Reasoning about Strategies under Partial Observability and Fairness Constraints

Simon Busard UCLouvain, Belgium
Charles Pecheur UCLouvain, Belgium
Hongyang Qu University of Sheffield, UK
Franco Raimondi Middlesex University, UK

CFV Seminar, Brussels, November 29, 2013

Running Example: A simple card game [1]

Three cards: A, K, Q
(A wins over K, K over Q, Q over A);

A player, a dealer.

[1] W. Jamroga, W. van der Hoek. *Agents that Know How to Play.* (2004)

Running Example: A simple card game [1]

Three cards: A, K, Q
(A wins over K, K over Q, Q over A);

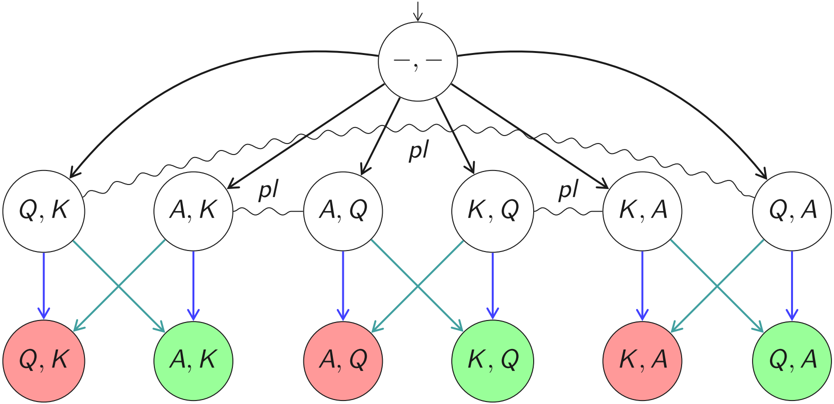
A player, a dealer.

The dealer gives a card and keeps one;

the player can change his card
with the one on table.

[1] W. Jamroga, W. van der Hoek. *Agents that Know How to Play.* (2004)

Running Example: A simple card game



Running Example: A simple card game [1]

Three cards: A, K, Q
(A wins over K, K over Q, Q over A);

A player, a dealer.

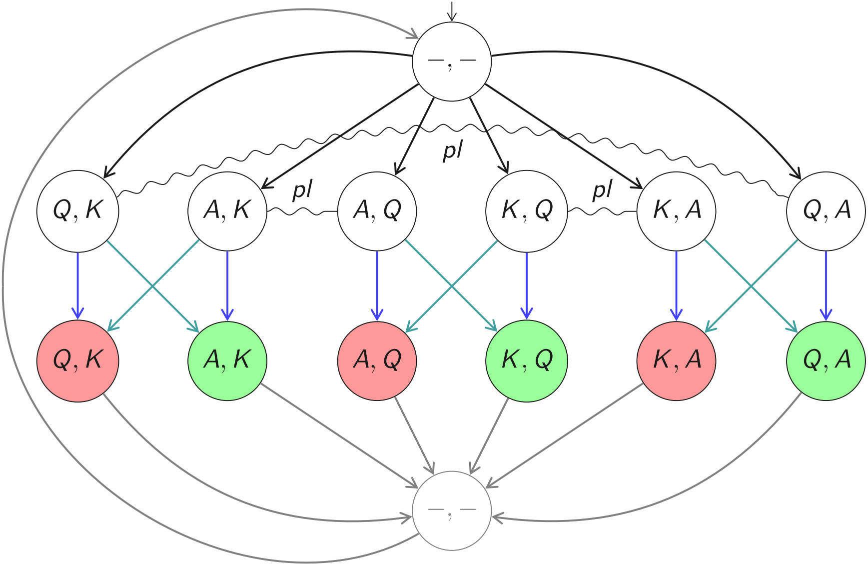
The dealer gives a card and keeps one;

the player can change his card
with the one on table.

Variant: the player can play infinitely.

[1] W. Jamroga, W. van der Hoek. *Agents that Know How to Play.* (2004)

Running Example: A simple card game



Reasoning about strategies

Model checking problem:
does the player have a strategy to win?

Reasoning about strategies

Model checking problem:
does the player have a strategy to win?

$\Rightarrow$ it depends on the semantics!

Reasoning about strategies

Model checking problem:
does the player have a strategy to win?

Under *ATL*, we consider all strategies.
The player has a strategy to win,
even if he cannot play it:
e.g., in $\langle A, K \rangle$, keep the card; in $\langle A, Q \rangle$, exchange it.

Reasoning about strategies

Model checking problem:
does the player have a strategy to win?

ATL: yes.

Under ATL_{ir}, we consider only **memoryless uniform** strategies.
There is no uniform strategy to win,
because the player cannot distinguish, e.g., $\langle A, K \rangle$ and $\langle A, Q \rangle$.

Reasoning about strategies

Model checking problem:
does the player have a strategy to win?

ATL: yes.

ATL_{ir}: no.

If we consider ATL_{ir} with a **fair dealer** and an **infinite play**,
the player can eventually win:
just use one uniform strategy, the right pair will finally come.

Reasoning about strategies

Model checking problem:
does the player have a strategy to win?

ATL: yes.

ATL_{ir}: no.

ATL_{ir} + fair dealer and infinite play: yes.

$\Rightarrow$ $ATLK_{irF}$: branching time, knowledge, **memoryless uniform** strategies and unconditional **fairness constraints**.

Outline

ATL, reasoning about **strategies** of the agents. [2]

Syntax: Strategic modalities: $\langle\Gamma\rangle \mathbf{X}\ \phi$, $[\Gamma]\mathbf{G}\ \phi$, $\langle\Gamma\rangle[\phi_1\ \mathbf{U}\ \phi_2]$, etc.

Semantics: A state s satisfies $\langle\Gamma\rangle\ \pi$ iff there exists a set of **strategies** for agents in Γ such that **all enforced paths satisfy** π.

[2] Alur et al. *Alternating-time temporal logic.* (2002)

ATL, reasoning about **strategies** of the agents. [2]

Syntax: Strategic modalities: $\langle\Gamma\rangle\mathbf{X}\ \phi$, $[\Gamma]\mathbf{G}\ \phi$, $\langle\Gamma\rangle\mathbf{[}\phi_1\ \mathbf{U}\ \phi_2\mathbf{]}$, etc.

Semantics: A state *s* satisfies $\langle\Gamma\rangle\ \pi$ iff there exists a set of **strategies** for agents in Γ such that **all enforced paths satisfy** π.

Model checking:

$$eval_{ATL}([\Gamma]\mathbf{G}\ \phi) = \nu Z.eval_{ATL}(\phi) \cap Pre_{[\Gamma]}(Z)$$

where $Pre_{[\Gamma]}(Z)$ is the set of states from which Γ cannot avoid to reach Z in one step.

[2] Alur et al. *Alternating-time temporal logic.* (2002)

ATL_{ir}, memoryless uniform strategies [3]

Only **memoryless uniform** strategies:

$$f_a : S \to Act_a \text{ such that } s \sim_a s' \implies f_a(s) = f_a(s')$$

Semantics: A state s satisfies $\langle\Gamma\rangle\ \pi$ iff there exists a set of **memoryless uniform** strategies for agents in Γ such that all paths enforced **from all** $s' \sim_\Gamma s$ satisfy π.

[3] Schobbens. *Alternating-time logic with imperfect recall*. (2004).

FairCTL: time and fairness constraints [4]

Add a set of **fairness constraints** $FC \subseteq 2^S$ to the model;
$\Rightarrow$ unconditional state-based fairness.

Only **fair paths** are considered:
$s \models$ **E** π iff there exists a **fair** path from s satisfying π;
$s \models$ **A** π iff all **fair** paths from s satisfy π.

[4] Clarke, Grumberg, Peled. *Model checking*. (2000).

FairCTL: time and fairness constraints [4]

Add a set of **fairness constraints** $FC \subseteq 2^S$ to the model;
$\Rightarrow$ unconditional state-based fairness.

Only **fair paths** are considered:
$s \models \mathbf{E}\ \pi$ iff there exists a **fair** path from s satisfying π;
$s \models \mathbf{A}\ \pi$ iff all **fair** paths from s satisfy π.

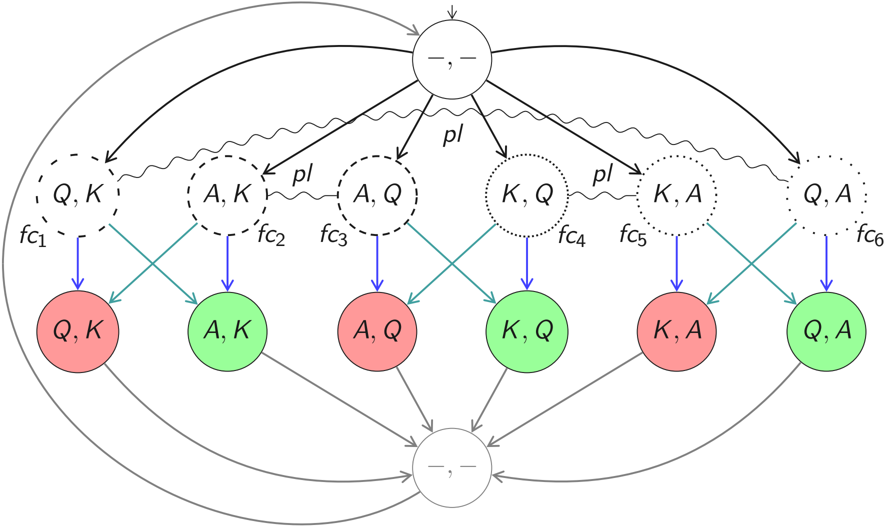
Model checking:

$$eval_{FCTL}(\mathbf{EG}\ \phi) = \nu Z.\Phi \cap \bigcap_{fc \in FC} Pre(\mu Y.(Z \cap fc) \cup (\Phi \cap Pre(Y)))$$

where $Pre(Z)$ is the set of states having a successor in Z
and $\Phi = eval_{FCTL}(\phi)$.

[4] Clarke, Grumberg, Peled. *Model checking*. (2000).

Adding fairness constraints to the card game



Outline

$ATLK_{irF} = FairCTL$, knowledge and ATL_{ir} with fairness

Syntax: CTL (**EX**, **AG**, etc.), knowledge ($\mathbf{K}_{ag}$, $\mathbf{C}_{g}$, etc.) and strategies ($\langle\Gamma\rangle\mathbf{F}$, $[\Gamma]\mathbf{U}$, etc.)

Semantics: A state s satisfies $\langle\Gamma\rangle\ \pi$ iff there exists a **memoryless uniform** strategy for Γ such that all **fair** paths enforced **from all** $s' \sim_\Gamma s$ satisfy π.

To model check $ATLK_{irF}$, we defined $ATLK_{IrF}$ and its model checking

$ATLK_{IrF} = FairCTL$ + knowledge + *ATL* with fairness

$ATLK_{IrF}$ **semantics:** A state s satisfies $\langle\Gamma\rangle\ \pi$ iff there exists a memoryless strategy (not necessarily uniform) for Γ such that all **fair** paths enforced (from s only) satisfy π.

To model check $ATLK_{irF}$, we defined $ATLK_{IrF}$ and its model checking

$ATLK_{IrF} = FairCTL$ + knowledge + ATL with fairness

$ATLK_{IrF}$ **semantics:** A state s satisfies $\langle\Gamma\rangle\ \pi$ iff there exists a memoryless strategy (not necessarily uniform) for Γ such that all **fair** paths enforced (from s only) satisfy π.

$ATLK_{IrF}$ **model checking:**

$$eval_{IrF}([\Gamma]\mathbf{G}\ \phi) = \nu Z.\Phi \cap \bigcap_{fc \in FC} Pre_{[\Gamma]}(\mu Y.(Z \cap fc) \cup (\Phi \cap Pre_{[\Gamma]}(Y)))$$

where $\Phi = eval_{IrF}(\phi)$.

$ATLK_{irF}$ model checking

A state s satisfies $\langle\Gamma\rangle\ \pi$ iff there exists a **memoryless uniform strategy** for Γ which allows Γ to enforce π in all **states indistinguishable from** s, considering only **fair paths**.

$ATLK_{irF}$ model checking

A state s satisfies $\langle\Gamma\rangle\ \pi$ iff there exists a **memoryless uniform strategy** for Γ which allows Γ to enforce π in all **states indistinguishable from** s, considering only **fair paths**.

To get all the states satisfying $\langle\Gamma\rangle\ \pi$:

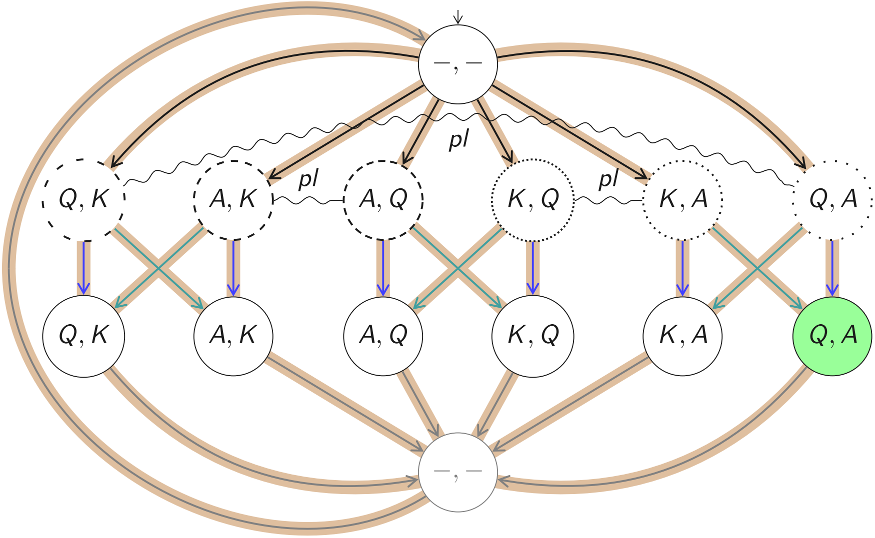
1. List all the memoryless uniform strategies;
2. Use $ATLK_{IrF}$ model checking to get states satisfying the property **in this strategy**;
3. Then restrict to set of undistinguishable states.

$ATLK_{irF}$ model checking: *Split* algorithm

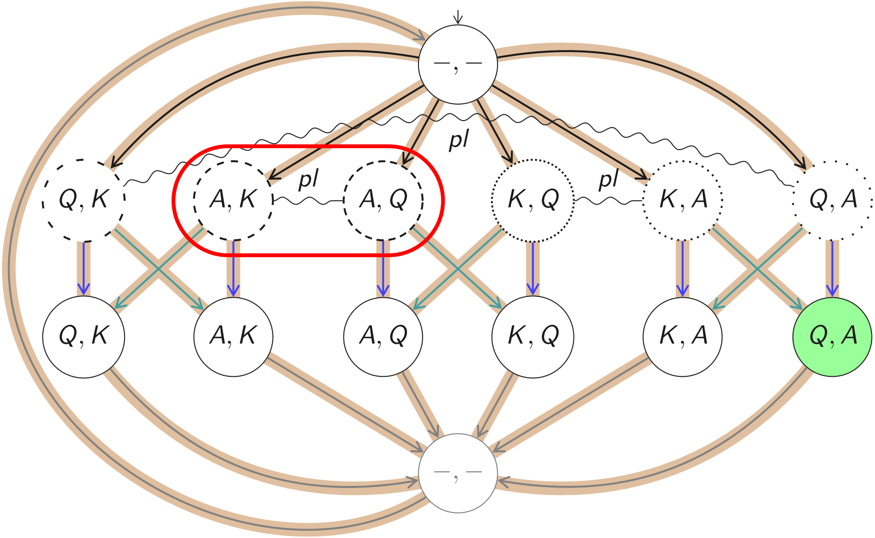
Split the state/action pairs into memoryless uniform strategies.

1. Get all conflicting equivalence classes;
2. If there are none, the set is itself a memoryless uniform strategy.
3. Otherwise, choose a conflicting equivalence class;
4. Split it;
5. and recursively call *Split* on the rest.

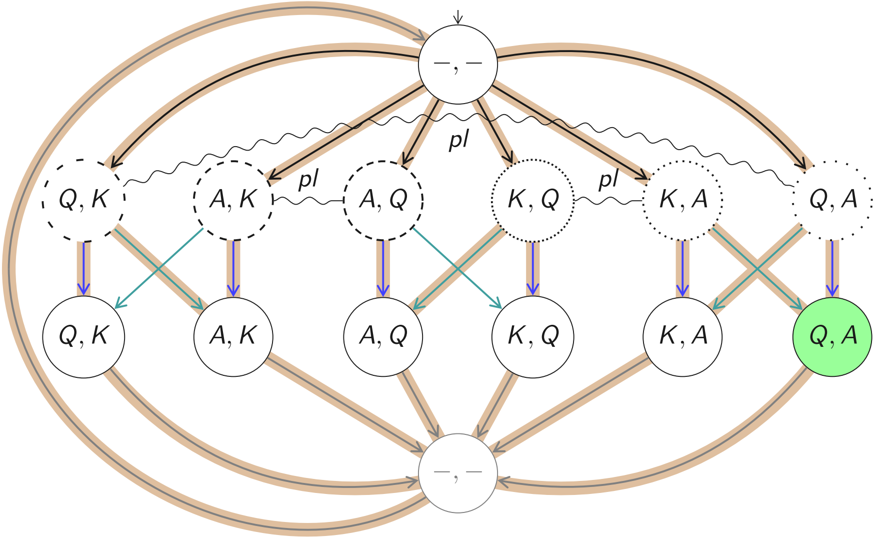
$ATLK_{irF}$ model checking example: $\langle player \rangle \mathbf{F}$ $win \wedge \langle Q, * \rangle$



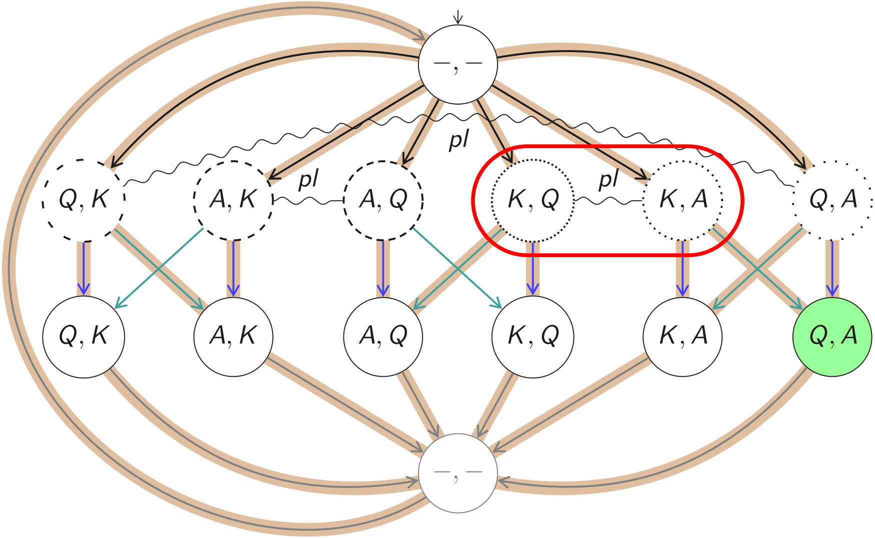
$ATLK_{irF}$ model checking example: $\langle player \rangle \mathbf{F}$ $win \wedge \langle Q, * \rangle$



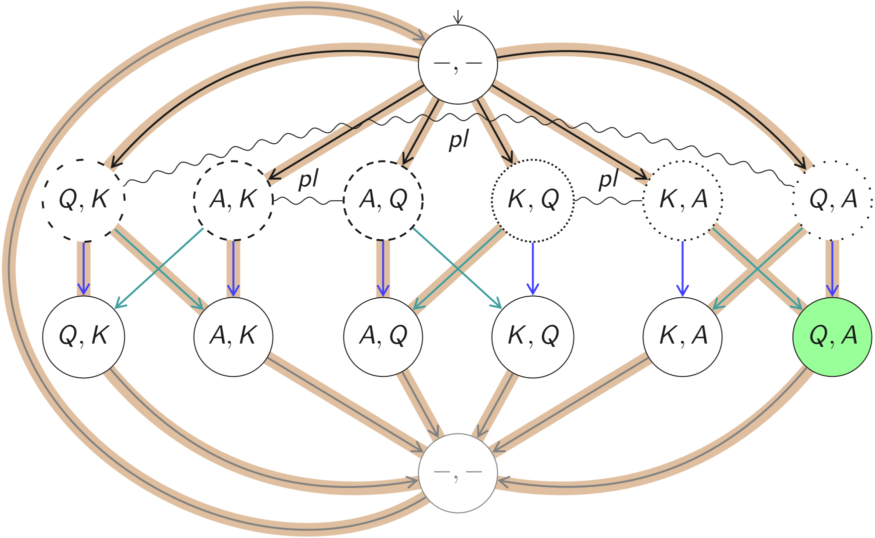
$ATLK_{irF}$ model checking example: $\langle player \rangle \mathbf{F}$ $win \wedge \langle Q, * \rangle$



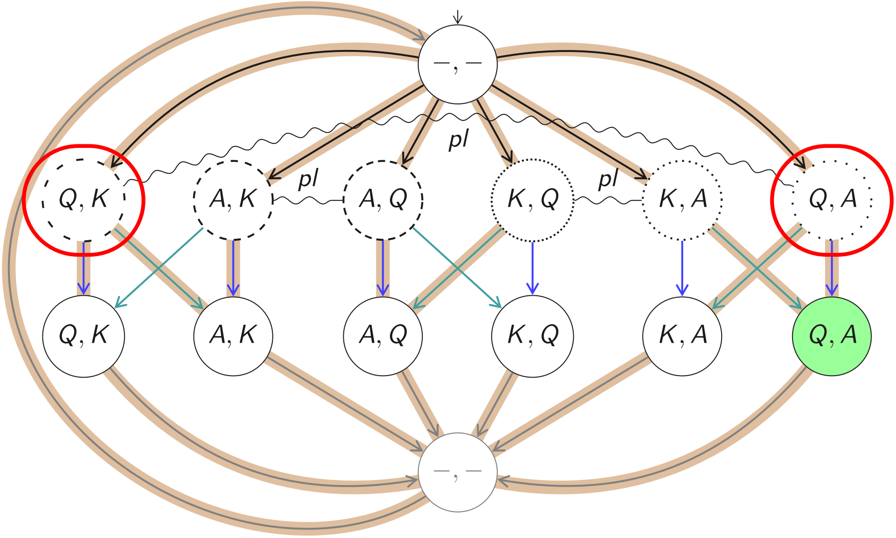
$ATLK_{irF}$ model checking example: $\langle player \rangle \mathbf{F}$ $win \wedge \langle Q, * \rangle$



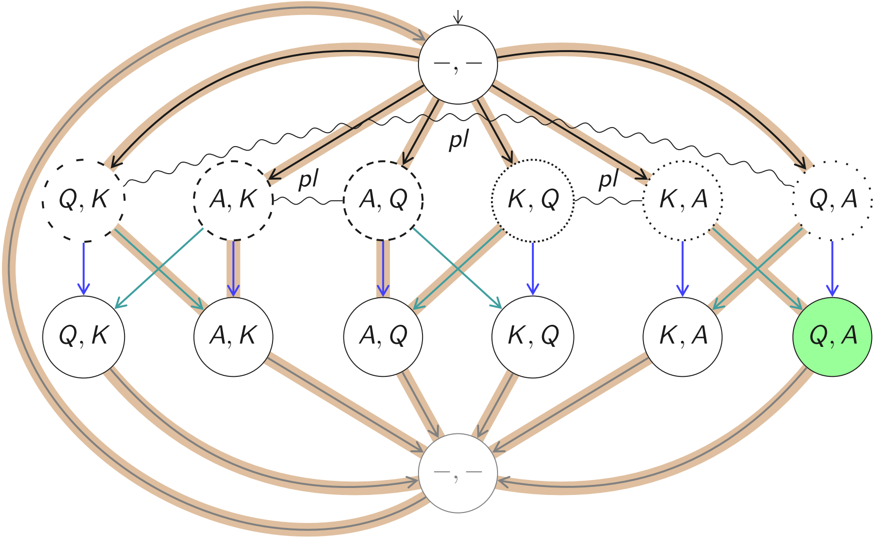
$ATLK_{irF}$ model checking example: $\langle player \rangle \mathbf{F}$ $win \wedge \langle Q, * \rangle$



$ATLK_{irF}$ model checking example: $\langle player \rangle \mathbf{F}$ $win \wedge \langle Q, * \rangle$



$ATLK_{irF}$ model checking example: $\langle player \rangle \mathbf{F}$ $win \wedge \langle Q, * \rangle$



Apply $ATLK_{IrF}$ model checking $\Rightarrow$ all states satisfy the property;

$\Rightarrow$ the strategy is winning for all.

Improving the algorithm with filtering

$$s \not\models_{IrF} \langle \Gamma \rangle\ \pi \implies s \not\models_{irF} \langle \Gamma \rangle\ \pi$$

Improving the algorithm with filtering

$$s \not\models_{IrF} \langle\Gamma\rangle\ \pi \implies s \not\models_{irF} \langle\Gamma\rangle\ \pi$$

$\Rightarrow$ Can only consider the states satisfying $\langle\Gamma\rangle\ \psi$ under $ATLK_{IrF}$;

Improving the algorithm with filtering

$$s \not\models_{IrF} \langle\Gamma\rangle\ \pi \implies s \not\models_{irF} \langle\Gamma\rangle\ \pi$$

$\Rightarrow$ Can only consider the states satisfying $\langle\Gamma\rangle\ \psi$ under $ATLK_{IrF}$;

$\Rightarrow$ Can only consider actions that allow Γ to win under $ATLK_{IrF}$;

Improving the algorithm with filtering

$$s \not\models_{IrF} \langle\Gamma\rangle\ \pi \implies s \not\models_{irF} \langle\Gamma\rangle\ \pi$$

$\Rightarrow$ Can only consider the states satisfying $\langle\Gamma\rangle\ \psi$ under $ATLK_{IrF}$;

$\Rightarrow$ Can only consider actions that allow Γ to win under $ATLK_{IrF}$;

$\Rightarrow$ Can alternate between filtering states and actions and splitting equivalence classes into non-conflicting subsets.

Outline

Complexity

$ATLK_{IrF}$ is in **P**: the proposed algorithm is polynomial.

Complexity

$ATLK_{IrF}$ is in **P**: the proposed algorithm is polynomial.

$ATLK_{irF}$ subsumes ATL_{ir} (in the case of two agents)
$\Rightarrow ATLK_{irF}$ is Δ_2^P-hard;

Split algorithm is in **NP**
$\Rightarrow ATLK_{irF}$ is Δ_2^P-complete.

Vacuous Strategies

If Γ have a strategy **producing no fair path**,
Γ can win any objective;
in particular, **unsatisfiable formulas** like $\langle\Gamma\rangle\mathbf{F}\ \mathit{false}$.

Vacuous Strategies

If Γ have a strategy **producing no fair path**,
Γ can win any objective;
in particular, **unsatisfiable formulas** like $\langle\Gamma\rangle\mathbf{F}\ \mathit{false}$.

Solutions

- consider only groups of agents that **cannot prevent fairness**;
- change the semantics to only consider **strategies producing at least one fair path**;
- ...

Knowledge relations

A state s satisfies $\langle\Gamma\rangle\ \pi$ under $ATLK_{irF}$ iff there exists a memoryless **uniform** strategy for Γ which allows Γ to enforce π in all **states indistinguishable from** s, considering only fair paths.

Distributed knowledge used for both relations
$\Rightarrow$ Γ is considered as a unique agent
$\Rightarrow$ the simplest form.

Knowledge relations

A state s satisfies $\langle\Gamma\rangle\ \pi$ under $ATLK_{irF}$ iff there exists a memoryless **uniform** strategy for Γ which allows Γ to enforce π in all **states indistinguishable from** s, considering only fair paths.

Distributed knowledge used for both relations
⇒ Γ is considered as a unique agent
⇒ the simplest form.

We could consider other knowledge relations:

- one knowledge relation per agent of Γ (used by ATL_{ir} for uniformity);
- group knowledge;
- common knowledge.

Outline

Tool

Prototype implemented with PyNuSMV,
a Python framework based on NuSMV [5].

Several tested implementations.

[5] S. Busard, C. Pecheur. *PyNuSMV: NuSMV as a Python Library.* (2013)

Implementation and tests

Basic algorithm:

1. splitting the entire system into uniform strategies;
2. checking each strategy.

$\Rightarrow$ explodes quickly, huge number of strategies
(huge number of combinations of choices for actions).

Implementation and tests

Improved algorithm:

Alternate between filtering out losing states and actions and splitting one conflicting equivalence class.

$\Rightarrow$ slower explosion,
especially when only a few states satisfy the property.

Implementation and tests

Mixing both:

1. filtering out losing states and actions;
2. splitting the rest into uniform strategies;
3. checking each strategy.

$\Rightarrow$ best solution:
most of the filtering work is performed by the first filtering.

More improvements (current work)

1. Partial strategies: check only strategies "that matter".

2. Implementation optimizations:
   - early termination: stop when a strategy is found for all states;
   - caching: remember states satisfying sub-formulas through different strategies;
   - ...

Outline

Conclusion

$ATLK_{irF}$: branching time, knowledge and strategies under partial observability and (unconditional state-based) fairness constraints.

(Symbolic) model checking algorithm based on $ATLK_{IrF}$ **model checking** and **splitting the graph** into memoryless uniform strategies.

Conclusion

$ATLK_{irF}$: branching time, knowledge and strategies under partial observability and (unconditional state-based) fairness constraints.

(Symbolic) model checking algorithm based on $ATLK_{IrF}$ **model checking** and **splitting the graph** into memoryless uniform strategies.

$\Rightarrow$ Still needs some improvements.

$\Rightarrow$ Work on counter-examples (controller synthesis,...)

Thank you.

Questions?