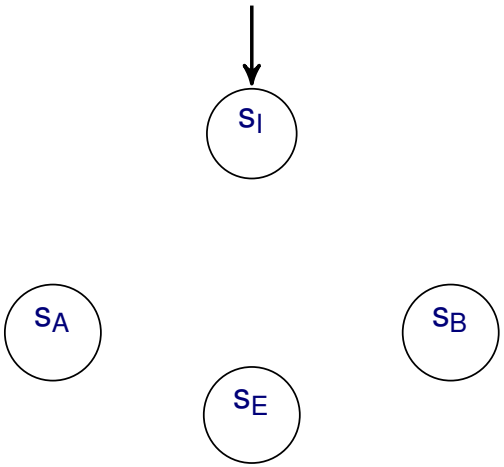
Depending on the memory, we distinguish between:

- *memoryless strategies* $\Rightarrow \sigma : St \to \mathrm{Ac}$; the players take a decision by considering the actual state of the game.
- *bounded strategies* $\Rightarrow \sigma : St^{<g} \to \mathrm{Ac}$; the players take a decision by considering a partial history of the game.
- *memoryfull strategies* $\Rightarrow \sigma : St^{+} \to \mathrm{Ac}$; the players take a decision by considering the full history of the game.

Example: paper, rock, and scissor

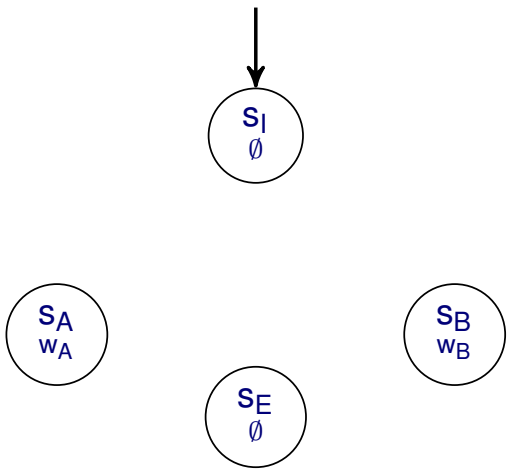
- $\mathrm{Ag} = \{\mathsf{A}: \mathit{Alice}, \mathsf{B}: \mathit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \mathit{Paper}, \mathsf{R}: \mathit{Rock}, \mathsf{S}: \mathit{Scissor}\}$

Example: paper, rock, and scissor



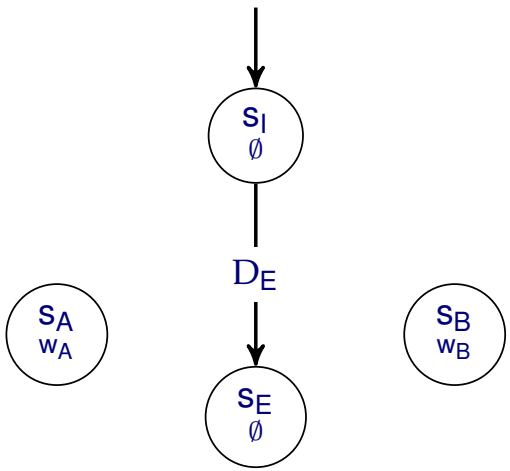
- $\mathrm{Ag} = \{\mathsf{A}: \textit{Alice}, \mathsf{B}: \textit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \textit{Paper}, \mathsf{R}: \textit{Rock}, \mathsf{S}: \textit{Scissor}\}$
- $\mathrm{St} = \{s_I, s_A, s_T, s_B\}$
- s_I: *initial state*

Example: paper, rock, and scissor



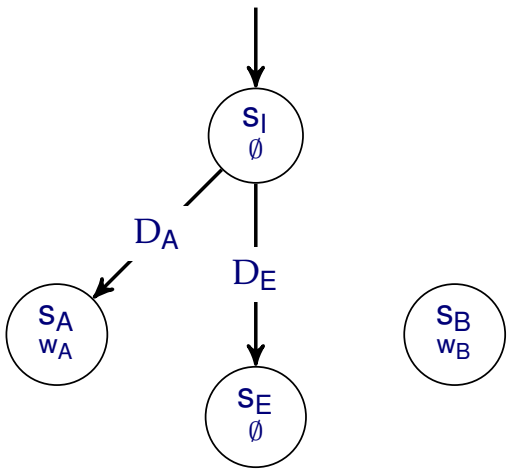
- $\mathrm{Ag} = \{\mathsf{A}: \textit{Alice}, \mathsf{B}: \textit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \textit{Paper}, \mathsf{R}: \textit{Rock}, \mathsf{S}: \textit{Scissor}\}$
- $\mathrm{St} = \{s_I, s_A, s_T, s_B\}$
- s_I: *initial state*
- $\mathrm{AP} = \{w_A: \textit{Alice wins}, w_B: \textit{Bob wins}\}$

Example: paper, rock, and scissor



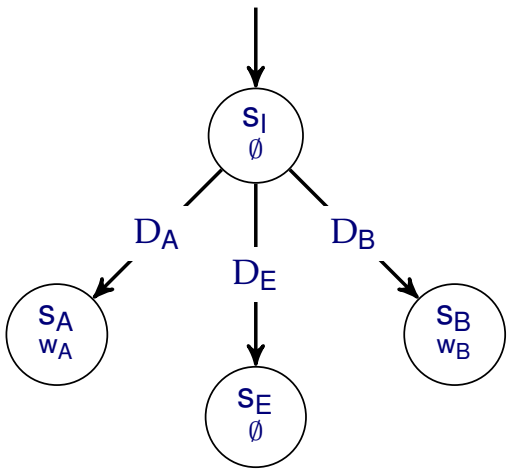
- $\mathrm{Ag} = \{\mathsf{A}: \textit{Alice}, \mathsf{B}: \textit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \textit{Paper}, \mathsf{R}: \textit{Rock}, \mathsf{S}: \textit{Scissor}\}$
- $\mathrm{St} = \{s_I, s_A, s_T, s_B\}$
- s_I: *initial state*
- $\mathrm{AP} = \{w_A: \textit{Alice wins}, w_B: \textit{Bob wins}\}$
- $D_T = \{(\mathsf{P}, \mathsf{P}), (\mathsf{R}, \mathsf{R}), (\mathsf{S}, \mathsf{S})\}$

Example: paper, rock, and scissor



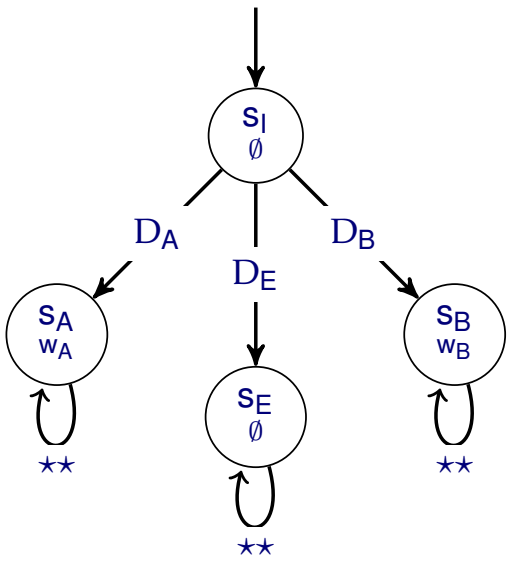
- $\mathrm{Ag} = \{\mathsf{A}: \textit{Alice}, \mathsf{B}: \textit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \textit{Paper}, \mathsf{R}: \textit{Rock}, \mathsf{S}: \textit{Scissor}\}$
- $\mathrm{St} = \{s_I, s_A, s_T, s_B\}$
- s_I: *initial state*
- $\mathrm{AP} = \{w_A: \textit{Alice wins}, w_B: \textit{Bob wins}\}$
- $D_T = \{(\mathsf{P},\mathsf{P}), (\mathsf{R},\mathsf{R}), (\mathsf{S},\mathsf{S})\}$
- $D_A = \{(\mathsf{P},\mathsf{R}), (\mathsf{R},\mathsf{S}), (\mathsf{S},\mathsf{P})\}$

Example: paper, rock, and scissor



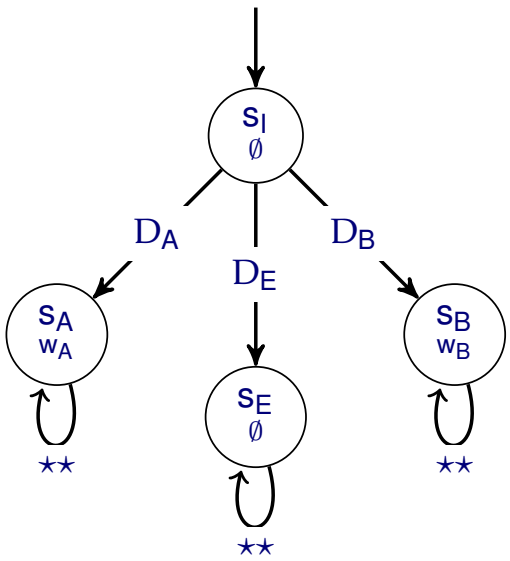
- $\mathrm{Ag} = \{\mathsf{A}: \textit{Alice}, \mathsf{B}: \textit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \textit{Paper}, \mathsf{R}: \textit{Rock}, \mathsf{S}: \textit{Scissor}\}$
- $\mathrm{St} = \{s_I, s_A, s_T, s_B\}$
- s_I: *initial state*
- $\mathrm{AP} = \{w_A: \textit{Alice wins}, w_B: \textit{Bob wins}\}$
- $D_T = \{(\mathsf{P},\mathsf{P}), (\mathsf{R},\mathsf{R}), (\mathsf{S},\mathsf{S})\}$
- $D_A = \{(\mathsf{P},\mathsf{R}), (\mathsf{R},\mathsf{S}), (\mathsf{S},\mathsf{P})\}$
- $D_B = \{(\mathsf{R},\mathsf{P}), (\mathsf{S},\mathsf{R}), (\mathsf{P},\mathsf{S})\}$

Example: paper, rock, and scissor



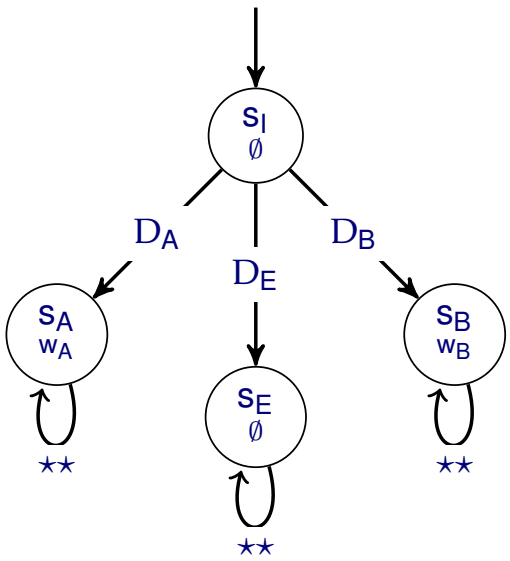
- $\mathrm{Ag} = \{\mathsf{A}: \textit{Alice}, \mathsf{B}: \textit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \textit{Paper}, \mathsf{R}: \textit{Rock}, \mathsf{S}: \textit{Scissor}\}$
- $\mathrm{St} = \{s_I, s_A, s_T, s_B\}$
- s_I: *initial state*
- $\mathrm{AP} = \{w_A: \textit{Alice wins}, w_B: \textit{Bob wins}\}$
- $D_T = \{(P,P),(R,R),(S,S)\}$
- $D_A = \{(P,R),(R,S),(S,P)\}$
- $D_B = \{(R,P),(S,R),(P,S)\}$

Example: paper, rock, and scissor



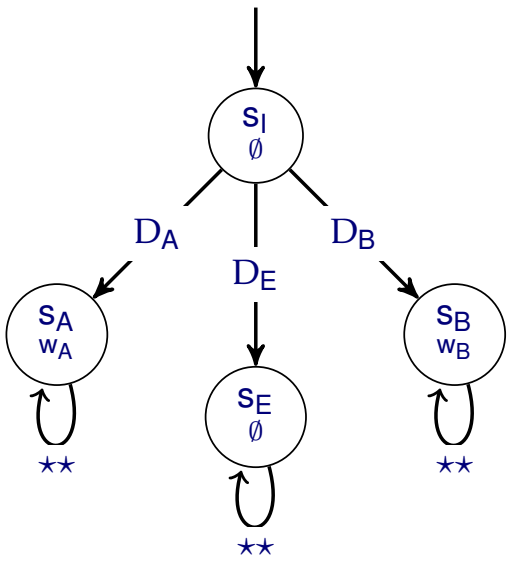
- $\mathrm{Ag} = \{\mathsf{A}: \textit{Alice}, \mathsf{B}: \textit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \textit{Paper}, \mathsf{R}: \textit{Rock}, \mathsf{S}: \textit{Scissor}\}$
- $\mathrm{St} = \{s_I, s_A, s_T, s_B\}$
- s_I: *initial state*
- $\mathrm{AP} = \{w_A: \textit{Alice wins}, w_B: \textit{Bob wins}\}$
- $D_T = \{(\mathsf{P},\mathsf{P}), (\mathsf{R},\mathsf{R}), (\mathsf{S},\mathsf{S})\}$
- $D_A = \{(\mathsf{P},\mathsf{R}), (\mathsf{R},\mathsf{S}), (\mathsf{S},\mathsf{P})\}$
- $D_B = \{(\mathsf{R},\mathsf{P}), (\mathsf{S},\mathsf{R}), (\mathsf{P},\mathsf{S})\}$
- A strategy for A is $\sigma_A(h) = \mathsf{P}, \forall h \in S^+$

Example: paper, rock, and scissor



- $\mathrm{Ag} = \{\mathsf{A}: \textit{Alice}, \mathsf{B}: \textit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \textit{Paper}, \mathsf{R}: \textit{Rock}, \mathsf{S}: \textit{Scissor}\}$
- $\mathrm{St} = \{s_I, s_A, s_T, s_B\}$
- s_I: *initial state*
- $\mathrm{AP} = \{w_A: \textit{Alice wins}, w_B: \textit{Bob wins}\}$
- $D_T = \{(\mathsf{P},\mathsf{P}), (\mathsf{R},\mathsf{R}), (\mathsf{S},\mathsf{S})\}$
- $D_A = \{(\mathsf{P},\mathsf{R}), (\mathsf{R},\mathsf{S}), (\mathsf{S},\mathsf{P})\}$
- $D_B = \{(\mathsf{R},\mathsf{P}), (\mathsf{S},\mathsf{R}), (\mathsf{P},\mathsf{S})\}$
- A strategy for A is $\sigma_A(h) = \mathsf{P}, \forall h \in S^+$
- A strategy for B is $\sigma_B(h) = \mathsf{R}, \forall h \in S^+$

Example: paper, rock, and scissor



- $\mathrm{Ag} = \{\mathsf{A}: \textit{Alice}, \mathsf{B}: \textit{Bob}\}$
- $\mathrm{Ac} = \{\mathsf{P}: \textit{Paper}, \mathsf{R}: \textit{Rock}, \mathsf{S}: \textit{Scissor}\}$
- $\mathrm{St} = \{s_I, s_A, s_T, s_B\}$
- s_I: *initial state*
- $\mathrm{AP} = \{w_A: \textit{Alice wins}, w_B: \textit{Bob wins}\}$
- $D_T = \{(\mathsf{P},\mathsf{P}), (\mathsf{R},\mathsf{R}), (\mathsf{S},\mathsf{S})\}$
- $D_A = \{(\mathsf{P},\mathsf{R}), (\mathsf{R},\mathsf{S}), (\mathsf{S},\mathsf{P})\}$
- $D_B = \{(\mathsf{R},\mathsf{P}), (\mathsf{S},\mathsf{R}), (\mathsf{P},\mathsf{S})\}$
- A strategy for A is $\sigma_A(h) = \mathsf{P}, \forall h \in S^+$
- A strategy for B is $\sigma_B(h) = \mathsf{R}, \forall h \in S^+$
- σ_A and σ_B induce the path $s_I \cdot (s_A)^\omega$

Multi-agent Systems (4)

Visibility of the agents

- Depending on the visibility of the players, we distinguish between:
  - *perfect information games (PI)*;
  - *imperfect information games (II)*.
- In PI case, the players have full knowledge of the evolution of the game, in every moment.
- In II case, the players often have to come to decisions without having all relevant information at hand.

Multi-agent Systems (5)

Specification

- *Internal* $\Rightarrow$ directly over the game structure (Examples: Reachability, Safety, Nash Equilibrium).
- *External* $\Rightarrow$ specific logics under the strategic reasoning (Examples: ATL, Strategic Logic).

Specifications (1)

Alternating-Time Temporal Logic (ATL*)

- ATL* generalizes CTL* by replacing the path quantification with strategy quantification over coalition of agents.
- It uses the strategy modality $\langle\langle A \rangle\rangle$ and $[\![A]\!]$, with A set of agents.
- The strategies are implicit.

Specifications (1)

Alternating-Time Temporal Logic (ATL*)

- ATL* generalizes CTL* by replacing the path quantification with strategy quantification over coalition of agents.
- It uses the strategy modality $\langle\langle A \rangle\rangle$ and $[\![A]\!]$, with A set of agents.
- The strategies are implicit.

Example

$\langle\langle \{\alpha, \beta\} \rangle\rangle \mathsf{G} \neg \mathit{fail}$: the agents α and β have a collective strategy such that the system does not reach the failure state, independently with the strategies of the other agents.

ATL Model checking Complexities

Strategies \ Visibility	**perfect information**	**imperfect information**
memoryless	*PTIME*-complete	P^{NP}-complete
memoryfull	*PTIME*-complete	undecidable

ATL* Model checking complexities

Strategies \ Visibility	**perfect information**	**imperfect information**
memoryless	*PSPACE*-complete	*PSPACE*-complete
memoryfull	2*EXPTIME*-complete	undecidable

SPECIFICATIONS (2)

Strategy Logic (SL)

- SL treats the strategies as first-order objects.
- SL extends LTL with:
  - *strategy quantifiers* $\langle\langle x \rangle\rangle$ and $[\![x]\!]$ read respectively as *"there exists a strategy x"* and *"for all strategies x"*;
  - *agent binding* (a, x) that links an agent a to a strategy associated with a variable x.
- It subsumes other logics for strategic reasoning such as ATL*.
- It allows to represent Nash equilibrium and similar concepts.

SPECIFICATIONS (2)

Strategy Logic (SL)

- SL treats the strategies as first-order objects.
- SL extends LTL with:
  - *strategy quantifiers* $\langle\langle x \rangle\rangle$ and $[\![x]\!]$ read respectively as *"there exists a strategy x"* and *"for all strategies x"*;
  - *agent binding* (a, x) that links an agent a to a strategy associated with a variable x.
- It subsumes other logics for strategic reasoning such as ATL*.
- It allows to represent Nash equilibrium and similar concepts.

Example

$\langle\langle \mathsf{x} \rangle\rangle \langle\langle \mathsf{y} \rangle\rangle [\![\mathsf{z}]\!] (\alpha, \mathsf{x})(\beta, \mathsf{y})(\gamma, \mathsf{z})(\mathsf{G} \neg \mathit{fail})$: α and β have strategies x and y, respectively, that guarantee to not reach a failure state, independently with the strategy z of γ.

SL Model Checking complexities

Strategies \ Visibility	**perfect information**	**imperfect information**
memoryless	*PSPACE*-complete	*PSPACE*-complete
memoryfull	non-elementary	undecidable

Why do we use logics for the strategic reasoning?

Why do we use logics for the strategic reasoning?

Exemple

- 2 vehicles: A et B;

Why do we use logics for the strategic reasoning?

Exemple

- 2 vehicles: A et B;
- B is in front of A;

Why do we use logics for the strategic reasoning?

Exemple

- 2 vehicles: A et B;
- B is in front of A;
- Goal: A wants to pass B (atomic proposition g);

Why do we use logics for the strategic reasoning?

Exemple

- 2 vehicles: A et B;
- B is in front of A;
- Goal: A wants to pass B (atomic proposition g);
- Actions for A: pass (p), not pass (np);

Why do we use logics for the strategic reasoning?

Exemple

- 2 vehicles: A et B;
- B is in front of A;
- Goal: A wants to pass B (atomic proposition g);
- Actions for A: pass (p), not pass (np);
- Actions for B: accelerate (a), not accelerate (na).

Why do we use logics for the strategic reasoning?

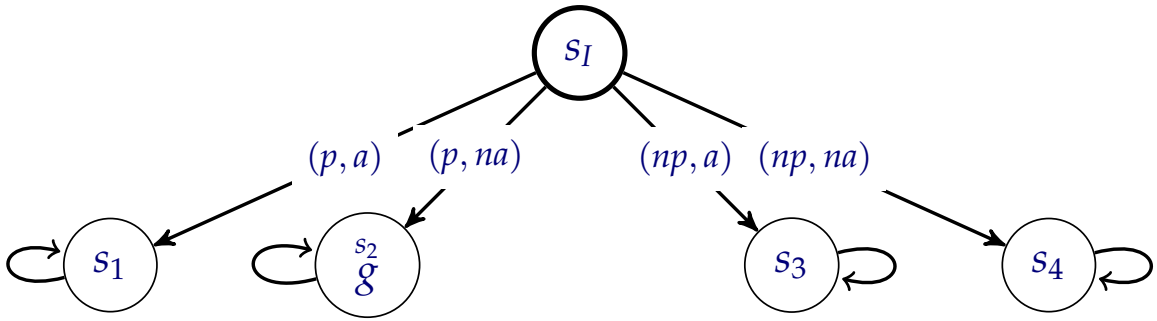
Exemple

- 2 vehicles: A et B;
- B is in front of A;
- Goal: A wants to pass B (atomic proposition g);
- Actions for A: pass (p), not pass (np);
- Actions for B: accelerate (a), not accelerate (na).

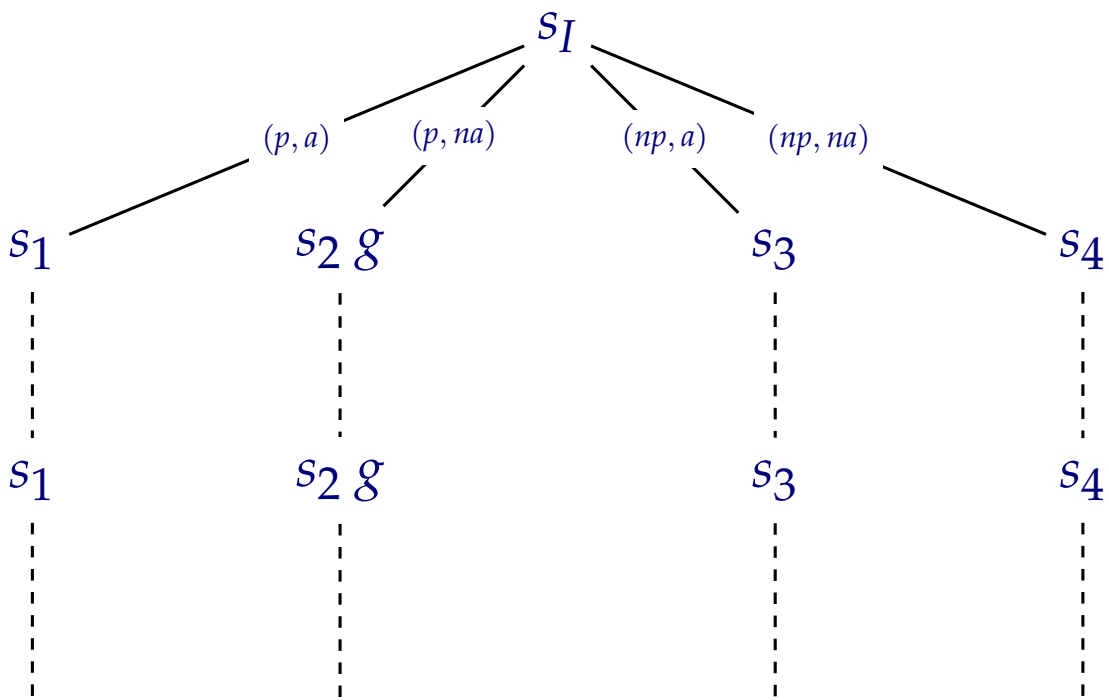
Why do we use logics for the strategic reasoning?

Exemple

- 2 vehicles: A et B;
- B is in front of A;
- Goal: A wants to pass B (atomic proposition g);
- Actions for A: pass (p), not pass (np);
- Actions for B: accelerate (a), not accelerate (na).



Unwinding of the model



Comparison between CTL and ATL (1)

- In the model there is a path where A can pass B but there is no strategy for A to pass B!

Comparison between CTL and ATL (1)

- In the model there is a path where A can pass B but there is no strategy for A to pass B!
- The formula in CTL $EF\, g$ is true.

Comparison between CTL and ATL (1)

- In the model there is a path where A can pass B but there is no strategy for A to pass B!
- The formula in CTL $EF\, g$ is true.
- The formula in ATL $\langle\langle A \rangle\rangle F\, g$ is false.

Comparison between CTL and ATL (2)

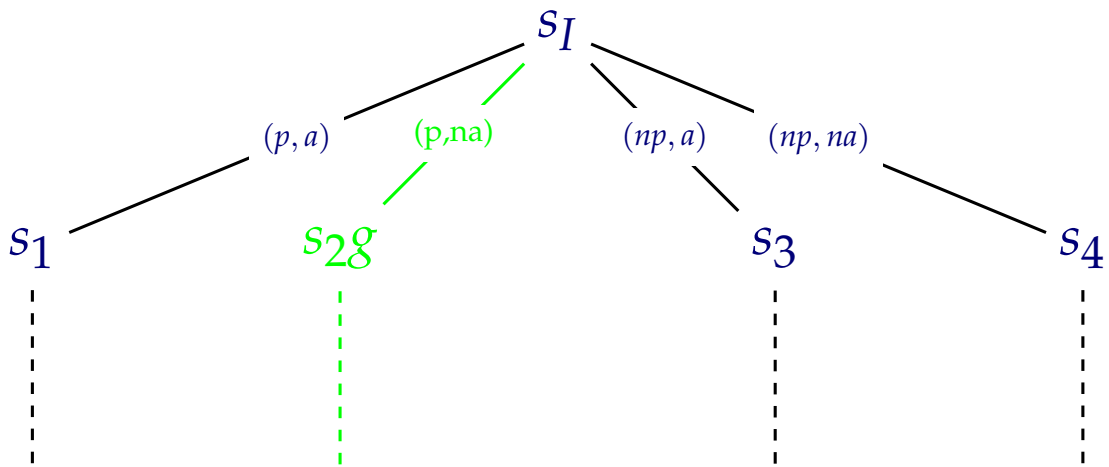
- The formula $EF\, g$ is verified in the path: $s_1, s_2, \ldots$

Comparison between CTL and ATL (2)

- The formula $EF\, g$ is verified in the path: $s_1, s_2, \ldots$

Comparison between CTL and ATL (2)

- The formula $EF\, g$ is verified in the path: $s_I, s_2, \ldots$

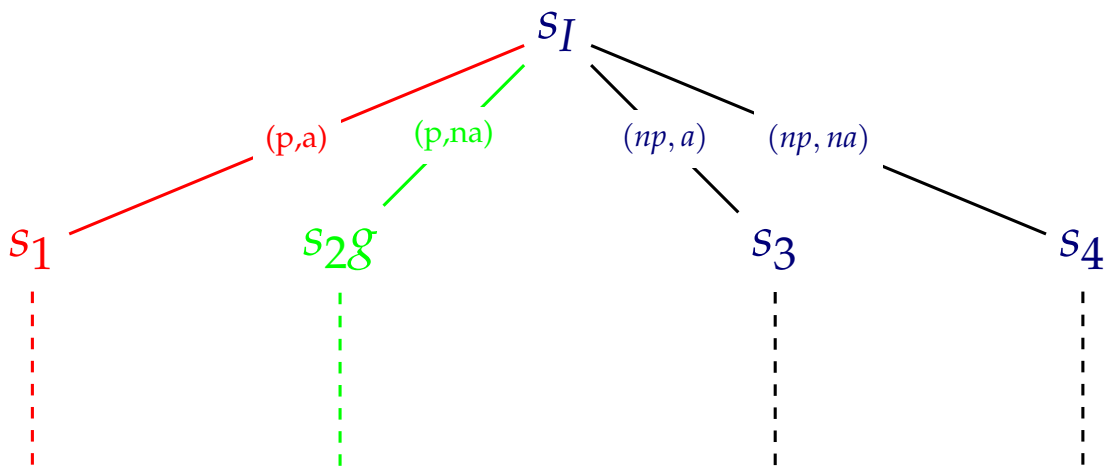


Comparison between CTL and ATL (3)

- The formula in ATL: $\langle\langle A \rangle\rangle F\, g$ is not verified. In fact:
  - if A select the action p then we need to verify the formula in the paths: $s_I, s_1, \ldots$ (false) and $s_I, s_2, \ldots$ (true);

Comparison between CTL and ATL (3)

- The formula in ATL: $\langle\langle A \rangle\rangle F\, g$ is not verified. In fact:
  - if A select the action p then we need to verify the formula in the paths: $s_I, s_1, \ldots$ (false) and $s_I, s_2, \ldots$ (true);



Comparison between CTL and ATL (4)

Equivalence between CTL and ATL

- $E\psi \equiv \langle\langle Ag \rangle\rangle \psi$
- $A\psi \equiv \langle\langle \emptyset \rangle\rangle \psi$

with ψ without quantifications.

Comparison between CTL and ATL (4)

Equivalence between CTL and ATL

- $E\psi \equiv \langle\langle Ag \rangle\rangle \psi$
- $A\psi \equiv \langle\langle \emptyset \rangle\rangle \psi$

with ψ without quantifications.

Summary

With CTL we cannot verify the properties on a single agent and on coalitions (subset) of agents.

Research topics

Imperfect information + full memory
vs
decidability

Approximate Memoryfull Strategies

Our contribution

A partial model checking technique under perfect recall and imperfect information based on ATL^* 3-valued semantics.

High level idea

1. A notion of bounded memory in the context of ATL^* and ATL specifications in multi-agent systems with imperfect information.
2. A novel 3-valued semantics for ATL^* and ATL under bounded memory and imperfect information.
3. We proved that 3-valued semantics constitutes an approximation w.r.t. the traditional 2-valued semantics.
4. A sound, even if partial, algorithm for the model checking in 2-valued semantics via its approximation as 3-valued semantics.

Approximate Imperfect Information

Our contribution

A partial model checking technique under perfect recall and imperfect information based on ATL^* specifications.

High level idea

Given a model M with imperfect information and an ATL^* formula φ:

1. We build an abstract model M_a with perfect information from M;
2. We check whether φ is defined in M_a; if that is the case then we are done, otherwise step (3);
3. We try to construct a refined model M_r from M_a;
4. We check whether φ is defined in M_r; if that is the case then we are done, otherwise step (3).

Natural Strategies

Natural Strategies

A natural strategy s_a for agent a is a list of condition-action rules

$$(cond,\ act)$$

such that:

- $cond$ is:
  - Memoryless $\Rightarrow$ a boolean combination of propositions;
  - With memory $\Rightarrow$ a regular expression over boolean formulas.
- act is an available action:
  - Memoryless $\Rightarrow$ such that $q \models cond$;
  - With memory $\Rightarrow$ such that $h \models cond$.

NatATL

Variant of ATL

NatATL updates ATL by replacing the quantification as follows:

$\langle\langle \mathrm{A} \rangle\rangle^{\leq k} \varphi$: *there is a natural strategy less or equal than* k *that satisfies* φ

NatATL

Variant of ATL

NatATL updates ATL by replacing the quantification as follows:

$\langle\langle A \rangle\rangle^{\leq k} \varphi$: *there is a natural strategy less or equal than* k *that satisfies* φ

NatATL Model Checking Complexities

Strategies \ Limit k	**fixed or bounded**	**parameter**
memoryless	*PTIME*	P^{NP}
with memory	P^{NP}	*PSPACE*

NatATL

Variant of ATL

NatATL updates ATL by replacing the quantification as follows:

$\langle\langle A \rangle\rangle^{\leq k} \varphi$: *there is a natural strategy less or equal than* k *that satisfies* φ

NatATL Model Checking Complexities

Strategies \ Limit k	**fixed or bounded**	**parameter**
memoryless	*PTIME*	P^{NP}
with memory	P^{NP}	*PSPACE*

The above results also hold in the case of imperfect information!

FIND DECIDABLE AND USEFUL FRAGMENTS IN STRATEGIC LOGICS

Strategy Logic with Simple Goals

Tradeoff: expressivity vs. complexity

- We want to get more expressivity than in ATL without sacrificing model checking complexity (at least not much).

Strategy Logic with Simple Goals

Tradeoff: expressivity vs. complexity

- We want to get more expressivity than in ATL without sacrificing model checking complexity (at least not much).
- Formulas are composed of 3 subsequent parts:

quantifiers | strategy binders | simple temporal goal

Strategy Logic with Simple Goals

Tradeoff: expressivity vs. complexity

- We want to get more expressivity than in ATL without sacrificing model checking complexity (at least not much).
- Formulas are composed of 3 subsequent parts:

quantifiers | strategy binders | simple temporal goal

Theorem (Expressivity)

SL[SG] is strictly more expressive and distinguishing than ATL.

Strategy Logic with Simple Goals

Tradeoff: expressivity vs. complexity

- We want to get more expressivity than in ATL without sacrificing model checking complexity (at least not much).
- Formulas are composed of 3 subsequent parts:

quantifiers | strategy binders | simple temporal goal

Theorem (Expressivity)

SL[SG] is strictly more expressive and distinguishing than ATL.

Theorem (Verification Complexity)

Model checking SL[SG] is PTIME-complete wrt the size of the model and the length of the formula $\rightsquigarrow$ same complexity class as for ATL!

Thank you for the attention! ☺

Appendix

Imperfect information + full memory
vs
decidability

Other ways to find decidability

Approximation via Runtime Verification

- Runtime verification is a computing system analysis based on extracting information from a running system.
- Runtime verification avoids the complexity of the model checking by analyzing only a few execution traces.
- Challenge: use Runtime Verification to approximate model checking for multi-agent systems.

Other ways to find decidability

Approximation via Runtime Verification

- Runtime verification is a computing system analysis based on extracting information from a running system.
- Runtime verification avoids the complexity of the model checking by analyzing only a few execution traces.
- Challenge: use Runtime Verification to approximate model checking for multi-agent systems.

Approximation via CTL

- Find the decidability via approximation of the specification.
- To be decidable: characterize what we can verify via ATL and what we need to abstract via CTL.

Natural Strategies

Natural Strategies and some applications

Voting Protocols

- Implementation of *Prêt à Voter* protocol.
- Used for remote voting and voting of handicapped persons in the Australian state of Victoria elections in November 2014.
- Examples of properties:
  - $\langle\langle Voter \rangle\rangle^{\leq k} F\ end\_voting\_procedure$
  - $\langle\langle Coercer, Voter \rangle\rangle^{\leq k} F\ coercer\_has\_receipt$

Natural Strategies and some applications

Voting Protocols

- Implementation of *Prêt à Voter* protocol.
- Used for remote voting and voting of handicapped persons in the Australian state of Victoria elections in November 2014.
- Examples of properties:
  - $\langle\langle Voter \rangle\rangle^{\leq k} F\ end\_voting\_procedure$
  - $\langle\langle Coercer, Voter \rangle\rangle^{\leq k} F\ coercer\_has\_receipt$

Auctions

- Natural strategies with memory in repeated auctions.
- Quantitative semantics to handle payoffs.

From Attack graphs to Formal methods

From Attack graphs to Formal methods

- Challenges involved:

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;
  - bounded resources;

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;
  - bounded resources;
  - continuous time;

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;
  - bounded resources;
  - continuous time;
  - stochastic setting.

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;
  - bounded resources;
  - continuous time;
  - stochastic setting.
- Find a way to model attack graphs as games:

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;
  - bounded resources;
  - continuous time;
  - stochastic setting.
- Find a way to model attack graphs as games:
  - 2-player vs multi-player;

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;
  - bounded resources;
  - continuous time;
  - stochastic setting.
- Find a way to model attack graphs as games:
  - 2-player vs multi-player;
  - turn-based vs concurrent;

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;
  - bounded resources;
  - continuous time;
  - stochastic setting.
- Find a way to model attack graphs as games:
  - 2-player vs multi-player;
  - turn-based vs concurrent;
- Find a way to define properties:

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;
  - bounded resources;
  - continuous time;
  - stochastic setting.
- Find a way to model attack graphs as games:
  - 2-player vs multi-player;
  - turn-based vs concurrent;
- Find a way to define properties:
  - internal vs external specifications;

From Attack graphs to Formal methods

- Challenges involved:
  - imperfect information;
  - bounded resources;
  - continuous time;
  - stochastic setting.
- Find a way to model attack graphs as games:
  - 2-player vs multi-player;
  - turn-based vs concurrent;
- Find a way to define properties:
  - internal vs external specifications;
  - qualitative vs quantitative objectives.

Formal Verification via SMT solvers

ATL Model Checking via SMT solvers

- We introduce a fragment of ATL, called ATL-live, that is suitable to express liveness properties in MAS.
- We introduce a sound translation from ATL-live to First-order Logic (FOL).
- We show how the universal model checking problem for ATL-live can be reduced to semantic entailment in FOL.
- We prove that ATL-live is maximal, i.e. if any other connective is added, non-FOL reasoning techniques would be required.
- These results are meant to be a first step towards the application of FOL reasoners to model check strategic abilities expressed in ATL.