Satisfiability of ATL with strategy contexts

François Laroussinie and Nicolas Markey
LIAFA LSV

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...
   - 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

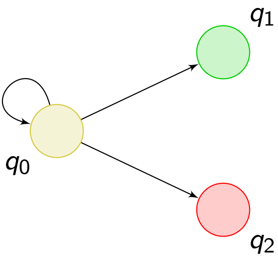
3. Satisfiability of ATL with strategy contexts
   - QCTL satisfiability is decidable, but...
   - ATL_{sc} satisfiability is not, except for turn-based games

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.



		player 1		
				
player 2		q_0	q_2	q_1
		q_1	q_0	q_2
		q_2	q_1	q_0

Reasoning about multi-agent systems

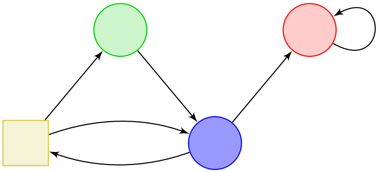
Concurrent games

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

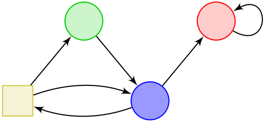
A strategy for a given player is a function telling what to play depending on what has happened previously.

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 .

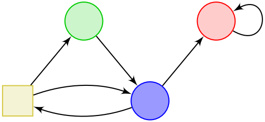


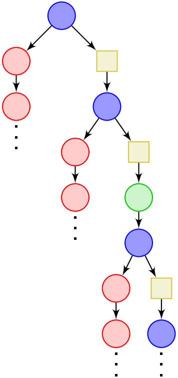
Reasoning about open systems

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

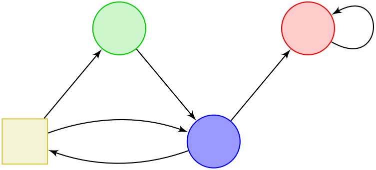
$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

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



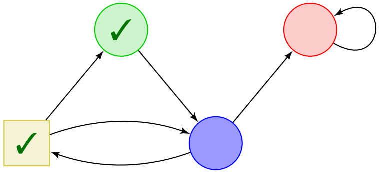
- $\langle\langle\square\rangle\rangle$ **F** 

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

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



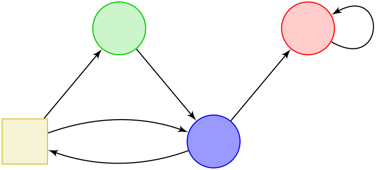
- $\langle\langle \square \rangle\rangle$ **F** ○

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

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



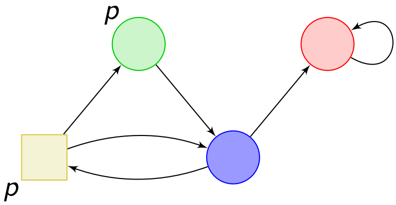
- $\langle\!\langle \square \rangle\!\rangle\ \mathbf{F}\ \bigcirc$
- $\langle\!\langle \bigcirc \rangle\!\rangle\ \mathbf{G}(\ \langle\!\langle \square \rangle\!\rangle\ \mathbf{F}\ \bigcirc)$

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

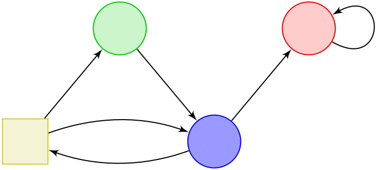
$\langle\!\langle A \rangle\!\rangle\, \varphi$ expresses that A has a strategy to enforce φ.



- $\langle\!\langle \square \rangle\!\rangle\ \mathbf{F}\ \bigcirc$
- $\langle\!\langle \bigcirc \rangle\!\rangle\ \mathbf{G}\,(\underbrace{\langle\!\langle \square \rangle\!\rangle\ \mathbf{F}\ \bigcirc}_{p}) \equiv \langle\!\langle \bigcirc \rangle\!\rangle\ \mathbf{G}\ p$

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

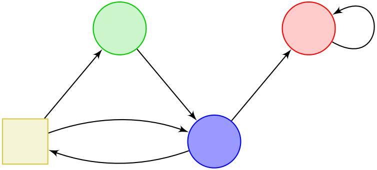
Another semantics: ATL with strategy contexts [BDLM09]



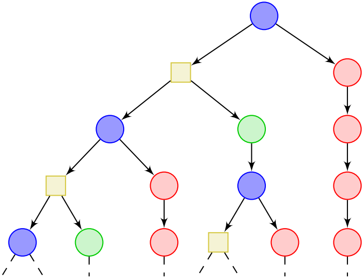
$$\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \square \rangle\rangle\ \mathbf{F}\ \bigcirc)$$

[BDLM09] Brihaye, Da Costa, Laroussinie, M. ATL with strategy contexts. LFCS, 2009.

Another semantics: ATL with strategy contexts [BDLM09]



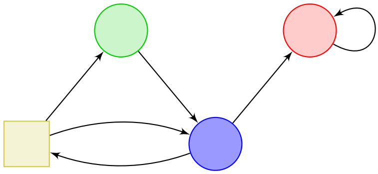
$\langle\!\langle \bigcirc \rangle\!\rangle\ \mathbf{G}(\ \langle\!\langle \square \rangle\!\rangle\ \mathbf{F}\ \bigcirc)$



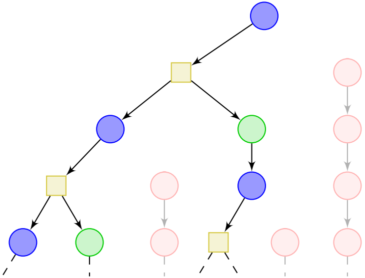
- consider the following strategy of Player ○: "always go to □";

[BDLM09] Brihaye, Da Costa, Laroussinie, M. ATL with strategy contexts. LFCS, 2009.

Another semantics: ATL with strategy contexts [BDLM09]



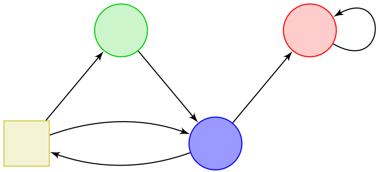
$\langle\!\langle \bigcirc \rangle\!\rangle\ \mathbf{G}(\ \langle\!\langle \Box \rangle\!\rangle\ \mathbf{F}\ \bigcirc)$



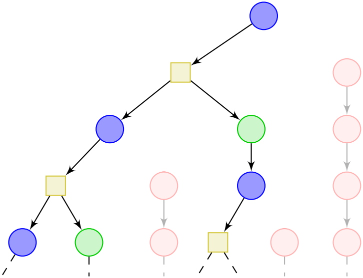
- consider the following strategy of Player ○: “always go to □”;

[BDLM09] Brihaye, Da Costa, Laroussinie, M. ATL with strategy contexts. LFCS, 2009.

Another semantics: ATL with strategy contexts [BDLM09]



$\langle\!\langle \bigcirc \rangle\!\rangle\ \mathbf{G}(\ \langle\!\langle \square \rangle\!\rangle\ \mathbf{F}\ \bigcirc)$



- consider the following strategy of Player $\bigcirc$: "always go to $\square$";
- in the remaining tree, Player $\square$ can always enforce a visit to $\bigcirc$.

[BDLM09] Brihaye, Da Costa, Laroussinie, M. ATL with strategy contexts. LFCS, 2009.

What ATL_{sc} can express

- All ATL* properties:

What ATL_{sc} can express

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

$$\langle\cdot \mathrm{Server} \cdot\rangle\ \mathbf{G} \left[\begin{array}{ll} & \displaystyle\bigwedge_{c \in \mathrm{Clients}} \langle\cdot c \cdot\rangle\ \mathbf{F}\, \mathrm{access}_c \\ \wedge & \\ & \neg \displaystyle\bigwedge_{c \neq c'} \mathrm{access}_c \wedge \mathrm{access}_{c'} \end{array} \right]$$

What ATL_{sc} can express

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

$$\langle\!\cdot\text{Server}\cdot\!\rangle\ \mathbf{G}\left[\begin{array}{ll} & \bigwedge_{c\in\text{Clients}} \langle\!\cdot c\cdot\!\rangle\ \mathbf{F}\,\text{access}_c \\ \wedge & \\ & \neg \bigwedge_{c\neq c'} \text{access}_c \wedge \text{access}_{c'} \end{array}\right]$$

- Existence of Nash equilibria:

$$\langle\!\cdot A_1,\ldots,A_n\cdot\!\rangle \bigwedge_i \left(\langle\!\cdot A_i\cdot\!\rangle\,\varphi_{A_i} \Rightarrow \varphi_{A_i}\right)$$

- Existence of dominating strategy:

$$\langle\!\cdot A\cdot\!\rangle\ [\!\cdot B\cdot\!]\,(\neg\varphi \Rightarrow [\!\cdot A\cdot\!]\,\neg\varphi)$$

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

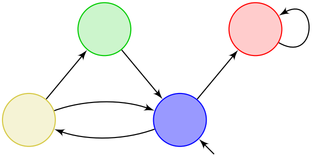
$\exists p.\ \varphi$ means that there exists a labelling of the model with p under which φ holds.

[Kup95] Kupferman. Augmenting Branching Temporal Logics with Existential Quantification over Atomic Propositions. CAV, 1995.
[Fre01] French. Decidability of Quantifed Propositional Branching Time Logics. AJCAI, 2001.

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 φ holds.



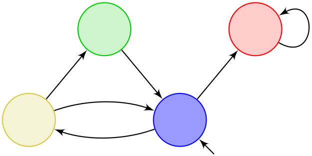
- $\mathbf{E\,F}\ \bigcirc \wedge \forall p.\ \big[\mathbf{E\,F}(p \wedge \bigcirc) \Rightarrow \mathbf{A\,G}(\bigcirc \Rightarrow p)\big]$

[Kup95] Kupferman. Augmenting Branching Temporal Logics with Existential Quantification over Atomic Propositions. CAV, 1995.
[Fre01] French. Decidability of Quantifed Propositional Branching Time Logics. AJCAI, 2001.

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 φ holds.



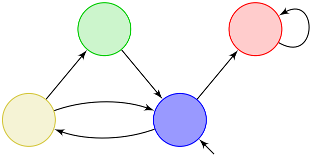
- $\mathbf{E\,F}\bigcirc \wedge \forall p.\ \big[\mathbf{E\,F}(p \wedge \bigcirc) \Rightarrow \mathbf{A\,G}(\bigcirc \Rightarrow p)\big] \equiv \text{uniq}(\bigcirc)$

[Kup95] Kupferman. Augmenting Branching Temporal Logics with Existential Quantification over Atomic Propositions. CAV, 1995.
[Fre01] French. Decidability of Quantifed Propositional Branching Time Logics. AJCAI, 2001.

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 φ holds.

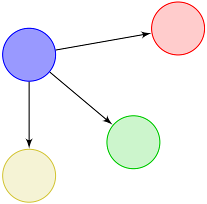


- $\mathbf{EF}\,\bigcirc \wedge \forall p.\ \left[\mathbf{EF}(p \wedge \bigcirc) \Rightarrow \mathbf{AG}(\bigcirc \Rightarrow p)\right] \equiv \mathrm{uniq}(\bigcirc)$

  $\rightsquigarrow$ true if we label the Kripke structure;
  $\rightsquigarrow$ false if we label the computation tree;

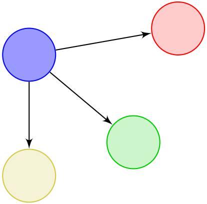
[Kup95] Kupferman. Augmenting Branching Temporal Logics with Existential Quantification over Atomic Propositions. CAV, 1995.
[Fre01] French. Decidability of Quantifed Propositional Branching Time Logics. AJCAI, 2001.

Translating ATL$_{sc}$ into QCTL



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

Translating ATL$_{sc}$ into QCTL



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

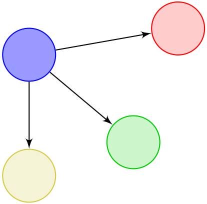
$\langle\cdot A\cdot\rangle\, \varphi$ can be encoded as follows:

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

- this corresponds to a strategy: $\quad \mathbf{A}\,\mathbf{G}(m_i^A \Leftrightarrow \bigwedge \neg\, m_j^A)$;
- the outcomes all satisfy φ:

$$\mathbf{A}\big[\mathbf{G}(q \wedge m_i^A \Rightarrow \mathbf{X}\,\mathrm{Next}(q, A, m_i^A)) \Rightarrow \varphi\big].$$

Translating ATL$_{sc}$ into QCTL



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

Theorem (DLM12)

QCTL model checking is decidable (in the tree semantics).

Corollary

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

[DLM12] Da Costa, Laroussinie, M. Quantified CTL: expressiveness and model checking. CONCUR, 2012.

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

What about satisfiability?

Theorem (LM13a)

QCTL satisfiability is decidable.

[LM13a] Laroussinie, M. Quantified CTL: expressiveness and complexity. Submitted, 2013.

What about satisfiability?

Theorem (LM13a)

QCTL satisfiability is decidable.

But

Theorem (TW12)

ATL_{sc} satisfiability is undecidable.

[LM13a] Laroussinie, M. Quantified CTL: expressiveness and complexity. Submitted, 2013.
[TW12] Troquard, Walther. On Satisfiability in ATL with Strategy Contexts. JELIA, 2012.

What about satisfiability?

Theorem (LM13a)

QCTL satisfiability is decidable.

But

Theorem (TW12)

ATL_{sc} satisfiability is undecidable.

Why?

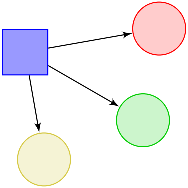
The translation from ATL_{sc} to QCTL assumes that the game structure is fixed!

[LM13a] Laroussinie, M. Quantified CTL: expressiveness and complexity. Submitted, 2013.
[TW12] Troquard, Walther. On Satisfiability in ATL with Strategy Contexts. JELIA, 2012.

Satisfiability for turn-based games

Theorem (LM13b)

When restricted to turn-based games, ATL_{sc} satisfiability is decidable.



- player $\square$ has moves (green circle), (yellow circle) and (red circle).
- a strategy can be encoded by marking some of the nodes of the tree with proposition mov_A.

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

$\exists mov_A.$

- it corresponds to a strategy: $\mathbf{A}\,\mathbf{G}(turn_A \Rightarrow \mathbf{E}\,\mathbf{X}_1\, mov_A)$;
- the outcomes all satisfy φ: $\mathbf{A}\big[\mathbf{G}(turn_A \wedge \mathbf{X}\, mov_A) \Rightarrow \varphi\big]$.

[LM13b] Laroussinie, M. Satisfiability of ATL with strategy contexts. Gandalf, 2013.

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 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 assignement

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.*

[CHP07] Chatterjee, Henzinger, Piterman. Strategy Logic. CONCUR, 2007.
[MMV10] Mogavero, Murano, Vardi. Reasoning about strategies. FSTTCS, 2010.

Conclusions and future works

Conclusions

- ATL_{sc} is a very powerful logic for reasoning about games.
- 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).

Conclusions and future works

Conclusions

- ATL_{sc} is a very powerful logic for reasoning about games.
- 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.