Logical Aspects of Artificial Intelligence

$\text{ATL}^+ \text{ in } \text{PSPACE} + \text{Adding resources}$

Stéphane Demri

dem⋆i@lmf.cn⋆s.f⋆

December 13th, 2023
Plan of the lecture

- Recapitulation of previous lecture.
- $\text{ATL}^+$ model-checking problem in $\text{PSPACE}$.
- By-product: model-checking for $\text{ATL}$ in $\text{PTIME}$.
- Introduction to concurrent game structures with resources.
- More exercises to prepare the exam (on January 10th).
Recapitulation of the Previous Lecture
Model-checking for ATL in \textsc{PTime}

- Iterative algorithm that consists in labelling each state in $S$ by the set of state formulae it satisfies.

- For each $r \in S$, the algorithm builds a set of state formulae $l(r)$ such that
  - for every subformula $\varphi$, $\varphi \in l(r)$, or $\neg \varphi \in l(r)$, but not both at the same time,
  - for every subformula or its negation $\psi$, $\psi \in l(r)$ iff $\mathcal{M}, r \models \psi$.

- Fixpoint characterisation of the strategy modalities is used to handle strategy formulae in \textsc{PTime}.
Introduction to ATL

▶ Path formulae $\Phi$ and state formulae $\varphi$

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

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

▶ CGS $M$ with two agents in $\{1, 2\}$.

$$M, s_0 \models \langle \{1\} \rangle (Gp_1 \lor Fp_2)$$
Model-checking for $\text{ATL}^+$ is PSPACE-hard

$$\psi = \forall p_1 \exists p_2 \forall p_3 \cdots \forall p_{2n-1} \exists p_{2n} \varphi$$

$\psi$ is QBF satisfiable iff $\mathcal{M}_\psi, s_0 \models \langle 1 \rangle \varphi[p_i \leftarrow F p_i]$
ATL$^+$ Model-Checking in PSPACE
A simplified $\text{ATL}^+$ fragment

For the PSPACE upper bound proof, restriction to

\[ \varphi ::= p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \langle A \rangle \Phi \quad \Phi ::= \neg \Phi \mid \Phi \land \Phi \mid \Phi \lor \Phi \mid G \varphi \mid F \varphi \]

(no $X$, no unrestricted $U$)

PSPACE-hardness proof uses formulae from this fragment.

Since $\neg F \varphi$ logically equivalent to $G \neg \varphi$, we can restrict ourselves to positive Boolean combinations:

\[ \Phi ::= \Phi \mid \Phi \land \Phi \mid \Phi \lor \Phi \mid G \varphi \mid F \varphi \]

Extension with atomic path formulae of the form $\varphi$, $X \varphi$ and $\varphi U \varphi'$ obtained by adapting the forthcoming developments.
Preliminaries about trees

- Tree $t$ is a subset of $\mathbb{N}^*$ such that ($n \in \mathbb{N}^*$ and $i \in \mathbb{N}$)
  - $n \cdot i \in t$ implies $n \in t$,
  - $n \cdot (i + 1) \in t$ implies $n \cdot i \in t$.

- Labelled tree is defined as a map $t : \text{dom}(t) \to \Sigma$ for some alphabet $\Sigma$. 
Given $n \in t$, the label of the branch leading to $n$ is $t^{br}(n) \overset{\text{def}}{=} t(\varepsilon) \cdot t(i_1) \cdot t(i_1 i_2) \cdots t(i_1 \cdots i_k)$ assuming $n = i_1 \cdots i_k$.

$t^{br}(20) = a \cdot c \cdot c$.

An infinite branch in $t$ is an infinite sequence $i_1 i_2 \cdots \in \mathbb{N}^\omega$ in $t$ with label $t(\varepsilon) \cdot t(i_1) \cdot t(i_1 i_2) \cdots$.

Tree $t$ is finite-branching if for all $n \in t$ there is $i \in \mathbb{N}$ such that $n \cdot i \notin t$. 
Strategy tree $t_{\sigma}$

- Strategy $\sigma$ for the coalition $A^*$.
- Set of computations starting from $s^*$ and respecting $\sigma$ can be organised as an infinite tree $t_{\sigma}$.

$\sigma$: agent 1 chooses $b$ on $s_1$
Definition of the strategy tree $t_\sigma$

- $t_\sigma$ smallest labelled tree over the finite alphabet $S$.
  - $\varepsilon \in \text{dom}(t_\sigma)$ and $t_\sigma(\varepsilon) \overset{\text{def}}{=} s_0$ with $s_0 = s^*$.

- $\text{out}(s^*, \sigma(s^*)) = \{r_1, \ldots, r_\alpha\}$ implies $0, \ldots, \alpha - 1 \in \text{dom}(t_\sigma)$. For all $i \in \{0, \ldots, \alpha - 1\}$, $t(i) \overset{\text{def}}{=} r_{i+1}$.
  - $0, \ldots, \alpha - 1$ only children of $\varepsilon$.

- Assume $n \in t_\sigma$ with $n = i_1 \cdots i_k$ and the label of the finite branch leading to $n$ is $s_0 \cdots s_k$.
  - $\text{out}(s_k, \sigma(s_0 \cdots s_k)) = \{r_1, \ldots, r_\alpha\}$ implies $n \cdot 0, \ldots, n \cdot (\alpha - 1) \in t_\sigma$.
  - For all $i \in \{0, \ldots, \alpha - 1\}$, $t_\sigma(n \cdot i) \overset{\text{def}}{=} r_{i+1}$.
  - $n \cdot 0, \ldots, n \cdot (\alpha - 1)$ only children of $n$. 
\( t_\sigma \) is the right structure!

- The structure \( t_\sigma \) is very naturally related to \( \text{comp}(s^*, \sigma) \) by recording only the histories (finite computations) obtained from \( s^* \) following \( \sigma \).

- For every computation \( \lambda \in \text{comp}(s^*, \sigma) \), there is an infinite branch in \( t_\sigma \) whose label is \( \lambda \).

- For every infinite branch \( \mathcal{B} = i_1i_2 \cdots \) of \( t_\sigma \), its label \( t^{br}(\mathcal{B}) = t(\varepsilon) \cdot t(i_1) \cdot t(i_1i_2) \cdots \) is a computation in \( \text{comp}(s^*, \sigma) \).

- The propositions below are equivalent.
  1. For all the computations \( \lambda \in \text{comp}(s^*, \sigma) \), we have \( M^*, \lambda \models \Phi^* \).
  2. For every infinite branch \( \mathcal{B} = i_1i_2 \cdots \) of \( t_\sigma \), we have \( M^*, t^{br}(\mathcal{B}) \models \Phi^* \).
Updating current path formula (objective)

- $\Phi^*$ positive Boolean combination of $F\varphi$ or $G\varphi$ with propositional $\varphi$.

- Suppose $\Phi^* = Gp_1 \lor Fp_2$, and $\lambda = s_0s_1s_2 \cdots$ in

  $\Phi^*$ can be transformed into $\bot \lor Fp_2$ when visiting $s_1$,

  and then into $\bot \lor \top$ when visiting $s_2$.

- As $\bot \lor \top \equiv \top$ for all the states visited from $s_2$, $\Phi^*$ holds true on $\lambda$. 
We write $U(s, \Psi)$ to denote the Boolean combination of atomic path formulae obtained from $\Psi$ according to the following clauses.

1. If $G\phi$ occurs in $\Psi$ and $M, s \not\models \phi$, then replace every occurrence of $G\phi$ in $\Psi$ by $\bot$.

2. If $F\phi$ occurs in $\Psi$ and $M, s \models \phi$, then replace every occurrence of $F\phi$ in $\Psi$ by $\top$.

$M, \lambda \models \Psi$ iff $M, \lambda \models U(\lambda(0), \Psi)$.

$M, \lambda \models \Psi$ iff $M, \lambda_{\geq 1} \models U(\lambda(1), U(\lambda(0), \Psi))$. 
Introducing the set $\text{pf}(M^*, \Phi^*)$ of path formulae

- Given a finite sequence $\lambda \in S^+$, define $U(\lambda, \Phi^*)$ as follows.
  - If $\lambda$ has length 1 (i.e. $\lambda = \lambda(0)$), then
    \[ U(\lambda, \Phi^*) \overset{\text{def}}{=} U(\lambda(0), \Phi^*) \]
  - If $\lambda$ has length $n \geq 2$, then
    \[ U(\lambda, \Phi^*) \overset{\text{def}}{=} U(\lambda(n), U(\lambda[0, n-1], \Phi^*)). \]

$\text{pf}(M^*, \Phi^*) \overset{\text{def}}{=} \{ U(\lambda, \Phi^*) \mid \lambda \in S^+ \}$. 
Example

\[ \lambda = s_0 s_1 s_1 s_2^\omega \text{ and } \Phi^* = Gp_1 \lor Fp_2. \]

- \( U(s_0, \Phi^*) = \Phi^* \).
- \( U(s_0 s_1, \Phi^*) = \perp \lor Fp_2 \).
- \( U(s_0 s_1 s_1, \Phi^*) = \perp \lor Fp_2 \).
- \( U(s_0 s_1 s_1 s_2^i, \Phi^*) = \perp \lor \top \). \quad (i \geq 1)

\[ pf(M^*, \Phi^*) = \{ \Phi^*, \perp \lor Fp_2, \perp \lor \top \} . \]
Properties about $U$

Infinite computation $\lambda$ in $\mathcal{M}^*$.

(correctness) For all $n \in \mathbb{N}$,

$$\mathcal{M}, \lambda \models \Phi^* \text{ iff } \mathcal{M}, \lambda_{\geq n} \models U(\lambda[0, n], \Phi^*).$$

(small size) $\text{card}(\rho f(\mathcal{M}^*, \Phi^*))$ bounded by $3^{\text{size}(\Phi^*)}$.

(stabilisation) There is $n \geq 0$ such that for all $n' \geq n$, we have

$$U(\lambda[0, n], \Phi^*) = U(\lambda[0, n'], \Phi^*).$$
Using $U$ to characterise $\mathcal{M}^*$, $\lambda \models \Phi^*$

- Infinite computation $\lambda$ in $\mathcal{M}^*$.

- Equivalence between

  (I) $\mathcal{M}^*$, $\lambda \models \Phi^*$.

  (II) There is $I \geq 0$ such that

    (a) for all $J \geq I$, we have $U(\lambda[0, J], \Phi^*) = U(\lambda[0, I], \Phi^*)$,
    (b) replacing in $\Phi_I$ all the G-formulae by $\top$ leads to a formula propositionnally equivalent to $\top$.

- $\models_\infty \Phi^*$: replacing in $\Phi^*$ all the G-formulae by $\top$ leads to a formula propositionnally equivalent to $\top$.

  ((b) can be replaced by $\models_\infty \Phi^*$)

- $Fp \lor G\neg p$ logically equivalent to $\top$ but not propositionally equivalent to $\top$. 
Proof (I)

- First suppose that $M^*, \lambda \models \Phi^*$.

- By (stabilisation), there are $\Psi_{\lambda, \Phi^*}$ and $I ∈ \mathbb{N}$ such that for all $J ≥ I$, we have $U(\lambda[0, J], \Phi^*) = \Psi_{\lambda, \Phi^*}$.

- For all $J ≥ I$ and all $G\varphi$ occurring in $\Psi_{\lambda, \Phi^*}$, $\lambda(J) \models \varphi$.

- For all $J ∈ [0, I − 1]$ and all $G\varphi$ occurring in $\Psi_{\lambda, \Phi^*}$, $\lambda(J) \models \varphi$.

- Similarly, for all $J ≥ 0$ and for all $F\varphi$ occurring in $\Psi_{\lambda, \Phi^*}$, $\lambda(J) \models \neg \varphi$. 
Proof (II)

- Consequently, for all $G\varphi$ occurring in $\Psi_{\lambda,\Phi^*}$, we have $\lambda \models G\varphi$ and for all $F\varphi'$ occurring in $\Psi_{\lambda,\Phi^*}$, we have $\lambda \not\models F\varphi'$.

- By (correctness), equivalence between
  - $\lambda \models \Phi^*$,
  - $\lambda_{\geq l} \models \Psi_{\lambda,\Phi^*}$,
  - $\lambda_{\geq l} \models \Psi'_{\lambda,\Phi^*}$, where $\Psi'_{\lambda,\Phi^*}$ is computed from $\Psi_{\lambda,\Phi^*}$ by replacing each $F\varphi'$ by $\bot$ and each $G\varphi$ by $\top$.

- Since $\Psi_{\lambda,\Phi^*}$ is a positive Boolean formula, this means that replacing in $\Psi_{\lambda,\Phi^*}$ all the $G$-formulae by $\top$ leads to a formula propositionally equivalent to $\top$.

- The proof for the other direction easier.
Enriched tree $t_\sigma$

Idea: to add the element of $pf(M^*, \Phi^*)$ leading to the final state

$\sigma$: agent 1 chooses $b$ on $s_1$
Enriching the tree $t_\sigma$: formal definition

$t_\sigma$ smallest labelled tree defined over the alphabet $S \times pf(M^*, \Phi^*)$.

- $\varepsilon \in \text{dom}(t_\sigma)$ and $t_\sigma(\varepsilon) \overset{\text{def}}{=} (s_0, \Phi_0)$ with $s_0 = s^*$ and $\Phi_0 = U(s^*, \Phi^*)$.

- Assuming that $\text{out}(s^*, \sigma(s^*)) = \{r_1, \ldots, r_\alpha\}$ for some $\alpha \geq 1$, we have $0, \ldots, \alpha - 1 \in \text{dom}(t_\sigma)$ and for all $i \in \{0, \ldots, \alpha - 1\}$, $t_\sigma(i) \overset{\text{def}}{=} (r_{i+1}, U(r_{i+1}, U(s^*, \Phi^*)))$.

- Assume that $n \in \text{dom}(t_\sigma)$ with $n = i_1 \cdots i_k$ for some $k \geq 1$, the label of the finite branch leading to $n$ is $s_0 \cdots s_k$ and $t_\sigma(n) = (s_k, \Phi_k)$.

  If $\text{out}(s_k, \sigma(s_0 \cdots s_k)) = \{r_1, \ldots, r_\alpha\}$, then $n \cdot 0, \ldots, n \cdot (\alpha - 1) \in \text{dom}(t_\sigma)$ and for all $i \in \{0, \ldots, \alpha - 1\}$, $t_\sigma(n \cdot i) \overset{\text{def}}{=} (r_{i+1}, U(r_{i+1}, \Phi_k))$. 
Basic properties

- For every infinite branch $\mathcal{B}$ of $t_\sigma$ with label $(s_0, \Phi_0) \cdot (s_1, \Phi_1) \cdot (s_2, \Phi_2) \cdot \cdots$,
  
  - $\text{card} \{\Phi_0, \Phi_1, \Phi_2, \ldots\} \leq \text{size}(\Phi^*)$,
  
  - for all $i \geq 0$, we have $\Phi_i = U(s_0 \cdot \ldots \cdot s_i, \Phi^*)$. \hspace{1cm} (by construction)
  
  - there is $l \geq 0$ such that $\Phi_l = \Phi_{l+1} = \Phi_{l+2} = \cdots$.

- $t_\sigma$ is finite-branching (since $\text{card}(S)$ is finite).
Almost final characterisation lemma

(I) For all $\lambda \in \text{comp}(s^*, \sigma)$, there is an infinite branch in $t_\sigma$ with label $(s_0, \Phi_0) \cdot (s_1, \Phi_1) \cdots$ such that $\lambda = s_0 s_1 \cdots$.

(II) For every infinite branch $B = i_1 i_2 \cdots$ of $t_\sigma$ with label $(s_0, \Phi_0) \cdot (s_1, \Phi_1) \cdots$, we have $s_0 s_1 \cdots \in \text{comp}(s^*, \sigma)$.

(III) The propositions below are equivalent.

(a) For all $\lambda \in \text{comp}(s^*, \sigma)$, we have $\mathcal{M}^*, \lambda \models \Phi^*$.
(b) For every infinite branch $B = i_1 i_2 \cdots$ of $t_\sigma$, with label $(s_0, \Phi_0) \cdot (s_1, \Phi_1) \cdots$, there is $I \geq 0$ such that
   - for all $J \geq l$, we have $\Phi_J = \Phi_l$,
   - $\models_{\infty} \Phi_l$. 


Definition of $t^*_\sigma$: towards finite characterisation

$\triangleright$ $t^*_\sigma$: subtree of $t_\sigma$ such that

$$\text{dom}(t^*_\sigma) = \{\varepsilon\} \cup \{n \cdot i \in \text{dom}(t_\sigma) \mid \text{no letter in } t^{br}_\sigma(n) \text{ occurs at two distinct positions}\}.$$
Properties of $t^*_\sigma$

- For every $n \in \text{dom}(t^*_\sigma)$, the length of $n$ is at most $\text{card}(S \times \text{pf}(\mathcal{M}^*, \Phi^*))$, i.e. it is bounded by $\text{card}(S) \times 3^{\text{size}(\Phi^*)}$.

- For every infinite $\mathcal{B}$ with label $(s_0, \Phi_0) \cdot (s_1, \Phi_1) \cdot (s_2, \Phi_2) \cdot \ldots$, we have $\text{card}([\{\Phi_0, \Phi_1, \Phi_2, \ldots\}]) \leq \text{size}(\Phi^*)$.
  The length of $n$ is actually bounded by $\text{card}(S) \times \text{size}(\Phi^*)$.

- $t^*_\sigma$ is a finite tree.
Main characterisation lemma

The propositions below are equivalent.

(I) $\mathcal{M}^*, s^* \models \langle \mathcal{A}^* \rangle \Phi^*$.

(II) There is a strategy $\sigma$ for $\mathcal{A}^*$ such that

- the depth of $t^*_\sigma$ is at most $\text{card}(S) \times \text{size}(\Phi^*)$,

- for every maximal branch $i_1 \cdots i_K$ of $t^*_\sigma$ with label $(s_0, \Phi_0) \cdots (s_K, \Phi_K)$ there is a unique $l < K$ such that
  
  (a) $\Phi_l = \cdots = \Phi_K$,

  (b) $(s_l, \Phi_l) = (s_K, \Phi_K)$,

  (c) $\models_\infty \Phi_l$. 

Example

\[ (a, a) \xrightarrow{(a, b)} s_1 \xrightarrow{(b, a)} s_2 \]

\[ M^*, s_1 \models \langle \{1\} \rangle (Gp_1 \lor Fp_2). \]

\[ \sigma: \text{unique positional strategy for the agent 1 such that the agent 1 chooses the action } b \text{ on } s_1. \]
Proof idea

- Assume $\mathcal{M}^*, s^* \models \langle A^* \rangle \Phi^*$ with strategy $\sigma$.
  - $t^*_\sigma$ does not necessarily satisfy (II).
  - Strategy $\sigma$ is modified to avoid redundant parts.
  - If on a branch of $t_\sigma$, $(s, \Phi)$ is seen twice without $\Phi$ being the final one, the part between the two occurrences of $(s, \Phi)$ can be short-circuited.

- Assume $t^*_\sigma$ satisfies (II).
  - Strategy $\sigma'$ witnessing $\mathcal{M}^*, s^* \models \langle A^* \rangle \Phi^*$ defined by $t^*_\sigma$ by “replicating subtrees.”
\( \mathcal{M}^*, s^* \models \langle A^* \rangle \Phi^* \) in \( \text{PSPACE} \)

- \( \mathcal{M}^*, s^* \models \langle A^* \rangle \Phi^* \) equivalent to existence of a tree of branching factor at most \( \text{card}(S) \) and of depth at most \( \text{card}(S) \times \text{size}(\Phi^*) \).

- Termination argument requires to keep in memory a branch of polynomial length, which can be computed with an alternating Turing machine running in polynomial-time.

- \( \text{APTIME} = \text{PSPACE} \).

- Guessing the children of a node (there are at most \( \text{card}(S) \) nodes) may require to visit almost all the CGS \( \mathcal{M}^* \), which is polynomial in \( \text{size}(\mathcal{M}^*) \).
Model-checking problem for $\mathit{ATL}^+$ in $\mathit{PSPACE}$

- Finite $\mathcal{M} = (\mathit{Agt}, S, \mathit{Act}, \mathit{act}, \delta, L)$, $s \in S$ and $\mathit{ATL}^+$ formula $\varphi$ as before.

- Iterative algorithm that consists in labelling each state in $S$ by the set of state formulae it satisfies.

- $\varphi_1, \ldots, \varphi_N$ state formulae occurring in $\varphi$ ordered by increasing size. Clearly, $N \leq \text{size}(\varphi)$.

- $\mathcal{M}$ interprets only the propositional variables occurring in $\varphi$ and $\text{size}(\mathcal{M})$ can be defined from a reasonably succinct encoding.
Labelling algorithm

- For each $r \in S$, the algorithm builds a set of state formulae $l(r)$ such that
  - for every $i \in [1, N]$, either $\varphi_i \in l(r)$, or $\neg \varphi_i \in l(r)$, but not both at the same time,
  - for every $\psi \in \{\varphi_1, \ldots, \varphi_N, \neg \varphi_1, \ldots, \neg \varphi_N\}$, $\psi \in l(r)$ iff $\mathcal{M}, r \models \psi$.

- Determining whether $\mathcal{M}, s \models \varphi$ consists then in checking whether $\varphi \in l(s)$ once the labelling algorithm terminates.
Labelling algorithm (II)

- For each $i \in [1, N]$ and each $r \in S$, we insert either $\varphi_i$ in $l(r)$ or $\neg \varphi_i$ dans $l(r)$.

- Total number of insertions in $O(N \times \text{card}(S))$, which is polynomial in $\text{size}(\mathcal{M}) + \text{size}(\varphi)$.

- Each single insertion can be done in polynomial space too.

- For all $r \in S$, $l(r)$ is initialized to the empty set.
Labelling algorithm (III)

Case 1: \( \varphi_i \) is a propositional variable.
If \( \varphi_i \in L(r) \) by definition of \( \mathcal{M} \), then insert \( \varphi_i \) in \( l(r) \) otherwise insert \( \neg \varphi_i \) in \( l(r) \).

Case 2: \( \varphi_i = \varphi_{i_1} \land \varphi_{i_2} \) for some \( i_1, i_2 < i \).
Insert \( \varphi_i \) in \( l(r) \) if \( \{ \varphi_{i_1}, \varphi_{i_2} \} \subseteq l(r) \) otherwise insert \( \neg \varphi_i \) in \( l(r) \).

Case 3: \( \varphi_i = \langle \langle A \rangle \rangle \Phi \) where the maximal state formulae in \( \Phi \) are \( \varphi_{i_1}, \ldots, \varphi_{i_\alpha} \) with \( i_1, \ldots, i_\alpha < i \).
Let \( p_1, \ldots, p_\alpha \) be fresh variables and \( \mathcal{M}' \) variant of \( \mathcal{M} \) such that \( p_n \in L'(r) \) iff \( \varphi_{i_n} \in l(r) \).

\[ \mathcal{M}, r \models \langle \langle A \rangle \rangle \Phi \text{ iff } \mathcal{M}', r \models \langle \langle A \rangle \rangle \Phi[\psi_{i_n} \leftarrow p_n], \]

For each state \( r \in S \), \( \varphi_i \) is inserted to \( l(r) \) whenever the above test (adapted for each state \( r \)) is positive, otherwise \( \neg \varphi_i \) is inserted to \( l(r) \).
Büchi Games and $\text{ATL}^+$ Fragments
Büchi games

Büchi game \((V, V_1, V_2, E, F)\):

\[ V = V_1 \cup V_2 \quad E \subseteq V \times V \quad F \subseteq V \]

\[ V_1 = \{v_2, v_3\} \quad V_2 = \{v_1\} \quad F = \{v_1, v_3\} \]

A play \(\Pi\) is a sequence \(v_0 v_1 \cdots\) such that for all \(j \geq 1\), \((v_{j-1}, v_j) \in E\).

Strategy for player \(i\) \((i = 1, 2)\) is a map \(\sigma : (V^* \cdot V_i) \to V\) such that \((v, \sigma(\Pi \cdot v)) \in E\) for all plays \(\Pi \cdot v\).

A play \(v_0 v_1 \cdots\) is consistent with \(\sigma\) iff for all \(j\) such that \(v_j \in V_i\), we have \(\sigma(v_0 v_1 \cdots v_j) = v_{j+1}\).
Büchi games (II)

- Player 1 has a winning strategy \( \Leftrightarrow \) there is a strategy \( \sigma \) for 1 such that for every infinite play \( \Pi \) consistent with \( \sigma \), one element in \( F \) occurs infinitely often in \( \Pi \).

- Player 1 has a winning strategy in

  ![Diagram](image)

- Checking whether a Player 1 has a winning strategy in a Büchi game starting from a node \( v_0 \) is \( \text{PTIME-complete} \).

- Checking whether \( \mathcal{M}^*, s^* \models \langle A^* \rangle \Phi^* \) holds is equivalent to the existence of a winning strategy in a Büchi game.
Remarks about complexity

- In the worst-case, the size of the Büchi game is exponential in the size of $\mathcal{M}^*$ and $\Phi^*$.

- When the Büchi game can be solved in PTIME for a class of instances $\mathcal{M}^*$, $s^* \models \langle A^* \rangle \Phi^*$, the model-checking for the fragment can be solved in PTIME.
Büchi game for $\mathcal{M}^\ast$, $s^\ast \models \langle A^\ast \rangle \Phi^\ast$

- $V_1 \overset{\text{def}}{=} S \times pf(\mathcal{M}^\ast, \Phi^\ast)$.
  
  (Player 1 represents the coalition $A^\ast$)

- $V_2$ is the subset of $S \times (\bigcup_{s \in S} D_A(s)) \times pf(\mathcal{M}^\ast, \Phi^\ast)$ such that $(s, g, \Phi) \in V_2$ iff $g \in D_A(s)$.

- For all $(s, \Phi) \in V_1$ and $g \in D_A(s)$,

  $$((s, \Phi), (s, g, \Phi)) \in E.$$  

- For all $(s, g, \Phi) \in V_2$ with $\text{out}(s, g) = \{r_1, \ldots, r_\alpha\}$, for all $i \in [1, \alpha]$,

  $$((s, g, \Phi), (r_i, U(r_i, \Phi))) \in E.$$  

- $F$ is the set of nodes $(s, \Phi)$ such that $\models_\infty \Phi$.  

Correctness

- $M^*, s^* \models \langle A^* \rangle \Phi^*$ iff Player 1 has a winning strategy for the Büchi game $(V, V_1, V_2, E, F)$ starting from $(s^*, U(s^*, \Phi^*))$.
  
  (also true for the full language)

- For the fragment ATL of $\text{ATL}^+$, $(V, V_1, V_2, E, F)$ is of polynomial size in the size of $M^*, \Phi^*$.

  ($\text{pf}(M^*, \Phi^*)$ contains two elements)

- Consequently, the model-checking problem for $\text{ATL}$ is $\text{PTIME}$.

- $\text{PTIME}$ also holds for any fragment of $\text{ATL}^+$ in which $\text{card}(\text{pf}(M^*, \Phi^*))$ is bounded.
Concurrent Game Structures with Resources
Motivations

- In ATL-like logics, actions are normally modelled as *abstract objects* that bear no computational cost.

- It is natural to reason about *resources produced or consumed by actions*.

- The resources can be of any nature (money, energy, etc.).

- Resource-aware logics:
  - actions have costs/weights,
  - formulae may specify constraints about such (cumulative) costs/weights.

- Dealing with resources can lead to high complexity, even undecidable model-checking problems.
Toy example

- A rover is exploring an unknown area.
- At any time it can move around or recharge its battery, but not at the same time.

Moving around consumes one energy unit at every time step, whereas the rover can recharge of one energy unit at a time.

Switching between modes also requires one energy unit.

**Specification**: Is it always the case that, given an energy budget of \( b \) units, the rover will be able to move?
Concurrent game structures with resources

Concurrent game structures + resources (counters)

- Number $r$ of **resource types** (a.k.a. resources/counters).

- Partial "weight" function $wf : S \times Agt \times Act \rightarrow \mathbb{Z}^r$.

- Given a joint action $f : Agt \rightarrow Act$,

\[
wf(s, f) \overset{\text{def}}{=} \sum_{a \in Agt} wf(s, a, f(a))
\]

- $wf(s, f)$: weight/cost to trigger a transition from $s$ with the (total) joint action $f$. 
Concurrent Game Structures with One Resource

- Partial “weight” function $wf : S \times Agt \times Act \rightarrow \mathbb{Z}$.

$\begin{array}{c}
(a/1, a/2) \xrightarrow{p_1} (a/1, b/ - 3) \xrightarrow{(a/5, a/0)} (b/ - 8, a/0) \xrightarrow{(a/0, a/0)} (a/0, a/0) \\
\end{array}$

(two agents, $wf(s_1, 2, b) = -3$)

- $wf(s, f)$: weight/cost to trigger a transition from $s$ with the (total) joint action $f : Agt \rightarrow Act$:

$$wf(s, f) \overset{\text{def}}{=} \sum_{a \in Agt} wf(s, a, f(a))$$

\[(wf(s_0, [1 \mapsto a, 2 \mapsto b]) = -2)\]
Resource availability vectors

- **Strategy** $\sigma : (s_0 \xrightarrow{f_0} s_1 \xrightarrow{f_1} s_2 \cdots \xrightarrow{f_{n-1}} s_n) \mapsto (g : A \rightarrow \text{Act})$.

- **Initial budget** $\vec{b} \in \mathbb{N}^r$.

- $\lambda = s_0 \xrightarrow{f_0} s_1 \xrightarrow{f_1} s_2 \ldots$ in $\text{comp}(s, \sigma)$.

- **Resource availability vectors** $(\vec{v}_i)_{i \in \mathbb{N}}$
  - $\vec{v}_0 \overset{\text{def}}{=} \vec{b}$,
  - $\vec{v}_{i+1} \overset{\text{def}}{=} \vec{v}_i + \text{wf}(s_i, f_i)$. \hspace{1cm} $(\sigma(s_0 \xrightarrow{f_0} s_1 \cdots \xrightarrow{f_{i-1}} s_i) \subseteq f_i)$

- **$RAV(\vec{b}, \lambda) \overset{\text{def}}{=} (s_0, \vec{v}_0) \rightarrow (s_1, \vec{v}_1) \rightarrow (s_2, \vec{v}_2) \rightarrow \cdots (s_i, \vec{v}_i) \cdots$**
\( \vec{b} \)-consistent strategies

\[ \lambda \in \text{comp}(s, \sigma) \ \overset{\text{def}}{\implies} \ \vec{b} \text{-consistent} \ \iff \ \text{for all } i, \vec{0} \preceq \vec{v}_i. \]

(in \( \text{RAV}(\vec{b}, \lambda) \))

(we stay within the budget)

\[ \sigma \text{ is a } \vec{b} \text{-consistent w.r.t. } s \ \overset{\text{def}}{\iff} \ \text{all the computations from } s \ \text{respecting } \sigma \ \text{are } \vec{b} \text{-consistent.} \]

\[ \text{In ATL with } r \text{ resources (written } \text{ATL}(\text{Agt}, r)) \text{, we restrict the strategies to } \vec{b} \text{-consistent ones.} \]
ATL with resources: ATL\((Agt, r)\)

- **ATL\((Agt, r)\)**: extension of ATL to reason about resources.

- **Formulae**
  \((p \in PROP, A \subseteq Agt, \vec{b} \in \mathbb{N}^r)\)
  \[
  \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle \langle \vec{A}^\vec{b} \rangle \rangle \ X \varphi \mid \langle \langle \vec{A}^\vec{b} \rangle \rangle \ G \varphi \mid \langle \langle \vec{A}^\vec{b} \rangle \rangle \ U \varphi
  \]

- **\(\langle \langle \vec{A}^\vec{b} \rangle \rangle \varphi\)**: existential strategy quantifier with initial budget \(\vec{b}\).

- **Satisfaction relation:**
  \[
  \mathcal{M}, s \models p \iff s \in L(p)
  \]
  \[
  \mathcal{M}, s \models \langle \langle \vec{A}^\vec{b} \rangle \rangle \Phi \iff \text{there is a } \vec{b}\text{-consistent } \sigma \text{ w.r.t. } s \text{ s.t. } \forall \lambda = s_0 \xrightarrow{f_0} s_1 \ldots \in \text{comp}(s, \sigma), \mathcal{M}, \lambda \models \Phi
  \]
Example

\[
\begin{align*}
\langle a/1, a/2 \rangle & \xrightarrow{p_1} \langle a/1, b/ -3 \rangle & \langle a/5, a/0 \rangle & \xrightarrow{p_2} \langle b/ -8, a/0 \rangle & \langle a/0, a/0 \rangle
\end{align*}
\]

\[\text{\textbullet} \quad M, s_1 \models \neg \langle \{1\}^7 \rangle X p_2 \land \langle \{1\}^7 \rangle F p_2.\]

\[\text{\textbullet} \quad M, s_0 \models \langle \{2\}^1 \rangle G p_1 \land \neg \langle \{2\}^5 \rangle F p_2.\]
Upward-closed sets

- \( \vec{b} \preceq \vec{b}' \iff \text{for all } i \in [1, r], \vec{b}[i] \leq \vec{b}'[i]. \)

- A set \( X \subseteq \mathbb{N}^r \) is **upward-closed** \( \iff \) for all \( \vec{b}, \vec{b}' \in \mathbb{N}^r \), \( \vec{b} \in X \) and \( \vec{b} \preceq \vec{b}' \) imply \( \vec{b}' \in X. \)

- The set \( \{ \vec{b} \in \mathbb{N}^r \mid \mathcal{M}, s \models \langle A\vec{b} \rangle \Phi \} \) is upward-closed.

- (see today’s Exercise 1)

- By Dickson’s Lemma (\((\mathbb{N}^r, \preceq) \) is a wqo), there is a **finite** set \( X \subseteq \mathbb{N}^r \) such that for all \( \vec{b} \in \mathbb{N}^r \), we have
  \[ \mathcal{M}, s \models \langle A\vec{b} \rangle \Phi \iff \text{there is } \vec{b}_{min} \in X \text{ such that } \vec{b}_{min} \preceq \vec{b}. \]
Variant semantics

- In the semantics of $\text{ATL}(\text{Agt}, r)$, in the clause for $\mathcal{M}, s \models \langle A \vec{b} \rangle \Phi$, one aims the existence of a strategy $\sigma$ such that all the computations in $\text{comp}(s, \sigma)$ are $\vec{b}$-consistent.

- This could be viewed as too strong.

- $\text{ATL}'(\text{Agt}, r)$: variant of $\text{ATL}(\text{Agt}, r)$ such that $\mathcal{M}, s \models \langle A \vec{b} \rangle \Phi$ holds iff for some strategy $\sigma$, for all $\vec{b}$-consistent computations $\lambda \in \text{comp}(s, \sigma)$, we have $\mathcal{M}, \lambda \models \Phi$.

- In short, the objective $\Phi$ is satisfied only for the infinite computations that are $\vec{b}$-consistent.
Reduction from the halting problem

- The model-checking problem for ATL′(2) (2 resources) is undecidable.

- Reduction from the halting problem for Minsky machines.

- Minsky machine: set of $n$ instructions on two counters $c_1$ and $c_2$.

- The $l$th instruction has one of the form below ($i \in \{1, 2\}$, $l' \in [1, n]$):
  - $l$: $c_i := c_i + 1; \text{goto } l'$
  - $l$: if $c_i = 0$ then goto $l'$ else $c_i := c_i - 1; \text{goto } l''$.

- Configurations are elements of $[1, n] \times \mathbb{N} \times \mathbb{N}$ and the initial configuration is $(1, 0, 0)$. 
Halting problem

Computation is a sequence of configurations starting from the initial configuration and such that two successive configurations respect the instructions.

Consider the Minsky machine described by the two instructions below:

1: \( c_1 := c_1 + 1; \) goto 2
2: \( c_2 := c_2 + 1; \) goto 1

Here is the unique computation:

\[(1, 0, 0) \rightarrow (2, 1, 0) \rightarrow (1, 1, 1) \rightarrow (2, 2, 1) \rightarrow (1, 2, 2) \rightarrow (2, 3, 2) \ldots\]

Halting problem

Input: a Minsky machine \( M; \)
Question: is there a finite computation that reaches the instruction \( n?\)

The halting problem for Minsky machines is undecidable.
Undecidability proof (I)

Given a Minsky machine $M$, we build
\[ M = \langle \{1, 2\}, 2, S, Act, act, wf, \delta, L \rangle \]
such that
\[ M \text{ reaches the instruction } n \text{ iff } M, 1 \models \langle \{1\} \vec{0} \rangle F p. \]

Without any loss of generality, we assume that the instruction $n$ is equal to “$n$: $c_1 := c_1 + 1 ; \text{goto } n$”.

The set of states $S$ is equal to $[1, 2n] \cup \{\bot\}$ where $\bot$ is a sink state.

Propositional variable $p$ holds only on the state $n$.

We write $e_i$ ($i = 1, 2$) to denote the unit vector in $\mathbb{N}^2$ with the $i$th entry equal to 1.
Undecidability proof (II)

- For each instruction “$l$: $c_i := c_i + 1$; goto $l’$”, we have the transition
  \[ I \xrightarrow{(a/ + e_i, b/\vec{0})} I’ \]

- For each instruction “$l$: if $c_i = 0$ then goto $l_1$ else $c_i := c_i - 1$; goto $l_2$”, we have the transitions:
  \[ (a/0, z_i/-e_i)/ \]
  \[ (a/0, z_i/0) \]
  \[ (a/0, z_i/\vec{0}) \]
  \[ (a/\vec{0}, b/\vec{0}) \]
  \[ (a/\vec{0}, b/\vec{0}) \]
  \[ (a/\vec{0}, b/\vec{0}) \]
  \[ (a/\vec{0}, b/\vec{0}) \]
Undecidability proof (III)

- If $M$ reaches the instruction $n$, then the strategy $\sigma_1$ for the agent 1 consists in following the satisfaction of the zero-tests.

- Action $z_i$ triggered at state $l$ whenever a zero-test is positive.

- In that case, the opponent agent 2 can only choose the action $z_i$ at the state $n + l$.
  (otherwise a negative value is reached and the computation is discarded)

- There is a unique infinite computation in $\text{comp}(1, \sigma_1)$ that is $\vec{0}$-consistent and it corresponds to the computation that allows $M$ to reach the instruction $n$.

- Proof in the other direction similar.
Model-checking problem for $\text{ATL}(\text{Agt}, r)$

- $M, s \models \langle A^\vec{b} \rangle \Phi$ can be solved by a procedure dedicated to the energy parity game problem.

- $\text{MC}(\text{ATL}(\text{Agt}, r))$ is $2\text{EXP\,TIME}$-complete.
  
  $\quad (r \geq 3, \text{card}(\text{Agt}) \geq 2)$

- $2\text{EXP\,TIME}$ upper bound preserved with variants.
  - Extensions $\text{ATL}^+(\text{Agt}, r)$ and $\text{ATL}^*(\text{Agt}, r)$.
  - Idle action with zero weight available from any state for any agent.
  - The costs of the actions taken from the agents of the opponent coalition (if any) are ignored.

$$wf_A(s, f) \overset{\text{def}}{=} \sum_{a \in A} wf(s, a, f(a))$$

- Adding the value $\omega$, representing an infinite supply of the corresponding resources (in initial budgets).
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 etc.)
Other logical aspects of AI

- In this course DLs, ATL-like logics, SAT, SMT (Part I), ...
- Nonmonotonic reasoning.
- Public announcement logics.
- Temporal logics with data values.
- and much more ...