Alternating-time Temporal Logic

Rodica Condurache

Lecture 7
A closed system is a system whose evolution depends only on the internal actions.

It is modeled using Labeled Transition Systems (see Lecture 4)

Linear-time Temporal Logic (LTL) expresses properties on the acceptable executions of the system
- ALL executions have to satisfy the property!

Computation Tree Logic (CTL) can express existential and universal properties on executions
- $EF(q \land \neg p)$: there is a computation that visits a state satisfying $q \land \neg p$
(Recall) Computation Tree Logic (CTL)

- CTL formulas are evaluated on states of transition systems
- Contains universal ($A$) and existential ($E$) operators to reason on the properties of the computation tree
- The computation tree of a labeled transition system is its acyclic unfolding
(Recall) Computation Tree Logic (CTL)

Definition (CTL syntax)

Given a set $AP$ of atomic propositions, a CTL formula is defined by the following syntax:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid EX\varphi \mid EG\varphi \mid E\varphi U \varphi$$

where $p \in AP$.

- $EX\varphi$: there is an execution from the current state s.t. at next state holds $\varphi$
- $EG\varphi$: there is an execution from the current state s.t. always holds $\varphi$
- $E\varphi_1 U \varphi_2$: there is an execution from the current state s.t. holds $\varphi_1$ until $\varphi_2$ holds
Definition

We say that a state $v$ satisfies the CTL formula $\varphi$ in the labeled transition system $T$ ($T, v \models \varphi$) iff

- $T, v \models p$ iff $p \in \tau(v)$, for $p \in AP$
- $T, v \models \neg \varphi$ iff $T, v \not\models \varphi$
- $T, v \models \varphi_1 \lor \varphi_2$ iff $T, v \models \varphi_1$ or $T, v \models \varphi_2$
- $T, v \models \text{EX} \varphi$ iff there exists a path $\rho$ starting from $v$ s.t. it holds that $T, \rho[1] \models \varphi$
- $T, v \models \text{EG} \varphi$ iff there exists a path $\rho$ starting from $v$ s.t. it holds that $\forall n \geq 0$, $T, \rho[n] \models \varphi$
- $T, v \models \text{E} \varphi_1 \text{U} \varphi_2$ iff there exists a path $\rho$ starting from $v$ s.t. it holds that $\exists n \neq 0$ s.t. $T, \rho[n] \models \varphi_2$ and $\forall m < n$, holds $T, \rho[m] \models \varphi_1$
We use the equivalences:

\[ AF \phi \equiv \neg EG \neg \phi \]
\[ AG \phi \equiv \neg EF \neg \phi \]
Compositional modeling and design of reactive systems

⇒ Each component is a system interacting with an environment.

Environment
external choices

System
internal choices

The behavior of each system depends on the internal state as well as the behavior of the environment.
Open Systems

- Compositional modeling and design of reactive systems
  - Each component is a system interacting with an environment

**Environment**

- external choices

**System**

- internal choices

- The behavior of each system depends on the internal state as well as the behavior of the environment.

- Question: Can the system resolve its internal choices so that the satisfaction of a property is guaranteed no matter how the environment resolves the external choices?

  *This can be seen as a winning condition in a two-player game between the system and the environment!*
Compositional architectures can be seen as multiagent systems (or games)

- each component is an agent (player)
- the environment of each component consists of the other components in the architecture

A state in the multiagent system contains the global state of the architecture
e.g., the state of each car (and maybe more)
Example: Robots and Carriage

Two robots push a carriage from opposite sides.

We assume that each robot can either push (action *push*) or refrain from pushing (action *wait*).

both robots use the same force when pushing

the carriage can move clockwise or anticlockwise, or it can remain in the same place - depending on who pushes
In multiagent systems, we may express the fact that some agents (in the set $A$) can cooperate and ensure some property $\varphi$ against any behavior of the other agents

\[ \langle A \rangle \varphi \]

\( \langle A \rangle \varphi \) is a path quantifier which ranges over all computations that the agents in $A$ can force the game into, irrespective of how the other players proceed.

The quantifier $\langle A \rangle$ is a generalization of quantifiers in CTL:
- the existential path quantifier $\exists$ corresponds to $\langle A^g \rangle$
- the universal path quantifier $\forall$ corresponds to $\langle \emptyset \rangle$
Example: Robots and Carriage

We can express properties as:

- Can Robot 1 ensure that the carriage eventually reaches \( pos_1 \)?
  
  \( \langle 1 \rangle F pos_1 \)

- Can Robot 1 ensure that \( pos_1 \) is never reached?
  
  \( \langle 1 \rangle G \neg pos_1 \)
Example: Robots and Carriage

We can express properties as:

- Can Robot 1 ensure that the carriage eventually reaches $pos_1$?  
  \[ \langle 1 \rangle F \ pos_1 \]  
  \[ \text{No!} \]

- Can Robot 1 ensure that $pos_1$ is never reached?  
  \[ \langle 1 \rangle G \neg \ pos_1 \]
We can express properties as:

- Can Robot 1 ensure that the carriage eventually reaches $pos_1$?  
  $\langle 1 \rangle F pos_1$  
  \[ \text{No!} \]

- Can Robot 1 ensure that $pos_1$ is never reached?  
  $\langle 1 \rangle G \neg pos_1$  
  \[ \text{Yes! } \sigma_1(q_0) = \text{wait}, \sigma_1(q_2) = \text{push} \]
Definition (ATL syntax)

Given a set $\mathcal{AP}$ of atomic propositions and a finite set $\text{Ag}$ of agents, an ATL formula is defined by the following syntax:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle A \rangle X \varphi \mid \langle A \rangle G \varphi \mid \langle A \rangle \varphi U \varphi$$

where $p \in \mathcal{AP}$ and $A \subseteq \text{Ag}$.

We write $\langle A \rangle F \varphi$ for $\langle A \rangle \text{true} U \varphi$.

The dual of $\langle A \rangle$ is $[A] \varphi := \neg \langle A \rangle \neg \varphi$

We read:

- $\langle A \rangle \varphi$: Players in $A$ can cooperate to make $\varphi$ true
- $[A] \varphi$: Players in $A$ cannot cooperate to make $\varphi$ false
Semantic of $\mathcal{M}$, $v \models \langle A \rangle \phi$:

- Consider the two-player game over the same state space as $\mathcal{M}$ where:
  - Player A (protagonist) chooses actions for agents in A
  - Player B (antagonist) chooses actions for the other agents
  - Objective of Player A is to satisfy $\phi$

- $s \models \langle A \rangle \phi$ iff there is a winning strategy $\sigma_A$ for Player A in the above game from $v$. 
ATL - Semantics

- ATL formulas are evaluated over states $v$ in a multiagent system $\mathcal{M} = \langle \mathcal{AP}, \mathcal{Ag}, (\text{Act}_i)_{i \in \mathcal{Ag}}, \mathcal{V}, v_0, \tau, E \rangle$

Definition

We say that a state $v$ satisfies the ATL formula $\varphi$ in the multiagent system $\mathcal{M}$ $(\mathcal{M}, v \models \varphi)$ iff

- $\mathcal{M}, v \models p$ iff $p \in \tau(v)$, for $p \in \mathcal{AP}$
- $\mathcal{M}, v \models \neg \varphi$ iff $q \not\models \varphi$
- $\mathcal{M}, v \models \varphi_1 \lor \varphi_2$ iff $v \models \varphi_1$ or $v \models \varphi_2$
- $\mathcal{M}, v \models \langle A \rangle X \varphi$ iff there exists a strategy for each player in $A$, such that for all computations $\rho$ starting from $v$ and following these strategies, it holds that $\rho[1] \models \varphi$
- $\mathcal{M}, v \models \langle A \rangle G \varphi$ iff for $\rho$ as defined above, it holds that $\forall n \geq 0$, $\rho[n] \models \varphi$
- $\mathcal{M}, v \models \langle A \rangle \varphi_1 U \varphi_2$ iff for $\rho$ as defined above, it holds that $\exists n \neq 0$ s.t. $\rho[n] \models \varphi_2$ and $\forall m < n$, holds $\rho[m] \models \varphi_1$
Example: Train-crossing Problem

- One train trying to pass a gate and one controller
- turn-based:
  - the train requests an authorization
  - the controller grants it or refuses the request
  - if the grant is given, the train decides if it accepts it
  - if grant accepted, the train enters in the gate
  - if the train is in_gate, the controller can force it out
Example: Train-crossing Problem

Properties expressible in ATL:

- Whenever the train is outside the gate and has not been granted permission to enter, then the controller can prevent it from entering:
  \[
  \langle \Box \rangle \, G((\text{out}\_\text{of}\_\text{gate} \land \neg \text{grant}) \rightarrow \langle \text{ctr} \rangle \, G \, \text{out}\_\text{of}\_\text{gate})
  \]

- Whenever the train is outside the gate, the controller cannot force it to enter:
  \[
  \langle \Box \rangle \, G(\text{out}\_\text{of}\_\text{gate} \rightarrow [\text{ctr}] \, G \, \text{out}\_\text{of}\_\text{gate})
  \]

- Whenever the train is outside the gate, the train and the controller can cooperate so that the train will enter the gate:
  \[
  \langle \Box \rangle \, G(\text{out}\_\text{of}\_\text{gate} \rightarrow \langle \text{train}, \text{ctr} \rangle \, F \, \text{in}\_\text{gate})
  \]
ATL in different settings

The semantics of ATL can be defined using restrictions on
- observation of agents
  - perfect information (I)
  - imperfect information on states (i)
- quantification on strategies
  - "no recall" (r) or
  - "perfect recall" (R) for agents

<table>
<thead>
<tr>
<th>ATL</th>
<th>$ir$</th>
<th>$iR$</th>
<th>$Ir$</th>
<th>$IR$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>$P^{NP}$</td>
<td>Undecidable</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
</tbody>
</table>

**Table:** Complexity of Model Checking problem for ATL formulas
Model Checking ATL under Perfect Information ($\text{ATL}_{Ir}$ and $\text{ATL}_{IR}$)

Given a multiagent system $\mathcal{M} = \langle \mathcal{AP}, \text{Ag}, (\text{Act}_i)_{i \in \text{Ag}}, V, v_0, \tau, E \rangle$,

- Let $\text{Reg}_\mathcal{M} : \mathcal{AP} \to 2^Q$, where $q \in \text{Reg}(p) \iff p \in \tau(q)$
- Let $\text{Pre}(2^\text{Ag} \times 2^V) \to 2^V$, where $q \in \text{Pre}(A, Y)$ iff $\exists \sigma_A$ s.t. $E(q, \sigma_A) \subseteq Y$

The recursive algorithm is defined as follows:

$\text{MC}(\mathcal{M}, \varphi)$:

- case $\varphi = p$: return $\text{Reg}(\varphi)$;
- case $\varphi = \neg \phi$: return $V \setminus \text{MC}(\mathcal{M}, \phi)$;
- case $\varphi = \phi_1 \lor \phi_2$: return $\text{MC}(\mathcal{M}, \phi_1) \cup \text{MC}(\mathcal{M}, \phi_2)$;
- case $\varphi = \langle A \rangle X \phi$: return $\text{Pre}(A, \text{MC}(\mathcal{M}, \phi))$;
- case $\varphi = \langle A \rangle G \phi$:
  - $Y := V$; $Z := \text{MC}(\mathcal{M}, \phi)$;
  - while $Y \notin Z$ do:
    - $Y = Z$;
    - $Z = \text{Pre}(A, Y) \cap \text{MC}(\mathcal{M}, \phi)$
  - return $Y$
- case $\varphi = \langle A \rangle \phi_1 U \phi_2$:
  - $Y := \emptyset$; $Z := \text{MC}(\mathcal{M}, \phi_2)$;
  - while $Z \notin Y$ do:
    - $Y = Y \cup Z$;
    - $Z = \text{Pre}(A, Y) \cap \text{MC}(\mathcal{M}, \phi_1)$
  - return $Y$

The opposite of an attractor!!!
Stop when cannot remove nodes from $Y$!

A sort of attractor!!!
Stop when cannot add nodes in $Y$!
Example: ATL Model Checking under Perfect Information

\[ \varphi = \langle \emptyset \rangle G (\text{out_of_gate} \rightarrow \lbrack \text{ctr} \rbrack G \text{out_of_gate}) \]

On blackboard!
ATL and Imperfect Information ($\text{ATL}_{ir}$ and $\text{ATL}_{iR}$)

- Each agent $i \in \text{Ag}$ has a set of observations (recall Lecture 6)
- The strategies considered when evaluating $\langle A \rangle$ are based on observations!

Can Robot 1 ensure that $\text{pos}_1$ is never reached?
$\mathcal{M}, q_0 \models \langle 1 \rangle G \neg \text{pos}_1$?

Can both robots cooperate and ensure that $\text{pos}_1$ is never reached?
$\mathcal{M}, q_0 \models \langle 1, 2 \rangle G \neg \text{pos}_1$?

Can both robots cooperate and ensure that $\text{pos}_1$ is eventually reached?
$\mathcal{M}, q_0 \models \langle 1, 2 \rangle F \text{ pos}_1$?
ATL and Imperfect Information ($\text{ATL}_i$ and $\text{ATL}_{iR}$)

- Each agent $i \in Ag$ has a set of observations (recall Lecture 6)
- The strategies considered when evaluating $\langle A \rangle$ are based on observations!

Schematic view

Multiagent system

- Can Robot 1 ensure that $pos_1$ is never reached?
  $\mathcal{M}, q_0 \models \langle 1 \rangle G \neg pos_1$? No!

- Can both robots cooperate and ensure that $pos_1$ is never reached?
  $\mathcal{M}, q_0 \models \langle 1, 2 \rangle G \neg pos_1$? Yes! (agree to play the same action forever)

- Can both robots cooperate and ensure that $pos_1$ is eventually reached?
  $\mathcal{M}, q_0 \models \langle 1, 2 \rangle F pos_1$? Yes! (one pushes all the time, the other waits)
Theorem

The model-checking problem for ATL under imperfect information and no recall (ATL_{ir}) can be solved in \( P^{NP} \) time.

Proof.

Idea: For each subformula of the type \( \langle A \rangle \phi \), with \( \phi \) starting with a temporal operator:

- guess a strategy for each agent in \( A \) (one action for each observation)
- verify in linear time that the resulting labeled transition system satisfies \( \phi \)

Each formula \( \phi \) has only one temporal operator after evaluating its subformulas and replacing with fresh atomic propositions!

Theorem (C. Dima, F.L. Țiplea 2011)

The model-checking problem for ATL under imperfect information and perfect recall (ATL_{iR}) is undecidable.
Relaxing ATL: $ATL^*$

- No restriction on operators: We can write $\varphi = \ll A \gg X X p$

**Definition ($ATL^*$ syntax)**

Given a set $\mathcal{AP}$ of atomic propositions and a set $Ag$ of agents, the formulas in $ATL^*$ are defined by the following syntax:

- **state formula:** $\varphi := p | \neg \varphi | \varphi \lor \varphi | \ll A \gg \psi$
- **path formula:** $\psi := \varphi | \neg \psi | \psi \lor \psi | X \psi | \psi U \psi$

- $ATL$ is a fragment of $ATL^*$

<table>
<thead>
<tr>
<th>$ATL^*$</th>
<th>$ir$</th>
<th>$iR$</th>
<th>$Ir$</th>
<th>$IR$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>PSPACE</td>
<td>Undecidable</td>
<td>PSPACE</td>
<td>2ExpTime</td>
</tr>
</tbody>
</table>

**Table:** Complexity of Model Checking problem for $ATL^*$ formulas
Games and $ATL^*$ formulas

We can use $ATL$ and $ATL^*$ to express properties in concurrent games:

Does Agent $i$ have a strategy to win with the condition:

- **Reach**($R_i$): \( M, v_0 \models \langle i \rangle F (\forall v \in R_i \, v) \)
- **Safe**($S_i$): \( M, v_0 \models \langle i \rangle G (\forall v \in S_i \, v) \)
- **Büchi**($T_i$): \( M, v_0 \models \langle i \rangle GF (\forall v \in T_i \, v) \)
- **coBüchi**($T_i$): \( M, v_0 \models \langle i \rangle FG \neg (\forall v \in T_i \, v) \)
- **LTL**($\varphi_i$): \( M, v_0 \models \langle i \rangle \varphi_i \)

This does not necessarily give better algorithms!!!
Extending ATL with Knowledge operators: ATEL

- Recall: in imperfect information, each agent has a set $O_i$ of observations.
- The Alternating-time Temporal Epistemic Logic (ATEL) extends ATL with the knowledge operators $K_i$.

**Definition (ATEL syntax)**

Given a set $\mathcal{AP}$ of atomic propositions and a finite set $Ag$ of agents, an ATEL formula is defined by the following syntax:

$$\varphi ::= p \mid \lnot \varphi \mid \varphi \lor \varphi \mid \langle A \rangle X \varphi \mid \langle A \rangle G \varphi \mid \langle A \rangle \varphi U \varphi \mid K_i \varphi$$

where $p \in \mathcal{AP}$, $A \subseteq Ag$ and $i \in Ag$.

- $K_i \varphi$ reads as "Agent $i$ knows that the formula $\varphi$ holds".

$$M, v \models K_i \varphi \text{ iff for all } v' \text{ s.t. } obs_i(v) = obs_i(v'), \text{ holds } M, v' \models \varphi$$
Extending ATL with Knowledge operators: ATEL

- Recall: in imperfect information, each agent has a set $O_i$ of observations
- The Alternating-time Temporal Epistemic Logic (ATEL) extends ATL with the knowledge operators $K_i$

**Definition (ATEL syntax)**

Given a set $\mathcal{AP}$ of atomic propositions and a finite set $Ag$ of agents, an ATEL formula is defined by the following syntax:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle A \rangle X \varphi \mid \langle A \rangle G \varphi \mid \langle A \rangle \varphi U \varphi \mid K_i \varphi$$

where $p \in \mathcal{AP}$, $A \subseteq Ag$ and $i \in Ag$.

- $K_i \varphi$ reads as "Agent $i$ knows that the formula $\varphi$ holds"

$$\mathcal{M}, v \models K_i \varphi \iff \text{for all } v' \text{ s.t. } obs_i(v) = obs_i(v'), \text{ holds } \mathcal{M}, v' \models \varphi$$

Not much is known about algorithms and complexities!
Bibliography

- Fabienne Sophie Eigner, "Alternating-Time Temporal Logic" -

- Wojciech Jamroga and Wojciech Penczek, "Specification and Verification of Multi-Agent Systems" - http:
  //krak.ipipan.waw.pl/~wjamroga/papers/verification12esslli-lncs.pdf

- Pierre-Yves Schobbens, "Alternating-time logic with imperfect recall" -