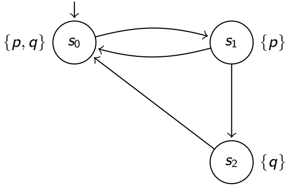
Alternating-time Temporal Logic

Rodica Condurache

Lecture 8

Closed Systems

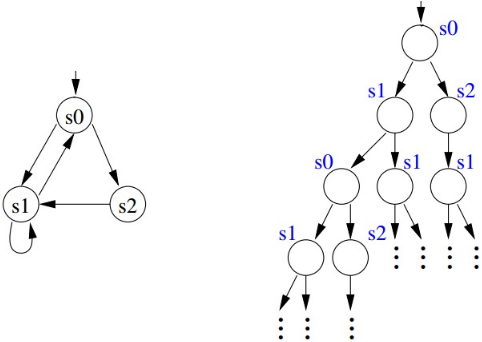
- A closed system is a system whose evolution depends only on the internal actions.
- It is modeled using Labeled Transition Systems (see Lecture 4)



- 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 $\mathcal{AP}$ of atomic propositions, a CTL formula is defined by the following syntax:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid EX\varphi \mid EG\varphi \mid E\varphi\mathcal{U}\varphi$$

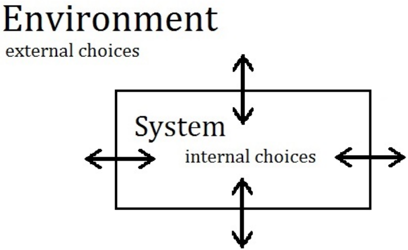
where $p \in \mathcal{AP}$.

- $EX\varphi$: there is an execution from the current state s.t. at next state holds φ
- $EG\varphi$: there is an execution from the current state s.t. always holds φ
- $E\varphi_1\mathcal{U}\varphi_2$: there is an execution from the current state s.t. holds φ_1 until φ_2 holds

Open Systems

- Compositional modeling and design of reactive systems

  ⇒ Each component is a system interacting with an environment

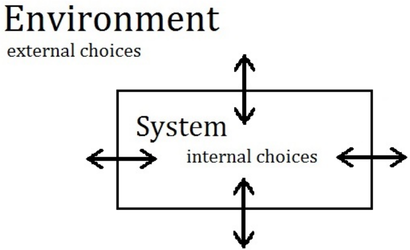


- The behavior of each system depends on the internal state as well as the behavior of the environment.

Open Systems

- Compositional modeling and design of reactive systems

  ⇒ Each component is a system interacting with an environment

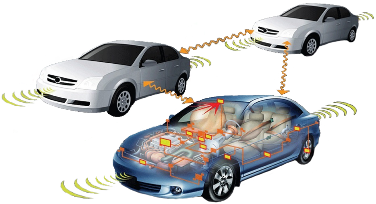


- The behavior of each system depends on the internal state as well as the behavior of the environment.
- Question: Can the system resolve its internal choices so that the satisfaction of a property is guaranteed no matter how the environment resolves the external choices?

  This can be seen as a winning condition in a two-player game between the system and environment!

Compositional architectures as multiagent systems (games)

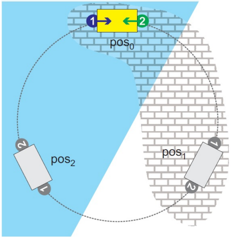
- Compositional architectures can be seen as multiagent systems (or games)
  - each component is an agent (player)
  - the environment of each component consists of the other components in the architecture



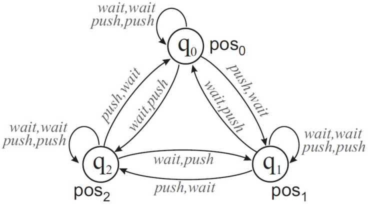
  - A state in the multiagent system contains the global state of the architecture

    e.g., the state of each car (and maybe more)

Example: Robots and Carriage



Schematic view



Multiagent system

- 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

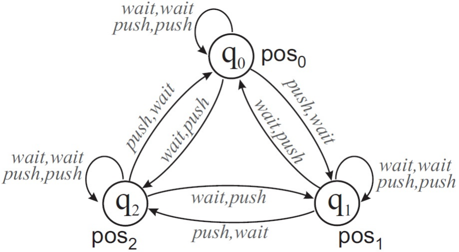
Alternating-time Temporal Logic (ATL)

- In multiagent systems, we may express the fact that some agents (in the set A) can cooperate and ensure some property φ against any behavior of the other agents

$$\langle\langle A \rangle\rangle \varphi$$

- $\langle\langle A \rangle\rangle \varphi$ is a *path quantifier* which ranges over all computations that agents in A can force the game into, **irrespective of how the other players proceed.**
- The quantifier $\langle\langle A \rangle\rangle$ is a generalization of quantifiers in CTL:
  - the existential path quantifier E corresponds to $\langle\langle \mathrm{Ag} \rangle\rangle$
  - the universal path quantifier A corresponds to $\langle\langle \emptyset \rangle\rangle$

Example: Robots and Carriage



We can express properties as:

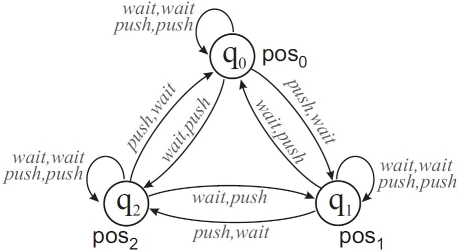
- Can Robot 1 ensure that the carriage eventually reaches pos_1?

$$\langle\langle 1 \rangle\rangle F\ pos_1$$

- Can Robot 1 ensure that pos_1 is never reached?

$$\langle\langle 1 \rangle\rangle G \neg pos_1$$

Example: Robots and Carriage



We can express properties as:

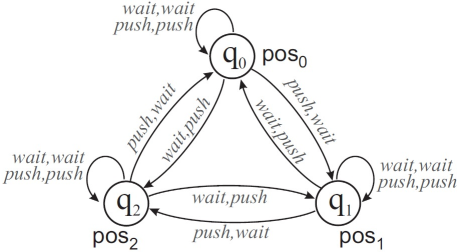
- Can Robot 1 ensure that the carriage eventually reaches pos_1? No!

$$\langle\!\langle 1 \rangle\!\rangle F \ pos_1$$

- Can Robot 1 ensure that pos_1 is never reached?

$$\langle\!\langle 1 \rangle\!\rangle G \neg pos_1$$

Example: Robots and Carriage



We can express properties as:

- Can Robot 1 ensure that the carriage eventually reaches pos_1? No!

$$\langle\langle 1 \rangle\rangle F \; pos_1$$

- Can Robot 1 ensure that pos_1 is never reached? Yes! $\sigma_1(q_0) = wait$, $\sigma_1(q_2) = push$

$$\langle\langle 1 \rangle\rangle G \neg pos_1$$

ATL - Syntax

Definition (ATL syntax)

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

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle A \rangle\!\rangle X\varphi \mid \langle\!\langle A \rangle\!\rangle G\varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathcal{U} \varphi$$

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

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

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

We read:

- $\langle\!\langle A \rangle\!\rangle\varphi$: Players in A can cooperate to make φ true
- $[\![A]\!]\varphi$: Players in A cannot cooperate to make φ false

ATL - Semantics : Intuition

Semantic of $\mathcal{M}, v \models \langle\langle A \rangle\rangle \phi$:

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

ATL - Semantics

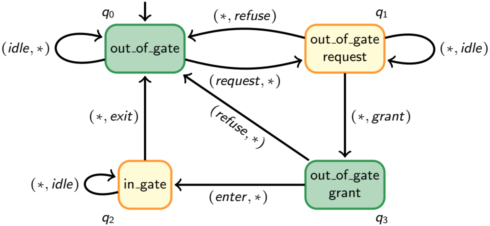
- ATL formulas are evaluated over states v in a multiagent system $\mathcal{M} = \langle \mathcal{AP}, \mathrm{Ag}, (\mathrm{Act}_i)_{i \in \mathrm{Ag}}, V, v_0, \tau, E \rangle$
- Let $\rho = v_0 v_1 v_2 \ldots$ be a path in $\mathcal{M}$. Then, $\rho[i] = v_i$.

Definition

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

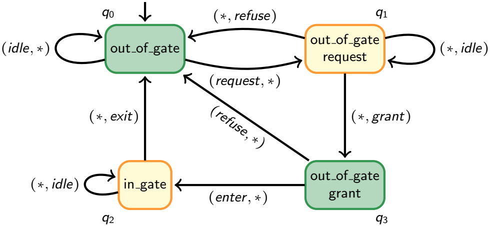
- $\mathcal{M}, v \vDash p$ iff $p \in \tau(v)$, for $p \in \mathcal{AP}$
- $\mathcal{M}, v \vDash \neg\varphi$ iff $\mathcal{M}, q \nvDash \varphi$
- $\mathcal{M}, v \vDash \varphi_1 \vee \varphi_2$ iff $\mathcal{M}, v \vDash \varphi_1$ or $\mathcal{M}, v \vDash \varphi_2$
- $\mathcal{M}, v \vDash \langle\!\langle A \rangle\!\rangle X\varphi$ iff there exists a strategy for each player in A, such that for all computations ρ starting from v and following these strategies, it holds that $\mathcal{M}, \rho[1] \vDash \varphi$
- $\mathcal{M}, v \vDash \langle\!\langle A \rangle\!\rangle G\varphi$ iff for ρ as defined above, it holds that $\forall n \geq 0$, $\mathcal{M}, \rho[n] \vDash \varphi$
- $\mathcal{M}, v \vDash \langle\!\langle A \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$ iff for ρ as defined above, it holds that $\exists n \neq 0$ s.t. $\mathcal{M}, \rho[n] \vDash \varphi_2$ and $\forall m < n$, holds $\mathcal{M}, \rho[m] \vDash \varphi_1$

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:

$$\langle\langle \varnothing \rangle\rangle G((out\_of\_gate \land \neg grant) \rightarrow \langle\langle ctr \rangle\rangle G\ out\_of\_gate)$$

- Whenever the train is outside the gate, the controller cannot force it to enter:

$$\langle\langle \varnothing \rangle\rangle G(out\_of\_gate \rightarrow [\![ctr]\!] G\ out\_of\_gate)$$

- Whenever the train is outside the gate, the train and the controller can cooperate so that the train will enter the gate:

$$\langle\langle \varnothing \rangle\rangle G(out\_of\_gate \rightarrow \langle\langle train, ctr \rangle\rangle F\ in\_gate)$$

ATL in different settings

Definition (Model Checking Problem for ATL)

Given a multiagent system $\mathcal{M} = \langle \mathcal{AP}, \mathrm{Ag}, (\mathrm{Act}_i)_{i \in \mathrm{Ag}}, V, v_0, \tau, E \rangle$, a state $v \in V$ and an ATL formula φ, is it the case that $\mathcal{M}, v \models \varphi$?

The semantics of ATL can be defined using restrictions on

- observation of agents
  - perfect information (I)
  - imperfect information on states (i)
- types of strategies
  - *"no recall"* (r) or
  - *"perfect recall"* (R) for agents

ATL	**ir**	**iR**	**Ir**	**IR**
Complexity	P^{NP}	Undecidable	PTIME	PTIME

Table: Complexity of Model Checking problem for ATL formulas

Model Checking ATL under Perfect Information (ATL_{Ir} and ATL_{IR})

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

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

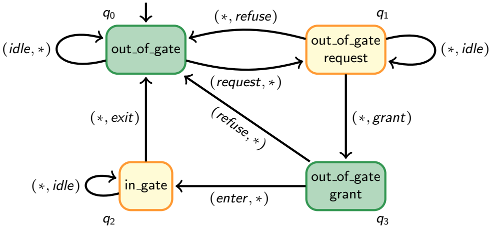
The recursive algorithm is defined as follows:

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

The oposite 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



$$\varphi = \langle\!\langle \emptyset \rangle\!\rangle G\,(out\_of\_gate \rightarrow [\![ctr]\!]\,G\ out\_of\_gate)$$

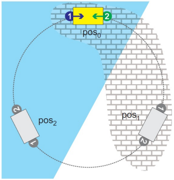
On blackboard!

ATL and Imperfect Information (ATL_{ir} and ATL_{iR})

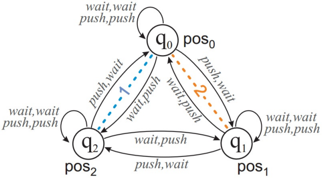
- Each agent $i \in \mathrm{Ag}$ has a set of observations
- The strategies considered when evaluating $\langle\!\langle A \rangle\!\rangle$ are based on observations!

Example

- Robot 1 cannot distinguish between pos_0 and pos_2: $\mathcal{O}_1 = \{\{q_0, q_2\}, \{q_1\}\}$
- Robot 2 cannot distinguish between pos_0 and pos_1: $\mathcal{O}_2 = \{\{q_0, q_1\}, \{q_2\}\}$



Schematic view



Multiagent system

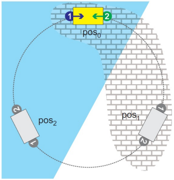
- Can Robot 1 ensure that pos_1 is never reached?
  $\mathcal{M}, q_0 \models \langle\!\langle 1 \rangle\!\rangle G \neg pos_1$?
- Can both robots cooperate and ensure that pos_1 is never reached?
  $\mathcal{M}, q_0 \models \langle\!\langle 1, 2 \rangle\!\rangle G \neg pos_1$?
- Can both robots cooperate and ensure that pos_1 is eventually reached?
  $\mathcal{M}, q_0 \models \langle\!\langle 1, 2 \rangle\!\rangle F\ pos_1$?

ATL and Imperfect Information (ATL_{ir} and ATL_{iR})

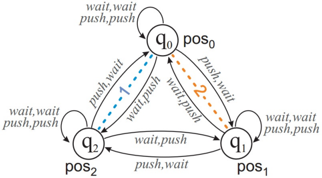
- Each agent $i \in \text{Ag}$ has a set of observations
- The strategies considered when evaluating $\langle\langle A \rangle\rangle$ are based on observations!

Example

- Robot 1 cannot distinguish between pos_0 and pos_2: $\mathcal{O}_1 = \{\{q_0, q_2\}, \{q_1\}\}$
- Robot 2 cannot distinguish between pos_0 and pos_1: $\mathcal{O}_2 = \{\{q_0, q_1\}, \{q_2\}\}$



Schematic view



Multiagent system

- Can Robot 1 ensure that pos_1 is never reached?

  $\mathcal{M}, q_0 \models \langle\langle 1 \rangle\rangle G \neg pos_1$? No!

- Can both robots cooperate and ensure that pos_1 is never reached?

  $\mathcal{M}, q_0 \models \langle\langle 1, 2 \rangle\rangle G \neg pos_1$? Yes! (agree to play the same action forever)

- Can both robots cooperate and ensure that pos_1 is eventually reached?

  $\mathcal{M}, q_0 \models \langle\langle 1, 2 \rangle\rangle F\ pos_1$? Yes! (one pushes all the time, the other waits)

Model Checking ATL under Imperfect Information

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 $\langle\!\langle A \rangle\!\rangle \phi$, with ϕ 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 ϕ

Each formula ϕ has only one temporal operator after evaluating its subformulas and replacing with fresh atomic propositions!

☐

Theorem (C. Dima, F.L. Ţiplea 2011)

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

Relaxing ATL : ATL^*

- No restriction on operators: Wa can write $\varphi = \langle\langle A \rangle\rangle XXp$

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:

$$\text{state formula:}\quad \varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle A \rangle\rangle \psi$$

$$\text{path formula:}\quad \psi := \varphi \mid \neg\psi \mid \psi \vee \psi \mid X\psi \mid \psi \mathcal{U} \psi$$

- ATL is a fragment of ATL^*

ATL^*	**ir**	**iR**	**Ir**	**IR**
Complexity	PSPACE	Undecidable	PSPACE	2EXPTIME

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:

- $\textsc{Reach}(R_i)$: $\mathcal{M}, v_0 \models \langle\!\langle i \rangle\!\rangle F(\bigvee_{v \in R_i} v)$
- $\textsc{Safe}(S_i)$: $\mathcal{M}, v_0 \models \langle\!\langle i \rangle\!\rangle G(\bigvee_{v \in S_i} v)$
- $\textsc{Büchi}(T_i)$: $\mathcal{M}, v_0 \models \langle\!\langle i \rangle\!\rangle GF(\bigvee_{v \in T_i} v)$
- $\textsc{coBüchi}(T_i)$: $\mathcal{M}, v_0 \models \langle\!\langle i \rangle\!\rangle FG\neg(\bigvee_{v \in T_i} v)$
- $LTL(\varphi_i)$: $\mathcal{M}, v_0 \models \langle\!\langle i \rangle\!\rangle \varphi_i$

This does not necessarily give better algorithms!!!

Extending ATL with Knowledge operators : ATEL

- Recall: in imperfect information, each agent has a set $\mathcal{O}_i$ of observations
- The Alternating-time Temporal Epistemic Logic (ATEL) extends ATL with the knowkedge 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:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle A \rangle\!\rangle X \varphi \mid \langle\!\langle A \rangle\!\rangle G \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathcal{U} \varphi \mid K_i \varphi$$

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

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

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

Extending ATL with Knowledge operators : ATEL

- Recall: in imperfect information, each agent has a set $\mathcal{O}_i$ of observations
- The Alternating-time Temporal Epistemic Logic (ATEL) extends ATL with the knowkedge 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:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle A\rangle\rangle X\varphi \mid \langle\langle A\rangle\rangle G\varphi \mid \langle\langle A\rangle\rangle \varphi\mathcal{U}\varphi \mid K_i\varphi$$

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

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

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

Not much is known about algorithms and complexities! ☹

Bibliography

- Fabienne Sophie Eigner, "Alternating-Time Temporal Logic" - `https://www.react.uni-saarland.de/teaching/games-verification-synthesis-08/downloads/eigner_handout.pdf`
- `http://www.dis.uniroma1.it/~bloisi/didattica/apprendimento0809/atl.pdf`
- Wojciech Jamroga and Wojciech Penczek, "Specification and Verification of Multi-Agent Systems" - `http://krak.ipipan.waw.pl/~wjamroga/papers/verification12esslli-lncs.pdf`
- `https://www.react.uni-saarland.de/teaching/automata-games-verification-12/downloads/notes15.pdf`
- Pierre-Yves Schobbens, "Alternating-time logic with imperfect recall" - `http://www.win.tue.nl/~evink/entcs-vol85.2-lcmas2003/85.2.002.pdf`