Temporal logics for multi-agent systems

Nicolas Markey
LSV – ENS Cachan

(based on joint works with Thomas Brihaye,
Arnaud Da Costa-Lopes, François Laroussinie)

Dagstuhl, February 2, 2015
Outline of the presentation

1 Basics of CTL and ATL
   - expressing properties of reactive systems
   - efficient verification algorithms

2 Temporal logics for multi-agent systems
   - specifying properties of complex interacting systems
   - expressive power of ATL$_{sc}$
   - translation into Quantified CTL (QCTL)
   - algorithms for ATL$_{sc}$
   - Strategy Logic

3 Conclusions and future works
Outline of the presentation

1 Basics of CTL and ATL
   - expressing properties of reactive systems
   - efficient verification algorithms

2 Temporal logics for multi-agent systems
   - specifying properties of complex interacting systems
   - expressive power of ATL$_{sc}$
   - translation into Quantified CTL (QCTL)
   - algorithms for ATL$_{sc}$
   - Strategy Logic

3 Conclusions and future works
Computation-Tree Logic (CTL)

- atomic propositions: \(\bigcirc\), \(\bigcirc\), ...

- boolean combinators: \(\neg\ \phi\), \(\phi \lor \psi\), \(\phi \land \psi\), ...

- temporal modalities: \(X\ \phi\) "next \(\phi\)", \(\phi \mathcal{U} \psi\) "\(\phi\) until \(\psi\)"

- path quantifiers: \(E\ \phi\) "eventually \(\phi\)", \(A\ \phi\) "always \(\phi\)"
Computation-Tree Logic (CTL)

- atomic propositions: \( \bigcirc, \bigcirc, \ldots \)
- boolean combinators: \( \neg \varphi, \varphi \lor \psi, \varphi \land \psi, \ldots \)
Computation-Tree Logic (CTL)

- atomic propositions: $\bigcirc$, $\bigcirc$, ...
- boolean combinators: $\neg \varphi$, $\varphi \lor \psi$, $\varphi \land \psi$, ...
- temporal modalities:
  
  \[ X \varphi \quad \varphi \mathbin{U} \psi \quad \varphi \mathbin{U} \psi \]

  "next $\varphi$"

  "$\varphi$ until $\psi$"
Computation-Tree Logic (CTL)

- atomic propositions: \(\bigcirc, \bigcirc, \ldots\)
- boolean combinators: \(\neg \varphi, \varphi \lor \psi, \varphi \land \psi, \ldots\)
- temporal modalities:
  - \(X \varphi\) \(\bigcirc \rightarrow \varphi \rightarrow \varnothing \rightarrow \varnothing \rightarrow \varnothing \rightarrow \varnothing \ldots\) “next \(\varphi\)"
  - \(\varphi \mathop{U} \psi\) \(\varnothing \rightarrow \varnothing \rightarrow \psi \rightarrow \varnothing \rightarrow \varnothing \rightarrow \varnothing \ldots\) “\(\varphi\) until \(\psi\)"
  - true \(U \varphi \equiv F \varphi\) \(\bigcirc \rightarrow \varnothing \rightarrow \varnothing \rightarrow \varnothing \rightarrow \varnothing \rightarrow \varnothing \ldots\) “eventually \(\varphi\)"
  - \(\neg F \neg \varphi \equiv G \varphi\) \(\varnothing \rightarrow \bigcirc \rightarrow \varnothing \rightarrow \varnothing \rightarrow \varnothing \rightarrow \varnothing \ldots\) “always \(\varphi\)"
Computation-Tree Logic (CTL)

- atomic propositions: \(\circ, \bigcirc, \ldots\)
- boolean combinators: \(\neg \varphi, \varphi \lor \psi, \varphi \land \psi, \ldots\)
- temporal modalities:
  - \(X \varphi\), \(\varphi \mathrel{U} \psi\), \(\text{true} \mathrel{U} \varphi \equiv F \varphi\), \(\neg F \neg \varphi \equiv G \varphi\)
    - “next \(\varphi\)”
    - “\(\varphi\) until \(\psi\)”
    - “eventually \(\varphi\)”
    - “always \(\varphi\)”
- path quantifiers:
  - \(E \varphi\), \(A \varphi\)
Examples of CTL formulas

In CTL, each temporal modality is in the immediate scope of a path quantifier.
Examples of CTL formulas

In CTL, each temporal modality is in the immediate scope of a path quantifier.

\[ EF \quad \text{is reachable} \]
Examples of CTL formulas

In CTL, each temporal modality is in the immediate scope of a path quantifier.

\[ EF \] is reachable
Examples of CTL formulas

In CTL, each temporal modality is in the immediate scope of a path quantifier.

\[ EG(\neg \bullet \land EF \bullet) \] there is a path along which \( \bullet \) is always reachable, but never reached
Examples of CTL formulas

In CTL, each temporal modality is in the immediate scope of a path quantifier.

$$\text{EG}(\neg \square p \land \text{EF} p)$$

there is a path along which $p$ is always reachable, but never reached
Examples of CTL formulas

In CTL, each temporal modality is in the immediate scope of a path quantifier.

$$\text{EG}(\neg \bigcirc \land \text{EF} \bigcirc_p)$$

there is a path along which $\bigcirc$ is always reachable, but never reached
Examples of CTL formulas

In CTL, each temporal modality is in the immediate scope of a path quantifier.

Theorem ([CE81, QS82])

**CTL model checking is PTIME-complete.**
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.

\[
\begin{array}{c|c|c}
q_0 & q_1 & q_2 \\
\hline
q_0 & \rightarrow & q_1 \\
q_1 & \rightarrow & q_0 \\
q_2 & \rightarrow & q_2 \\
\end{array}
\]

player 1

player 2
Concurrent games

A concurrent game is made of

- a transition system;

\[ q_0, q_1, q_2 \]
Concurrent games

A concurrent game is made of

- a transition system;
- a set of agents (or players);

\[
q_0 \xrightarrow{a} q_1 \\
q_0 \xrightarrow{b} q_2
\]
Reasoning about open 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 open 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.
Reasoning about open systems

** Strategies **

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

** Example **

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.

**Example**

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

ATL extends CTL with **strategy quantifiers**

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

Temporal logics for games: ATL

ATL extends CTL with **strategy quantifiers**

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

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

- Model checking ATL is \( \text{PTIME} \)-complete.

Temporal logics for games: ATL

ATL extends CTL with **strategy quantifiers**

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

Temporal logics for games: ATL

ATL extends CTL with strategy quantifiers

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

Temporal logics for games: ATL

ATL extends CTL with strategy quantifiers

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

Temporal logics for games: ATL

ATL extends CTL with **strategy quantifiers**

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

Theorem ([AHK02]) Model checking ATL is $\text{PTIME}$-complete.

Temporal logics for games: ATL

ATL extends CTL with strategy quantifiers

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

Temporal logics for games: ATL

ATL extends CTL with strategy quantifiers

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

\[\langle\langle p \rangle\rangle \equiv \langle\langle \Box \rangle\rangle (\langle\langle \Box \rangle\rangle \Box p) \equiv \langle\langle \Diamond \rangle\rangle G p\]

Theorem ([AHK02])

Model checking ATL is PTIME-complete.

Outline of the presentation

1 Basics of CTL and ATL
   - expressing properties of reactive systems
   - efficient verification algorithms

2 Temporal logics for multi-agent systems
   - specifying properties of complex interacting systems
   - expressive power of ATL_{sc}
   - translation into Quantified CTL (QCTL)
   - algorithms for ATL_{sc}
   - Strategy Logic

3 Conclusions and future works
Consider the following strategy of Player $G$: "always go to $F$"; in the remaining tree, Player $G$ can always enforce a visit to $F$. 

consider the following strategy of Player $\Diamond$: “always go to $\Box$;
consider the following strategy of Player $\ ● $: “always go to $\ □ $”;

$ ⟨\Diamond ⟩ \text{G}(⟨\Box ⟩ \text{F} \ ● )$
consider the following strategy of Player \( \bigcirc \): “always go to \( \square \);”

in the remaining tree, Player \( \square \) can always enforce a visit to \( \bigcirc \).
ATL with strategy contexts

Definition

$\text{ATL}_{sc}$ has two new strategy quantifiers: $\langle \cdot A \cdot \rangle \varphi$ and $\langle \neg A \neg \rangle \varphi$.

- $\langle \cdot A \cdot \rangle$ is similar to $\langle A \rangle$ but **assigns** the corresponding strategy to $A$ for evaluating $\varphi$;
ATL with strategy contexts

Definition

$\text{ATL}_{sc}$ has two new strategy quantifiers: $\langle A \rangle \varphi$ and $\langle -A \rangle \varphi$.

- $\langle A \rangle$ is similar to $\langle A \rangle$ but assigns the corresponding strategy to $A$ for evaluating $\varphi$;
- $\langle -A \rangle$ drops the assigned strategies for $A$.
ATL with strategy contexts

Definition

\( \text{ATL}_{sc} \) has two new strategy quantifiers: \( \langle \cdot A \cdot \rangle \varphi \) and \( \langle -A- \rangle \varphi \).

- \( \langle \cdot A \cdot \rangle \) is similar to \( \langle \langle A \rangle \rangle \) but **assigns** the corresponding strategy to \( A \) for evaluating \( \varphi \);
- \( \langle -A- \rangle \) **drops** the assigned strategies for \( A \).
- \( \lbrack A \rbrack \) is **dual** to \( \langle \cdot A \cdot \rangle \):

\[
\lbrack A \rbrack \varphi \equiv \neg \langle \cdot A \cdot \rangle \neg \varphi
\]

\( \lbrack A \rbrack \varphi \) which states that any strategy for \( A \) has an outcome along which \( \varphi \) holds.
What $\text{ATL}_{sc}$ can express

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

$$
\langle \cdot \text{Server} \rangle \ G \left[ \bigwedge_{c \in \text{Clients}} \langle \cdot c \rangle F \text{access}_c \land \\
\neg \bigwedge_{c \neq c'} \text{access}_c \land \text{access}_{c'} \right]
$$

[Diagram of client-server interactions with a server and multiple clients accessing a shared resource.]
What $\text{ATL}_{sc}$ can express

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

  $\langle \cdot \text{Server} \cdot \rangle \mathbf{G} \left[ \bigwedge_{c \in \text{Clients}} \langle \cdot c \cdot \rangle \mathbf{F} \text{access}_c \land \neg \bigwedge_{c \neq c'} \text{access}_c \land \text{access}_{c'} \right]$

- **Existence of Nash equilibria**:

  $\langle \cdot A_1, \ldots, A_n \cdot \rangle \bigwedge_i \left( \langle \cdot A_i \cdot \rangle \varphi_{A_i} \Rightarrow \varphi_{A_i} \right)$
What $\text{ATL}_{sc}$ can express

- **Client-server interactions** for accessing a shared resource:
  \[
  \langle \text{Server} \rangle \ G \left[ \bigwedge_{c \in \text{Clients}} \langle c \rangle \ F \ \text{access}_c \land \neg \bigwedge_{c \neq c'} \text{access}_c \land \text{access}_{c'} \right]
  \]

- **Existence of Nash equilibria**:
  \[
  \langle A_1, \ldots, A_n \rangle \ \bigwedge_i \left( \langle A_i \rangle \varphi_{A_i} \Rightarrow \varphi_{A_i} \right)
  \]

- **Existence of dominating strategy**:
  \[
  \langle A \rangle [B] (\neg \varphi \Rightarrow [A] \neg \varphi)
  \]
More expressiveness results

Theorem

- $\textit{\text{ATL}}_{\text{sc}}$ is strictly more expressive than $\textit{\text{ATL}}$,
- The operator $\langle \neg A \rangle$ does not add expressive power,
- $\textit{\text{ATL}}_{\text{sc}}$ is as expressive as $\textit{\text{ATL}}^*_{\text{sc}}$. 
More expressiveness results

**Theorem**
- \( \text{ATL}_{sc} \) is strictly more expressive than ATL,
- The operator \( \langle -A \rangle \) does not add expressive power,
- \( \text{ATL}_{sc} \) is as expressive as \( \text{ATL}_{sc}^* \).

**Proof**

\[
\langle A \rangle \varphi \equiv \langle \text{Agt} \rangle \langle A \rangle \hat{\varphi}
\]
More expressiveness results

Theorem

- **ATL\(_s\)c** is strictly more expressive than **ATL**, 
- The operator \(\langle -A\rangle\) does not add expressive power, 
- **ATL\(_s\)c** is as expressive as **ATL\(_s\)*c**.

Proof

\(\langle 1.\rangle (\langle 2.\rangle \textbf{X} a \land \langle 2.\rangle \textbf{X} b)\) is only true in the second game. But **ATL** cannot distinguish between these two games.
More expressiveness results

**Theorem**

- $\text{ATL}_{sc}$ is strictly more expressive than $\text{ATL}$,
- The operator $\langle -A\rangle$ does not add expressive power,
- $\text{ATL}_{sc}$ is as expressive as $\text{ATL}_{sc}^*$.

**Proof**

Replace implicit quantification with explicit one:

$$\langle \cdot 1 \cdot \rangle \varphi \equiv \langle \cdot 1 \cdot \rangle [\text{Agt} \setminus \{1\}] \langle \cdot \emptyset \cdot \rangle \hat{\varphi}$$

we can always assume that the context is full.
More expressiveness results

**Theorem**

- $\text{ATL}_{sc}$ is strictly more expressive than $\text{ATL}$,
- The operator $\langle -A - \rangle$ does not add expressive power,
- $\text{ATL}_{sc}$ is as expressive as $\text{ATL}_{sc}^*$.

**Proof**

Replace implicit quantification with explicit one:

$$\langle 1 \cdot \rangle \varphi \equiv \langle 1 \cdot \rangle [\text{Agt} \setminus \{1\}] \langle 0 \cdot \rangle \hat{\varphi}$$

$\sim$ we can always assume that the context is full.

- $\langle -A - \rangle \varphi$ is then equivalent to $[A:] \langle 0 \cdot \rangle \varphi$;
- $\langle 0 \cdot \rangle$ can be inserted between two temporal modalities.
Outline of the presentation

1 Basics of CTL and ATL
   - expressing properties of reactive systems
   - efficient verification algorithms

2 Temporal logics for multi-agent systems
   - specifying properties of complex interacting systems
   - expressive power of $\text{ATL}_{sc}$
   - translation into Quantified CTL (QCTL)
   - algorithms for $\text{ATL}_{sc}$
   - Strategy Logic

3 Conclusions and future works
Quantified CTL

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

QCTL extends CTL with propositional quantifiers

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

\[
\begin{align*}
&\text{EF} \bigcirc \land \forall p. \left[ \text{EF}(p \land \bigcirc) \Rightarrow \text{AG}(\bigcirc \Rightarrow p) \right]
\end{align*}
\]


Quantified CTL

QCTL extends CTL with propositional quantifiers

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

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

Quantified CTL

QCTL extends CTL with propositional quantifiers

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

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

\begin{itemize}
  \item true if we label the Kripke structure;
  \item false if we label the computation tree;
\end{itemize}

Semantics of QCTL

- structure semantics:

\[ \models_s \exists p. \varphi \iff \models \varphi \]
Semantics of QCTL

- structure semantics:

- tree semantics:
Expressiveness of QCTL

QCTL can “count”:

\[
\begin{align*}
E X_1 \varphi & \equiv E X \varphi \land \forall p. \ [E X(p \land \varphi) \Rightarrow A X(\varphi \Rightarrow p)] \\
E X_2 \varphi & \equiv \exists q. \ [E X_1(\varphi \land q) \land E X_1(\varphi \land \neg q)]
\end{align*}
\]

Expressiveness of QCTL

- QCTL can “count”:

\[
\begin{align*}
EX_1 \varphi & \equiv EX \varphi \land \forall p. \ [EX(p \land \varphi) \Rightarrow AX(\varphi \Rightarrow p)] \\
EX_2 \varphi & \equiv \exists q. \ [EX_1(\varphi \land q) \land EX_1(\varphi \land \neg q)]
\end{align*}
\]

- QCTL can express (least or greatest) fixpoints:

\[
\begin{align*}
\mu T. \varphi(T) & \equiv \exists t. \ [AG(t \iff \varphi(t)) \land \\
& \quad (\forall t'.(AG(t' \iff \varphi(t')) \Rightarrow AG(t \Rightarrow t')))
\end{align*}
\]

Expressiveness of QCTL

- QCTL can “count”:

\[
\text{EX}_1 \varphi \equiv \text{EX} \varphi \land \forall p. [\text{EX}(p \land \varphi) \Rightarrow \text{AX}(\varphi \Rightarrow p)]
\]

\[
\text{EX}_2 \varphi \equiv \exists q. [\text{EX}_1(\varphi \land q) \land \text{EX}_1(\varphi \land \neg q)]
\]

- QCTL can express (least or greatest) fixpoints:

\[
\mu T. \varphi(T) \equiv \exists t. [\text{AG}(t \iff \varphi(t)) \land
(\forall t'.(\text{AG}(t' \iff \varphi(t')) \Rightarrow \text{AG}(t \Rightarrow t')))]
\]

Theorem

QCTL and MSO are equally expressive (under both semantics).

Theorem

*Under the structure semantics:*

- *model checking* is PSPACE-complete;
- *satisfiability* is undecidable.
Decision problems for QCTL

Theorem

Under the structure semantics:
- model checking is PSPACE-complete;
- satisfiability is undecidable.

Under the tree semantics:
- model checking is \( k \)-EXPTIME-complete (for \( k \) nested quantifiers);
- satisfiability is \( k+1 \)-EXPTIME-complete (for \( k \) nested quantifiers).
Translating $\text{ATL}_{sc}$ into QCTL

- player $A$ has moves $m^A_1, \ldots, m^A_n$;
- from the transition table, we can compute the set $\text{Next}(\bigcirc, A, m^A_i)$ of states that can be reached from $\bigcirc$ when player $A$ plays $m^A_i$.

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 $\text{Next}(\bigcirc, A, m_i^A)$ of states that can be reached from $\bigcirc$ 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 $\varphi$:

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

Translating $\text{ATL}_{sc}$ into 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 $m_i^A$.

Corollary

$\text{ATL}_{sc}$ model checking is decidable, with non-elementary complexity (TOWER-complete).

Corollary

$\text{ATL}_{sc}^0$ (quantification restricted to memoryless strategies) model checking is PSPACE-complete.

What about satisfiability?

**Theorem**

*QCTL satisfiability is decidable (for the tree semantics).*

---

What about satisfiability?

**Theorem**

$QCTL$ satisfiability is decidable (for the tree semantics).

But

**Theorem ([TW12])**

$ATL_{sc}$ satisfiability is undecidable.

What about satisfiability?

**Theorem**

QCTL satisfiability is decidable (for the tree semantics).

But

**Theorem ([TW12])**

ATL\(_{sc}\) satisfiability is undecidable.

**Why?**

The translation from ATL\(_{sc}\) to QCTL assumes that the game structure is given!

Satisfiability for turn-based games

Theorem (LM13b)

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

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 $\bullet$.
- a strategy can be encoded by marking some of the nodes of the tree with proposition $\text{mov}_A$.

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

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

- it corresponds to a strategy: $\text{AG}(\text{turn}_A \Rightarrow \text{EX}_1 \text{mov}_A)$;
- the outcomes all satisfy $\varphi$: $\text{A}[\text{G}(\text{turn}_A \Rightarrow \text{X mov}_A) \Rightarrow \varphi]$.

What about Strategy Logic? [CHP07, MMV10]

<table>
<thead>
<tr>
<th>Strategy logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit quantification over strategies + strategy assignement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle A \rangle \varphi \equiv \exists \sigma_1. \text{assign}(\sigma_1, A).\varphi )</td>
</tr>
</tbody>
</table>

What about Strategy Logic? [CHP07, MMV10]

**Strategy logic**
Explicit quantification over strategies + strategy assignment

**Example**

\[
\langle \cdot A \rangle \varphi \equiv \exists \sigma_1. \text{assign}(\sigma_1, A) \cdot \varphi
\]

Strategy logic can also be translated into QCTL.

**Theorem**
- *Strategy-logic model-checking is decidable.*
- *Strategy-logic satisfiability is decidable when restricted to turn-based games.*

Conclusions and future works

Conclusions

- \( \text{ATL}_{sc} \) is a powerful logic for specifying properties of games on graphs;
- QCTL is a nice tool to understand such temporal logics for games (\( \text{ATL}_{sc} \), Strategy Logic, ...);

Future directions

- Defining interesting (expressive yet tractable) fragments of those logics (e.g. by bounding the number of alternations);
- Obtaining practicable algorithms.
- Adding quantitative aspects to \( \text{ATL}_{sc} \);
- Characterizing equivalent structures w.r.t \( \text{ATL}_{sc} \);
- Considering randomised strategies.
Conclusions

- **ATL\textsubscript{sc}** is a powerful logic for specifying properties of games on graphs;
- **QCTL** is a nice tool to understand such temporal logics for games (**ATL\textsubscript{sc}, Strategy Logic, ...**);

Future directions

- Defining interesting (expressive yet tractable) fragments of those logics (e.g. by bounding the number of alternations);
- Obtaining practicable algorithms.
- Adding quantitative aspects to ATL\textsubscript{sc};
- Characterizing equivalent structures w.r.t ATL\textsubscript{sc};
- Considering randomised strategies.