Satisfiability of ATL with strategy contexts

François Laroussinie and Nicolas Markey
LIAFA and LSV

Highlights of Logic, Games and Automata
Paris, 18-21 September 2013
Outline of the presentation

1. Temporal logics for games: ATL and extensions
   - expressing properties of complex interacting systems
   - extensions to non-zero-sum games

2. From ATL with strategy contexts to QCTL
   - QCTL is CTL with propositional quantification
   - strategies encoded as propositions on the computation tree

3. Satisfiability of ATL with strategy contexts
   - QCTL satisfiability is decidable, but...
   - $\text{ATL}_{sc}$ satisfiability is not, except for turn-based games
Outline of the presentation

1 Temporal logics for games: ATL and extensions
   - expressing properties of complex interacting systems
   - extensions to non-zero-sum games

2 From ATL with strategy contexts to QCTL
   - QCTL is CTL with propositional quantification
   - strategies encoded as propositions on the computation tree

3 Satisfiability of ATL with strategy contexts
   - QCTL satisfiability is decidable, but...
   - ATL\textsubscript{sc} satisfiability is not, except for turn-based games
Reasoning about multi-agent systems

Concurrent games

A concurrent game is made of:
- a transition system;
- a set of agents (or players);
- a table indicating the transition to be taken given the actions of the players.
Reasoning about multi-agent systems

Concurrent games
A concurrent game is made of
- a transition system;
- a set of agents (or players);
- a table indicating the transition to be taken given the actions of the players.

Turn-based games
A turn-based game is a game where only one agent plays at a time.
Reasoning about open systems

**Strategies**

A strategy for a given player is a function telling what to play depending on what has happened previously.
A strategy for a given player is a function telling what to play depending on what has happened previously.

Strategy for player □:
alternately go to ⊙ and ◦.
Reasoning about open systems

**Strategies**

A *strategy* for a given player is a function telling what to play depending on what has happened previously.

Strategy for player □:
alternately go to □ and □.
Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

\[ \langle A \rangle \varphi \text{ expresses that } A \text{ has a strategy to enforce } \varphi. \]
Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle A \rangle \varphi$ expresses that $A$ has a strategy to enforce $\varphi$. 

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

\[ \langle A \rangle \varphi \] expresses that \( A \) has a strategy to enforce \( \varphi \).

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

\( \langle A \rangle \varphi \) expresses that \( A \) has a strategy to enforce \( \varphi \).

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle A \rangle \varphi$ expresses that $A$ has a strategy to enforce $\varphi$.

Another semantics: ATL with strategy contexts [BDLM09]

Another semantics: ATL with strategy contexts [BDLM09]

\[\langle \Box \rangle G(\langle \square \rangle F \circ)\]

consider the following strategy of Player \(\bigcirc\): “always go to \(\square\)”;
Another semantics: ATL with strategy contexts [BDLM09]

\[ \langle \Diamond \rangle \textbf{G} (\langle \Box \rangle \textbf{F} \Diamond) \]

consider the following strategy of Player \( \bullet \): "always go to \( \square \);"
Another semantics: ATL with strategy contexts [BDLM09]

\[ \langle \Diamond \rangle G(\langle \Box \rangle F \circ) \]

- consider the following strategy of Player \( \bigcirc \): “always go to \( \Box \)”; 
- in the remaining tree, Player \( \Box \) can always enforce a visit to \( \bigcirc \).

What $\text{ATL}_{sc}$ can express

- All $\text{ATL}^*$ properties:
What $\text{ATL}_{sc}$ can express

- All $\text{ATL}^*$ properties:
- Client-server interactions for accessing a shared resource:

$$\langle \text{Server} \rangle \ G \bigwedge \left[ \bigwedge_{c \in \text{Clients}} \langle c \rangle F \text{access}_c \right.
\left. \quad \neg \bigwedge_{c \neq c'} \text{access}_c \land \text{access}_{c'} \right]$$
What ATL sc can express

- All ATL* properties:

- **Client-server interactions** for accessing a shared resource:

  \[
  \langle \text{Server} \rangle \ \text{G} \begin{cases} \\
  \bigwedge_{c \in \text{Clients}} \langle \cdot \rangle \ F \ access_c \\
  \neg \bigwedge_{c \neq c'} access_c \land access_{c'}
  \end{cases}
  \]

- Existence of **Nash equilibria**:

  \[
  \langle \cdot A_1, \ldots, A_n \rangle \ \bigwedge_i \left( \langle \cdot A_i \rangle \varphi A_i \Rightarrow \varphi A_i \right)
  \]

- Existence of **dominating strategy**:

  \[
  \langle \cdot A \rangle [B] \ (\neg \varphi \Rightarrow [A] \neg \varphi)
  \]
Outline of the presentation

1. Temporal logics for games: ATL and extensions
   - expressing properties of complex interacting systems
   - extensions to non-zero-sum games

2. From ATL with strategy contexts to QCTL
   - QCTL is CTL with propositional quantification
   - strategies encoded as propositions on the computation tree

3. Satisfiability of ATL with strategy contexts
   - QCTL satisfiability is decidable, but...
   - ATL_{sc} satisfiability is not, except for turn-based games
Quantified CTL [Kup95,Fre01]

QCTL extends CTL with propositional quantifiers

\[ \exists p. \varphi \] means that there exists a labelling of the model with \( p \) under which \( \varphi \) holds.


Quantified CTL [Kup95,Fre01]

QCTL extends CTL with propositional quantifiers

$\exists p. \varphi$ means that there exists a labelling of the model with $p$ under which $\varphi$ holds.

$\mathbf{E}F(p \land \lozenge) \Rightarrow A\mathbf{G}(\lozenge \Rightarrow p)$


Quantified CTL [Kup95,Fre01]

QCTL extends CTL with **propositional quantifiers**

\[ \exists p. \varphi \] means that **there exists a labelling** of the model with \( p \) under which \( \varphi \) holds.

\[ \bigcirc EF \bigcirc \land \forall p. \left[ EF(p \land \bigcirc) \Rightarrow AG(\bigcirc \Rightarrow p) \right] \equiv uniq(\bigcirc) \]


Quantified CTL [Kup95,Fre01]

QCTL extends CTL with **propositional quantifiers**

\[ \exists p. \varphi \] means that **there exists a labelling** of the model with \( p \) under which \( \varphi \) holds.

\[ \text{EF} \land \forall p. \left[ \text{EF}(p \land \lozenge) \Rightarrow \text{AG}(\lozenge \Rightarrow p) \right] \equiv \text{uniq}(\lozenge) \]

\[ \sim \) true if we label the Kripke structure; \]

\[ \sim \) false if we label the computation tree;


Translating $\text{ATL}_{sc}$ into QCTL

- player $A$ has moves $m_1^A$, ..., $m_n^A$;
- from the transition table, we can compute the set $\text{Next}(\bullet, A, m_i^A)$ of states that can be reached from $\bullet$ when player $A$ plays $m_i^A$. 

Translating $\text{ATL}_{sc}$ into QCTL

- player $A$ has moves $m_1^A$, ..., $m_n^A$;
- from the transition table, we can compute the set $\text{Next}(q, A, m_i^A)$ of states that can be reached from $q$ when player $A$ plays $m_i^A$.

$\langle A \rangle \varphi$ can be encoded as follows:

$$\exists m_1^A. \exists m_2^A \ldots \exists m_n^A.$$

- this corresponds to a strategy: $A \ G(m_i^A \Leftrightarrow \bigwedge \neg m_j^A)$;
- the outcomes all satisfy $\varphi$:

$$A\left[ G(q \land m_i^A \Rightarrow X \text{Next}(q, A, m_i^A)) \Rightarrow \varphi \right].$$

Translating $\text{ATL}_{sc}$ into $\text{QCTL}$

- player $A$ has moves $m_1^A, \ldots, m_n^A$;
- from the transition table, we can compute the set $\text{Next}(\bigcirc, A, m_i^A)$ of states that can be reached from $\bigcirc$ when player $A$ plays $m_i^A$.

**Theorem (DLM12)**

$\text{QCTL}$ model checking is decidable (in the tree semantics).

**Corollary**

$\text{ATL}_{sc}$ model checking is decidable.

Outline of the presentation

1. Temporal logics for games: ATL and extensions
   - expressing properties of complex interacting systems
   - extensions to non-zero-sum games

2. From ATL with strategy contexts to QCTL
   - QCTL is CTL with propositional quantification
   - strategies encoded as propositions on the computation tree

3. Satisfiability of ATL with strategy contexts
   - QCTL satisfiability is decidable, but...
   - $\text{ATL}_{sc}$ satisfiability is not, except for turn-based games
What about satisfiability?

**Theorem (LM13a)**

*QCTL satisfiability is decidable.*

---

What about satisfiability?

Theorem (LM13a)

\[ \text{QCTL satisfiability is decidable.} \]

But

Theorem (TW12)

\[ \text{ATL}_{sc} \text{ satisfiability is undecidable.} \]

What about satisfiability?

**Theorem (LM13a)**

QCTL satisfiability is decidable.

**Theorem (TW12)**

$\text{ATL}_{sc}$ satisfiability is undecidable.

Why?

The translation from $\text{ATL}_{sc}$ to QCTL assumes that the game structure is fixed!

Satisfiability for turn-based games

Theorem (LM13b)

When restricted to turn-based games, $\text{ATL}_{sc}$ satisfiability is decidable.

Player $\square$ has moves $\bigcirc$, $\bigcirc$, and $\bigcirc$. A strategy can be encoded by marking some of the nodes of the tree with proposition $\text{mov}_A$.

$\langle \cdot A \cdot \rangle \varphi$ can be encoded as follows:

\[ \exists \text{mov}_A. \]

- it corresponds to a strategy: $A G(\text{turn}_A \Rightarrow E X_1 \text{mov}_A)$;
- the outcomes all satisfy $\varphi$: $A [G(\text{turn}_A \land X \text{mov}_A) \Rightarrow \varphi]$.

Restricting to memoryless strategies

Memoryless strategies
One move in each state of the structure (not of its execution tree).

Our reduction to QCTL is still valid!
(but we now label the structure)
Restricting to memoryless strategies

Memoryless strategies
One move in each state of the structure (not of its execution tree).

Our reduction to QCTL is still valid!
(but we now label the structure)

Theorem

Model checking $\text{ATL}_{sc}$ with only memoryless quantification is PSPACE-complete.
Restricting to memoryless strategies

**Memoryless strategies**

One move in each state of the structure (not of its execution tree).

Our reduction to QCTL is still valid!
(but we now label the structure)

**Theorem**

*Model checking ATL\(_{sc}\) with only memoryless quantification is PSPACE-complete.*

However:

**Theorem**

*Satisfiability of ATL\(_{sc}\) with memoryless quantification is undecidable (even on turn-based structures).*
What about Strategy Logic [CHP07, MMV10]?

**Strategy logic**

Explicit quantification over strategies + strategy assignment

Strategy logic can also be translated into QCTL.

**Theorem**

- *Strategy-logic satisfiability is decidable when restricted to turn-based games.*
- *Memoryless strategy-logic satisfiability is undecidable.*

Conclusions and future works

Conclusions

- $\text{ATL}_{sc}$ is a very powerful logic for reasoning about games.
- $\text{QCTL}$ is a nice tool to understand such logics.
- Satisfiability is undecidable, except when looking for turn-based games (or when fixing the set of moves).
- Restricting to memoryless strategies does not help (actually, it is even worse).

Future directions

- Defining interesting (expressive yet tractable) fragments of those logics;
- Obtaining practicable algorithms.
- Considering randomised strategies.
Conclusions and future works

Conclusions

- $\text{ATL}_{sc}$ is a very powerful logic for reasoning about games.
- $\text{QCTL}$ is a nice tool to understand such logics.
- Satisfiability is undecidable, except when looking for turn-based games (or when fixing the set of moves).
- Restricting to memoryless strategies does not help (actually, it is even worse).

Future directions

- Defining interesting (expressive yet tractable) fragments of those logics;
- Obtaining practicable algorithms.
- Considering randomised strategies.