Formal Aspects of Strategic Reasoning

Aniello Murano

Università degli Studi di Napoli Federico II

ESSAI 2024
Athens Greece
July 2024

Alternating-Time Temporal-Logic

[…from two…to multi-player games…]

From Two-Player to Multi-Player

- Module Checking is a basic setting to check for system correctness against an adversarial environment: two-player game «**Sys vs. Env**»
- It is suitable for several system verification scenario, but very specific:

  The system has only **one strategy**

  The environment has the ability to **non-deterministically** disable possible evolution of the game

- Nowadays systems are composed of **several agents**, autonomous and rational, each one with its own goal, interacting among them and sensing the other agents.
- An important contribution in this field:

Alternating-Time Temporal Logic

[Alur, Kupferman, Henzinger. J. of ACM 2002]

Agents in ATL

- ATL generalizes CTL: temporal operators are indexed by coalitions of agents.
- Formally, path quantifiers A and E are replaced with the strategic cooperative quantifiers **≪A≫** and **[[A]]**, where **A is a team of agents**.
- ≪A≫ϕ means that coalition A has a (collective) strategy to enforce ϕ, no matter what the other agents (not in A) will behave.
- Strategic quantifiers allow for a selective extraction of paths over a (game) model.
- As for CTL*, we can have ATL*

Syntax of ATL and ATL*

- ATL* contains state-formulas and path-formulas.
- ATL* state-formulas are formed according to the grammar:
  - $\phi := true \mid p \mid \phi \wedge \phi \mid \neg\phi \mid \langle\langle A \rangle\rangle\, \psi$
    where $p \in AP$ and ψ is a path-formula
- ATL* path-formulas are as in CTL*:
  - $\psi := \phi \mid \psi \wedge \psi \mid \neg\psi \mid X\psi \mid \psi\, U\, \psi$
    where ϕ is a state-formula, and ψ a path-formula

Syntax of ATL and ATL*

- ATL* contains state-formulas and path-formulas.
- ATL* state-formulas are formed according to the grammar:
  - $\phi := true \mid p \mid \phi \wedge \phi \mid \neg\phi \mid \langle\langle A \rangle\rangle\, \psi$

    where $p \in AP$ and ψ is a path-formula
- ATL* path-formulas are as in CTL*:
  - $\psi := \phi \mid \psi \wedge \psi \mid \neg\psi \mid X\psi \mid \psi\, U\, \psi$

    where ϕ is a state-formula, and ψ a path-formula
- In ATL path-formulas are reduced to:
  - $\psi := X\, \phi \mid \phi\, U\, \phi$

    where ϕ is a state-formula.
- Note that in ATL, X and U alternate with $\langle\langle A \rangle\rangle$ and its dual [[A]]

Example

Example

- Agents ={  ,  } 

Example

- Agents ={  ,  } 

- $\langle\langle$Lupin$\rangle\rangle$ (G run away $\wedge$ F diamods)

  "Lupin has a strategy to stay (always) away from Zenigata and eventually get some diamonds"

Example

- Agents ={  ,  } 

- ≪Lupin≫ (G run away ∧ F diamods)

  "Lupin has a strategy to stay (always) away from Zenigata and eventually get some diamonds"

- ≪Lupin, Margot≫ fun Until caught

  "Lupin and Margot have fun with money until they get caught from Zenigata"

ATL Semantics: CGS

- ATL can be interpreted over **Concurrent Game Structures** (CGS):

$C = (AP, Ag, Ac, S, S_0, R, Lab)$

- AP is a set of atomic propositions
- Ag is a set of agents
- Ac is a set of Actions
- S is a set of states
- $S_0 \subseteq S$ is the set of initial states
- $Lab : S \rightarrow 2^{AP}$ labels each state with propositions true in the state
- Let $Dc: Ag \rightarrow Ac$ be the set of agent's decisions (action choices). Then, we have $R : S \times Dc \rightarrow S$

Example

- Let M = (AP, Ag, Ac, S, S_0, R, Lab) be a GCS representing a city with a Bank and a Museum in which Lupin and Zenigata, startiting from Home, compete

Example

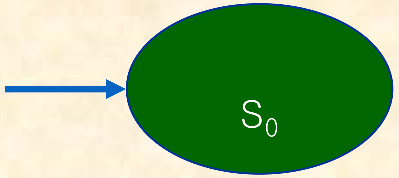
- Let M = (AP, Ag, Ac, S, S_0, R, Lab) be a GCS representing a city with a Bank and a Museum in which Lupin and Zenigata, startiting from Home, compete
- We have:
  - Ag = {L, Z}

Example

- Let M = (AP, Ag, Ac, S, S_0, R, Lab) be a GCS representing a city with a Bank and a Museum in which Lupin and Zenigata, startiting from Home, compete
- We have:
  - Ag = {L, Z}
  - Ac = {B, M, H}

Example

- Let M = (AP, Ag, Ac, S, S_0, R, Lab) be a GCS representing a city with a Bank and a Museum in which Lupin and Zenigata, startiting from Home, compete
- We have:
  - Ag = {L, Z}
  - Ac = {B, M, H}
  - S = Ag x Ac
  - S_0 = Ag x {H}

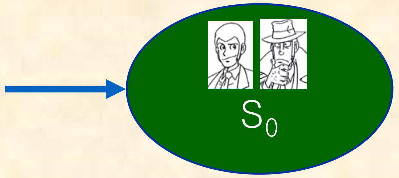


Example

- Let M = (AP, Ag, Ac, S, S_0, R, Lab) be a GCS representing a city with a Bank and a Museum in which Lupin and Zenigata, startiting from Home, compete
- We have:
  - Ag = {L, Z}
  - Ac = {B, M, H}
  - S = Ag x Ac
  - S_0 = Ag x {H}
  - AP ={Win_L, Win_z}
  - Lab:

    Lupin's winning states → Win_L

    Lupin's losing states → Win_Z

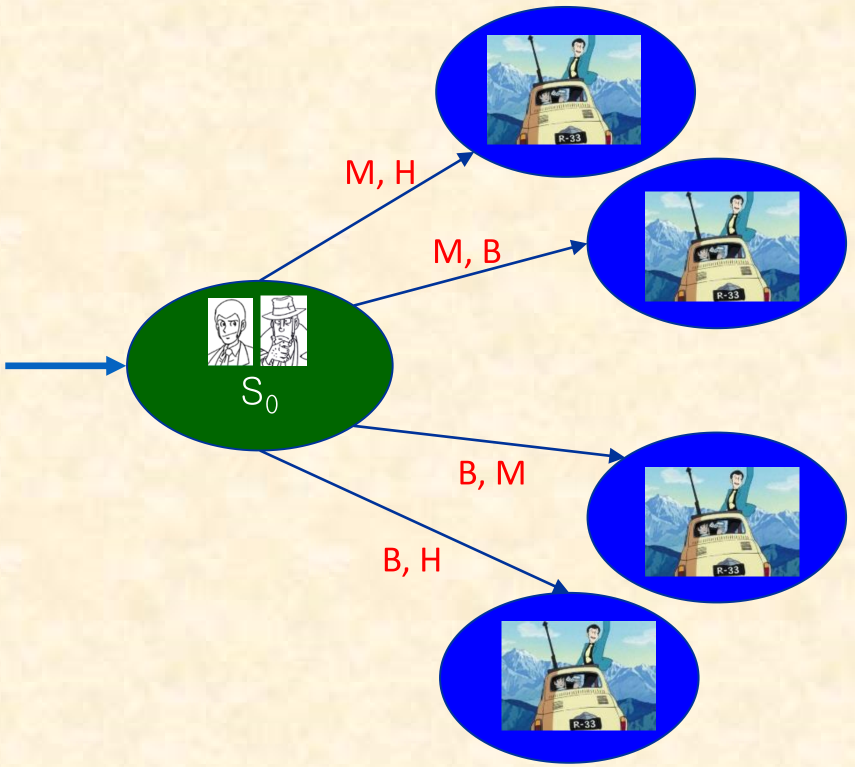


Example

- Let M = (AP, Ag, Ac, S, S_0, R, Lab) be a GCS representing a city with a Bank and a Museum in which Lupin and Zenigata, startiting from Home, compete
- We have:
  - Ag = {L, Z}
  - Ac = {B, M, H}
  - S = Ag x Ac
  - S_0 = Ag x {H}
  - AP ={Win_L, Win_z}
  - Lab:

    Lupin's winning states → Win_L

    Lupin's losing states → Win_Z
  - R is as in the figure

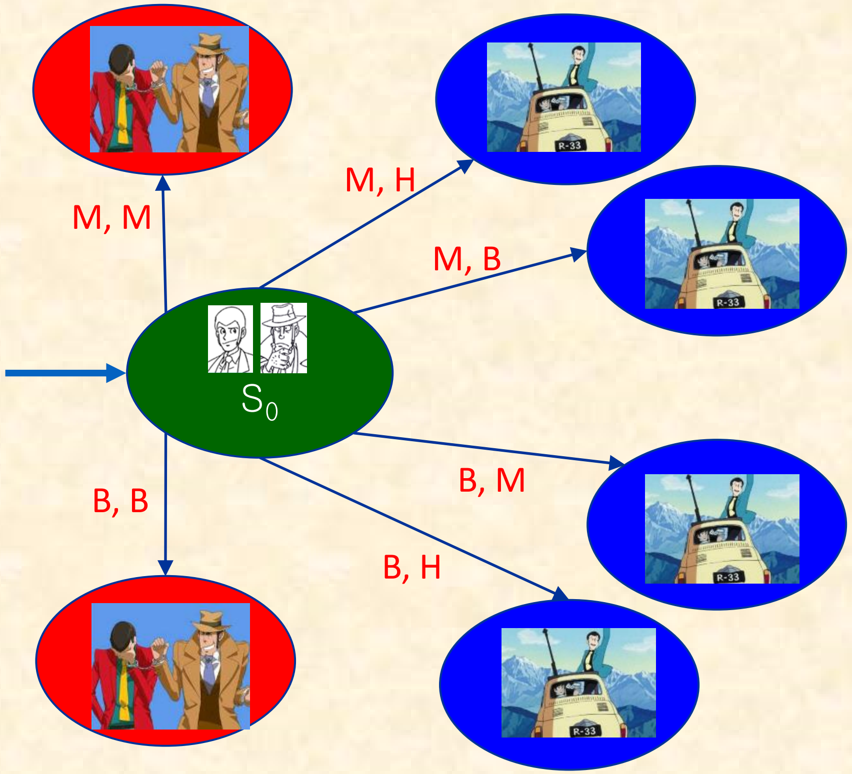


Example

- Let M = (AP, Ag, Ac, S, S_0, R, Lab) be a GCS representing a city with a Bank and a Museum in which Lupin and Zenigata, startiting from Home, compete
- We have:
  - Ag = {L, Z}
  - Ac = {B, M, H}
  - S = Ag x Ac
  - S_0 = Ag x {H}
  - AP ={Win_L, Win_z}
  - Lab:

    Lupin's winning states → Win_L

    Lupin's losing states → Win_Z
  - R is as in the figure

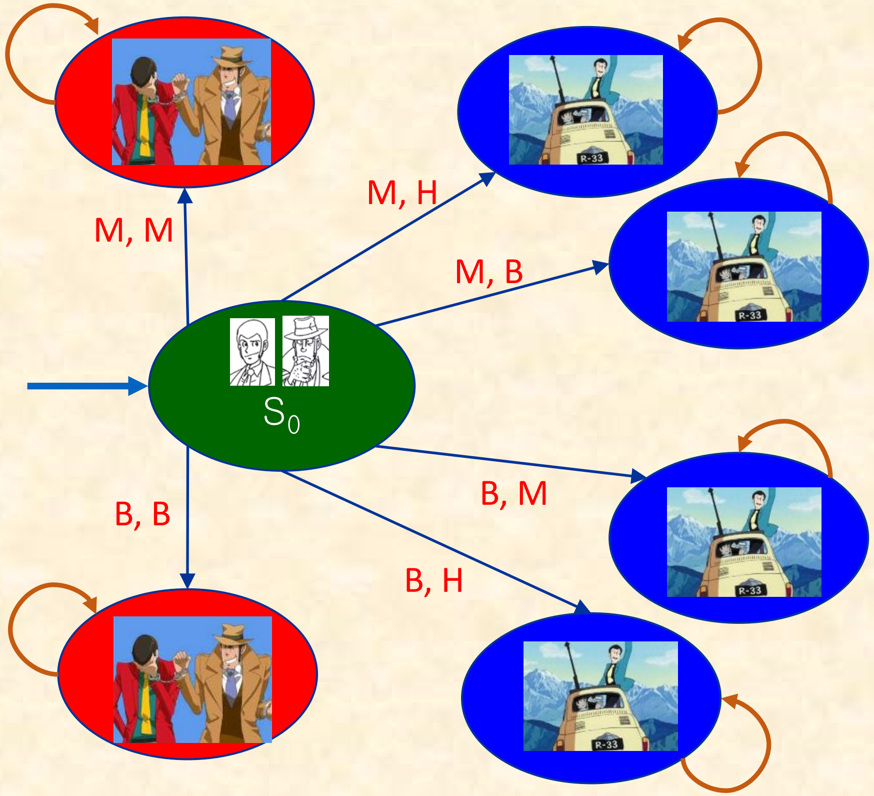


Example

- Let M = (AP, Ag, Ac, S, S_0, R, Lab) be a GCS representing a city with a Bank and a Museum in which Lupin and Zenigata, startiting from Home, compete
- We have:
  - Ag = {L, Z}
  - Ac = {B, M, H}
  - S = Ag x Ac
  - S_0 = Ag x {H}
  - AP ={Win_L, Win_z}
  - Lab:

    Lupin's winning states → Win_L

    Lupin's losing states → Win_Z
  - R is as in the figure

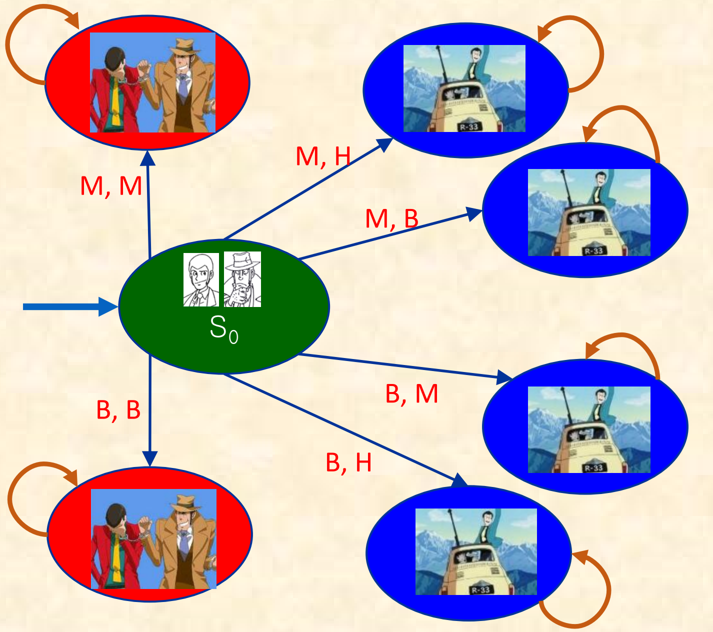


ATL Semantics: Strategies

- A strategy for an agent a is s_a: $St^+ \rightarrow Act$
- It is a **memoryfull** conditional plan that specifies which decision the agent a has to take in every possible situation.
- Formally, it is considered a **perfect recall** strategy
- As in Module Checking, we can have a memoryless (imperfect recall) strategy is$_a$: $St \rightarrow Act$. We will come back on this later…
- A collective strategy S_A for a group of agents A is a tuple of strategies, one for each agent in A.
- The **outcome** of the team A from a state q, $out(q,S_A)$, is the set of all paths that result from agents A executing S_A (concurrently)
- $M, q \models \langle\langle A \rangle\rangle \phi$ iff there is S_A, such that $M, \pi \models \phi$ for every $\pi \in out(q,S_A)$.
- CTL path quantifiers can be embedded in ATL:
  - $E\phi \equiv \langle\langle Agt \rangle\rangle \phi$
  - $A\phi \equiv \langle\langle \emptyset \rangle\rangle \phi$

Examples

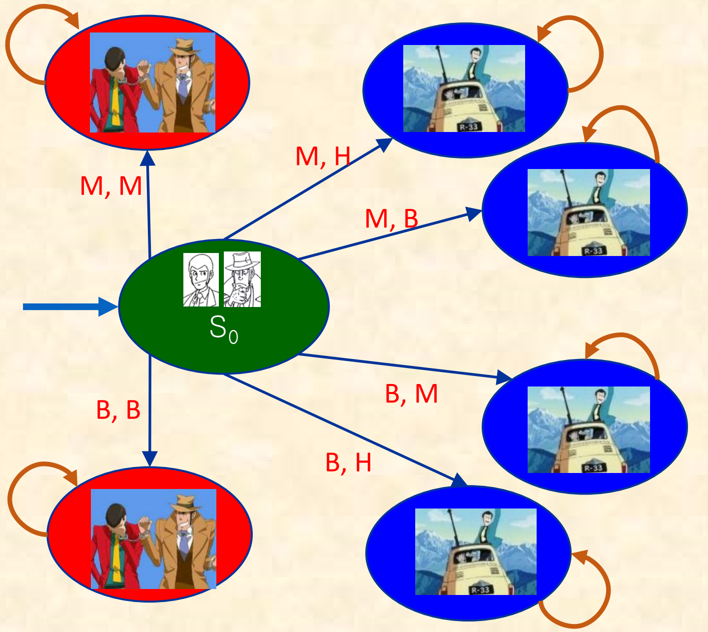
- Let us consider again the Lupin-Zenigata game M



Examples

- Let us consider again the Lupin-Zenigata game M

- Recall that M, q ⊨ ≪A≫ ϕ iff there exists a collective strategy S_A, such that G, π ⊨ ϕ for every π ∈ out(q,S_A)

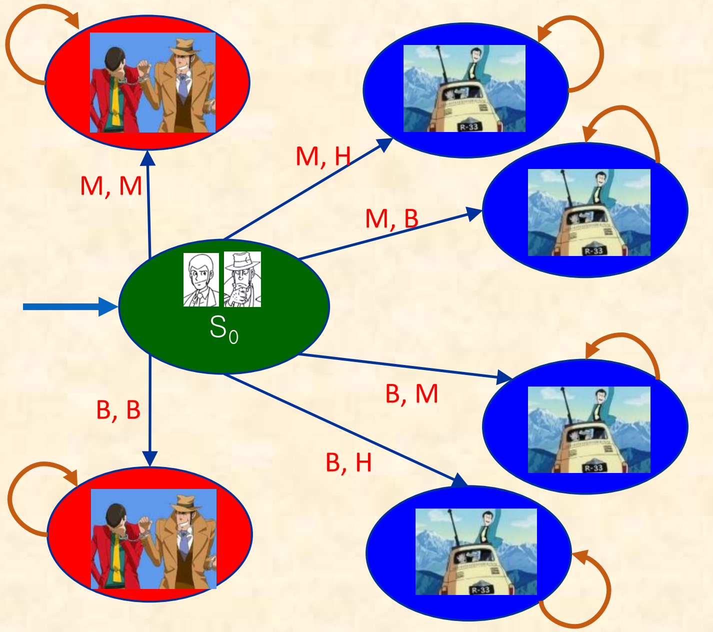


Examples

- Let us consider again the Lupin-Zenigata game M

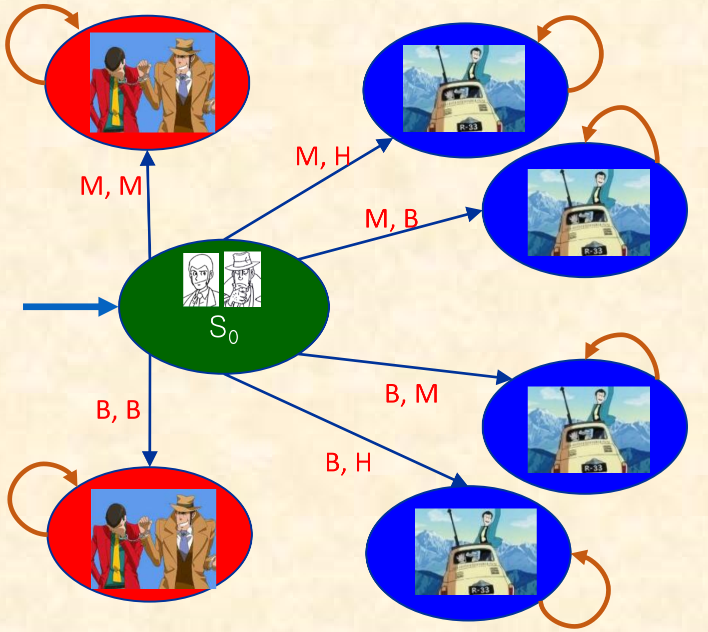
- Recall that $M, q \models \langle\langle A \rangle\rangle \phi$ iff there exists a collective strategy S_A, such that $G, \pi \models \phi$ for every $\pi \in out(q, S_A)$

- $M \not\models \langle\langle Lupin \rangle\rangle G\ Win_L$
  - Lupin does not have a strategy to Win



Examples

- Let us consider again the Lupin-Zenigata game M
- Recall that M, q $\models \langle\langle A \rangle\rangle\ \phi$ iff there exists a collective strategy S_A, such that G, $\pi \models \phi$ for every $\pi \in out(q, S_A)$
- M $\not\models \langle\langle$ Lupin $\rangle\rangle$ G Win_L
  - Lupin does not have a strategy to Win
- M $\models \langle\langle$Lupin, Zenigata$\rangle\rangle$ F Win_L
  - Lupin wins if he cooperates with Zenigata
  - Note: this is a Liveness property, e.g., something good will happen



ATL and ATL* Model checking

- For ATL, a fix-point algorithm is easy and effective:
  - You need to calculate Pre(A, Q): the states q from which the coalition A con force the game to reach Q, no matter how the other agents will play.

- For ATL*, one can reduce to parity games, or use and automata-theoretic approach via Parity condition. The latter extends the one used for CTL*

function $mcheck(\mathcal{M}, \varphi)$.
Global model checking formulae of ATL.
Returns the exact subset of St for which formula φ holds.

case $\varphi \equiv p$ **:** return $\mathcal{V}(p)$
case $\varphi \equiv \neg\psi$ **:** return $St \setminus mcheck(\mathcal{M}, \psi)$
case $\varphi \equiv \psi_1 \wedge \psi_2$ **:** return $mcheck(\mathcal{M}, \psi_1) \cap mcheck(\mathcal{M}, \psi_2)$
case $\varphi \equiv \langle\langle A\rangle\rangle \mathrm{X}\psi$ **:** return $pre(A, mcheck(\mathcal{M}, \psi))$
case $\varphi \equiv \langle\langle A\rangle\rangle \mathrm{G}\psi$ **:**
$Q_1 := Q; \quad Q_2 := Q_3 := mcheck(\mathcal{M}, \psi);$
while $Q_1 \not\subseteq Q_2$ **do** $Q_1 := Q_1 \cap Q_2; \; Q_2 := pre(A, Q_1) \cap Q_3$ **od**;
return Q_1
case $\varphi \equiv \langle\langle A\rangle\rangle \psi_1 \mathrm{U} \psi_2$ **:**
$Q_1 := \emptyset; \quad Q_2 := mcheck(\mathcal{M}, \psi_2); \quad Q_3 := mcheck(\mathcal{M}, \psi_1);$
while $Q_2 \not\subseteq Q_1$ **do** $Q_1 := Q_1 \cup Q_2; \; Q_2 := pre(A, Q_1) \cap Q_3$ **od**;
return Q_1
end case

$$\text{pre}(A, Q) = \{q \mid \exists \alpha_A \forall \alpha_{\mathrm{Agt}\setminus A}\, o(q, \alpha_A, \alpha_{\mathrm{Agt}\setminus A}) \in Q\}$$

ATL decision problems

Complexity Results for ATL			
Logic	**Model Checking w.r.t.system**	**Model Checking**	**Satisfiability**
LTL	NLOGSPACE [4]	PSPACE [5]	PSPACE [4]
CTL	NLOGSPACE [6]	PTIME [5]	EXPTIME [2]
CTL*	NLOGSPACE [6]	PSPACE [5]	2EXPTIME [4]
ATL	PTIME [3]	PTIME [3]	EXPTIME [7]
ATL*	PTIME [3]	2EXPTIME [3]	2EXPTIME [8]

[2] Clarke, Emerson: Logics of Programs 1981

[3] Alur, Henzinger, Kupferman: JACM 2002

[4] Emerson: Temporal and modal logic. MIT Press 1990

[5] Clarke, Emerson, Sistla. TOPLAS 1986

[6] Kupferman, Vardi, Wolper. JACM 2000

[7] Walther, Lutz, Wolter, Wooldridge: J. of Logic and Computation 2006

[8] Schewe. ICALP 2008

ATL vs. Module Checking

- Module checking
  - Two-player game (system vs. environment)
  - Environment strategies come through Exec(M)
  - CTL Module Ckecking is EXPTIME-complete (PTime in the model)

- ATL
  - Multi-player
  - Strategies come from coalition of agents.
  - ATL model checking is PTIME in |states| of M and |ϕ|, but notice that **|M| is exponential in the number of agents**

Part 2

- We keep talking about logics for strategic reasoning
- We introduce Strategy Logic as a powerful extension of ATL
- In ATL
  - Strategies are treated implicitly
  - Agents cannot share strategies nor reuse some from the past.
  - Every time an agent appears in a formula, previous strategies are reset

- In Strategy Logic
  - Strategies are unpacked from agents and used as first order objects.
  - Strategies can be reused and shared among agents.
  - Several complex and useful specifications can be expressed without effecting the overall decision complexities. Among the others: **Nash Equilibrium**.

Let’s have a break!