Reasoning about Strategies under Partial Observability and Fairness Constraints

Simon Busard  
Charles Pecheur  
Hongyang Qu  
Franco Raimondi  

UCLouvain, Belgium  
UCLouvain, Belgium  
University of Sheffield, UK  
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.

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.

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.

Running Example: A simple card game

\[
\begin{align*}
Q, K & \quad A, K & \quad A, Q & \quad K, Q & \quad K, A & \quad Q, A \\
Q, K & \quad A, K & \quad A, Q & \quad K, Q & \quad K, A & \quad Q, A \\
\end{align*}
\]

\[pl\]
Reasoning about strategies

Model checking problem:

*does the player have a strategy to win?*
Reasoning about strategies

Model checking problem: 

\textbf{does the player have a strategy to win?}

\implies \text{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.

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

Strategies, Temporal Logics and Fairness

Strategies under Partial Observability and Fairness Constraints

Discussions

Implementation

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

**Syntax:** Strategic modalities: \( \langle \Gamma \rangle X \phi, [\Gamma] G \phi, \langle \Gamma \rangle [\phi_1 U \phi_2], \) etc.

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

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

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

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

**Model checking:**

$$eval_{ATL}(\lbrack \Gamma \rbrack G \phi) = \nu Z. eval_{ATL}(\phi) \cap Pre[\Gamma](Z)$$

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

$\text{ATL}_{ir}$, memoryless uniform strategies [3]

Only **memoryless uniform** strategies:

$$f_a : S \rightarrow \text{Act}_a \text{ such that } s \sim_a s' \implies f_a(s) = f_a(s')$$

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

FairCTL: time and fairness constraints [4]

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

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

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

**Model checking:**

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

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

Adding fairness constraints to the card game
Outline

Strategies, Temporal Logics and Fairness

Strategies under Partial Observability and Fairness Constraints

Discussions

Implementation

Conclusion
\( \text{ATL}K_{irF} = \text{FairCTL} \), knowledge and \( \text{ATL}_{ir} \) with fairness

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

**Semantics:** A state \( s \) satisfies \( \langle \Gamma \rangle \pi \) iff there exists a **memoryless uniform** strategy for \( \Gamma \) such that all **fair paths** enforced from all \( s' \sim_{\Gamma} s \) satisfy \( \pi \).
To model check $ATLK_{irF}$,
we defined $ATLK_{irF}$ and its model checking

\[
ATLK_{irF} = FairCTL + \text{knowledge} + ATL \text{ with fairness}
\]

$ATLK_{irF}$ semantics: A state $s$ satisfies $\langle \Gamma \rangle \pi$ iff there exists a memoryless strategy (not necessarily uniform) for $\Gamma$ such that all fair paths enforced (from $s$ only) satisfy $\pi$. 
To model check $ATLK_{irF}$, we defined $ATLK_{irF}$ and its model checking

$$ATLK_{irF} = FairCTL + \text{knowledge} + ATL \text{ with fairness}$$

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

$ATLK_{irF}$ model checking:

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

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 $\Gamma$ which allows $\Gamma$ to enforce $\pi$ in all **states indistinguishable from** $s$, considering only **fair paths**.
A state $s$ satisfies $\langle \Gamma \rangle \pi$ iff there exists a memoryless uniform strategy for $\Gamma$ which allows $\Gamma$ to enforce $\pi$ 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\textsubscript{irF} model checking example: $\langle \text{player} \rangle F \text{win} \land \langle Q, * \rangle$
$\text{ATL}K_{irF}$ model checking example: $\langle \text{player} \rangle F \ \text{win} \land \langle Q, * \rangle$
ATLK_{irF} model checking example: $\langle \text{player} \rangle F \ (\text{win} \land \langle Q, \ast \rangle)$
Apply ATLK\textsubscript{irF} model checking example: $\langle \text{player} \rangle F \text{ win} \land \langle Q, * \rangle$
\( \text{ATLK}_{irF} \) model checking example: \( \langle \text{player} \rangle F \text{ win} \land \langle Q, * \rangle \)
ATLK\textsubscript{irF} model checking example: \( \langle \text{player} \rangle F \) \( \text{win} \wedge \langle Q, * \rangle \)
$\text{ATLK}_{irF}$ model checking example: $\langle \text{player} \rangle F \, \text{win} \land \langle Q, * \rangle$

Apply $\text{ATLK}_{irF}$ 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

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

\[
\implies \text{Can only consider the states satisfying } \langle \Gamma \rangle \psi \text{ under } ATLK_{IrF};
\]
Improving the algorithm with filtering

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

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

\[ \Rightarrow \text{Can only consider actions that allow } \Gamma \text{ to win under } ATLK_{IrF}; \]
Improving the algorithm with filtering

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

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

\[ \Rightarrow \text{Can only consider actions that allow } \Gamma \text{ to win under } ATLK_{IrF}; \]

\[ \Rightarrow \text{Can alternate between filtering states and actions and splitting equivalence classes into non-conflicting subsets.} \]
Outline

Strategies, Temporal Logics and Fairness

Strategies under Partial Observability and Fairness Constraints

Discussions

Implementation

Conclusion
ATLK_{irF} is in P: the proposed algorithm is polynomial.
ATLK$_{irF}$ is in $\textbf{P}$: the proposed algorithm is polynomial.

ATLK$_{irF}$ subsumes ATL$_{ir}$ (in the case of two agents)
$\Rightarrow$ ATLK$_{irF}$ is $\Delta^P_2$-hard;

*Split* algorithm is in $\textbf{NP}$
$\Rightarrow$ ATLK$_{irF}$ is $\Delta^P_2$-complete.
Vacuous Strategies

If $\Gamma$ have a strategy producing no fair path, $\Gamma$ can win any objective; in particular, unsatisfiable formulas like $\langle \Gamma \rangle F \ false$. 
Vacuous Strategies

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

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 $\Gamma$ which allows $\Gamma$ to enforce $\pi$ in all states indistinguishable from $s$, considering only fair paths.

Distributed knowledge used for both relations
$\Rightarrow$ $\Gamma$ 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 $\Gamma$ which allows $\Gamma$ to enforce $\pi$ in all states indistinguishable from $s$, considering only fair paths.

Distributed knowledge used for both relations

$\Rightarrow \Gamma$ is considered as a unique agent

$\Rightarrow$ the simplest form.

We could consider other knowledge relations:

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

Strategies, Temporal Logics and Fairness

Strategies under Partial Observability and Fairness Constraints

Discussions

Implementation

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

Several tested implementations.

Implementation and tests

Basic algorithm:

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

⇒ 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.

⇒ 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.

⇒ 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

Strategies, Temporal Logics and Fairness

Strategies under Partial Observability and Fairness Constraints

Discussions

Implementation

Conclusion
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?