Towards efficient model checking for variants of ATL under different semantics

Wojciech Penczek

a joint work with
W. Jamroga, B. Konikowska, M. Knapik, L. Petrruci and A. Etienne

Institute of Computer Sciences, PAS, Warsaw, and Siedlce University, Poland

Bordeaux, Talence, WG2.2 Meeting, the 20th of September

Outline

- Introduction to specification of strategic abilities in ATL*,
- Model checking multi-valued version of ATL*,
- Partial order reductions for sATL*,
- Simpler strategies for Timed ATL (if time permits).

Specification and Verification of Strategic Ability

- Many important properties are based on **strategic ability**
- Functionality ≈ ability of authorized users to complete some tasks
- Security ≈ inability of unauthorized users to complete certain tasks
- One can try to formalize such properties in modal logics of strategic ability, such as ATL or Strategy Logic
- ...and verify them by model checking

Specification and Verification of Strategic Ability

- Many important properties are based on **strategic ability**
- Functionality ≈ ability of authorized users to complete some tasks
- Security ≈ inability of unauthorized users to complete certain tasks
- One can try to formalize such properties in modal logics of strategic ability, such as ATL or Strategy Logic
- ...and verify them by model checking

Specification and Verification of Strategic Ability

- Many important properties are based on **strategic ability**
- Functionality $\approx$ ability of authorized users to complete some tasks
- Security $\approx$ inability of unauthorized users to complete certain tasks
- One can try to formalize such properties in modal logics of strategic ability, such as ATL or Strategy Logic
- ...and verify them by model checking

Motivation: VoteVerif

- New project has just began between the Polish Academy of Sciences and University of Luxembourg
- **VoteVerif**: Verification of Voter-Verifiable Voting Protocols
- Example properties: ballot confidentiality, coercion-resistance, end-to-end voter-verifiability
- Underpinned by existence (or nonexistence) of a suitable strategy for the voter and/or the coercer

Motivation: VoteVerif

- New project has just began between the Polish Academy of Sciences and University of Luxembourg
- **VoteVerif**: Verification of Voter-Verifiable Voting Protocols
- Example properties: ballot confidentiality, coercion-resistance, end-to-end voter-verifiability
- Underpinned by existence (or nonexistence) of a suitable strategy for the voter and/or the coercer

Papers introducing ATL* and TATL

- Alternating-time temporal logic [Alur et al. 1997-2002]
- Timed alternating-time temporal logic [Henzinger and Prabhu, LAMAS 2006]
- Model checking timed ATL for durational concurrent game structures [Laroussinie, Markey, Oreiby, LAMAS 2006]

ATL: What Agents Can Achieve

- ATL: Alternating-time Temporal Logic
- Temporal logic meets game theory
- Main idea: cooperation modalities

$\langle\langle A \rangle\rangle \phi$: coalition A has a collective strategy to enforce ϕ

$\leadsto$ ϕ can include temporal operators: X (next), F (sometime in the future), G (always in the future), U (strong until)

Semantic Variants of ATL

- Basic semantics of ATL assumes perfect information - not very realistic
- Semantic variants for more realistic cases defined in (Jamroga 2003), (Jonker 2003), (Schobbens 2004), (Jamroga & van der Hoek 2004), (Agotnes 2004), ...
- Encapsulate different assumptions about agents and abilities

Semantic Variants of ATL*

Memory of agents:

- Perfect Recall (R) vs. imperfect recall strategies (r)

Available information:

- Perfect Information (I) vs. imperfect information strategies (i)

ATL: What Agents Can Achieve

Example formulae:

- $\bigwedge_{i \in Candidates} \langle\langle v \rangle\rangle \mathrm{F}\ \mathrm{voted}_{\mathrm{v,i}}$:
  "The voter can cast her vote in an arbitrary way"
- $\neg\langle\langle c, v \rangle\rangle \mathrm{F} \bigvee_{i \in Candidates} K_c \mathrm{voted}_{\mathrm{v,i}}$:
  "The coercer cannot learn how the voter voted even if the voter cooperates with the coercer" (in ATL + K)

So, **let's specify and model-check**!

Not that easy...

ATL: What Agents Can Achieve

Example formulae:

- $\bigwedge_{i \in Candidates} \langle\langle v \rangle\rangle \mathrm{F}\ \mathrm{voted}_{\mathrm{v,i}}$:
  "The voter can cast her vote in an arbitrary way"
- $\neg\langle\langle c, v \rangle\rangle \mathrm{F} \bigvee_{i \in Candidates} K_c \mathrm{voted}_{\mathrm{v,i}}$:
  "The coercer cannot learn how the voter voted even if the voter cooperates with the coercer" (in ATL + K)

So, **let's specify and model-check**!

Not that easy...

ATL: What Agents Can Achieve

Example formulae:

- $\bigwedge_{i \in Candidates} \langle\langle v \rangle\rangle \mathrm{F}\ \mathrm{voted}_{\mathrm{v,i}}$:
  "The voter can cast her vote in an arbitrary way"
- $\neg\langle\langle c, v \rangle\rangle \mathrm{F} \bigvee_{i \in Candidates} K_c \mathrm{voted}_{\mathrm{v,i}}$:
  "The coercer cannot learn how the voter voted even if the voter cooperates with the coercer" (in ATL + K)

So, **let's specify and model-check**!

Not that easy...

ATL: What Agents Can Achieve

Example formulae:

- $\bigwedge_{i \in Candidates} \langle\langle v \rangle\rangle \mathrm{F}\, \mathrm{voted}_{\mathrm{v,i}}$:
  "The voter can cast her vote in an arbitrary way"
- $\neg\langle\langle c, v \rangle\rangle \mathrm{F} \bigvee_{i \in Candidates} K_c \mathrm{voted}_{\mathrm{v,i}}$:
  "The coercer cannot learn how the voter voted even if the voter cooperates with the coercer" (in ATL + K)

So, **let's specify and model-check**!

Not that easy...

Not That Easy...

Caveat: there are serious **complexity obstacles**:

- Model checking agent logics for agents with perfect information ranges from **P-complete** to **EXPTIME-compl.**,
- Model checking agent logics for agents with imperfect information ranges from **NP-complete** to **undecidable**, depending on the exact syntax, semantics, and representation of models.
- Model checking ATL under imperfect information and imperfect recall is Δ_2^P**-complete** (in the size of a model and a formula).

Not That Easy...

These manifest in:

- State-space explosion,
- Transition-space explosion,
- Invalidity of fixpoint equivalences for ATL under imperfect information (see N. Bulling, C. Dima, V. Goranko, W. Jamroga, ...).

What to do ?

Possible ways out:...

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - **multi-valued model checking** over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- **Simpler strategies** - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible ways out:...

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - **multi-valued model checking** over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- **Simpler strategies** - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible ways out:...

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - **multi-valued model checking** over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- **Simpler strategies** - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible ways out:...

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - **multi-valued model checking** over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- **Simpler strategies** - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible ways out:...

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - **multi-valued model checking** over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- **Simpler strategies** - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible ways out:...

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - **multi-valued model checking** over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- **Simpler strategies** - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Motivation: Multi-Valued Abstraction

State abstraction:

- Cluster similar states into new abstract states
- Model checking over new abstract models

Possible problems:

- Even the values of some basic properties can be hard to compute in some states ⇝ undefined truth values
- Clustered states may disagree on some basic properties ⇝ inconsistent truth values

This leads to **multi-valued verification**

Motivation: Multi-Valued Abstraction

State abstraction:

- Cluster similar states into new abstract states
- Model checking over new abstract models

Possible problems:

- Even the values of some basic properties can be hard to compute in some states $\rightsquigarrow$ undefined truth values
- Clustered states may disagree on some basic properties $\rightsquigarrow$ inconsistent truth values

This leads to **multi-valued verification**

Motivation: Multi-Valued Abstraction

State abstraction:

- Cluster similar states into new abstract states
- Model checking over new abstract models

Possible problems:

- Even the values of some basic properties can be hard to compute in some states $\rightsquigarrow$ undefined truth values
- Clustered states may disagree on some basic properties $\rightsquigarrow$ inconsistent truth values

This leads to **multi-valued verification**

Syntax

ATL* syntax in Negation Normal Form, augmented with constants for logical values L, and operator $\preccurlyeq$ for comparing truth values:

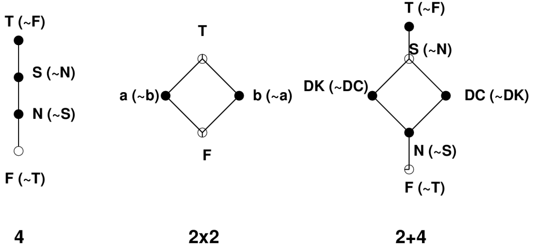
$$\phi ::= \mathrm{c} \mid \mathrm{p} \mid \neg \mathrm{p} \mid \phi \wedge \phi \mid \phi \vee \phi \mid \langle\langle A \rangle\rangle \gamma \mid \overline{\langle\langle A \rangle\rangle}\, \gamma \mid \phi \preccurlyeq \phi,$$

$$\gamma ::= \phi \mid \gamma \wedge \gamma \mid \gamma \vee \gamma \mid \mathrm{X}\,\gamma \mid \gamma\,\mathrm{U}\,\gamma \mid \gamma R \gamma,$$

where $c \in L$ and $p \in \mathcal{AP}$.

Models

ATL models with atomic propositions are interpreted in a distributive quasi-Boolean algebra (DM algebra) of truth values



Every element *x* in a DM algebra can be represented by the join of the join-irreducible elements smaller or equal than x.

Models - synchronous semantics

A Concurrent Game Structure is a 7 –tuple
$\mathcal{A} = (Agents, \Sigma, \mathcal{Q}, \mathcal{AP}, \mathcal{V}, protocol, trans)$, where:

- *Agents* is a finite set of all the agents,
- Σ is a finite set of actions,
- $\mathcal{Q}$ is a finite set of global locations,
- $\mathcal{AP}$ is a set of atomic propositions,
- $\mathcal{V} \colon \mathcal{Q} \times \mathcal{AP} \to \{\bot, \top\}$ is a valuation function,
- $protocol \colon Agents \times \mathcal{Q} \to \mathcal{P}(\Sigma) \setminus \{\emptyset\}$ is a protocol function,
- $trans \colon \mathcal{Q} \times \Sigma^{|Agents|} \to \mathcal{Q}$ is a transition function consistent with *protocol* for each agent of *Agents*.

Models - synchronous semantics

A MV-Concurrent Game Structure is a 7 –tuple $\mathcal{A} = (Agents, \Sigma, \mathcal{Q}, \mathcal{AP}, \mathcal{V}, protocol, trans)$, where:

- *Agents* is a finite set of all the agents,
- Σ is a finite set of actions,
- $\mathcal{Q}$ is a finite set of global locations,
- $\mathcal{AP}$ is a set of atomic propositions,
- $\mathcal{V}: \mathcal{Q} \times \mathcal{AP} \rightarrow L$ is a valuation function,
- $protocol: Agents \times \mathcal{Q} \rightarrow \mathcal{P}(\Sigma) \setminus \{\emptyset\}$ is a protocol function,
- $trans: \mathcal{Q} \times \Sigma^{|Agents|} \rightarrow \mathcal{Q}$ is a transition function consistent with *protocol* for each agent of *Agents*.

Models - synchronous semantics

A Tight Durational Concurrent Game Structure is a 7 –tuple $\mathcal{A} = (Agents, \Sigma, \mathcal{Q}, \mathcal{AP}, \mathcal{V}, protocol, trans)$, where:

- *Agents* is a finite set of all the agents,
- Σ is a finite set of actions,
- $\mathcal{Q}$ is a finite set of global locations,
- $\mathcal{AP}$ is a set of atomic propositions,
- $\mathcal{V} : \mathcal{Q} \times \mathcal{AP} \to \{\bot, \top\}$ is a valuation function,
- $protocol : Agents \times \mathcal{Q} \to \mathcal{P}(\Sigma) \setminus \{\emptyset\}$ is a protocol function,
- $trans : \mathcal{Q} \times \Sigma^{|Agents|} \to \mathcal{Q} \times \mathbb{N}_+$ is a transition function consistent with *protocol* for each agent of *Agents*.

Models - synchronous semantics

A Concurrent Game Structure is a 7 –tuple $\mathcal{A} = (Agents, \Sigma, \mathcal{Q}, \mathcal{AP}, \mathcal{V}, protocol, trans)$, where:

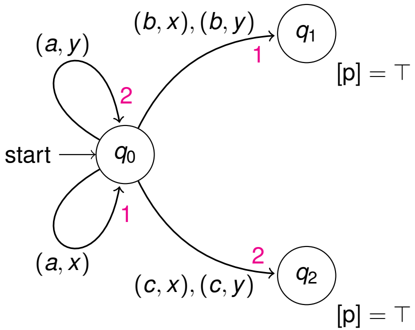
- *Agents* is a finite set of all the agents,
- Σ is a finite set of actions,
- $\mathcal{Q}$ is a finite set of global locations,
- $\mathcal{AP}$ is a set of atomic propositions,
- $\mathcal{V} \colon \mathcal{Q} \times \mathcal{AP} \to \{\bot, \top\}$ is a valuation function,
- $protocol \colon Agents \times \mathcal{Q} \to \mathcal{P}(\Sigma) \setminus \{\emptyset\}$ is a protocol function,
- $trans \colon \mathcal{Q} \times \Sigma^{|Agents|} \to \mathcal{Q}$ is a transition function consistent with *protocol* for each agent of *Agents*.

Models - synchronous semantics

A Concurrent Game Structure is a 8–tuple $\mathcal{A} = (Agents, \Sigma, \mathcal{Q}, \mathcal{AP}, \mathcal{V}, protocol, trans, \{\sim_a \mid a \in Agents\})$, where:

- $Agents$ is a finite set of all the agents,
- Σ is a finite set of actions,
- $\mathcal{Q}$ is a finite set of global locations,
- $\mathcal{AP}$ is a set of atomic propositions,
- $\mathcal{V}\colon \mathcal{Q} \times \mathcal{AP} \to \{\bot, \top\}$ is a valuation function,
- $protocol\colon Agents \times \mathcal{Q} \to \mathcal{P}(\Sigma) \setminus \{\emptyset\}$ is a protocol function,
- $trans\colon \mathcal{Q} \times \Sigma^{|Agents|} \to \mathcal{Q}$ is a transition function consistent with *protocol* for each agent of *Agents*.
- $\sim_a \subseteq \mathcal{Q} \times \mathcal{Q}$, for each $a \in Agents$, is an indistinguishability relation.

Example of a Model



Perfect Information Strategies - *I*

Let $a \in Agents$:

Perfect recall (R), perfect information strategies (I) ($\Sigma_{R,I}$)

Functions $\sigma_a \colon \mathcal{Q}^+ \to \Sigma$ s.t., $\forall_{\pi \in \mathcal{Q}^+} \sigma_a(\pi) \in protocol(a, \pi_F)$.

(Intuition: no constraints, apart from the protocol of agent *a*)

Imperfect recall (r), perfect information strategies (I) ($\Sigma_{r,I}$)

Strategies $\sigma_a \in \Sigma_{r,I}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi_F = \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on the final location)

π_F: the final global location of π

Perfect Information Strategies - I

Let $a \in Agents$:

Perfect recall (R), perfect information strategies (I) ($\Sigma_{R,I}$)

Functions $\sigma_a\colon \mathcal{Q}^+ \to \Sigma$ s.t., $\forall_{\pi \in \mathcal{Q}^+} \sigma_a(\pi) \in protocol(a, \pi_F)$.

(Intuition: no constraints, apart from the protocol of agent a)

Imperfect recall (r), perfect information strategies (I) ($\Sigma_{r,I}$)

Strategies $\sigma_a \in \Sigma_{r,I}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi_F = \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent a selects an action based on the final location)

π_F: the final global location of π

Perfect Information Strategies - *I*

Let $a \in Agents$:

Perfect recall (R), perfect information strategies (I) ($\Sigma_{R,I}$)

Functions $\sigma_a \colon \mathcal{Q}^+ \to \Sigma$ s.t., $\forall_{\pi \in \mathcal{Q}^+} \sigma_a(\pi) \in protocol(a, \pi_F)$.

(Intuition: no constraints, apart from the protocol of agent *a*)

Imperfect recall (r), perfect information strategies (I) ($\Sigma_{r,I}$)

Strategies $\sigma_a \in \Sigma_{r,I}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi_F = \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on the final location)

π_F: the final global location of π

Perfect Information Strategies - I

Let $a \in Agents$:

Perfect recall (R), perfect information strategies (I) ($\Sigma_{R,I}$)

Functions $\sigma_a \colon \mathcal{Q}^+ \to \Sigma$ s.t., $\forall_{\pi \in \mathcal{Q}^+} \sigma_a(\pi) \in protocol(a, \pi_F)$.

(Intuition: no constraints, apart from the protocol of agent a)

Imperfect recall (r), perfect information strategies (I) ($\Sigma_{r,I}$)

Strategies $\sigma_a \in \Sigma_{r,I}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi_F = \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent a selects an action based on the final location)

π_F: the final global location of π

Imperfect Information Strategies - *i*

Let $a \in Agents$:

Perfect recall (R), imperfect information strategies (i) ($\Sigma_{R,i}$)

Strategies $\sigma_a \in \Sigma_{R,i}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi(0) \sim_a \pi'(0), \ldots, \pi_F \sim_a \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on its view of the history)

Imperfect recall (r), imperfect information strategies (i) ($\Sigma_{r,i}$)

Strategies $\sigma_a \in \Sigma_{r,I}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi_F \sim_a \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on its view of the final location)

Imperfect Information Strategies - *i*

Let $a \in Agents$:

Perfect recall (R), imperfect information strategies (i) ($\Sigma_{R,i}$)

Strategies $\sigma_a \in \Sigma_{R,i}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi(0) \sim_a \pi'(0), \ldots, \pi_F \sim_a \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on its view of the history)

Imperfect recall (r), imperfect information strategies (i) ($\Sigma_{r,i}$)

Strategies $\sigma_a \in \Sigma_{r,I}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi_F \sim_a \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on its view of the final location)

Imperfect Information Strategies - *i*

Let $a \in Agents$:

Perfect recall (R), imperfect information strategies (i) ($\Sigma_{R,i}$)

Strategies $\sigma_a \in \Sigma_{R,i}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi(0) \sim_a \pi'(0), \ldots, \pi_F \sim_a \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on its view of the history)

Imperfect recall (r), imperfect information strategies (i) ($\Sigma_{r,i}$)

Strategies $\sigma_a \in \Sigma_{r,I}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi_F \sim_a \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on its view of the final location)

Imperfect Information Strategies - *i*

Let $a \in Agents$:

Perfect recall (R), imperfect information strategies (i) ($\Sigma_{R,i}$)

Strategies $\sigma_a \in \Sigma_{R,i}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi(0) \sim_a \pi'(0), \ldots, \pi_F \sim_a \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on its view of the history)

Imperfect recall (r), imperfect information strategies (i) ($\Sigma_{r,i}$)

Strategies $\sigma_a \in \Sigma_{r,I}$ s.t., for each $\pi, \pi' \in \mathcal{Q}^+$, if $\pi_F \sim_a \pi'_F$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: agent *a* selects an action based on its view of the final location)

Joint Strategies

- A joint strategy σ_A for agents $A \subseteq Agents$ is a tuple of strategies, one per agent $a \in A$.
- The outcome of σ_A in location $q \in \mathcal{Q}$ is the set $out(q, \sigma_A) \subseteq \mathcal{Q}^\omega$ s.t. $\pi \in out(q, \sigma_A)$ iff $\pi(0) = q$ and for each $i \in \mathbb{N}$: $\pi(i) \xrightarrow{\text{act}'} \pi(i+1)$ for some $\text{act}' \in \Sigma$ s.t. $\text{act}'|_A = \sigma_A(\pi_i)$ and $\text{act}'|_{\overline{A}} \in protocol_{\overline{A}}(\pi(i))$.

Intuition: when coalition A follows σ_A, then in every global location, coalition A selects actions according to the joint strategy while the remaining agents $\overline{A}$ can choose any actions.

Joint Strategies

- A joint strategy σ_A for agents $A \subseteq \mathit{Agents}$ is a tuple of strategies, one per agent $a \in A$.
- The outcome of σ_A in location $q \in \mathcal{Q}$ is the set $\mathit{out}(q, \sigma_A) \subseteq \mathcal{Q}^\omega$ s.t. $\pi \in \mathit{out}(q, \sigma_A)$ iff $\pi(0) = q$ and for each $i \in \mathbb{N}$: $\pi(i) \xrightarrow{\mathrm{act}'} \pi(i+1)$ for some $\mathrm{act}' \in \Sigma$ s.t. $\mathrm{act}'|_A = \sigma_A(\pi_i)$ and $\mathrm{act}'|_{\overline{A}} \in \mathit{protocol}_{\overline{A}}(\pi(i))$.

Intuition: when coalition A follows σ_A, then in every global location, coalition A selects actions according to the joint strategy while the remaining agents $\overline{A}$ can choose any actions.

Joint Strategies

- A joint strategy σ_A for agents $A \subseteq Agents$ is a tuple of strategies, one per agent $a \in A$.
- The outcome of σ_A in location $q \in \mathcal{Q}$ is the set $out(q, \sigma_A) \subseteq \mathcal{Q}^\omega$ s.t. $\pi \in out(q, \sigma_A)$ iff $\pi(0) = q$ and for each $i \in \mathbb{N}$: $\pi(i) \xrightarrow{\text{act}'} \pi(i+1)$ for some $\text{act}' \in \Sigma$ s.t. $\text{act}'|_A = \sigma_A(\pi_i)$ and $\text{act}'|_{\overline{A}} \in protocol_{\overline{A}}(\pi(i))$.

Intuition: when coalition A follows σ_A, then in every global location, coalition A selects actions according to the joint strategy while the remaining agents $\overline{A}$ can choose any actions.

Joint Strategies

- A joint strategy σ_A for agents $A \subseteq \mathit{Agents}$ is a tuple of strategies, one per agent $a \in A$.

- The outcome of σ_A in location $q \in \mathcal{Q}$ is the set $\mathit{out}(q, \sigma_A) \subseteq \mathcal{Q}^\omega$ s.t. $\pi \in \mathit{out}(q, \sigma_A)$ iff $\pi(0) = q$ and for each $i \in \mathbb{N}$: $\pi(i) \xrightarrow{\mathrm{act}'} \pi(i+1)$ for some $\mathrm{act}' \in \Sigma$ s.t. $\mathrm{act}'|_A = \sigma_A(\pi_i)$ and $\mathrm{act}'|_{\overline{A}} \in \mathit{protocol}_{\overline{A}}(\pi(i))$.

Intuition: when coalition A follows σ_A, then in every global location, coalition A selects actions according to the joint strategy while the remaining agents $\overline{A}$ can choose any actions.

Joint Strategies

- A joint strategy σ_A for agents $A \subseteq Agents$ is a tuple of strategies, one per agent $a \in A$.
- The outcome of σ_A in location $q \in \mathcal{Q}$ is the set $out(q, \sigma_A) \subseteq \mathcal{Q}^\omega$ s.t. $\pi \in out(q, \sigma_A)$ iff $\pi(0) = q$ and for each $i \in \mathbb{N}$: $\pi(i) \xrightarrow{act'} \pi(i+1)$ for some $act' \in \Sigma$ s.t. $act'|_A = \sigma_A(\pi_i)$ and $act'|_{\overline{A}} \in protocol_{\overline{A}}(\pi(i))$.

Intuition: when coalition A follows σ_A, then in every global location, coalition A selects actions according to the joint strategy while the remaining agents $\overline{A}$ can choose any actions.

Semantics

We use **denotational semantics** that interprets Boolean and modal operators as either maximizers or minimizers

$\bigcup$, $\bigcap$ - the least upper bound, the greatest lower bound.

Σ_A - a set of joint strategies for A (variants: IR, iR, Ir, or ir)

$$[X\,\gamma]_{M,\pi} = [\gamma]_{M,\pi[1..\infty]};$$

.....

$$[\langle\langle A\rangle\rangle\gamma]_{M,q} = \bigcup_{\sigma_A\in\Sigma_A}\bigcap_{\pi\in out(q,\sigma_A)}\{[\gamma]_{M,\pi}\};$$

$$[\overline{\langle\langle A\rangle\rangle}\,\gamma]_{M,q} = \bigcap_{\sigma_A\in\Sigma_A}\bigcup_{\pi\in out(q,\sigma_A)}\{[\gamma]_{M,\pi}\};$$

$[\varphi_1 \preccurlyeq \varphi_2]_{M,q} = \top$ if $[\varphi_1]_{M,q} \leq [\varphi_2]_{M,q}$ and $\bot$ otherwise.

Semantics

We use **denotational semantics** that interprets Boolean and modal operators as either maximizers or minimizers

$\bigcup$, $\bigcap$ - the least upper bound, the greatest lower bound.

Σ_A - a set of joint strategies for A (variants: IR, iR, Ir, or ir)

$$[X\,\gamma]_{M,\pi} = [\gamma]_{M,\pi[1..\infty]};$$

.....

$$[\langle\langle A\rangle\rangle\gamma]_{M,q} = \bigcup_{\sigma_A\in\Sigma_A}\bigcap_{\pi\in out(q,\sigma_A)}\{[\gamma]_{M,\pi}\};$$

$$[\overline{\langle\langle A\rangle\rangle}\,\gamma]_{M,q} = \bigcap_{\sigma_A\in\Sigma_A}\bigcup_{\pi\in out(q,\sigma_A)}\{[\gamma]_{M,\pi}\};$$

$[\varphi_1 \preccurlyeq \varphi_2]_{M,q} = \top$ if $[\varphi_1]_{M,q} \leq [\varphi_2]_{M,q}$ and $\bot$ otherwise.

Semantics

We use **denotational semantics** that interprets Boolean and modal operators as either maximizers or minimizers
$\bigcup$, $\bigcap$ - the least upper bound, the greatest lower bound.
Σ_A - a set of joint strategies for A (variants: IR, iR, Ir, or ir)

$$[\mathrm{X}\,\gamma]_{M,\pi} = [\gamma]_{M,\pi[1..\infty]};$$

.....

$$[\langle\!\langle A\rangle\!\rangle\gamma]_{M,q} = \bigcup_{\sigma_A\in\Sigma_A}\bigcap_{\pi\in out(q,\sigma_A)}\{[\gamma]_{M,\pi}\};$$

$$[\overline{\langle\!\langle A\rangle\!\rangle}\,\gamma]_{M,q} = \bigcap_{\sigma_A\in\Sigma_A}\bigcup_{\pi\in out(q,\sigma_A)}\{[\gamma]_{M,\pi}\};$$

$[\varphi_1 \preccurlyeq \varphi_2]_{M,q} = \top$ if $[\varphi_1]_{M,q} \leq [\varphi_2]_{M,q}$ and $\bot$ otherwise.

Multi-Valued ATL* Extends 2-Valued ATL*

Theorem

The logic mv-$ATL^*_{\preccurlyeq}$ is a **conservative extension** *of ATL*, i.e.:*

for every 2-valued model M, ATL formula φ, and state (path) ι:*

$$[\varphi]_{M,\iota} = \top \quad \text{iff} \quad M, \iota \models_{ATL*} \varphi.$$
$$[\varphi]_{M,\iota} = \bot \quad \text{iff} \quad M, \iota \not\models_{ATL*} \varphi.$$

Translation to Simpler Lattices

Theorem

Let $f : L \to L'$ be a mapping that preserves bounds, i.e.,

$$f\Big(\bigcap_{i \in I} x_i\Big) = \bigcap_{i \in I} f(x_i), \qquad \textit{and} \qquad f\Big(\bigcup_{i \in I} x_i\Big) = \bigcup_{i \in I} f(x_i).$$

Then, for any mv-ATL formula φ and any state (resp. path) ι:*

$$[\varphi]_{\mathbf{f(M)},\iota} = \mathbf{x} \qquad \textit{iff} \qquad [\varphi]_{\mathbf{M},\iota} \in \mathbf{f}^{-1}(\mathbf{x})$$

Translation to 2-valued Lattices

Corollary

There exists a simple translation of checking whether $[\varphi]_{M,\iota} = x$ in mv-ATL* to several instances of 2-valued model checking of φ in ATL.*

$[\varphi]_{M,\iota} = \bigcup\{j \in \text{Join-irreducible(L)} \mid [\varphi]_{f_j(M),\iota} = \top\}$

$f_j(M)$ - the model M translated by $f_j : \mathrm{L} \longrightarrow \{\bot, \top\}$:

$f_j(\uparrow j) = \top, \quad f_j(\mathrm{L} \setminus \uparrow j) = \bot.$

Complexity of Multi-Valued ATL* Model Checking: Perfect Information

Theorem

Multi-valued verification of ATL* incurs only **polynomial increase** *in the complexity compared to the 2-valued case.*

Specifically, model checking mv-ATL$_{Ir\preccurlyeq}$ is **P***-complete, and model checking mv-ATL$^*_{Ir\preccurlyeq}$ is* **2EXPTIME***-complete in the size of the model and the formula, and the number of logical values.*

Imperfect Information

The method does not depend on the actual definition of strategy sets Σ_A!

Thus, we have:

Theorem

Model checking mv-ATL$_{ir\preccurlyeq}$ is Δ_2^P-complete, and model checking mv-ATL$^*_{ir\preccurlyeq}$ is **PSPACE***-complete in the size of the model and the formula, and the number of logical values.*

Imperfect Information

Theorem

Model checking $mv\text{-}ATL^*_{iR\preccurlyeq}$ *and* $mv\text{-}ATL_{iR\preccurlyeq}$ *is undecidable in general.*

For the fragment of $mv\text{-}ATL_{iR\preccurlyeq}$ *with singleton coalitions only, model checking is* **EXPTIME***-complete in the size of the model and the formula, and the number of logical values.*

Making model checking more efficient

- Abstraction - **multi-valued model checking** over smaller models,
- **Partial order reductions** - model checking over smaller models
- **Simpler strategies** - counting strategies for TATL

Making model checking more efficient

- Abstraction - **multi-valued model checking** over smaller models,
- **Partial order reductions** - model checking over smaller models
- **Simpler strategies** - counting strategies for TATL

Making model checking more efficient

- Abstraction - **multi-valued model checking** over smaller models,
- **Partial order reductions** - model checking over smaller models
- **Simpler strategies** - counting strategies for TATL

Making model checking more efficient

- Abstraction - **multi-valued model checking** over smaller models,
- **Partial order reductions** - model checking over smaller models
- **Simpler strategies** - counting strategies for TATL

What to do ?

Idea behind POR

POR is a method of generating reduced state spaces, preserving some temporal formula ψ, that exploits:

- Independency of actions, restricted to the pairs of actions such that one of them is invisible, i.e., does not change valuations of the atomic propositions used in ψ,
- Infinite sequences of global locations that differ in the ordering of independent actions only are called ψ-equivalent,
- ψ does not distinguish between ψ-equivalent sequences,

A reduced state space contains for each infinite sequence at least one ψ-equivalent, but as few as possible.

Networks of automata - asynchronous semantics

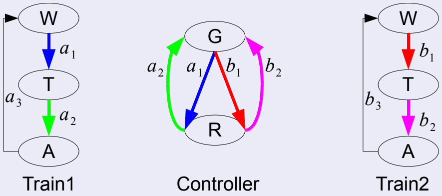


Figure: TC composed of two trains and the controler

Experimental Results - Trains and controler (TC)

Property: if the train 1 is in the tunnel, then no other train is in the tunnel at the same time: $AG(\text{in\_tunnel}_1 \rightarrow \bigwedge_{i=2}^{n} \neg\text{in\_tunnel}_i)$,

State spaces for n trains

$F(n)$ - the size of the full state space.
$R(n)$ - the size of the reduced state space.

- $F(n) = c_n \times 2^{n+1}$, for some $c_n > 1$,
- $R(n) = 2n + 1$.

The reduced state space is *exponentially smaller* than the original one, for both LTL-X and CTL-X.

Experimental Results - Trains and controler (TC)

Property: if the train 1 is in the tunnel, then no other train is in the tunnel at the same time: $AG(\text{in\_tunnel}_1 \rightarrow \bigwedge_{i=2}^{n} \neg\text{in\_tunnel}_i)$,

State spaces for n trains

$F(n)$ - the size of the full state space.
$R(n)$ - the size of the reduced state space.

- $F(n) = c_n \times 2^{n+1}$, for some $c_n > 1$,
- $R(n) = 2n + 1$.

The reduced state space is *exponentially smaller* than the original one, for both LTL-X and CTL-X.

Experimental Results - Trains and controler (TC)

Property: if the train 1 is in the tunnel, then no other train is in the tunnel at the same time: $AG(\text{in\_tunnel}_1 \rightarrow \bigwedge_{i=2}^{n} \neg\text{in\_tunnel}_i)$,

State spaces for n trains

$F(n)$ - the size of the full state space.
$R(n)$ - the size of the reduced state space.

- $F(n) = c_n \times 2^{n+1}$, for some $c_n > 1$,
- $R(n) = 2n + 1$.

The reduced state space is *exponentially smaller* than the original one, for both LTL-X and CTL-X.

Networks of automata

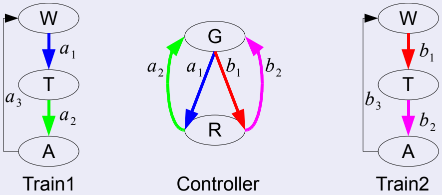


Figure: TC composed of two trains and the controler

Interleaved Interpreted Systems - asynchronous semantics

Assume we have n agents.

Definition

- $Act = A_1 \cup \ldots \cup A_n$ - a set of the actions,
- $\mathcal{Q} = L_1 \times \ldots \times L_n$ - a set of the global locations,
- $t_i : L_i \times A_i \rightarrow L_i$ for $i = 1, \ldots, n$ - an i-local evolution function,
- $Inttrans : \mathcal{Q} \times Act \rightarrow \mathcal{Q}$ - an interleaved evolution function:
  $Inttrans((q_1, \ldots, q_n), \mathrm{act}) = (q'_1, \ldots, q'_n)$ iff
  $t_i(q_i, \mathrm{act}) = q'_i$ if $\mathrm{act} \in A_i$ and $q_i = q'_i$ if $\mathrm{act} \notin A_i$,
- $q \sim_i q'$ iff $q_i = q'_i$ for $i = 1, \ldots, n$ - the indistinguishabilty relations.

sATL* over interleaved models

Restrictions of ATL*

- sATL* (simple ATL*) - ATL* without the next state operator and without nested strategic operators,
- $sATL_{ir}$, $sATL^*_{ir}$, $sATL_{Ir}$, $sATL^*_{Ir}$
- Model checking $sATL_{ir}$ and $sATL^*_{ir}$ is PSPACE-complete in the size of the model representation and the length of a formula.

Theorem

Partial order reductions preserving LTL-X preserve also $sATL^*_{ir}$.

Remark: the theorem does not hold for $sATL^*_{Ir}$.
Partial order reduction methods for LTL-X can be used for $sATL^*_{ir}$.

Making model checking more efficient

- Abstraction - **multi-valued model checking** over smaller models,
- **Partial order reductions** - model checking over smaller models,
- **Simpler strategies** - counting strategies for TATL

Making model checking more efficient

-
-
- **Simpler strategies** - counting strategies for TATL

What to do ?

Syntax of TATL

Timed Alternating-Time Temporal Logic (TATL)

The language of TATL is defined by the following grammar:

$$\phi ::= \mathsf{p} \mid \neg\phi \mid \phi \vee \phi \mid \langle\langle A\rangle\rangle X\phi \mid \langle\langle A\rangle\rangle \phi U_{\sim\eta}\phi \mid \langle\langle A\rangle\rangle \phi R_{\sim\eta}\phi,$$

where $\mathsf{p} \in \mathcal{AP}$, $A \subseteq Agents$, $\sim \in \{\leq, =, \geq\}$, and $\eta \in \mathbb{N}$.

Syntax of TATL

Timed Alternating-Time Temporal Logic (TATL)

The language of TATL is defined by the following grammar:

$$\phi ::= \mathrm{p} \mid \neg\phi \mid \phi \vee \phi \mid \langle\!\langle A \rangle\!\rangle X\phi \mid \langle\!\langle A \rangle\!\rangle \phi U_{\sim\eta}\phi \mid \langle\!\langle A \rangle\!\rangle \phi R_{\sim\eta}\phi,$$

where $\mathrm{p} \in \mathcal{AP}$, $A \subseteq \mathit{Agents}$, $\sim \in \{\leq, =, \geq\}$, and $\eta \in \mathbb{N}$.

Syntax of TATL

Timed Alternating-Time Temporal Logic (TATL)

The language of TATL is defined by the following grammar:

$$\phi ::= \mathrm{p} \mid \neg\phi \mid \phi \vee \phi \mid \langle\langle A\rangle\rangle X\phi \mid \langle\langle A\rangle\rangle \phi U_{\sim\eta}\phi \mid \langle\langle A\rangle\rangle \phi R_{\sim\eta}\phi,$$

where $\mathrm{p} \in \mathcal{AP}$, $A \subseteq \mathit{Agents}$, $\sim \in \{\leq, =, \geq\}$, and $\eta \in \mathbb{N}$.

TATL, cont'd

$\text{TATL}_{\leq,\geq}$: a subset of TATL with only $\leq, \geq$ allowed,
e.g., $\langle\langle A\rangle\rangle G_{\geq 42}\text{safe} \in \text{TATL}_{\leq,\geq}$, $\langle\langle A\rangle\rangle F_{=13}\text{finish} \notin \text{TATL}_{\leq,\geq}$.

Examples of properties:

- $\langle\langle A\rangle\rangle G_{\geq 42}\text{safe}$: "Coalition *A* **has a strategy to enforce** that safe holds always after reaching 42 time units".
- $\langle\langle A\rangle\rangle F_{=13}\text{finish}$: "Coalition *A* **has a strategy to enforce** that finish is reached in exactly 13 time units".

Counting Strategies: perfect information

Counting strategies ($\Sigma_{\#}$)

Strategies $\sigma_a \in \Sigma_T$ s.t. for each $\pi, \pi' \in \mathcal{S}^+$, if $loc(\pi_F) = loc(\pi'_F)$ and $\#_F(\pi) = \#_F(\pi')$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

(Intuition: action selection depends on the number of visits to the location of π_F)

Alternative notation

A counting strategy is a function $\sigma_a^{\#} : \mathcal{Q} \times \mathbb{N} \to \Sigma$ s.t. $\sigma_a^{\#}(q, k) := \sigma_a(\pi)$ if $q = loc(\pi_F)$ and $k = \#_F(\pi)$.

$\#_F(\pi)$: the number of states of π whose location is $loc(\pi_F)$.

Counting Strategies: perfect information

Threshold strategies ($\Sigma_{\#_n}$)

A counting strategy $\sigma_a^\# \in \Sigma_\#$ is called *n*–**threshold** for some $n \in \mathbb{N}_+$ iff for each location $q \in \mathcal{Q}$ there exist:

- actions $\text{act}_1, \ldots, \text{act}_{n+1} \in \Sigma$, and
- integer intervals $I_1 = [1, i_1), I_2 = [i_1, i_2), \ldots, I_{n+1} = [i_n, \infty)$

s.t. for all $1 \leq j \leq n+1$: $\sigma_a^\#(q, k) = \text{act}_j$ if $k \in I_j$.

Example: a counting strategy is 2–threshold if for any location $q \in \mathcal{Q}$ there are **three** actions $\text{act}_1, \text{act}_2, \text{act}_3$ s.t. first only act_1 is used when q is visited, then only act_2, and finally only act_3, ad infinitum.

Counting Strategies: perfect information

Threshold strategies ($\Sigma_{\#_n}$)

A counting strategy $\sigma_a^\# \in \Sigma_\#$ is called *n*–**threshold** for some $n \in \mathbb{N}_+$ iff for each location $q \in \mathcal{Q}$ there exist:

- actions $\text{act}_1, \ldots, \text{act}_{n+1} \in \Sigma$, and
- integer intervals $I_1 = [1, i_1), I_2 = [i_1, i_2), \ldots, I_{n+1} = [i_n, \infty)$

s.t. for all $1 \leq j \leq n+1$: $\sigma_a^\#(q, k) = \text{act}_j$ if $k \in I_j$.

Example: a counting strategy is 2–threshold if for any location $q \in \mathcal{Q}$ there are **three** actions $\text{act}_1, \text{act}_2, \text{act}_3$ s.t. first only act_1 is used when q is visited, then only act_2, and finally only act_3, ad infinitum.

Counting Strategies: perfect information

Threshold strategies ($\Sigma_{\#_n}$)

A counting strategy $\sigma_a^\# \in \Sigma_\#$ is called *n*–**threshold** for some $n \in \mathbb{N}_+$ iff for each location $q \in \mathcal{Q}$ there exist:

- actions $\text{act}_1, \ldots, \text{act}_{n+1} \in \Sigma$, and
- integer intervals $I_1 = [1, i_1), I_2 = [i_1, i_2), \ldots, I_{n+1} = [i_n, \infty)$

s.t. for all $1 \leq j \leq n+1$: $\sigma_a^\#(q, k) = \text{act}_j$ if $k \in I_j$.

Example: a counting strategy is 2–threshold if for any location $q \in \mathcal{Q}$ there are **three** actions $\text{act}_1, \text{act}_2, \text{act}_3$ s.t. first only act_1 is used when q is visited, then only act_2, and finally only act_3, ad infinitum.

Counting Strategies: perfect information

Threshold strategies ($\Sigma_{\#_n}$)

A counting strategy $\sigma_a^\# \in \Sigma_\#$ is called *n*–**threshold** for some $n \in \mathbb{N}_+$ iff for each location $q \in \mathcal{Q}$ there exist:

- actions $\mathrm{act}_1, \ldots, \mathrm{act}_{n+1} \in \Sigma$, and
- integer intervals $I_1 = [1, i_1), I_2 = [i_1, i_2), \ldots, I_{n+1} = [i_n, \infty)$

s.t. for all $1 \leq j \leq n+1$: $\sigma_a^\#(q, k) = \mathrm{act}_j$ if $k \in I_j$.

Example: a counting strategy is 2–threshold if for any location $q \in \mathcal{Q}$ there are **three** actions $\mathrm{act}_1, \mathrm{act}_2, \mathrm{act}_3$ s.t. first only act_1 is used when q is visited, then only act_2, and finally only act_3, ad infinitum.

Counting Strategies: perfect information

Threshold strategies ($\Sigma_{\#_n}$)

A counting strategy $\sigma_a^\# \in \Sigma_\#$ is called *n*–**threshold** for some $n \in \mathbb{N}_+$ iff for each location $q \in \mathcal{Q}$ there exist:

- actions $\text{act}_1, \ldots, \text{act}_{n+1} \in \Sigma$, and
- integer intervals $I_1 = [1, i_1), I_2 = [i_1, i_2), \ldots, I_{n+1} = [i_n, \infty)$

s.t. for all $1 \leq j \leq n+1$: $\sigma_a^\#(q, k) = \text{act}_j$ if $k \in I_j$.

Example: a counting strategy is 2–threshold if for any location $q \in \mathcal{Q}$ there are **three** actions $\text{act}_1, \text{act}_2, \text{act}_3$ s.t. first only act_1 is used when q is visited, then only act_2, and finally only act_3, ad infinitum.

Threshold

Theorem. Threshold for $\mathrm{TATL}_{\leq,\geq}$ is 2

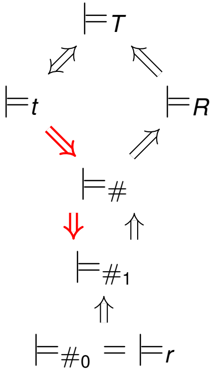
For each $q \in \mathcal{Q}$ and $\phi \in \mathrm{TATL}_{\leq,\geq}$, if $q \models_{I,T} \phi$, then $q \models_{\#_1} \phi$.

This may help to alleviate the explosion of strategies.

Theorem

There is no threshold for TATL.

Hierarchy of satisfaction relations (for I)



The Red implications hold only for $\text{TATL}_{\leq,\geq}$.

Conclusions

Alleviating state/transition/strategy explosions:

- Model checking for ATL^*_{Ir}, ATL^*_{ir}, and $TATL_{\leq,\geq}$ is difficult, but:
- In practical applications one can successfully use:

  Multi-valued model checking over abstract models,
  Partial order reduction methods,
  Counting strategies rather than timed ones.

Lecture based on the papers:

- Partial Order Reductions for Model Checking Temporal-epistemic Logics over Interleaved Multi-agent Systems [A. Lomuscio, W. Penczek, H. Qu: Fundamenta Informaticae, 2010]
- Specification and Verification of Multi-Agent Systems [W. Jamroga, W. Penczek: ESSLLI, 2011]
- Multi-Valued Verification of Strategic Ability [W. Jamroga, B. Konikowska, W. Penczek: AAMAS, 2016]
- Timed ATL: Forget Memory, Just Count [E. Andre, L. Petrucci, W. Jamroga, M. Knapik, W.Penczek, AAMAS, 2017]
- Towards Partial Order Reductions for Fragments of Alternating-Time Temporal Logic [P. Dembiński, W. Jamroga, A. Mazurkiewicz, W. Penczek, ICS PAS Report 1036, 2017]

Thank you!