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

Stéphane Demri  demri@lsv.fr

January 06th, 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

- **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** $L : S \rightarrow \mathcal{P}(\text{PROP})$. 

**Diagram:**

- States: $s_1, s_2, s_3, s_4$. 
- Transitions: 
  - $(b, a) \rightarrow s_2 \rightarrow (a, a), (a, b) \rightarrow s_3 \rightarrow (c, c)$ 
  - $(a, b), (a, a) \rightarrow s_3 \rightarrow (b, b)$ 
  - $(b, a) \rightarrow s_1 \rightarrow (b, b)$ 
  - $p \rightarrow s_2 \rightarrow (a, a), (a, b) \rightarrow s_3 \rightarrow (c, c)$ 
  - $q \rightarrow s_3 \rightarrow (c, c)$ 

**Parameters:** 
- $\text{Agt} = \{1, 2\}$ 
- $S = \{s_1, s_2, s_3, s_4\}$ 
- $\text{Act} = \{a, b, c\}$
Basic concepts: strategies

- $\mathfrak{f}: A \rightarrow \text{Act}$: joint action by $A \subseteq \text{Agt}$ in $s$. $\mathfrak{f}$ 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:

$$\text{out}(s, \mathfrak{f}) \overset{\text{def}}{=} \{ s' \in S \mid \exists g \in D_{\text{Agt}}(s) \text{ s.t. } \mathfrak{f} \subseteq g \text{ and } s' = \delta(s, g) \}$$

- Strategy $F_A$ for $A$: map from the set of histories to the set of joint actions by $A$ such that

$$F_A(s_0 \xrightarrow{\mathfrak{f}_0} s_1 \cdots \xrightarrow{\mathfrak{f}_{n-1}} s_n) \in D_A(s_n)$$
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 $L(s_0) \rightarrow L(s_1) \rightarrow L(s_2) \ldots$.

- 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

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

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

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

\[ \mathcal{M}, s \models \langle\langle A\rangle\rangle \varphi_1 U \varphi_2 \overset{\text{def}}{=} \text{there is a strategy } F_A \text{ s.t. for all } \lambda = s_0 \xrightarrow{f_0} s_1 \ldots \in \text{Comp}(s, F_A), \]
\[ \text{there is some } i \text{ s.t. } \mathcal{M}, s_i \models \varphi_2 \text{ and for all } j \in [0, i - 1], \]
\[ \text{we have } \mathcal{M}, s_j \models \varphi_1. \]

\[ \mathcal{M}, s \models \langle\langle A\rangle\rangle G \varphi \overset{\text{def}}{=} \text{there is a strategy } F_A \text{ s.t. for all } \lambda = s_0 \xrightarrow{f_0} s_1 \ldots \in \text{Comp}(s, F_A), \]
\[ \text{for all } i, \text{ we have } \mathcal{M}, s_i \models \varphi. \]
Decision problems

- Model-checking problem for ATL:
  Input: $\phi$ in ATL, a finite CGS $M$ and a state $s$,
  Question: $M, s \models \phi$?

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

- Satisfiability problem for ATL:
  Input: $\phi$ in ATL,
  Question: Is there a CGS $M$ and $s$ in $M$ such that $M, s \models \phi$?

- Satisfiability and validity problems are EXPTIME-complete.

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

- $\text{CGS } \mathcal{M} = (\text{Agt}, S, \text{Act}, \text{act}, \delta, L), A \subseteq \text{Agt}, \text{ and } Z \subseteq S.$

- $\text{pre}(\mathcal{M}, A, Z)$: set of states from which $A$ has a collective move that guarantees that the outcome to be in $Z$.

- Definition of $\text{pre}(\mathcal{M}, A, \cdot)$: $\mathcal{P}(S) \rightarrow \mathcal{P}(S)$

  \[
  \text{pre}(\mathcal{M}, A, Z) \overset{\text{def}}{=} \{ s \in S \mid \text{there is } f \in D_A(s) \text{ such that } \text{out}(s, f) \subseteq Z \}\]

- $[\llbracket A \rrbracket X \varphi]^\mathcal{M} = \text{pre}(\mathcal{M}, A, [\varphi]^\mathcal{M}).$
Knaster-Tarski Theorem: a restricted form

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

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

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

- 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 $\mathcal{G}_{A,\varphi} : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$:

$$\mathcal{G}_{A,\varphi}(Z) \overset{\text{def}}{=} [\varphi]^\mathcal{M} \cap \text{pre}(\mathcal{M}, A, Z).$$

- $\mathcal{J} \langle \langle A \rangle \rangle G_{A,\varphi} \mathcal{K}^\mathcal{M} = \nu Z.(\mathcal{J} \varphi \mathcal{K}^\mathcal{M} \cap \text{pre}(\mathcal{M}, A, Z))$ (greatest fixpoint)

- Given $A \subseteq \text{Agt}$, formulae $\varphi, \psi$, and a CGS $\mathcal{M}$, we define $\mathcal{O}_{A,\varphi,\psi} : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$:

$$\mathcal{O}_{A,\varphi,\psi}(Z) \overset{\text{def}}{=} [\psi]^\mathcal{M} \cup ([\varphi]^\mathcal{M} \cap \text{pre}(\mathcal{M}, A, Z)).$$

- $\mathcal{J} \langle \langle A \rangle \rangle \varphi U \psi \mathcal{K}^\mathcal{M} = \mu Z.(\mathcal{J} \psi \mathcal{K}^\mathcal{M} \cup ([\varphi]^\mathcal{M} \cap \text{pre}(\mathcal{M}, A, Z))).$ (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 (\(\omega\)-sequences):
  \[ \varphi, \psi ::= p \mid \neg \varphi \mid \varphi \land \psi \mid \varphi \lor \psi \mid X \varphi \mid \varphi U \psi \]

- \(\langle A \rangle (G p \lor (\varphi_1 U \varphi_2))\) 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 \text{ is satisfied, on every computation respecting that strategy”}.

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

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

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

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

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

\[ \text{we have } \mathcal{M}, \lambda \models \Phi \]

\[ \mathcal{M}, \lambda \models \varphi \quad \text{iff} \quad \mathcal{M}, \lambda(0) \models \varphi, \text{ for every state formula } \varphi \]

\[ \mathcal{M}, \lambda \models \psi_1 \land \psi_2 \quad \text{iff} \quad \mathcal{M}, \lambda \models \psi_1 \text{ and } \mathcal{M}, \lambda \models \psi_2 \]

\[ \mathcal{M}, \lambda \models X\psi \quad \text{iff} \quad \mathcal{M}, \lambda[1, \infty) \models \psi \]

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

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

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

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

\[ \text{... but } M, s_2 \not\models \langle \langle \text{Agt} \rangle \rangle (Fp \land Fq) \text{ with positional strategies only.} \]
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 \( \text{PTIME} \)-complete. *(forthcoming labelling algorithm)*

- Model-checking problem for ATL is \( 2\text{EXPTIME} \)-complete. (non-positional strategies are required)

- CTL\(^*\) = fragment of ATL\(^*\).
  
  E.g., \( \langle \langle \text{Agt} \rangle \rangle \Phi \approx \text{E} \Phi \), \( \langle \langle \emptyset \rangle \rangle \Phi \approx \text{A} \Phi \), \ldots 

- Model-checking problem for CTL\(^*\) is \( \text{PSPACE} \)-complete.
Model-checking algorithm for ATL in PTIME
Principles of the labelling algorithm

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

- 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 $s \in \llbracket \psi \rrbracket^\mathcal{M}$
1: procedure GMC($\mathcal{M}$, $\varphi$)
2:   case $\varphi$ of
3:     $p$:  return $\{s \in S \mid p \in L(s)\}$  ▷ some atomic proposition
4:     $\neg\psi$: return $S \setminus \text{GMC}(\mathcal{M}, \psi)$
5:     $\psi_1 \lor \psi_2$:  return $\text{GMC}(\mathcal{M}, \psi_1) \cup \text{GMC}(\mathcal{M}, \psi_2)$
6:     $\langle\langle A\rangle\rangle X\psi$: return $\text{pre}(\mathcal{M}, A, \text{GMC}(\mathcal{M}, \psi))$
7:     $\langle\langle A\rangle\rangle G\psi$:  
8:        $X \leftarrow S; Y \leftarrow \text{GMC}(\mathcal{M}, \psi);$  ▷ $X$: previous value; $Y$: next value
9:     while $X \not\subseteq Y$ do  ▷ equivalent to $X \neq Y$ as always $Y \subseteq X$
10:        $X \leftarrow Y; Y \leftarrow \text{pre}(\mathcal{M}, A, X) \cap \text{GMC}(\mathcal{M}, \psi)$
11:     end while; return $Y$
12:     $\langle\langle A\rangle\rangle \psi_1 U\psi_2$:  
13:        $X \leftarrow \emptyset; Y \leftarrow \text{GMC}(\mathcal{M}, \psi_2);$  ▷ $X$: previous value; $Y$: next value
14:     while $Y \not\subseteq X$ do  ▷ equivalent to $X \neq Y$ as always $X \subseteq Y$
15:        $X \leftarrow Y;$
16:        $Y \leftarrow \text{GMC}(\mathcal{M}, \psi_2) \cup (\text{pre}(\mathcal{M}, A, X) \cap \text{GMC}(\mathcal{M}, \psi_1))$
17:     end while; return $Y$
18: end case
19: 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 $\text{GMC}(\mathcal{M}, \psi)$ 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 $\text{GMC}(\mathcal{M}, \psi)$ is performed.

- This technique is standard and herein we use it so that for each subformula $\psi$, $\text{GMC}(\mathcal{M}, \psi)$ is called at most once.
Computing $[[\{2\}] \times q \land [[\{1\}] pUq]^m$

$\begin{align*}
\quad & (b, a) \quad (a, a), (a, b) \\
\uparrow & \quad (b, b) \quad (c, c) \\
\quad & (a, b), (a, a) \quad (b, b) \\
\downarrow & \quad (b, a) \quad \quad (b, b)
\end{align*}$

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

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

- $\text{pre}(M, \{2\}, [[q]^m) = \text{pre}(M, \{2\}, \{s_3\}) = \{s_3, s_4\}$. 
Computing $[[\{2\}]Xq \land [\{1\}]pUq]^m$ (bis)

$\begin{array}{c}
\begin{tikzpicture}[node distance=1.5cm,thick,main node/.style={circle,draw}]
\node[main node] (1) {$s_1$};
\node[main node] (2) [right of=1] {$s_2$};
\node[main node] (3) [right of=2] {$s_3$};
\node[main node] (4) [right of=3] {$s_4$};
\path
(1) edge [loop above] node {$(b, a)$} (1)
(1) edge [above] node {$(a, b), (a, a)$} (2)
(2) edge [above] node {$(a, a), (a, b)$} (1)
(2) edge [above] node {$(b, b)$} (3)
(3) edge [above] node {$(c, c)$} (2)
(3) edge [loop above] node {$(c, c)$} (3)
(3) edge [loop above] node {$(b, a)$} (4)
(4) edge [loop above] node {$(b, a)$} (4)
(4) edge [below] node {$(b, b)$} (2)
(4) edge [above] node {$(c, c)$} (3);
\end{tikzpicture}
\end{array}$

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

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

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

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

$[[\{2\}]Xq \land [\{1\}]pUq]^m = [[\{2\}]Xq]^m \cap [[\{1\}]pUq]^m = \{s_3\}$
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 PTIME-complete.
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

\[ S_0 \xrightarrow{\text{idle, head}} S_1 \xrightarrow{\text{head, idle}} \text{win} \]
\[ S_0 \xrightarrow{\text{idle, tail}} S_2 \xrightarrow{\text{tail, idle}} \text{lose} \]

So \[ \prec \{1\} \prec \text{Fwin} \]
So \[ \not\prec \{1\} \prec \text{Fwin} \] with imperfect information
Concurrent game structures with imperfect information: iCGS

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

- \textbf{Agt}, \textbf{S}, \textbf{Act}, \delta, \textbf{L} as in a CGS.

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

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

\[ A = \bigcap_{a \in \text{Agt}} \sim a. \]

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

\[ \iff \]

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<sub>ir</sub>: ATL with imperfect information and memoryless strategies

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

- $\langle A \rangle G \varphi \Rightarrow \varphi \land \langle A \rangle X \langle A \rangle G \varphi$ is valid in ATL<sub>ir</sub>.

- $\varphi \land \langle A \rangle X \langle A \rangle G \varphi \Rightarrow \langle A \rangle G \varphi$ is not valid in ATL<sub>ir</sub>.

The model-checking problem for ATL<sub>ir</sub> 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)
Complexity of model-checking problems

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

- **R**: memoryful strategies ; **r**: memoryless strategies.

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

- **Imperfect information:**
  - $\text{ATL}_{ir}$: $\Delta_2^p$-complete.
  - $\text{ATL}^*_{ir}$: $\text{PSPACE}$-complete.
  - $\text{ATL}^*_{iR}$, $\text{ATL}_{iR}$: undecidable.
Model-checking: $\Delta^p_2$ upper bound

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

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

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

- The model-checking problem for $\text{ATL}_{ir}$ is in $\Delta^p_2$. 
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 SAT, SMT, DLs, ATL-like logics...
- Nonmonotonic reasoning.
- Public announcement logics.
- and much more...