

On the Expressiveness and Complexity of ATL

François Laroussinie, Nicolas Markey, Ghassan Oreiby

LSV, CNRS & ENS-Cachan

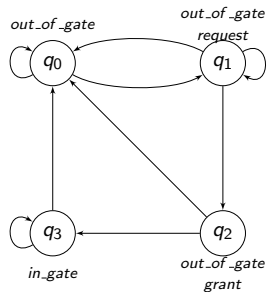
Recherches en vérification automatique

March 14, 2006

Overview of CTL

CTL

- A Kripke structure
- Quantification over **paths**
(**E**/ **A**)



Example

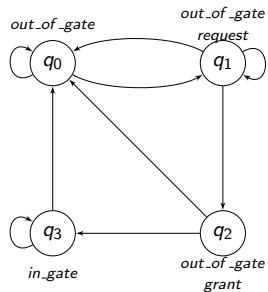
There exists a run such that the train is always out of gate.

EG out_of_gate

Overview of CTL

CTL

- A Kripke structure
- Quantification over **paths**
(**E**/ **A**)



Example

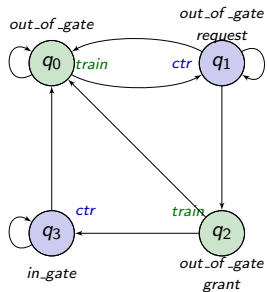
There exists a run such that the train is always out of gate.

EG *out_of_gate*

Overview of ATL

ATL

- A **multi-agent** system.
- Quantification over **strategies** of agents



Example

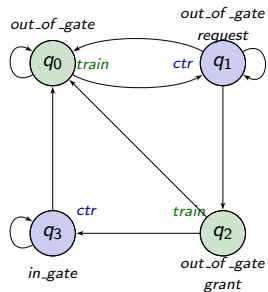
Whenever the train is out of gate, the controller cannot force it to enter the gate.

$$\text{AG } (out_of_gate \implies \neg \langle\langle ctr \rangle\rangle \text{F } in_gate)$$

Overview of ATL

ATL

- A **multi-agent** system.
- Quantification over **strategies** of agents



Example

Whenever the train is out of gate, the controller cannot force it to enter the gate.

$$\mathbf{AG} (out_of_gate \implies \neg \langle\langle ctr \rangle\rangle \mathbf{F} in_gate)$$

Outline of the talk

- 1 Introduction
 - Overview of CTL and ATL
- 2 Definitions
 - Multi-agent models
 - Strategy and outcomes
 - ATL (Alternating-time Temporal Logic)
- 3 Expressiveness
 - Weak Until
- 4 Complexity
 - Model checking ATL on CGSs
 - Model checking ATL on ATs
- 5 Conclusion

Outline of the talk

- 1 Introduction
 - Overview of CTL and ATL
- 2 Definitions
 - Multi-agent models
 - Strategy and outcomes
 - ATL (Alternating-time Temporal Logic)
- 3 Expressiveness
 - Weak Until
- 4 Complexity
 - Model checking ATL on CGSs
 - Model checking ATL on ATsS
- 5 Conclusion

CGS definition

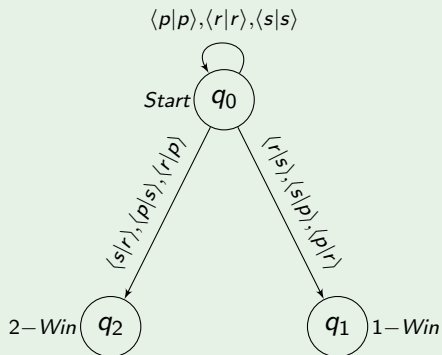
Definition

A CGS \mathcal{C} is a 5-tuple $(\text{Loc}, \text{Lab}, \text{Agt}, \text{Chc}, \text{Edg})$ s.t:

- **Loc**: a finite set of *locations*;
- **Lab**: $\text{Loc} \rightarrow 2^{\text{AP}}$: a labeling function;
- **Agt** = $\{A_1, \dots, A_k\}$: a set of *agents* (or *players*);
- **Chc**: $\text{Loc} \times \text{Agt} \rightarrow \mathbb{N}_{\geq 1}$ the choice function.
Chc(ℓ, A_i) = *number of possible moves* for A_i from ℓ .
- **Edg**: $\text{Loc} \times \mathbb{N}^k \rightarrow \text{Loc}$: the *transition table*.

Example of a CGS

Example



		Player 2		
		<i>p</i>	<i>r</i>	<i>s</i>
Player 1	<i>q</i> ₀			
	<i>p</i>	<i>q</i> ₀	<i>q</i> ₁	<i>q</i> ₂
	<i>r</i>	<i>q</i> ₂	<i>q</i> ₀	<i>q</i> ₁
<i>s</i>	<i>q</i> ₁	<i>q</i> ₂	<i>q</i> ₀	

Figure: Paper, rock and scissors

Semantics of CGSs

- From a location ℓ , each agent A_i chooses some m_{A_i} with

$$m_{A_i} < \text{Chc}(\ell, A_i).$$

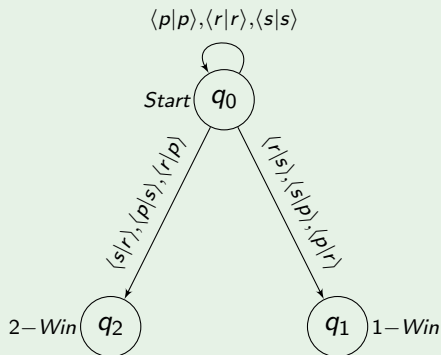
- $\text{Edg}(\ell, m_{A_1}, \dots, m_{A_k})$ gives the new location.

Notations:

- $\text{Next}(\ell) = \{\text{Edg}(\ell, \dots m_{A_i} \dots) \mid \forall m_{A_i} \cdot 1 \leq i \leq k\}$
- $\text{Next}(\ell, A_j, m) = \{\text{Edg}(\ell, \dots, m_{A_{j-1}}, m, m_{A_{j+1}} \dots)\}$

CGS example

Example

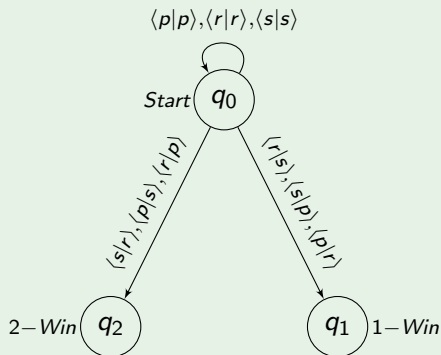


		Player 2			
		q_0	p	r	s
Player 1	p	q_0	q_1	q_2	
	r	q_2	q_0	q_1	
	s	q_1	q_2	q_0	

Figure: Paper, rock and scissors

CGS example

Example

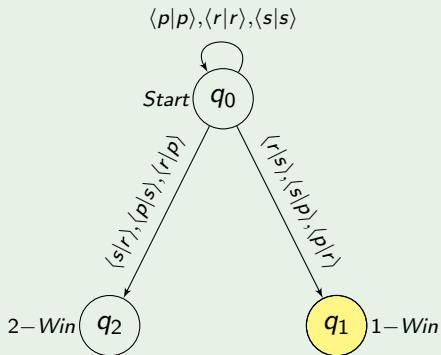


		Player 2			
		q_0	p	r	s
Player 1	q_0	q_0	q_1	q_2	q_2
	p	q_2	q_0	q_1	q_1
	r	q_1	q_2	q_0	q_1
	s	q_1	q_2	q_2	q_0

Figure: Paper, rock and scissors

CGS example

Example



		Player 2			
		q_0	p	r	s
Player 1	q_0	p	r	s	
	p	q_0	q_1	q_2	
	r	q_2	q_0	q_1	
	s	q_1	q_2	q_0	

Figure: Paper, rock and scissors

ATS definition

Definition

An *ATS* \mathcal{A} is a 4-tuple $(\text{Loc}, \text{Lab}, \text{Agt}, \text{Chc})$ where:

- Loc , Lab and Agt are the same as in CGSs;
- a *move* is a *set of locations*:

$$\text{Chc}: \text{Loc} \times \text{Agt} \rightarrow \mathcal{P}(\mathcal{P}(\text{Loc}))$$

with the following requirement: for any location ℓ and for moves $Q_i \in \text{Chc}(\ell, A_i)$,

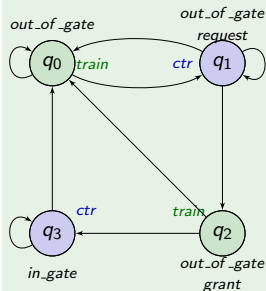
$$\bigcap_{i \leq k} Q_i \text{ must be a singleton.}$$

The *next location* is precisely the location that belongs to all the choices of the agents.

$\text{Next}(\ell)$ and $\text{Next}(\ell, A_i, m)$ are defined in the obvious way.

ATS example

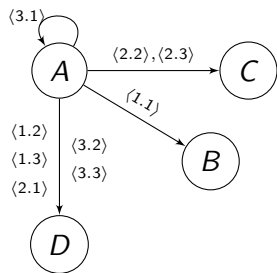
Example



- $\delta(q_0, \text{train}) = \{\{q_0\}, \{q_1\}\}$.
- $\delta(q_1, \text{ctr}) = \{\{q_0\}, \{q_1\}, \{q_2\}\}$.
- $\delta(q_2, \text{train}) = \{\{q_0\}, \{q_3\}\}$.
- $\delta(q_3, \text{ctr}) = \{\{q_0\}, \{q_3\}\}$.
- $\delta(q_0, \text{ctr}) = \delta(q_1, \text{train}) =$
 $\delta(q_2, \text{ctr}) = \delta(q_3, \text{train}) = \{\text{Loc}\}$.

Figure: Train controller

Translation $CGS \leftrightarrow ATS$



Naive approach

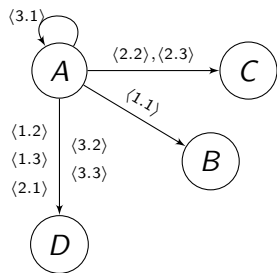
Move	Player 1	Player 2
1	{ B , D }	{ A , B , D }
2	{ C , D }	{ C , D }
3	{ A , D }	{ C , D }

Figure: Converting an CGS into an ATS

Cost of the translation:



Translation $CGS \leftrightarrow ATS$



Correct approach

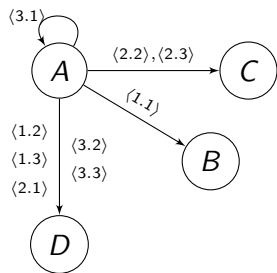
Move	Player 1	Player 2
1	$\{B_{1.1}, D_{1.2}, D_{1.3}\}$	$\{A_{3.1}, B_{1.1}, D_{2.1}\}$
2	$\{C_{2.2}, C_{2.3}, D_{2.1}\}$	$\{C_{2.2}, D_{1.2}, D_{3.2}\}$
3	$\{A_{3.1}, D_{3.2}, D_{3.3}\}$	$\{C_{2.3}, D_{1.3}, D_{3.3}\}$

Figure: Converting an CGS into an ATS

Cost of the translation:



Translation $CGS \leftrightarrow ATS$

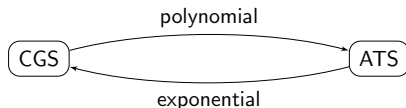


Correct approach

Move	Player 1	Player 2
1	$\{B_{1.1}, D_{1.2}, D_{1.3}\}$	$\{A_{3.1}, B_{1.1}, D_{2.1}\}$
2	$\{C_{2.2}, C_{2.3}, D_{2.1}\}$	$\{C_{2.2}, D_{1.2}, D_{3.2}\}$
3	$\{A_{3.1}, D_{3.2}, D_{3.3}\}$	$\{C_{2.3}, D_{1.3}, D_{3.3}\}$

Figure: Converting an CGS into an ATS

Cost of the translation:



Strategies and outcomes

Definition

- A **computation** is an infinite sequence $\rho = l_0 l_1 \dots$ such that $\forall i, l_{i+1} \in \text{Next}(l_i)$.
- A **strategy** is a function f_{A_i} s.t.
 $f_{A_i}(l_0, \dots, l_m) =$ a possible move for A_i from l_m .
- The **outcomes** $\text{Out}(l, f_{A_i})$ are the set of computations from l that agree with the strategy f_{A_i} of A_i .
- Notations: given $A \subseteq \text{Agt}$, we note:
 - $F_A = \{f_{A_i} \mid A_i \in A\}$
 - $\text{Out}(l, F_A)$

Syntax of ATL

Definition ([AHK97])

The syntax of **ATL** is defined by the following grammar:

$$\begin{aligned} \text{ATL} \ni \varphi_s, \psi_s &::= p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \langle\langle A \rangle\rangle \varphi_p \\ \varphi_p &::= \mathbf{X} \varphi_s \mid \mathbf{G} \varphi_s \mid \varphi_s \mathbf{U} \psi_s. \end{aligned}$$

where p ranges over the set AP and A over the subsets of Agt.

ATL subsumes CTL, since we have:

$$\mathbf{E} \varphi_p \equiv \langle\langle \text{Agt} \rangle\rangle \varphi_p,$$

$$\mathbf{A} \varphi_p \equiv \langle\langle \emptyset \rangle\rangle \varphi_p.$$

Semantics

- Semantics

$$l \models \langle\langle A \rangle\rangle \varphi_p \quad \text{iff} \quad \exists F_A \in \text{Strat}(A). \forall \rho \in \text{Out}(l, F_A). \rho \models \varphi_p$$

$$\rho \models \varphi_s \mathbf{U} \psi_s \quad \text{iff} \quad \exists i. \rho[i] \models \psi_s \text{ and } \forall 0 \leq j < i. \rho[j] \models \varphi_s$$

- We have $\langle\langle A \rangle\rangle \varphi \Rightarrow \neg \langle\langle \text{Agt} \setminus A \rangle\rangle \neg \varphi$,
but

$$\neg \langle\langle A \rangle\rangle \varphi \not\Rightarrow \langle\langle \text{Agt} \setminus A \rangle\rangle \neg \varphi.$$

- We denote $\llbracket A \rrbracket \varphi$ for $\neg \langle\langle A \rangle\rangle \neg \varphi$

Outline of the talk

- 1 Introduction
 - Overview of CTL and ATL
- 2 Definitions
 - Multi-agent models
 - Strategy and outcomes
 - ATL (Alternating-time Temporal Logic)
- 3 Expressiveness
 - Weak Until
- 4 Complexity
 - Model checking ATL on CGSs
 - Model checking ATL on ATs
- 5 Conclusion

Can ATL express weak until?

Definition

$$\varphi \mathbf{W} \psi \equiv \varphi \mathbf{U} \psi \vee \mathbf{G} \varphi$$

CTL

- $\mathbf{E} \varphi \mathbf{W} \psi \equiv \mathbf{E} \mathbf{G} \varphi \vee \mathbf{E} \varphi \mathbf{U} \psi$
- $\mathbf{A} \varphi \mathbf{W} \psi \equiv \neg \mathbf{E}(\neg \psi) \mathbf{U} (\neg \varphi \wedge \neg \psi)$

Question

Can we express $\langle\langle A \rangle\rangle a \mathbf{W} b$ in ATL?

- $\langle\langle A \rangle\rangle (\mathbf{G} \varphi \vee \varphi \mathbf{U} \psi)$ is not an ATL formula,
- $\langle\langle A \rangle\rangle \varphi \mathbf{W} \psi \not\equiv \langle\langle A \rangle\rangle \mathbf{G} \varphi \vee \langle\langle A \rangle\rangle \varphi \mathbf{U} \psi$.

Can ATL express weak until?

Definition

$$\varphi \mathbf{W} \psi \equiv \varphi \mathbf{U} \psi \vee \mathbf{G} \varphi$$

CTL

- $\mathbf{E}\varphi \mathbf{W} \psi \equiv \mathbf{E}\mathbf{G} \varphi \vee \mathbf{E}\varphi \mathbf{U} \psi$
- $\mathbf{A}\varphi \mathbf{W} \psi \equiv \neg \mathbf{E}(\neg\psi) \mathbf{U} (\neg\varphi \wedge \neg\psi)$

Question

Can we express $\langle\langle A \rangle\rangle a \mathbf{W} b$ in ATL?

- $\langle\langle A \rangle\rangle (\mathbf{G} \varphi \vee \varphi \mathbf{U} \psi)$ is not an ATL formula,
- $\langle\langle A \rangle\rangle \varphi \mathbf{W} \psi \not\equiv \langle\langle A \rangle\rangle \mathbf{G} \varphi \vee \langle\langle A \rangle\rangle \varphi \mathbf{U} \psi$.

Can ATL express weak until?

Question

Can we express $\langle\langle A \rangle\rangle a \mathbf{W} b$ in ATL?

- Answer: **No**

Theorem

Formula $\varphi = \langle\langle A \rangle\rangle a \mathbf{W} b$ cannot be expressed in ATL.

Idea:

- We present two families of models that cannot be distinguished by ATL formulae of any given size.
- One model satisfies $\langle\langle A \rangle\rangle a \mathbf{W} b$ while the other does not.

Can ATL express weak until?

Question

Can we express $\langle\langle A \rangle\rangle a \mathbf{W} b$ in ATL?

- Answer: **No**

Theorem

Formula $\varphi = \langle\langle A \rangle\rangle a \mathbf{W} b$ cannot be expressed in ATL.

Idea:

- We present two families of models that cannot be distinguished by ATL formulae of any given size.
- One model satisfies $\langle\langle A \rangle\rangle a \mathbf{W} b$ while the other does not.

Can ATL express weak until?

Question

Can we express $\langle\langle A \rangle\rangle a \mathbf{W} b$ in ATL?

- Answer: **No**

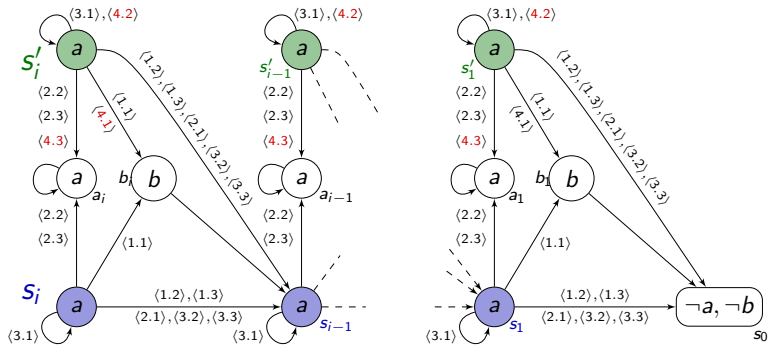
Theorem

Formula $\varphi = \langle\langle A \rangle\rangle a \mathbf{W} b$ cannot be expressed in ATL.

Idea:

- We present two families of models that cannot be distinguished by ATL formulae of any given size.
- One model satisfies $\langle\langle A \rangle\rangle a \mathbf{W} b$ while the other does not.

Can ATL express weak until?



Lemma

$\forall i > 0, \forall \psi \in \text{ATL}$ with $|\psi| \leq i$ we have: $s_i \models \psi$ iff $s'_i \models \psi$.

Outline of the talk

- 1 Introduction
 - Overview of CTL and ATL
- 2 Definitions
 - Multi-agent models
 - Strategy and outcomes
 - ATL (Alternating-time Temporal Logic)
- 3 Expressiveness
 - Weak Until
- 4 Complexity
 - Model checking ATL on CGSs
 - Model checking ATL on ATsS
- 5 Conclusion

ATL model checking over CGSs

Theorem ([AHK02])

Model checking ATL over CGSs is *PTIME-complete*.

- $Pre(A, L) = \{\ell \mid \exists m_A \cdot \text{Next}(\ell, A, m_A) \subseteq L\}$

$$\varphi = \langle\langle A \rangle\rangle \theta_1 \text{ U } \theta_2$$

$L := [false]; T := [\theta_2];$

while $T \not\subseteq L$ **do**

$L := L \cup T;$

$T := Pre(A, L) \cap [\theta_1]$

od;

$[\varphi] := T$

Overall complexity: $O(|\text{Edg}| \cdot |\varphi|)$, thus *PTIME*.

Implicit CGS

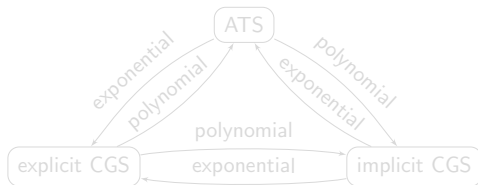
Definition

An **implicit CGS** is a CGS where:

- The transition function: in each ℓ it's given $((\varphi_0, \ell_0), \dots, (\varphi_n, \ell_n))$ where $\ell_i \in \text{Loc}$, φ_i is a boolean combination of propositions $A_j = c$.
- $\text{Edg}(\ell, m_{A_1}, \dots, m_{A_k}) = \ell_j$ s.t.

$$j = \min(i \mid \varphi_i(\ell, m_{A_1}, \dots, m_{A_k}) = \top).$$

- It is required that the last formula, φ_n , be \top .



Implicit CGS

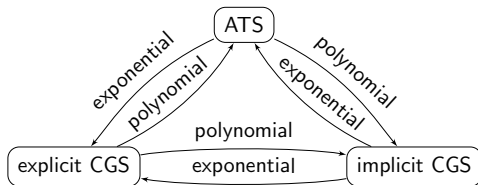
Definition

An **implicit CGS** is a CGS where:

- The transition function: in each ℓ it's given $((\varphi_0, \ell_0), \dots, (\varphi_n, \ell_n))$ where $\ell_i \in \text{Loc}$, φ_i is a boolean combination of propositions $A_j = c$.
- $\text{Edg}(\ell, m_{A_1}, \dots, m_{A_k}) = \ell_j$ s.t.

$$j = \min(i \mid \varphi_i(\ell, m_{A_1}, \dots, m_{A_k}) = \top).$$

- It is required that the last formula, φ_n , be \top .



ATL model checking over implicit CGSs

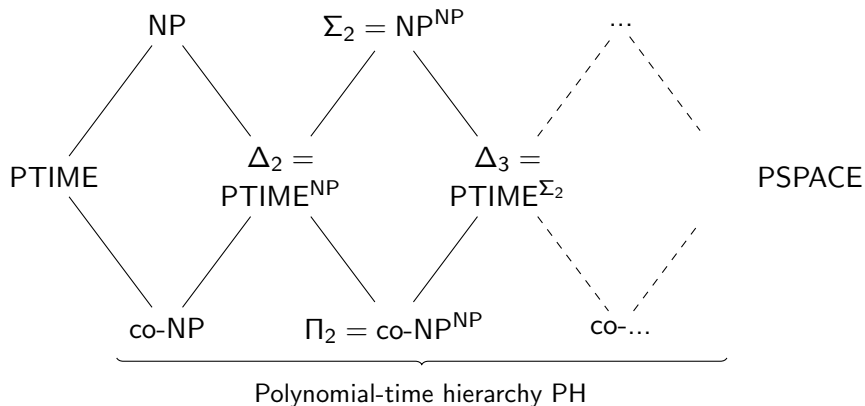
Theorem

Model checking ATL over implicit CGSs in Δ_3 -complete.

ATL model checking over implicit CGSs

Theorem

Model checking ATL over implicit CGSs in Δ_3 -complete.



ATL model checking over implicit CGSs

Theorem

Model checking ATL over implicit CGSs in Δ_3 -complete.

Membership in Δ_3 .

- Σ_2 algorithm proposed in [JD05]: correctly handles positive formulas (*i.e.* of the form $\langle\langle A \rangle\rangle \varphi$).
- That algorithm is used as an oracle, called a polynomial number of times.

ATL model checking over implicit CGSs

Theorem

Model checking ATL over implicit CGSs in Δ_3 -complete.

Hardness in Σ_2 . [JD05]

EQSAT₂:

Input: a boolean formula φ over variables in $X \cup Y$.

Output: true iff $\exists X. \forall Y. \varphi(X, Y)$

ATL model checking over implicit CGSs

Theorem

Model checking ATL over implicit CGSs in Δ_3 -complete.

Hardness in Σ_2 . [JD05]

EQSAT₂:

Input: a boolean formula φ over variables in $X \cup Y$.

Output: true iff $\exists X. \forall Y. \varphi(X, Y)$

\rightsquigarrow 1 player A_i per variable in X , 1 player B_j per variable in Y .

ATL model checking over implicit CGSs

Theorem

Model checking ATL over implicit CGSs in Δ_3 -complete.

Hardness in Σ_2 . [JD05]

EQSAT₂:

Input: a boolean formula φ over variables in $X \cup Y$.

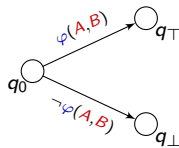
Output: true iff $\exists X. \forall Y. \varphi(X, Y)$

\rightsquigarrow 1 player A_i per variable in X , 1 player B_j per variable in Y .

Lemma

The instance of EQSAT₂ is positive iff

$$q_0 \models \langle\langle A_1, \dots, A_n \rangle\rangle \mathbf{X} q_{\top}.$$



ATL model checking over implicit CGSs

Theorem

Model checking ATL over implicit CGSs in Δ_3 -complete.

Hardness in Π_2 .

AQSAT₂:

Input: a boolean formula φ over variables in $X \cup Y$.

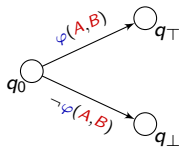
Output: true iff $\forall X. \exists Y. \varphi(X, Y)$

\rightsquigarrow 1 player A_i per variable in X , 1 player B_j per variable in Y .

Lemma

The instance of AQSAT₂ is positive iff

$$q_0 \models \neg \langle\langle A_1, \dots, A_n \rangle\rangle \mathbf{X} q_{\perp}.$$



ATL model checking over implicit CGSs

Theorem

Model checking ATL over implicit CGSs in Δ_3 -complete.

Hardness in Δ_3 (sketch).

SNSAT₂:

Input: formulas φ_i over variables in $X_i \cup Y_i \cup \{z_1, \dots, z_{i-1}\}$.

Output: the value of z_m defined by:

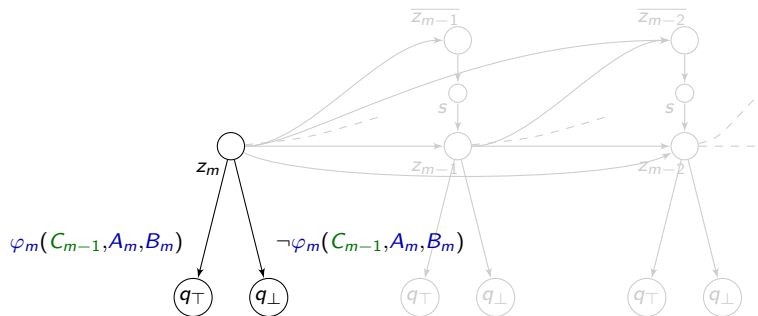
$$\left\{ \begin{array}{l} z_1 \stackrel{\text{def}}{=} \exists X_1. \forall Y_1. \varphi_1(X_1, Y_1) \\ z_2 \stackrel{\text{def}}{=} \exists X_2. \forall Y_2. \varphi_2(z_1, X_2, Y_2) \\ z_3 \stackrel{\text{def}}{=} \exists X_3. \forall Y_3. \varphi_3(z_1, z_2, X_3, Y_3) \\ \dots \\ z_m \stackrel{\text{def}}{=} \exists X_m. \forall Y_m. \varphi_m(z_1, \dots, z_{m-1}, X_m, Y_m) \end{array} \right.$$

ATL model checking over implicit CGSs

Theorem

Model checking ATL over implicit CGSs in Δ_3 -complete.

Hardness in Δ_3 (sketch).



$$\psi_m = \langle\langle AC \rangle\rangle (\neg s) \mathbf{U} (q \wedge \mathbf{EX} (s \wedge \mathbf{EX} \neg\psi_{m-1})).$$

ATL model checking over ATSs

Theorem ([AHK97])

Model checking ATL over ATSs is *PTIME-complete*.

Proof. Similar to the case of CGSs. □

But... Transitions of an ATS are not given explicitly.

The algorithm is polynomial *in the size of the underlying CGS* (which might be *exponential*).

Theorem ([JD05])

Model checking ATL over ATSs is *PTIME-complete if the number of agents is fixed*.

ATL model checking over ATSs

Theorem ([AHK97])

Model checking ATL over ATSs is *PTIME-complete*.

Proof. Similar to the case of CGSs. □

But... Transitions of an ATS are not given explicitly.

The algorithm is polynomial *in the size of the underlying CGS* (which might be *exponential*).

Theorem ([JD05])

Model checking ATL over ATSs is *PTIME-complete* if the number of agents is fixed.

ATL model checking over ATSs

Theorem ([AHK97])

Model checking ATL over ATSs is *PTIME-complete*.

Proof. Similar to the case of CGSs. □

But... Transitions of an ATS are not given explicitly.

The algorithm is polynomial *in the size of the underlying CGS* (which might be *exponential*).

Theorem ([JD05])

Model checking ATL over ATSs is *PTIME-complete if the number of agents is fixed*.

ATL model checking over ATSSs

Theorem

Model checking ATL over ATSSs is Δ_2 -complete.

ATL model checking over ATSSs

Theorem

Model checking ATL over ATSSs is Δ_2 -complete.

NP-hardness: Reduction from 3-SAT.

ATL model checking over ATs

Theorem

Model checking ATL over ATs is Δ_2 -complete.

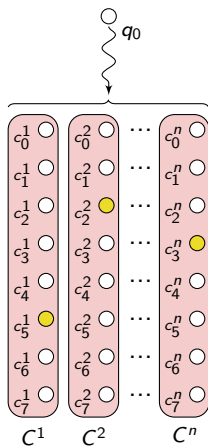
NP-hardness: Reduction from 3-SAT.

$$C = p \vee \neg q \vee r \quad \rightsquigarrow \quad \left\{ \begin{array}{l} c_0 = \neg p \vee \neg q \vee \neg r \\ c_1 = \neg p \vee \neg q \vee r \\ c_2 = \neg p \vee q \vee \neg r \\ c_3 = \neg p \vee q \vee r \\ c_4 = p \vee \neg q \vee \neg r \\ c_5 = p \vee \neg q \vee r \\ c_6 = p \vee q \vee \neg r \\ c_7 = p \vee q \vee r \end{array} \right.$$

ATL model checking over ATSS

Theorem

Model checking ATL over ATSS is Δ_2 -complete.



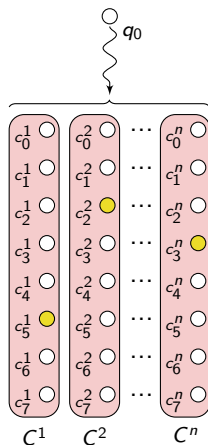
ATL model checking over ATSSs

Theorem

Model checking ATL over ATSSs is Δ_2 -complete.

1 player (P_1 to P_k) per atomic proposition:

- $p \rightsquigarrow \{c_j^i \mid c_j^i \text{ not made true by } p\}$
- $\neg p \rightsquigarrow \{c_j^i \mid c_j^i \text{ not made true by } \neg p\}$



ATL model checking over ATs

Theorem

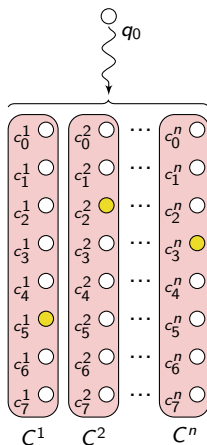
Model checking ATL over ATs is Δ_2 -complete.

1 player (P_1 to P_k) per atomic proposition:

- $p \rightsquigarrow \{c_j^i \mid c_j^i \text{ not made true by } p\}$
- $\neg p \rightsquigarrow \{c_j^i \mid c_j^i \text{ not made true by } \neg p\}$

Once those players have chosen their moves, **exactly one** clause c_j^i per original clause C^i belongs to the intersection of the chosen sets.

$$\text{E.g. } \left. \begin{array}{l} p = \top \\ q = \top \\ r = \perp \end{array} \right\} \Rightarrow \neg p \wedge \neg q \wedge r$$



ATL model checking over ATSS

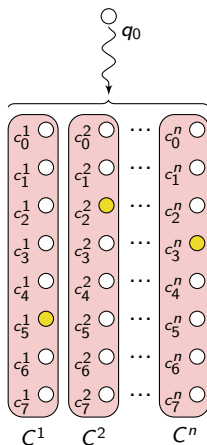
Theorem

Model checking ATL over ATSS is Δ_2 -complete.

1 player (P_1 to P_k) per atomic proposition:

- $p \rightsquigarrow \{c_j^i \mid c_j^i \text{ not made true by } p\}$
- $\neg p \rightsquigarrow \{c_j^i \mid c_j^i \text{ not made true by } \neg p\}$

1 extra player chooses one set among $\{c_0^1, \dots, c_7^1\}$ to $\{c_0^n, \dots, c_7^n\}$



ATL model checking over ATs

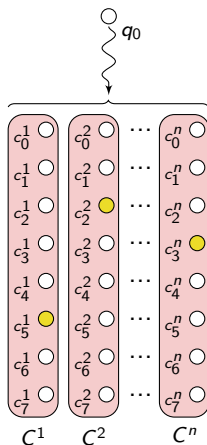
Theorem

Model checking ATL over ATs is Δ_2 -complete.

1 player (P_1 to P_k) per atomic proposition:

- $p \rightsquigarrow \{c_j^i \mid c_j^i \text{ not made true by } p\}$
- $\neg p \rightsquigarrow \{c_j^i \mid c_j^i \text{ not made true by } \neg p\}$

1 extra player chooses one set among $\{c_0^1, \dots, c_7^1\}$ to $\{c_0^n, \dots, c_7^n\}$



Lemma

The 3-SAT instance is true iff

$$q_0 \models \langle\langle P_1, \dots, P_k \rangle\rangle \mathbf{X} \neg \bullet$$

Conclusion

Expressiveness

$$\text{ATL}_W > \text{ATL}$$

Complexity results

	$\text{CGS}_{\text{fixed}}$	CGS	ATS
ATL	PTIME	Δ_3	Δ_2
ATL^+	Δ_3	Δ_3	Δ_3

Future Work

- Fairness constraints
- Timed models