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

December 9, 2010
Model checking

system:

⇒

property:

G(request⇒F grant)

model-checking algorithm

⇒

G(request⇒F grant)

yes/no
Model checking and control

system:

property:

G(request ⇒ F grant)

model-checking algorithm

yes/no

G(request ⇒ F grant)

model-checking algorithm

yes/no
Model checking and control

system:

property:

G(request ⇒ F grant)

model-checking
algorithm

yes/no
Game models

Definition

Concurrent game structures (CGS):
Game models

Definition

Concurrent game structures (CGS):
- labelled transition system;

\[ q_0 \rightarrow q_1, q_2 \]
\[ p, r, s \]
Game models

Definition

Concurrent game structures (CGS):
- labelled transition system;
- for each state, a table indicating the transitions to be taken depending on the choices of the players.
Game models

Definition

Concurrent game structures (CGS):
- labelled transition system;
- for each state, a table indicating the transitions to be taken depending on the choices of the players.

Player 1

<table>
<thead>
<tr>
<th>Player 1</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>q0</td>
<td>p</td>
<td>r</td>
<td>s</td>
</tr>
<tr>
<td>p</td>
<td>q0</td>
<td>q1</td>
<td>q2</td>
</tr>
<tr>
<td>r</td>
<td>q2</td>
<td>q0</td>
<td>q1</td>
</tr>
<tr>
<td>s</td>
<td>q1</td>
<td>q2</td>
<td>q0</td>
</tr>
</tbody>
</table>

Player 2

<table>
<thead>
<tr>
<th>Player 2</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>q0</td>
<td>p</td>
<td>r</td>
<td>s</td>
</tr>
<tr>
<td>p</td>
<td>q0</td>
<td>q1</td>
<td>q2</td>
</tr>
<tr>
<td>r</td>
<td>q2</td>
<td>q0</td>
<td>q1</td>
</tr>
<tr>
<td>s</td>
<td>q1</td>
<td>q2</td>
<td>q0</td>
</tr>
</tbody>
</table>
Game models

**Definition**

Concurrent game structures (CGS):
- labelled transition system;
- for each state, a table indicating the transitions to be taken depending on the choices of the players.

\[
\langle p|p\rangle, \langle r|r\rangle, \langle s|s\rangle
\]

\[
\langle r|s\rangle, \langle s|p\rangle, \langle p|r\rangle
\]

\[
\langle s|r\rangle, \langle p|s\rangle, \langle r|p\rangle
\]

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>q0</td>
</tr>
<tr>
<td>r</td>
<td>q2</td>
</tr>
<tr>
<td>s</td>
<td>q1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>q0</th>
<th>p</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>q0</td>
<td>q0</td>
<td>q1</td>
<td>q2</td>
</tr>
<tr>
<td>q1</td>
<td>q1</td>
<td>q2</td>
<td>q0</td>
</tr>
<tr>
<td>q2</td>
<td>q0</td>
<td>q1</td>
<td>q0</td>
</tr>
</tbody>
</table>
Game models

Definition
Concurrent game structures (CGS):
- labelled transition system;
- for each state, a table indicating the transitions to be taken depending on the choices of the players.

Remark
*Turn-based games* form a subclass of CGSs where at each state, all the moves are equivalent for all but one player.
Game models

**Definition**

Concurrent game structures (CGS):
- labelled transition system;
- for each state, a table indicating the transitions to be taken depending on the choices of the players.

**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 \( \text{last}(\rho) \).
Alternating-time temporal logic

Definition

ATL extends CTL with \textit{strategy quantifiers}:

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

(\text{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)
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)
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)

\[ \checkmark \ \langle \bigcirc \rangle \ F \bigcirc \]

\[ \times \ \langle \square \rangle \ F \bigcirc \]

\[ \times \ \langle \bigcirc \rangle \ G ( \langle \square \rangle \ F \bigcirc ) \]
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)

**Theorem**

*ATL model checking is PTIME-complete.*
ATL with strategy contexts

Definition

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

- \( \langle \cdot A \cdot \rangle \) is similar to \( \langle A \rangle \) but assigns the corresponding strategy to \( A \) for evaluating \( \phi \);
- \( \|A\| \) 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 $\llbracket A \rrbracket \phi$.

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

$\langle \cdot \cdot \rangle F \times \langle \cdot \cdot \rangle F$

$\llbracket \cdot \cdot \cdot \rrbracket F$

$\checkmark \ \llbracket \cdot \cdot \cdot \rrbracket G(\langle \Box \rangle F)$
ATL with strategy contexts

Definition

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

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

Definition

\[
G, s \models_F \langle A \rangle \phi \iff \exists \sigma_A. \forall \rho' \in \text{Out}(s, F[A \mapsto \sigma_A]).
G, \rho' \models_{F[A \mapsto \sigma_A]} \phi
\]

\[
G, \rho \models_F \phi \mathbf{U} \psi \iff \exists j. G, \rho \geq j \models_{\text{shift}(F, \rho[0..j])} \psi
\]

and \(\forall 0 \leq k < j. G, \rho \geq k \models_{\text{shift}(F, \rho[0..k])} \phi\)
Outline of the talk

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

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

- $\text{ATL}_{sc}$ encompasses ATL:

$$\langle \langle A \rangle \phi \equiv \langle \text{Agt} \rangle \langle \cdot \rangle \phi$$

- Client-server interactions for accessing a shared resource:

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

- (Boolean-objective, pure-strategy) Nash equilibria:

$$\langle A_1, ..., A_n \rangle \bigwedge_{i} (\langle \cdot \rangle \phi_{A_i} \Rightarrow \phi_{A_i})$$
Example of ATL_{sc} formulas

- **ATL_{sc}** encompases ATL:

\[
\langle \langle A \rangle \rangle \phi \equiv \langle \text{Agt} \rangle \langle A \rangle \hat{\phi}
\]

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

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

- *(Boolean-objective, pure-strategy)* Nash equilibria:

\[
\langle A_1, \ldots, A_n \rangle \ \land \ \left( \ \land_{i} \langle \cdot \rangle \ \phi_{A_i} \ \Rightarrow \ \phi_{A_i} \ \right)
\]
Example of $\text{ATL}_{sc}$ formulas

- $\text{ATL}_{sc}$ encompasses $\text{ATL}$:
  \[
  \langle \langle A \rangle \rangle \phi \equiv \langle \langle \text{Agt} \rangle \langle \cdot \rangle \rangle \hat{\phi}
  \]

- Client-server interactions for accessing a shared resource:
  \[
  \langle \cdot \rangle \text{Server} \cdot \mathbf{G} \left[ \bigwedge_{c \in \text{Clients}} \langle \cdot \rangle \text{F access}_c \right]
  \]

- (Boolean-objective, pure-strategy) Nash equilibria:
  \[
  \langle \cdot \rangle \langle A_1, \ldots, A_n \rangle \left( \bigwedge_i (\langle \cdot \rangle \phi_{A_i} \Rightarrow \phi_{A_i}) \right)
  \]
Theorem

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

Proof. The first statement is obtained (roughly) by replacing $\langle\cdot A\cdot \rangle$ with $\langle\cdot \cdot A\cdot \rangle$, which is the dual of $\langle\cdot A\cdot \rangle$.

The second statement is obtained (roughly) by inserting $\langle\cdot \cdot \emptyset\cdot \rangle$ between any two nested temporal modalities.
Expressiveness of $\text{ATL}_{sc}$

**Theorem**

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

**Proof.**

- The first statement is obtained (roughly) by replacing $\langle A \rangle$ with $[A]$, which is the dual of $\langle \cdot A \rangle$.
- The second statement is obtained (roughly) by inserting $\langle \cdot \emptyset \cdot \rangle$ between any two nested temporal modalities.
Expressiveness of ATL$_{sc}$

Theorem

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

Proof.

- $\langle 1 \rangle (\langle 2 \rangle X a \land \langle 2 \rangle X b)$. 

$s$ and $s'$ are alternating-bisimilar, hence undistinguishable by ATL$^*$. 
Outline of the talk

1. Introduction
2. Expressiveness issues
3. Related approaches
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 of handling the strategy context. 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)

- **QDₘ** (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 (hence only works for ATSs).
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 stochastic games. Model checking is undecidable (both deterministic and mixed strategies), but becomes 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 only on 2-player turn-based games. Algorithm similar to ours but in a simpler setting.
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): new version of SL with separate strategy quantifications and strategy assignments. Model-checking in 2EXPTIME over the full class of $n$-player CGSs. Satisfiability is undecidable.
Outline of the talk

1. Introduction
2. Expressiveness issues
3. Related approaches
4. Model checking
5. Conclusions
The unwinding tree is accepted by a deterministic tree automaton;
Model checking $\text{ATL}_{sc}$

Tree-automata approach

- 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;
We can mark outcomes corresponding to selected strategies, and check that they satisfy subformula $\varphi$;
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.
Theorem

Given a CGS $C$, a state $\ell_0$ and an ATL$_{sc}$ formula $\varphi$, we can build an APT $A$ s.t.

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

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

Theorem

Model-checking ATL$_{sc}$ can be achieved in $(d + 1)$-EXPTIME, where $d$ is the maximal number of nested quantifiers in the formula.
Recent advances: hardness

QPTL extends LTL with quantification over atomic propositions:

Example

$$\forall a. \exists b. \ G(b \leftrightarrow X a)$$
Recent advances: hardness

QPTL extends LTL with quantification over atomic propositions:

Example

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

Theorem (SVW87)
Satisfiability of a QPTL formula is \(k\)-EXPSPACE-complete, where \(k\) is the alternation-depth of the formula.
Recent advances: hardness

QPTL extends LTL with quantification over atomic propositions:

Example

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

\[ a \quad a \quad a \quad a \quad a \quad a \]

\[ b \quad b \quad b \quad b \quad b \quad b \]
Recent advances: hardness

QPTL extends LTL with quantification over atomic propositions:

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

Theorem (SVW87)
Satisfiability of a QPTL formula is \( k \)-EXPSPACE-complete, where \( k \) is the alternation-depth of the formula.
Recent advances: hardness

Example

\[ [A] \langle B \rangle \quad [G \bigcirc \Rightarrow G(\langle Z \rangle X X b \iff X \langle Z \rangle X X a)] \]
Recent advances: hardness

Example

Theorem

\[ [A] \langle B \rangle \quad [G \bigcirc \Rightarrow G(\langle Z \rangle XX b \Leftrightarrow X \langle Z \rangle XX a) ] \]

ATLsc model checking is \( k\)-EXPSPACE-hard for formulas with \( k + 1 \) nested quantifiers.
Conclusions

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

- Future works:
  - close the complexity gap in the model-checking problem;
  - study satisfiability of $\text{ATL}_{sc}$. 