Logical Aspects of Artificial Intelligence
Introduction to ATL (Part II)

Stéphane Demri  
demri@lsv.fr

October 27th, 2021
Plan of the lecture

- Recapitulation of previous lecture.
- Introduction to ATL*.
- Model-checking algorithm for ATL in PTIME.
- CGS with imperfect information.
- Exercises session.
Recapitulation of the previous lecture
Concurrent game structure: an example

\[ \begin{align*}
\text{Agt} &= \{1, 2\} \\
S &= \{s_1, s_2, s_3, s_4\} \\
\text{Act} &= \{a, b, c\}
\end{align*} \]

- **Action manager** \( \text{act} : \text{Agt} \times S \rightarrow \mathcal{P}(\text{Act}) \setminus \{\emptyset\} \).
  \( \text{act}(1, s_3) = \{c\} \).

- **Transition function** \( \delta : S \times (\text{Agt} \rightarrow \text{Act}) \rightarrow S \).
  \( \delta(s_4, [1 \mapsto c, 2 \mapsto c]) = s_3 \) — undef.
  \( \delta(s_4, [1 \mapsto c, 2 \mapsto a]) \).

- **Labelling** \( L : S \rightarrow \mathcal{P}(\text{PROP}) \).
Basic concepts: strategies

- \( f : A \rightarrow \text{Act} \): joint action by \( A \subseteq \text{Agt} \) in \( s \).
  \( f \) can be viewed as a tuple of actions of length \( \text{card}(A) \).

- \( D_A(s) \): set of joint actions by \( A \) in \( s \).

- Set of outcomes:
  \[
  \text{out}(s, f) \overset{\text{def}}{=} \{ s' \in S \mid \exists g \in D_{\text{Agt}}(s) \text{ s.t. } f \sqsubseteq g \text{ } \& \text{ } s' = \delta(s, g) \}
  \]
  \[
  (a_1, a_2, -, -, -) \sqsubseteq (a_1, a_2, a_3, a_4)
  \]

- Strategy \( F_A \) for \( A \): map from the set of histories to the set of joint actions by \( A \) such that
  \[
  F_A(s_0 \xrightarrow{f_0} s_1 \cdots \xrightarrow{f_{n-1}} s_n) \in D_A(s_n)
  \]
Basic concepts: computations

- Computation $\lambda = s_0 \xrightarrow{f_0} s_1 \xrightarrow{f_1} s_2 \ldots$ such that for all $i$, we have $s_{i+1} \in \delta(s_i, f_i)$. 
  ($f_i \in D_{Agt}(s_i)$)

- History = finite computation.

- For all ATL-like logics, we can restrict ourselves to computations of the form $s_0 s_1 s_2 \ldots$ (without joint actions).

- Linear model $L(s_0) \rightarrow L(s_1) \rightarrow L(s_2) \ldots$ 
  ($\omega$-sequence of propositional valuations)

- Notations with $\lambda = s_0 s_1 s_2 \ldots$
  - $\lambda(i) \overset{\text{def}}{=} s_i$.
  - $\lambda[i, \infty) \overset{\text{def}}{=} s_i s_{i+1} \ldots$. 
Alternating-time temporal Logic ATL

\[ \varphi ::= \top | p | \neg \varphi | \varphi \land \varphi | \langle \langle A \rangle \rangle X \varphi | \langle \langle A \rangle \rangle G \varphi | \langle \langle A \rangle \rangle \varphi \backslash U \varphi \]

\[ \langle \langle A \rangle \rangle F \varphi \overset{\text{def}}{=} \langle \langle A \rangle \rangle \top \backslash U \varphi. \]

\[ M, s \models \langle \langle A \rangle \rangle X \varphi \iff \text{there is } F_A \text{ s.t. for all } \lambda \in \text{Comp}(s, F_A), \]
we have \( M, \lambda(1) \models \varphi \)

\[ M, s \models \langle \langle A \rangle \rangle \varphi_1 \backslash U \varphi_2 \iff \text{there is a strategy } F_A \text{ s.t. for all } \lambda = s_0 \overset{f_A}{\rightarrow} s_1 \ldots \in \text{Comp}(s, F_A), \]
there is some \( i \) s.t. \( M, s_i \models \varphi_2 \)
and for all \( j \in [0, i - 1] \),
we have \( M, s_j \models \varphi_1 \).

\[ M, s \models \langle \langle A \rangle \rangle G \varphi \iff \text{there is a strategy } F_A \text{ s.t. for all } \lambda = s_0 \overset{f_A}{\rightarrow} s_1 \ldots \in \text{Comp}(s, F_A), \]
for all \( i \), we have \( M, s_i \models \varphi \).
Decision problems

- Model-checking problem for ATL:
  \[\text{Input: } \varphi \text{ in ATL, a finite CGS } \mathcal{M} \text{ and a state } s,\]
  \[\text{Question: } \mathcal{M}, s \models \varphi?\]

- Model-checking problem for ATL is \(\text{PTIME}\)-complete. (see forthcoming labelling algorithm)

- Satisfiability problem for ATL:
  \[\text{Input: } \varphi \text{ in ATL,}\]
  \[\text{Question: Is there a CGS } \mathcal{M} \text{ and } s \text{ in } \mathcal{M} \text{ such that } \mathcal{M}, s \models \varphi?\]

- Satisfiability and validity problems are \(\text{EXP\hspace{1pt}TIME}\)-complete.

- Positional strategies are sufficient for ATL!
Predecessor operator \text{pre}

\begin{itemize}
\item CGS $\mathcal{M} = (\text{Agt}, S, \text{Act}, \text{act}, \delta, L)$, $A \subseteq \text{Agt}$, and $Z \subseteq S$.
\item $\text{pre}(\mathcal{M}, A, Z)$: set of states from which $A$ has a collective move that guarantees that the outcome to be in $Z$.
\item Definition of $\text{pre}(\mathcal{M}, A, \cdot)$: $\mathcal{P}(S) \to \mathcal{P}(S)$
\end{itemize}
\begin{align*}
\text{pre}(\mathcal{M}, A, Z) & \stackrel{\text{def}}{=} \\
& \{ s \in S \mid \text{there is } f \in D_A(s) \text{ such that } \text{out}(s, f) \subseteq Z \}
\end{align*}
\begin{itemize}
\item $\lbrack \langle A \rangle X \varphi \rbrack^\mathcal{M} = \text{pre}(\mathcal{M}, A, [\varphi]^\mathcal{M})$.
\end{itemize}
Knaster-Tarski Theorem: a restricted form

- Knaster-Tarski Theorem (a restricted form). Let $G : \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ be a monotone operator. Then: $G$ has a least fixpoint $\mu G$ and a greatest fixpoint $\nu G$.

- $\mu G$ obtained by applying the successive iterations of $G$ beginning with $\emptyset$ until a fixpoint is reached.
  
  
  $\emptyset \subseteq G(\emptyset) \subseteq G^2(\emptyset) \subseteq G^3(\emptyset) \cdots$

- $\nu G$ obtained by applying the successive iterations of $G$, beginning with $X$, until a fixpoint is reached.
  
  $X \supseteq G(X) \supseteq G^2(X) \supseteq G^3(X) \cdots$

- When $X$ is finite, then the number of steps to reach a fixpoint is at most $\text{card}(X)$. 


Characterisation with fixpoints

Given $A \subseteq \text{Agt}$, a formula $\varphi$, and a CGS $\mathcal{M}$, we define

$G_{A,\varphi} : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$:

$G_{A,\varphi}(Z) \overset{\text{def}}{=} \lceil \varphi \rceil^\mathcal{M} \cap \text{pre}(\mathcal{M}, A, Z)$.

$\lceil \langle A \rangle G \varphi \rceil^\mathcal{M} = \nu Z. (\lceil \varphi \rceil^\mathcal{M} \cap \text{pre}(\mathcal{M}, A, Z))$

(greatest fixpoint)

Given $A \subseteq \text{Agt}$, formulae $\varphi, \psi$, and a CGS $\mathcal{M}$, we define

$O_{A,\varphi,\psi} : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$:

$O_{A,\varphi,\psi}(Z) \overset{\text{def}}{=} \lceil \psi \rceil^\mathcal{M} \cup \left( \lceil \varphi \rceil^\mathcal{M} \cap \text{pre}(\mathcal{M}, A, Z) \right)$

$\lceil \langle A \rangle \varphi U \psi \rceil^\mathcal{M} = \mu Z. (\lceil \psi \rceil^\mathcal{M} \cup (\lceil \varphi \rceil^\mathcal{M} \cap \text{pre}(\mathcal{M}, A, Z)))$

(least fixpoint)
ATL* in a nutshell
**ATL**: including all LTL-like path formulae

- **Strategy modalities in ATL**:
  
  \[
  \langle A \rangle X \varphi \quad \langle A \rangle G \varphi \quad \langle A \rangle \varphi U \varphi
  \]

- **LTL interpreted on computations (ω-sequences)**:
  \[
  \varphi, \psi ::= p \mid \neg \varphi \mid \varphi \land \psi \mid \varphi \lor \psi \mid X \varphi \mid \varphi U \psi
  \]

- **\langle A \rangle (Gp \lor (\varphi_1 U \varphi_2))** not an ATL formula ! (why?)

- **ATL**: extension of ATL with LTL-like path formulae. (standard CTL* extends CTL similarly)

- **ATL** distinguishes **path formulae** from **state formulae**

State formulae: \[
\varphi ::= p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \langle A \rangle \Phi
\]

Path formulae: \[
\Phi ::= \varphi \mid \neg \Phi \mid (\Phi \land \Phi) \mid X \Phi \mid G \Phi \mid (\Phi U \Phi)
\]
Examples of properties in ATL

- $\langle A \rangle \text{GF } \Phi$: “The coalition $A$ has a collective strategy so that infinitely often $\Phi$ is satisfied, on every computation respecting that strategy”.

- $\langle A \rangle (\text{F } \Phi \land \text{F } \Psi)$: “The coalition $A$ has a collective strategy to eventually reach an outcome satisfying $\Phi$ and to eventually reach an outcome satisfying $\Psi$, on every computation respecting that strategy”.

- $\langle A \rangle \text{G } \Phi$ equivalent to $\langle A \rangle \neg (\top \text{U} \neg \Phi)$. 
Satisfaction relations for ATL*

\( M, s \models p \quad \overset{\text{def}}{\iff} \quad p \in L(s) \)

\( M, s \models \langle A \rangle \Phi \quad \overset{\text{def}}{\iff} \quad \text{there is a strategy } F_A \text{ s.t.} \)

\( \text{for all } \lambda = s_0 \xrightarrow{f_0} s_1 \ldots \in \text{Comp}(s, F_A), \)

\( \text{we have } M, \lambda \models \Phi \)

\( M, \lambda \models \varphi \quad \overset{\text{def}}{\iff} \quad M, \lambda(0) \models \varphi, \text{ for every state formula } \varphi \)

\( M, \lambda \models \psi_1 \land \psi_2 \quad \overset{\text{def}}{\iff} \quad M, \lambda \models \psi_1 \text{ and } M, \lambda \models \psi_2 \)

\( M, \lambda \models X\psi \quad \overset{\text{def}}{\iff} \quad M, \lambda[1, \infty) \models \psi \)

\( M, \lambda \models G\psi \quad \overset{\text{def}}{\iff} \quad M, \lambda[i, \infty) \models \psi \text{ holds for all positions } i \geq 0 \)

\( M, \lambda \models \psi_1 U \psi_2 \quad \overset{\text{def}}{\iff} \quad \text{there is } i \geq 0 \text{ s.t. } M, \lambda[i, \infty) \models \psi_2 \text{ and} \)

\( M, \lambda[j, \infty) \models \psi_1 \text{ holds for all } 0 \leq j < i \)
Positional strategies are not enough for ATL*!

- $M, s_2 \models \langle\text{Agt}\rangle (Fp \land Fq)$.

- ... but $M, s_2 \not\models \langle\text{Agt}\rangle (Fp \land Fq)$ with positional strategies only.
Model-checking problem

Model-checking problem for ATL*:  
**Input:** $\varphi$ in ATL*, a finite CGS $\mathcal{M}$ and a state $s$,  
**Question:** $\mathcal{M}, s \models \varphi$?

Reminder: model-checking problem for ATL is PTIME-complete.  
(forthcoming labelling algorithm)

Model-checking problem for ATL* is 2EXPTIME-complete.  
(non-positional strategies are required)

CTL* = fragment of ATL*.  
E.g., $\langle \text{Agt} \rangle \Phi \approx \text{E}\Phi$, $\langle \emptyset \rangle \Phi \approx \text{A}\Phi$, . . .

Model-checking problem for CTL* is PSPACE-complete.
ATL$^+$

- ATL$^+$ is a formalism between ATL and ATL$^*$. Similarly to CTL$^+$ between CTL and CTL$^*$

- $\langle A \rangle (Fp \land Fq)$ in ATL$^+$ but not in ATL.

- ATL$^+$ distinguishes also path formulae from state formulae

  State formulae: $\varphi ::= p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \langle A \rangle \Phi$

  Path formulae: $\Phi ::= \neg \Phi \mid (\Phi \land \Phi) \mid X\varphi \mid G\varphi \mid (\varphi U \varphi)$

- Model-checking problem for ATL$^+$ is PSPACE-complete.
ATL$_r^*$: ATL$^*$ with memoryless strategies

- In ATL$_r^*$, the strategies witnessing satisfaction of $\mathcal{M}, s \models \langle A \rangle \Phi$ are memoryless.

- Checking whether $\mathcal{M}, s \models \langle A \rangle \Phi$ done as follows:
  1. Guess a strategy $F: S \rightarrow$ set of joint actions for $A$.
  2. Shrink $\mathcal{M}$ accordingly using $F$, leading to $\mathcal{M}'$.
  3. Assuming for each outermost state formula $\psi$ in $\Phi$, $[\psi]^\mathcal{M}$ already computed, $\mathcal{M}'' \overset{\text{def}}{=} \mathcal{M}[\rho^\text{new}_\psi \leftarrow [\psi]^\mathcal{M}]$.
  4. Check whether $\mathcal{M}'', s \models A \Phi[\rho^\text{new}_\psi / \psi]$ (CTL$^*$ model-checking instance).

- Ingredients for model-checking problem for ATL$_r^*$ in PSPACE
  - CTL$^*$ model-checking in PSPACE,
  - NPSPACE = PSPACE,
  - polynomial number of requests $\mathcal{M}, s \models \langle A \rangle \Phi$,
  - PTIME$^{PSPACE} = PSPACE$. 

Model-checking algorithm for ATL in PTIME
Principles of the labelling algorithm

- GMC($\mathcal{M}$, $\varphi$) computes the set $[\varphi]^\mathcal{M} \subseteq S$.

- Recursion depth linear in the size of $\varphi$.

- Dynamic programming is used to store GMC($\mathcal{M}$, $\varphi$).
  (omitted in the algorithm on the next slide).

- GMC($\mathcal{M}$, $\langle A \rangle \psi_1 U \psi_2$) and GMC($\mathcal{M}$, $\langle A \rangle G \psi$) computed thanks to their respective fixpoint characterisation.

- By structural induction, one can show that $s \in \text{GMC}(\mathcal{M}, \psi)$ iff $s \in [\psi]^\mathcal{M}$
procedure GMC(\(\mathcal{M}, \varphi\))

case \(\varphi\) of

\(\rho\): return \(\{s \in S \mid p \in L(s)\}\) ▷ some atomic proposition

\(\neg\psi\): return \(S \setminus \text{GMC}(\mathcal{M}, \psi)\)

\(\psi_1 \lor \psi_2\): return \(\text{GMC}(\mathcal{M}, \psi_1) \cup \text{GMC}(\mathcal{M}, \psi_2)\)

\(\langle A\rangle X\psi\): return \(\text{pre}(\mathcal{M}, A, \text{GMC}(\mathcal{M}, \psi))\)

\(\langle A\rangle G\psi\):

\(X \leftarrow S; Y \leftarrow \text{GMC}(\mathcal{M}, \psi)\);

▷ \(X\): previous value; \(Y\): next value

while \(X \not\subseteq Y\) do ▷ equivalent to \(X \neq Y\) as always \(Y \subseteq X\)

\(X \leftarrow Y; Y \leftarrow \text{pre}(\mathcal{M}, A, X) \cap \text{GMC}(\mathcal{M}, \psi)\)

end while; return \(Y\)

\(\langle A\rangle \psi_1 U \psi_2\):

\(X \leftarrow \emptyset; Y \leftarrow \text{GMC}(\mathcal{M}, \psi_2)\);

▷ \(X\): previous value; \(Y\): next value

while \(Y \not\subseteq X\) do ▷ equivalent to \(X \neq Y\) as always \(X \subseteq Y\)

\(X \leftarrow Y;\)

\(Y \leftarrow \text{GMC}(\mathcal{M}, \psi_2) \cup (\text{pre}(\mathcal{M}, A, X) \cap \text{GMC}(\mathcal{M}, \psi_1))\)

end while; return \(Y\)

end case

end procedure
Dynamic programming

- Array $T$ where $T[\psi]$ takes either the value $\perp$ (undefined) or a subset of $S$.

- Initially, all the values of $T$ are undefined.

- Whenever GMC($\mathcal{M}, \psi$) is invoked in the algorithm, we operate a slight change in the code: we first check whether $T[\psi]$ is defined.

- In the case $T[\psi]$ is undefined, a recursive call GMC($\mathcal{M}, \psi$) is performed.

- This technique is standard and herein we use it so that for each subformula $\psi$, GMC($\mathcal{M}, \psi$) is called at most once.
Computing $\lhd \{2\} \rhd Xq \land \lhd \{1\} \rhd pUq \rhd M$

- $[q]^M = \{s_3\}$ and $[p]^M = \{s_1, s_2\}$.

- For the first conjunct: $[\lhd \{2\} \rhd Xq]^M = \text{pre}(M, \{2\}, [q]^M)$.

- $\text{pre}(M, \{2\}, [q]^M) = \text{pre}(M, \{2\}, \{s_3\}) = \{s_3, s_4\}$. 
Computing $\{\langle\{2\}\rangle X q \land \langle\{1\}\rangle p U q\}^m$ (bis)

$J_{\langle\{2\}\rangle X q}^m = \{s_3, s_4\}$ ; $J_q^m = \{s_3\}$ and $J_p^m = \{s_1, s_2\}$.

$X_0 = [q]^m \cup (\text{pre}(M, \{1\}, \emptyset) \cap [p]^m) = \{s_3\}$.

$X_1 = [q]^m \cup (\text{pre}(M, \{1\}, X_0) \cap [p]^m) = \{s_3, s_2\}$.

$X_2 = [q]^m \cup (\text{pre}(M, \{1\}, X_1) \cap [p]^m) = \{s_3, s_2, s_1\}$.

$X_3 = [q]^m \cup (\text{pre}(M, \{1\}, X_2) \cap [p]^m) = X_2$.

$\{\langle\{2\}\rangle X q \land \langle\{1\}\rangle p U q\}^m = [\langle\{2\}\rangle X q]^m \cap [\langle\{1\}\rangle p U q]^m = \{s_3\}$
Complexity analysis

- Recursion depth linear in the depth of $\varphi$.
- Dynamic programming to compute each $\text{GMC}(M, \psi)$ only once.
- The model-checking problem for ATL is $\text{PTIME}$-complete.
Model-checking problem for ATL in PTIME

- **GMC(ℳ, ϕ)** can be computed in PTIME.

- **T[ψ]** is computed for a polynomial amount of subformulae ψ of ϕ.

- In the worst-case, computing **T[ψ]** requires a number of steps in \( O(\text{card}(S)) \) and each step requires polynomial time in size(ℳ).

- The most expensive cases are for \( ⟨⟨A⟩⟩Gψ \) and \( ⟨⟨A⟩⟩ψ_1 Uψ_2 \).
Imperfect Information
Imperfect information in CGS

- So far, the coalitions are completely aware of the structure of the CGS as well as the current state of the play.

- In concrete multi-agent systems, it is more common that the agents have only partial information.

- Imperfect information: uncertainty about the current state of the CGS (i.e. game structure).

- Incomplete information: uncertainty about the CGS structure.
Imperfect information: tossing a coin

$S_0 \xrightarrow{\text{head}} S_2 \xrightarrow{\text{tail}} S_0$
$S_0 \xrightarrow{\text{tail}} S_2 \xrightarrow{\text{head}} S_0$

$S_0 \xrightarrow{\text{idle}} \text{win}$
$S_0 \xrightarrow{\text{idle}} \text{lose}$

So $\ll 1 \gg F_{\text{win}}$

So $\nmid \ll 1 \gg F_{\text{win}}$ with imperfect information
Concurrent game structures with imperfect information: iCGS

\[ M = (Agt, S, (\sim_a)_{a \in Agt}, Act, act, \delta, L) \]

- \( Agt, S, Act, \delta, L \) as in a CGS.

- For each \( a \in Agt \), \( \sim_a \) is an equivalence relation on \( S \) understood as an indistinguishability relation for the agent \( a \).

- \( act : Agt \times S \rightarrow \mathcal{P}(Act) \setminus \{\emptyset\} \) as in CGS except that \( s \sim_a s' \) implies \( act(a, s) = act(a, s') \) for all \( a, s \) and \( s' \). ("indistinguishable states perform the same actions, given an agent \( a \)"")
Uniform strategies

- $\sim A \overset{\text{def}}{=} \bigcap_{a \in \text{Agt}} \sim a$.

\[
h = s_0 \xrightarrow{f_0} s_1 \cdots \xrightarrow{f_{n-1}} s_n \sim_A h' = s'_0 \xrightarrow{f'_0} s'_1 \cdots \xrightarrow{f'_{n-1}} s'_n
\]

$\overset{\text{def}}{\iff}$

for all $i, s_i \sim_A s'_i$

- **Memoryless uniform strategy** $F_A$:
  $s \sim_A s'$ implies $F_A(s) = F_A(s')$.

- **Memoryful uniform strategy** $F_A$:
  $h \sim_A h'$ implies $F_A(h) = F_A(h')$. 
**ATL\textsubscript{ir}: ATL with imperfect information and memoryless strategies**

- \( M, s \models \langle \langle A \rangle \rangle \Phi \iff \) there is memoryless uniform strategy \( F_A \) for \( A \) such that for all \( \lambda \in \text{Comp}(s, F_A) \), we have \( M, \lambda \models \Phi \).

- \( \langle \langle A \rangle \rangle \text{G} \varphi \Rightarrow \varphi \land \langle \langle A \rangle \rangle \text{X} \langle \langle A \rangle \rangle \text{G} \varphi \) is valid in ATL\textsubscript{ir}.

- \( \varphi \land \langle \langle A \rangle \rangle \text{X} \langle \langle A \rangle \rangle \text{G} \varphi \Rightarrow \langle \langle A \rangle \rangle \text{G} \varphi \) is not valid in ATL\textsubscript{ir}.

- The model-checking problem for ATL\textsubscript{ir} is \( \Delta^p_2 \)-complete.

(\( \Delta^p_2 \): class of problems solvable in polynomial time with a deterministic Turing machine calling an oracle solving NP problems)
\( \varphi \wedge \langle A \rangle X \langle A \rangle G \varphi \Rightarrow \langle A \rangle G \varphi \) not valid for ATL_{ir}
Complexity of model-checking problems

- I: perfect information (CGS) ; i: imperfect information (iCGS + uniform strategies)

- R: memoryful strategies ; r: memoryless strategies.

Perfect information:
- \( \text{ATL} = \text{ATL}_{IR} = \text{ATL}_{ir} \): PTIME-complete.
- \( \text{ATL}^*_{IR} = \text{ATL}^*: 2\text{EXPTIME}-\text{complete} \).
- \( \text{ATL}^*_{ir} \): PSPACE-complete.

Imperfect information:
- \( \text{ATL}_{ir} \): \( \Delta^p_2 \)-complete.
- \( \text{ATL}^*_{ir} \): PSPACE-complete.
- \( \text{ATL}^*_{iR}, \text{ATL}_{iR} \): undecidable.
Model-checking: $\Delta^p_2$ upper bound

- In $\text{ATL}_{ir}$, the strategies witnessing satisfaction of $\mathcal{M}, s \models \langle A \rangle \Phi$ are memoryless and uniform.

- Checking whether $\mathcal{M}, s \models \langle A \rangle \Phi$ done as follows:
  1. Guess a memoryless and uniform strategy $F$.
  2. Shrink $\mathcal{M}$ accordingly using $F$, leading to $\mathcal{M}'$.
  3. Assuming for each outermost state formula $\psi$ in $\Phi$, $[\psi]^{\mathcal{M}}$ already computed, $\mathcal{M}'' \overset{\text{def}}{=} \mathcal{M}[p_{\psi}^{\text{new}} \leftarrow [\psi]^{\mathcal{M}}]$.
  4. Check whether $\mathcal{M}'', s \models A \Phi[p_{\psi}^{\text{new}} / \psi]$
     (CTL model-checking instance).

- The model-checking problem for $\text{ATL}_{ir}$ is in $\Delta^p_2$.

- Three main ingredients:
  - CTL model-checking in $\text{PTIME}$,
  - polynomial number of requests $\mathcal{M}, s \models \langle A \rangle \Phi$,
  - $\text{PTIME}^{\text{NP}} = \Delta^p_2$. 

$\text{PTIME}^{\text{NP}}$ denotes the class of decision problems that can be solved by a polynomial-time Turing machine with access to an NP oracle.
More details to compute $\mathcal{M}, s \models \langle A \rangle \Phi$

1. Guess a memoryless and uniform strategy $F$. (NP step)

   $$F : s \in S \mapsto f \in D_A(s)$$

   For all $s \sim_A s'$, we have $F(s) = F(s')$.

2. Shrink $\mathcal{M}$ accordingly using $F$, leading to $\mathcal{M}'$: (PTIME step)
   Remove the transitions $s \xrightarrow{g} s'$ with $F(s) \not\subseteq g$.

3. Assuming for each outermost state formula $\psi$ in $\Phi$, $[\psi]^\mathcal{M}$ already computed, $\mathcal{M}'' \overset{\text{def}}{=} \mathcal{M}[p_{\psi}^{\text{new}} \leftarrow [\psi]^\mathcal{M}]$. (PTIME step)

4. All computations from $s$ in $\mathcal{M}''$ should satisfy $\Phi[p_{\psi}^{\text{new}}/\psi]$. I.e. $\mathcal{M}''$, $s \models A \Phi[p_{\psi}^{\text{new}}/\psi]$ in CTL. (PTIME step)
Other topics related to ATL

- More results about model-checking and satisfiability problems for ATL-like logics between ATL and ATL*.
- Logical formalisms with time, knowledge and games.
- More strategy logics (coalition logics, adding ressources, etc.)
Other logical aspects of AI

- In this course DLs, ATL-like logics, SAT, SMT (Part II), ...
- Nonmonotonic reasoning.
- Public announcement logics.
- and much more ...