ATL with strategy contexts
— Expressiveness and model checking —

Arnaud Da Costa\textsuperscript{1}, François Laroussinie\textsuperscript{2}, Nicolas Markey\textsuperscript{1}

\textsuperscript{1} LSV, CNRS & ENS Cachan, France
\textsuperscript{2} LIAFA, CNRS & Univ. Paris7-Diderot, France

October 25, 2011
Model checking

system:

property:

\[ G (\text{request} \Rightarrow F \text{grant}) \]

model-checking algorithm

yes/no
Model checking and control

System:

Property:

\[ G(\text{request} \Rightarrow F \text{grant}) \]

Model-checking algorithm

Yes/no
Reasoning about open systems

Concurrent games

A concurrent game is made of

- a transition system;

\[ q_0 \rightarrow q_1 \rightarrow q_2 \]

player 1

player 2
Reasoning about open systems

Concurrent games

A concurrent game is made of
- a transition system;
- a set of agents;

\[ q_0 \rightarrow q_1 \rightarrow q_2 \]

player 1
player 2
Reasoning about open systems

Concurrent games

A concurrent game is made of

- a transition system;
- a set of agents;
- a table indicating the transition to be taken given the actions of the players.

<table>
<thead>
<tr>
<th>player 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image 268x96 to 290x119]</td>
<td>![Image 191x28 to 213x96]</td>
</tr>
<tr>
<td>![Image 229x96 to 252x119]</td>
<td>![Image 306x96 to 329x119]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>player 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image 268x96 to 290x119]</td>
<td>![Image 191x28 to 213x96]</td>
</tr>
<tr>
<td>![Image 229x96 to 252x119]</td>
<td>![Image 306x96 to 329x119]</td>
</tr>
<tr>
<td>![Image 268x96 to 290x119]</td>
<td>![Image 191x28 to 213x96]</td>
</tr>
</tbody>
</table>

q0 q1 q2

player 1

player 2
Reasoning about open systems

Concurrent games
A concurrent game is made of
- a transition system;
- a set of agents;
- 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

Concurrent games

A concurrent game is made of
- a transition system;
- a set of agents;
- 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.

Definition

A strategy for Player $i$ is a function associating, with each finite play $\rho$ of the game, a possible move for Player $i$ from $\text{last}(\rho)$. 
Definition

ATL extends CTL with strategy quantifiers:

$$\langle A \rangle \phi \iff A \text{ has a strategy } \sigma \text{ to enforce } \phi$$

(along all the outcomes)
Alternating-time temporal logic

Definition

ATL extends CTL with strategy quantifiers:

\[ \langle A \rangle \phi \iff \text{A has a strategy } \sigma \text{ to enforce } \phi \]

(along all the outcomes)
Alternating-time temporal logic

Definition

ATL extends CTL with strategy quantifiers:

\[ \langle A \rangle \phi \iff A \text{ has a strategy } \sigma \text{ to enforce } \phi \] (along all the outcomes)

\[ \checkmark \langle \bigcirc \rangle F \equiv \langle \bigcirc \rangle \text{true} U \bigcirc \]

\[ \times \langle \square \rangle F \]
Alternating-time temporal logic

**Definition**

ATL extends CTL with *strategy quantifiers*:

\[ ⟨⟨A⟩⟩φ \iff A \text{ has a strategy } σ \text{ to enforce } φ \]

(along all the outcomes)

\[
\checkmark \; ⟨⟨⟩⟩F \; \equiv \; ⟨⟨⟩⟩true \; U \; \square
\]

\[\times \; ⟨⟨⟩⟩F \square\]

\[⟨⟨⟩⟩G(⟨⟨⟩⟩F \square)\]
Alternating-time temporal logic

Definition

ATL extends CTL with strategy quantifiers:

\[ \langle A \rangle \phi \iff A \text{ has a strategy } \sigma \text{ to enforce } \phi \]

(along all the outcomes)
Alternating-time temporal logic

Definition

ATL extends CTL with *strategy quantifiers*:

\[ \langle A \rangle \phi \iff A \text{ has a strategy } \sigma \text{ to enforce } \phi \]

(along all the outcomes)

\[ p \]

\[ p \]

\[ \checkmark \quad \langle \Box \rangle F \equiv \langle \Diamond \rangle \text{true} \cup \]

\[ \times \quad \langle \Box \rangle F \]

\[ \times \quad \langle \Diamond \rangle G(\langle \Box \rangle F) \equiv \langle \Diamond \rangle G p \]
Alternating-time temporal logic

Definition

ATL extends CTL with strategy quantifiers:

\[ \langle \langle A \rangle \rangle \phi \iff A \text{ has a strategy } \sigma \text{ to enforce } \phi \]  
(along all the outcomes)

Theorem

ATL model checking is PTIME-complete.
Another semantics: ATL with strategy contexts [BDLM09]

Evaluate the formula on the execution tree:

\[ \langle \Diamond \rangle \mathbf{G} (\langle \Box \rangle \mathbf{F} \circ \circ) \]
Another semantics: ATL with strategy contexts [BDLM09]

Evaluate the formula on the execution tree:

$$\langle \Diamond \rangle G(\langle \Box \rangle F \bullet)$$
Another semantics: ATL with strategy contexts [BDLM09]

\[ \langle \bigcirc \rangle \, G( \langle \Box \rangle F \bigcirc ) \]

Evaluate the formula on the execution tree:
- apply a strategy of Player $\bigcirc$;
Another semantics: ATL with strategy contexts [BDLM09]

Evaluate the formula on the execution tree:
- apply a strategy of Player $\bigcirc$;
- in the remaining tree, check that Player $\square$ can always enforce a visit to $\bullet$. 

\[
\langle\bigcirc\rangle \text{ G}(\langle\square\rangle \text{ F } \bullet)
\]
ATL with strategy contexts

**Definition**

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

- \(\langle \cdot A \rangle \) is similar to \(\langle A \rangle \) but **assigns** the corresponding strategy to \(A\) for evaluating \(\phi\);
- \(\llbracket A \rrbracket\) drops the assigned strategies for \(A\).
ATL with strategy contexts

Definition

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

- $\langle \cdot A \cdot \rangle$ is similar to $\langle A \rangle$ but assigns the corresponding strategy to $A$ for evaluating $\phi$;
- $\langle A \rangle$ drops the assigned strategies for $A$.

\[
\begin{align*}
\langle \cdot \cdot \rangle F & \times \langle \cdot \cdot \rangle F \\
\checkmark & \langle \cdot \circ \cdot \rangle G(\langle \cdot \Box \cdot \rangle F) \\
\times & \langle \cdot \Box \cdot \rangle F
\end{align*}
\]
Outline of the talk

1. Introduction
2. Related approaches
3. Expressiveness issues
4. Model checking
5. Conclusions
Outline of the talk

1. Introduction
2. Related approaches
3. Expressiveness issues
4. Model checking
5. Conclusions
Related approaches

- **ATL with commitment** (van der Hoek, Jamroga, Wooldridge, 2005) extends ATL with an operator which restricts the behaviour of some players to a fixed *(memoryless)* strategy.

- **ATL with irrevocable strategies** (Ågostnes, Goranko, Jamroga, 2008) is a similar extension to ours, but with a different way of handling the strategy contexts. Again, only investigated in the memoryless case.
Related approaches

- **ATL with commitment** (van der Hoek, Jamroga, Wooldridge, 2005)
- **ATL with irrevocable strategies** (Ågostnes, Goranko, Jamroga, 2008)

- **QDμ** (Pinchinat, 2007): extension of the μ-calculus with a *decision modality*. A strategy is a labelling of a tree whose directions are the set of decisions of the agents (only works for a subclass of CGSs).
Related approaches

- **ATL with commitment** (van der Hoek, Jamroga, Wooldridge, 2005)
- **ATL with irrevocable strategies** (Ågostnes, Goranko, Jamroga, 2008)

- **QDµ** (Pinchinat, 2007)

- **Stochastic Game Logic** (Baier, Brázdil, Größer, Kučera, 2007): same extension as ours, in a probabilistic setting: games are turn-based and stochastic. Model checking is undecidable (both for deterministic and mixed strategies), but decidable when restricting to memoryless strategies.
Related approaches

- **ATL with commitment** (van der Hoek, Jamroga, Wooldridge, 2005)
- **ATL with irrevocable strategies** (Ågostnes, Goranko, Jamroga, 2008)

- **QDμ** (Pinchinat, 2007)

- **Stochastic Game Logic** (Baier, Brázdil, Größer, Kučera, 2007)

- **Strategy logic** (Chatterjee, Henzinger, Piterman, 2007): first-order quantification over strategies. Nested formulas must be closed. Defined and studied only on 2-player turn-based games. Algorithm similar to ours but in a simpler setting (non-elementary complexity).
Related approaches

- **ATL with commitment** (van der Hoek, Jamroga, Wooldridge, 2005)
- **ATL with irrevocable strategies** (Ågostnes, Goranko, Jamroga, 2008)

- **QDμ** (Pinchinat, 2007)

- **Stochastic Game Logic** (Baier, Brázdil, Größer, Kučera, 2007)

- **Strategy logic** (Chatterjee, Henzinger, Piterman, 2007)

- **Strategy logic** (Mogavero, Murano, Vardi, 2010): new version of SL with separate strategy quantifications and strategy assignments. Model-checking in 2EXPTIME-complete over the full class of $n$-player CGSs. Satisfiability is undecidable.
Related approaches

- **ATL with commitment** (van der Hoek, Jamroga, Wooldridge, 2005)
- **ATL with irrevocable strategies** (Ågostnes, Goranko, Jamroga, 2008)
- **$QD\mu$** (Pinchinat, 2007)
- **Stochastic Game Logic** (Baier, Brázdil, Größer, Kučera, 2007)
- **Strategy logic** (Chatterjee, Henzinger, Piterman, 2007)
- **Strategy logic** (Mogavero, Murano, Vardi, 2010)
  same approach as ours. PSPACE algorithm when nested formulas are requires to be closed.
Outline of the talk

1. Introduction
2. Related approaches
3. Expressiveness issues
4. Model checking
5. Conclusions
What $\text{ATL}_{sc}$ can express

- All $\text{ATL}^*$ properties;
What $ATL_{sc}$ can express

- All $ATL^*$ properties;
- Client-server interactions for accessing a shared resource:

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

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

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

- Existence of **Nash equilibria**:

$$
\langle \cdot A_1, \ldots, A_n \rangle \ \bigwedge \ i \ (\langle \cdot A_i \rangle \varphi_{A_i} \Rightarrow \varphi_{A_i})
$$
What ATL\textsubscript{sc} can express

- All ATL\textsuperscript{*} properties;
- **Client-server interactions** for accessing a shared resource:
  \[
  \langle \cdot \text{Server} \cdot \rangle \ G \quad \bigg[ \quad \bigwedge_{c \in \text{Clients}} \langle \cdot c \cdot \rangle \ \mathbf{F} \ \text{access}_c \\
  \quad \bigwedge_{c \neq c'} \ \neg \ \text{access}_c \ \land \ \text{access}_{c'}
  \bigg]
  \]
- Existence of **Nash equilibria**:
  \[
  \langle \cdot A_1, \ldots, A_n \cdot \rangle \ \bigwedge_i \left( \langle \cdot A_i \cdot \rangle \ \varphi_{A_i} \implies \varphi_{A_i} \right)
  \]
- Existence of **dominating strategy**:
  \[
  \langle \cdot A \cdot \rangle \ [B] \ (\neg \varphi \implies [\cdot A \cdot] \ \neg \varphi)
  \]
Expressiveness of ATL$_{sc}$

Theorem ([BDLM09])

- The $\langle A \rangle$ -operator is superfluous;
- $ATL_{sc}$ is strictly more expressive than ATL.
Expressiveness of $\text{ATL}_{sc}$

**Theorem ([BDLM09])**

- The $\langle A \rangle$ -operator is superfluous;
- $\text{ATL}_{sc}$ is strictly more expressive than $\text{ATL}$.

**Proof**

\[ \langle 1 \cdot \rangle (\langle 2 \cdot \rangle X a \land \langle 2 \cdot \rangle X b) \].

$s$ and $s'$ are alternating-bisimilar, hence undistinguishable by $\text{ATL}^*$. 
Expressiveness of $\text{ATL}_{sc}$

**Theorem ([BDLM09])**

- The $\mathcal{A}$-operator is superfluous;
- $\text{ATL}_{sc}$ is strictly more expressive than $\text{ATL}$.

**Theorem**

- $\text{ATL}_{sc}$ is as expressive as $\text{ATL}_{sc}^*$.
Expressiveness of $\text{ATL}_{sc}$

Theorem ([BDLM09])

- The $\lla A \rra$-operator is superfluous;
- $\text{ATL}_{sc}$ is strictly more expressive than $\text{ATL}$.

Theorem

- $\text{ATL}_{sc}$ is as expressive as $\text{ATL}^*_{sc}$.

Proof

Insert extra $\lla \emptyset \rra$ between nested modalities.
Outline of the talk

1 Introduction
2 Related approaches
3 Expressiveness issues
4 Model checking
5 Conclusions
Model checking $\text{ATL}_{sc}$

Tree-automata approach

The unwinding tree is accepted by a deterministic tree automaton;
The unwinding tree is accepted by a deterministic tree automaton;
Model checking $\text{ATL}_{sc}$

Tree-automata approach

- A strategy is encoded as a labelling of the unwinding tree;
Model checking $\text{ATL}_{sc}$

Tree-automata approach

- We can mark outcomes corresponding to selected strategies;
We mark the tree with extra propositions $p_I$ and $p_r$, and require that it satisfies $A(G p_o \Rightarrow p_I U p_r)$;
We require that subtrees rooted at a $p_l$ or $p_r$ node is accepted by the automaton for $\varphi$ or $\varphi'$, respectively;
We can build a tree automaton accepting all trees that can be labelled with correct strategies. This requires turning the alternating tree automaton into a non-deterministic one, which yields an exponential-size automaton.
Model checking $\text{ATL}_{sc}$

**Theorem**

Given a CGS $\mathcal{C}$, a state $\ell_0$ and an $\text{ATL}_{sc}$ formula $\varphi$, we can build an alternating parity tree automaton $\mathcal{A}$ s.t.

$$\mathcal{L}(\mathcal{A}) \neq \emptyset \iff \mathcal{C}, \ell_0 \models \varphi.$$ 

$\mathcal{A}$ has size $d$-exponential, where $d$ is the maximal number of nested quantifiers.

**Theorem**

Model-checking $\text{ATL}_{sc}$ can be achieved in $d$-$\text{EXPTIME}$, where $d$ is the maximal number of nested quantifiers in the formula.
QLTL extends LTL with quantification over atomic propositions:

\[ \forall a. \exists b. \ G(b \Leftrightarrow X a) \]
Hardness

QLTL extends LTL with quantification over atomic propositions:

Example

\[ \forall a. \exists b. \ G(b \iff X a) \]

Theorem (SVW87)

Satisfiability of a QLTL formula is \(k\)-EXPSPACE-complete, where \(k\) is the alternation-depth of the formula.
Hardness

QLTL extends LTL with quantification over atomic propositions:

Example

∀a. ∃b. \( G(b \Leftrightarrow X a) \)

Theorem (SVW87) Satisfiability of a QLTL formula is \( k \)-EXPSPACE-complete, where \( k \) is the alternation-depth of the formula.
Hardness

QLTL extends LTL with quantification over atomic propositions:

Example

\[ \forall a. \exists b. \ G(b \iff X a) \]

Theorem (SVW87)

Satisfiability of a QLTL formula is $k$-EXPSPACE-complete, where $k$ is the alternation-depth of the formula.
Theorem

ATL sc model checking is (d-1)-EXPSPACE-hard for formulas having at most d nested quantifiers.
Theorem

$ATL_{sc}$ model checking is $(d-1)$-EXPSPACE-hard for formulas having at most $d$ nested quantifiers.
Conclusions

- Our results on $\text{ATL}_{sc}$:
  - $\text{ATL}_{sc}$ is a natural semantical extension of the popular ATL;
  - $\text{ATL}_{sc}$ is much more expressive: equilibria, client-server interactions... Well-suited for non-zero-sum objectives;
  - There is a price for this expressiveness: high complexity of the model-checking algorithm.
Conclusions

- **Our results on $ATL_{sc}$:**
  - $ATL_{sc}$ is a **natural semantical extension** of the popular $ATL$;
  - $ATL_{sc}$ is **much more expressive**: equilibria, client-server interactions... Well-suited for non-zero-sum objectives;
  - There is a price for this expressiveness: **high complexity** of the model-checking algorithm.

- **Future works:**
  - links between $ATL_{sc}$ and QCTL;
  - study satisfiability of $ATL_{sc}$;
  - behavioural equivalence for $ATL_{sc}$. 