Timed ATL:

Forget Memory, Just Count

Wojciech Penczek

a joint work with
E. Andre, W. Jamroga, M. Knapik, and L. Petrucci

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

MeFoSyLoMa, LIPN, Paris, the 2nd of June 2017

Outline

Main Contributions

- New counting strategies for Timed ATL (TATL)
- Hierarchy of semantics for different strategies of Timed ATL
- Counting strategies avoiding tracking the passage of time have the same expressivity as timed strategies

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 (T)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 (T)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 (T)ATL or Strategy Logic
- ...and verify them by model checking

Motivation: Two Projects

- Project PAS - CNRS on Parametric Verification,
- Project PAS - University of Luxembourg on Verification of Voter-Verifiable Voting Protocols VoteVerif,
- 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: Two Projects

- Project PAS - CNRS on Parametric Verification,
- Project PAS - University of Luxembourg on Verification of Voter-Verifiable Voting Protocols VoteVerif,
- 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,

Related Work

Previous

- 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]

Current

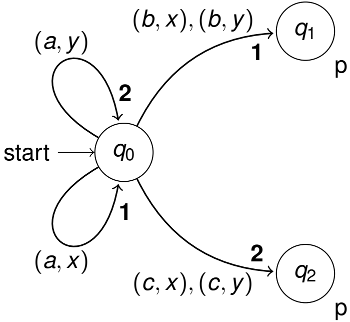
- Timed ATL: Forget Memory, Just Count [Andre, Petrucci, Jamroga, Knapik, Penczek, AAMAS 2017]

Models

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

- *Agents* is a finite set of all the agents,
- Σ is a finite set of actions,
- $\mathcal{Q}$ is a finite set of locations,
- $\mathcal{AP}$ is a set of atomic propositions,
- $\mathcal{L}\colon \mathcal{Q} \to \mathcal{P}(\mathcal{AP})$ is a location labeling function,
- $\mathit{protocol}\colon \mathit{Agents} \times \mathcal{Q} \to \mathcal{P}(\Sigma) \setminus \{\emptyset\}$ is a protocol function,
- $\mathit{trans}\colon \mathcal{Q} \times \Sigma^{|\mathit{Agents}|} \to \mathcal{Q} \times \mathbb{N}_{+}$ is a transition function.

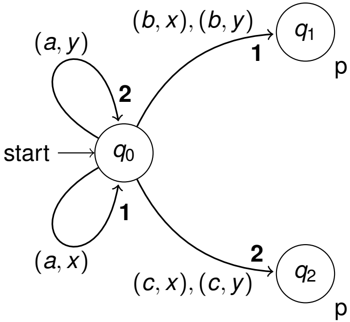
Models, cont'd



Runs are modeled in the space (locations x time) $S := Q \times \mathbb{N}$:

$$(q_0, 0) \xrightarrow{(a,y)} (q_0, 2) \xrightarrow{(a,x)} (q_0, 3) \xrightarrow{(c,y)} (q_2, 5)$$

Models, cont'd



Runs are modeled in the space (locations x time) $\mathcal{S} := \mathcal{Q} \times \mathbb{N}$:

$$(q_0, 0) \xrightarrow{(a,y)} (q_0, 2) \xrightarrow{(a,x)} (q_0, 3) \xrightarrow{(c,y)} (q_2, 5)$$

Notations

Let $q \in \mathcal{Q}$, $s \in \mathcal{S}$ and $\pi \in \mathcal{S}^+ \cup \mathcal{S}^\omega$.

- $loc(s)$ and $time(s)$: the location and time of s, resp.,
- $\pi(i)$: the i–th state of π,
- π_i: the prefix of π of length i,
- π^i: the suffix of π starting from $\pi(i)$,

Notations

- if π is finite:
  - π_F: its final state,
  - $\#_F(\pi)$: the number of states of π whose location is $loc(\pi_F)$.

Example. Count how many times the final location appears along π, e.g.:

$$\begin{aligned}\pi &= \big((q_0, 0), (q_0, 2)\big),\\ \pi' &= \big((q_0, 0), (q_0, 2), (q_0, 3)\big),\\ \pi'' &= \big((q_0, 0), (q_0, 2), (q_0, 3), (q_2, 5)\big),\end{aligned}$$

$\#_F(\pi) = 2$, $\#_F(\pi') = 3$, $\#_F(\pi'') = 1$.

Standard Strategies, cont'd

Let $a \in Agents$:

Timed perfect recall strategies (Σ_T)

Functions $\sigma_a \colon \mathcal{S}^+ \to \Sigma$ s.t., $\forall_{\pi \in \mathcal{S}^+} \sigma_a(\pi) \in protocol_a(loc(\pi_F))$.

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

Timed memoryless strategies (Σ_t)

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

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

Standard Strategies, cont'd

Let $a \in \mathit{Agents}$:

Timed perfect recall strategies (Σ_T)

Functions $\sigma_a \colon \mathcal{S}^+ \rightarrow \Sigma$ s.t., $\forall_{\pi \in \mathcal{S}^+} \sigma_a(\pi) \in \mathit{protocol}_a(\mathit{loc}(\pi_F))$.

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

Timed memoryless strategies (Σ_t)

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

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

Standard Strategies, cont'd

Timeless perfect recall strategies (Σ_R)

Strategies $\sigma_a \in \Sigma_T$ s.t., for each $n \in \mathbb{N}$ and $\pi, \pi' \in \mathcal{S}^n$, if $loc(\pi(i)) = loc(\pi'(i))$ for all $0 \leq i \leq n$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

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

Timeless memoryless recall strategies (Σ_r)

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

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

Standard Strategies, cont'd

Timeless perfect recall strategies (Σ_R)

Strategies $\sigma_a \in \Sigma_T$ s.t., for each $n \in \mathbb{N}$ and $\pi, \pi' \in \mathcal{S}^n$, if $loc(\pi(i)) = loc(\pi'(i))$ for all $0 \leq i \leq n$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

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

Timeless memoryless recall strategies (Σ_r)

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

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

Standard Strategies, cont'd

Timeless perfect recall strategies (Σ_R)

Strategies $\sigma_a \in \Sigma_T$ s.t., for each $n \in \mathbb{N}$ and $\pi, \pi' \in \mathcal{S}^n$, if $loc(\pi(i)) = loc(\pi'(i))$ for all $0 \leq i \leq n$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

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

Timeless memoryless recall strategies (Σ_r)

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

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

Standard Strategies, cont'd

Timeless perfect recall strategies (Σ_R)

Strategies $\sigma_a \in \Sigma_T$ s.t., for each $n \in \mathbb{N}$ and $\pi, \pi' \in \mathcal{S}^n$, if $loc(\pi(i)) = loc(\pi'(i))$ for all $0 \leq i \leq n$, then $\sigma_a(\pi) = \sigma_a(\pi')$.

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

Timeless memoryless recall strategies (Σ_r)

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

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

New Strategies

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

New Strategies, cont'd

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.

New Strategies, cont'd

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.

New Strategies, cont'd

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.

New Strategies, cont'd

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.

New Strategies, cont'd

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.

Joint Strategies

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

  Notation: if $A = \{a_1, \ldots, a_k\}$ for some $k \in \mathbb{N}$ and $\sigma_A = (\sigma_{a_1}, \ldots, \sigma_{a_k})$ is a joint strategy for A, then for each $i \in \mathbb{N}$ and $\pi \in \mathcal{S}^\omega$ denote $\sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i))$.

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

Intuition: when coalition A follows σ_A, then in every state it selects actions according to the joint strategy while the remaining agents 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 $a \in A$.

  Notation: if $A = \{a_1, \ldots, a_k\}$ for some $k \in \mathbb{N}$ and $\sigma_A = (\sigma_{a_1}, \ldots, \sigma_{a_k})$ is a joint strategy for A, then for each $i \in \mathbb{N}$ and $\pi \in \mathcal{S}^\omega$ denote $\sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i))$.

- The outcome of σ_A in state $s \in \mathcal{S}$ is the set $out(s, \sigma_A) \subseteq \mathcal{S}^\omega$ s.t. $\pi \in out(s, \sigma_A)$ iff $\pi(0) = s$ and for each $i \in \mathbb{N}$ $\pi(i) \stackrel{(\sigma_A(\pi_i), \mathrm{act}')}{\longrightarrow} \pi(i+1)$ for some $\mathrm{act}' \in \mathit{protocol}_{\overline{A}}(\mathit{loc}(\pi(i)))$.

Intuition: when coalition A follows σ_A, then in every state it selects actions according to the joint strategy while the remaining agents 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 $a \in A$.

  Notation: if $A = \{a_1, \ldots, a_k\}$ for some $k \in \mathbb{N}$ and $\sigma_A = (\sigma_{a_1}, \ldots, \sigma_{a_k})$ is a joint strategy for A, then for each $i \in \mathbb{N}$ and $\pi \in \mathcal{S}^\omega$ denote $\sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i))$.

- The outcome of σ_A in state $s \in \mathcal{S}$ is the set $out(s, \sigma_A) \subseteq \mathcal{S}^\omega$ s.t. $\pi \in out(s, \sigma_A)$ iff $\pi(0) = s$ and for each $i \in \mathbb{N}$ $\pi(i) \xrightarrow{(\sigma_A(\pi_i), act')} \pi(i+1)$ for some $act' \in protocol_{\overline{A}}(loc(\pi(i)))$.

Intuition: when coalition A follows σ_A, then in every state it selects actions according to the joint strategy while the remaining agents can choose any actions.

Joint Strategies

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

  Notation: if $A = \{a_1, \ldots, a_k\}$ for some $k \in \mathbb{N}$ and $\sigma_A = (\sigma_{a_1}, \ldots, \sigma_{a_k})$ is a joint strategy for A, then for each $i \in \mathbb{N}$ and $\pi \in \mathcal{S}^\omega$ denote $\sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i))$.

- The outcome of σ_A in state $s \in \mathcal{S}$ is the set $out(s, \sigma_A) \subseteq \mathcal{S}^\omega$ s.t. $\pi \in out(s, \sigma_A)$ iff $\pi(0) = s$ and for each $i \in \mathbb{N}$ $\pi(i) \overset{(\sigma_A(\pi_i), \text{act}')}{\longrightarrow} \pi(i+1)$ for some $\text{act}' \in protocol_{\overline{A}}(loc(\pi(i)))$.

Intuition: when coalition A follows σ_A, then in every state it selects actions according to the joint strategy while the remaining agents can choose any actions.

Joint Strategies

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

  Notation: if $A = \{a_1, \ldots, a_k\}$ for some $k \in \mathbb{N}$ and $\sigma_A = (\sigma_{a_1}, \ldots, \sigma_{a_k})$ is a joint strategy for A, then for each $i \in \mathbb{N}$ and $\pi \in \mathcal{S}^\omega$ denote $\sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i))$.

- The outcome of σ_A in state $s \in \mathcal{S}$ is the set $out(s, \sigma_A) \subseteq \mathcal{S}^\omega$ s.t. $\pi \in out(s, \sigma_A)$ iff $\pi(0) = s$ and for each $i \in \mathbb{N}$ $\pi(i) \stackrel{(\sigma_A(\pi_i), \mathrm{act}')}{\longrightarrow} \pi(i+1)$ for some $\mathrm{act}' \in protocol_{\overline{A}}(loc(\pi(i)))$.

Intuition: when coalition A follows σ_A, then in every state it selects actions according to the joint strategy while the remaining agents can choose any actions.

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 $\eta \in \mathbb{N}$.

We interpret $\langle\!\langle A \rangle\!\rangle\psi$ as "the coalition A has a strategy to enforce ψ", X stands for "in the next state", U for "until", and R for "release".

Derived modalities: F ("in the future") and G ("globally"): $\langle\!\langle A \rangle\!\rangle F_{\sim\eta}\phi := \langle\!\langle A \rangle\!\rangle \top U_{\sim\eta}\phi$, $\langle\!\langle A \rangle\!\rangle G_{\sim\eta}\phi := \langle\!\langle A \rangle\!\rangle \bot R_{\sim\eta}\phi$.

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 $\eta \in \mathbb{N}$.

We interpret $\langle\langle A \rangle\rangle\psi$ as "the coalition A has a strategy to enforce ψ", X stands for "in the next state", U for "until", and R for "release".

Derived modalities: F ("in the future") and G ("globally"): $\langle\langle A \rangle\rangle F_{\sim\eta}\phi := \langle\langle A \rangle\rangle \top U_{\sim\eta}\phi$, $\langle\langle A \rangle\rangle G_{\sim\eta}\phi := \langle\langle A \rangle\rangle \bot R_{\sim\eta}\phi$.

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 $\eta \in \mathbb{N}$.

We interpret $\langle\!\langle A \rangle\!\rangle\psi$ as "the coalition A has a strategy to enforce ψ", X stands for "in the next state", U for "until", and R for "release".

Derived modalities: F ("in the future") and G ("globally"): $\langle\!\langle A \rangle\!\rangle F_{\sim\eta}\phi := \langle\!\langle A \rangle\!\rangle \top U_{\sim\eta}\phi$, $\langle\!\langle A \rangle\!\rangle G_{\sim\eta}\phi := \langle\!\langle A \rangle\!\rangle \bot R_{\sim\eta}\phi$.

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 $\eta \in \mathbb{N}$.

We interpret $\langle\langle A \rangle\rangle\psi$ as "the coalition A has a strategy to enforce ψ", X stands for "in the next state", U for "until", and R for "release".

Derived modalities: F ("in the future") and G ("globally"): $\langle\langle A \rangle\rangle F_{\sim\eta}\phi := \langle\langle A \rangle\rangle \top U_{\sim\eta}\phi$, $\langle\langle A \rangle\rangle G_{\sim\eta}\phi := \langle\langle A \rangle\rangle \bot R_{\sim\eta}\phi$.

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 $\eta \in \mathbb{N}$.

We interpret $\langle\langle A \rangle\rangle \psi$ as "the coalition A has a strategy to enforce ψ", X stands for "in the next state", U for "until", and R for "release".

Derived modalities: F ("in the future") and G ("globally"): $\langle\langle A \rangle\rangle F_{\sim\eta}\phi := \langle\langle A \rangle\rangle \top U_{\sim\eta}\phi$, $\langle\langle A \rangle\rangle G_{\sim\eta}\phi := \langle\langle A \rangle\rangle \bot R_{\sim\eta}\phi$.

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 $\eta \in \mathbb{N}$.

We interpret $\langle\!\langle A \rangle\!\rangle\psi$ as "the coalition A has a strategy to enforce ψ", X stands for "in the next state", U for "until", and R for "release".

Derived modalities: F ("in the future") and G ("globally"): $\langle\!\langle A \rangle\!\rangle F_{\sim\eta}\phi := \langle\!\langle A \rangle\!\rangle \top U_{\sim\eta}\phi$, $\langle\!\langle A \rangle\!\rangle G_{\sim\eta}\phi := \langle\!\langle A \rangle\!\rangle \bot R_{\sim\eta}\phi$.

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 $\eta \in \mathbb{N}$.

We interpret $\langle\langle A\rangle\rangle\psi$ as "the coalition A has a strategy to enforce ψ", X stands for "in the next state", U for "until", and R for "release".

Derived modalities: F ("in the future") and G ("globally"): $\langle\langle A\rangle\rangle F_{\sim\eta}\phi := \langle\langle A\rangle\rangle \top U_{\sim\eta}\phi$, $\langle\langle A\rangle\rangle G_{\sim\eta}\phi := \langle\langle A\rangle\rangle \bot R_{\sim\eta}\phi$.

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 $\eta \in \mathbb{N}$.

We interpret $\langle\!\langle A \rangle\!\rangle\psi$ as "the coalition A has a strategy to enforce ψ", X stands for "in the next state", U for "until", and R for "release".

Derived modalities: F ("in the future") and G ("globally"):
$\langle\!\langle A \rangle\!\rangle F_{\sim\eta}\phi := \langle\!\langle A \rangle\!\rangle \top U_{\sim\eta}\phi$, $\langle\!\langle A \rangle\!\rangle G_{\sim\eta}\phi := \langle\!\langle A \rangle\!\rangle \bot R_{\sim\eta}\phi$.

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 $\eta \in \mathbb{N}$.

We interpret $\langle\langle A \rangle\rangle\psi$ as "the coalition A has a strategy to enforce ψ", X stands for "in the next state", U for "until", and R for "release".

Derived modalities: F ("in the future") and G ("globally"):
$\langle\langle A \rangle\rangle F_{\sim\eta}\phi := \langle\langle A \rangle\rangle \top U_{\sim\eta}\phi$, $\langle\langle A \rangle\rangle G_{\sim\eta}\phi := \langle\langle A \rangle\rangle \bot R_{\sim\eta}\phi$.

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".

TATL, cont'd

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

Examples of properties:

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

Semantics of TATL

For each type of strategy define the corresponding satisfaction relation, i.e., $\models_Y$ corresponds to Σ_Y, for $Y \in \{T, t, R, r, \#, \#_n\}$.

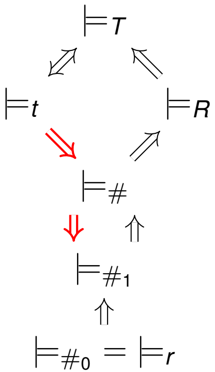
Satisfaction relation

$M, q \models_Y \langle\langle A \rangle\rangle \psi$ iff there exists a strategy $\sigma_A \in \Sigma_Y$ for A s.t. ψ holds **along each outcome** $\pi \in out((q, 0), \sigma_A)$.

Satisfaction relation over outcomes

- $\pi \models X\phi$ iff $loc(\pi(1)) \models \phi$,
- $\pi \models \phi U_{\sim\eta}\psi$ iff $loc(\pi(i)) \models \psi$ for some i s.t. $time(\pi(i)) \sim \eta$ and $loc(\pi(j)) \models \phi$ for all $j < i$,
- $\pi \models \phi R_{\sim\eta}\psi$ iff for all i: $time(\pi(i)) \sim \eta \Longrightarrow loc(\pi(i)) \models \psi$ or $loc(\pi(j)) \models \phi$ for some $j < i$,

Hierarchy of satisfaction relations



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

Key implications: timed strategies and memory

Theorem (1) Timed strategies do not need memory

For each $q \in \mathcal{Q}$ and $\phi \in$ TATL we have $q \models_T \phi$ iff $q \models_t \phi$.
(so we omit subscript in this case and write $\models$)

Lemma. Time limit

Let $\langle\langle A\rangle\rangle\psi \in$ TATL and $c \in \mathbb{N}$ be the greatest integer in ψ.
If $\sigma_A \in \Sigma_T$ implements $\langle\langle A\rangle\rangle\psi$, then there exists its reduction σ'_A s.t. $\forall_{q\in\mathcal{Q}}\forall_{t\geq c}\ \sigma'_A(q,t) = \sigma'_A(q,c+1)$, which also implements $\langle\langle A\rangle\rangle\psi$.
Intuitively, there is no need to track time after it exceeds c.

Key implications: timed strategies and memory

Theorem (1) Timed strategies do not need memory

For each $q \in \mathcal{Q}$ and $\phi \in$ TATL we have $q \models_T \phi$ iff $q \models_t \phi$.
(so we omit subscript in this case and write $\models$)

Lemma. Time limit

Let $\langle\!\langle A \rangle\!\rangle \psi \in$ TATL and $c \in \mathbb{N}$ be the greatest integer in ψ.
If $\sigma_A \in \Sigma_T$ implements $\langle\!\langle A \rangle\!\rangle \psi$, then there exists its reduction σ'_A s.t. $\forall_{q \in \mathcal{Q}} \forall_{t \geq c}\ \sigma'_A(q, t) = \sigma'_A(q, c+1)$, which also implements $\langle\!\langle A \rangle\!\rangle \psi$.
Intuitively, there is no need to track time after it exceeds c.

Key implications: timed strategies and memory

Theorem (1) Timed strategies do not need memory

For each $q \in \mathcal{Q}$ and $\phi \in$ TATL we have $q \models_T \phi$ iff $q \models_t \phi$.
(so we omit subscript in this case and write $\models$)

Lemma. Time limit

Let $\langle\langle A \rangle\rangle\psi \in$ TATL and $c \in \mathbb{N}$ be the greatest integer in ψ.
If $\sigma_A \in \Sigma_T$ implements $\langle\langle A \rangle\rangle\psi$, then there exists its reduction σ'_A s.t. $\forall_{q \in \mathcal{Q}} \forall_{t \geq c}\ \sigma'_A(q, t) = \sigma'_A(q, c+1)$, which also implements $\langle\langle A \rangle\rangle\psi$.
Intuitively, there is no need to track time after it exceeds c.

Key implications: time versus order

Theorem (2) $\models_{\#} \implies \models$

For each $q \in Q$ and $\phi \in$ TATL, if $q \models_{\#} \phi$, then $q \models \phi$.

Key implications: time versus order, cont'd

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

Theorem (3) $\models \implies \models_{\#}$

For each $q \in \mathcal{Q}$ and $\phi \in \text{TATL}_{\leq,\geq}$, if $q \models \phi$, then $q \models_{\#} \phi$.
(Just count locations, do not look at clock.)

(3) cannot be extended to TATL, see next slide.

Key implications: time versus order, cont'd

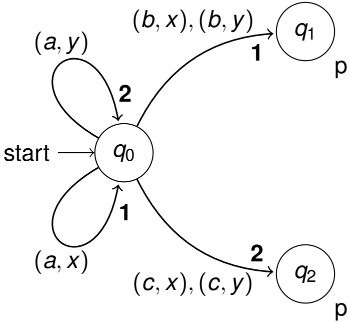
TATL$_{\leq,\geq}$: a subset of TATL with only $\leq, \geq$ allowed,
e.g., $\langle\langle A \rangle\rangle G_{\geq 42}$safe $\in$ TATL$_{\leq,\geq}$, $\langle\langle A \rangle\rangle F_{=13}$finish $\notin$ TATL$_{\leq,\geq}$.

Theorem (3) $\models \implies \models_{\#}$

For each $q \in \mathcal{Q}$ and $\phi \in$ TATL$_{\leq,\geq}$, if $q \models \phi$, then $q \models_{\#} \phi$.
(Just count locations, do not look at clock.)

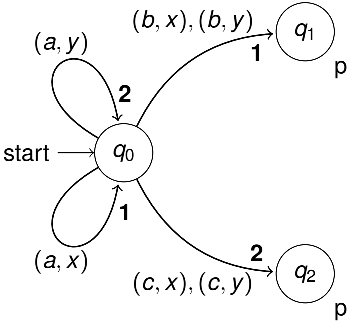
(3) cannot be extended to TATL, see next slide.

Key implications: time versus order, cot'd



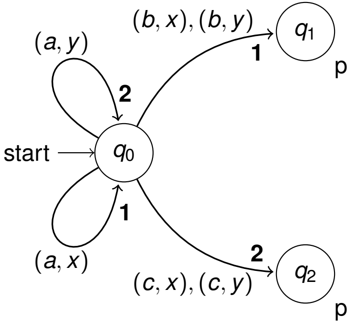
$q_0 \models_T \langle\langle 1 \rangle\rangle F_{=5}$p, but $q_0 \not\models_\# \langle\langle 1 \rangle\rangle F_{=5}$p, as there is no counting strategy that allows to decide when to leave q_0 for a location labeled with p and which branch to take in order to reach the target in 5 time units.

Key implications: time versus order, cot'd



$q_0 \models_T \langle\langle 1 \rangle\rangle F_{=5}$p, but $q_0 \not\models_\# \langle\langle 1 \rangle\rangle F_{=5}$p, as there is no counting strategy that allows to decide when to leave q_0 for a location labeled with p and which branch to take in order to reach the target in 5 time units.

Key implications: time versus order, cot'd



$q_0 \models_T \langle\langle 1 \rangle\rangle F_{=5}\mathrm{p}$, but $q_0 \not\models_\# \langle\langle 1 \rangle\rangle F_{=5}\mathrm{p}$, as there is no counting strategy that allows to decide when to leave q_0 for a location labeled with p and which branch to take in order to reach the target in 5 time units.

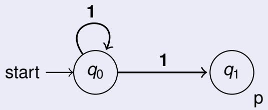
Key implications: counting up to...

Theorem (4) Threshold for $TATL_{\leq,\geq}$ is 2

For each $q \in Q$ and $\phi \in TATL_{\leq,\geq}$, if $q \models_{\#} \phi$, then $q \models_{\#_1} \phi$.

All modalities except for $U_{\geq\eta}$ need only one action, and $U_{\geq\eta}$ needs two.

...and cannot be lowered

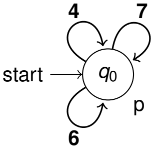


- $q_0 \models_{\#_1} \langle\langle 1 \rangle\rangle F_{\geq 5}$p: loops four times and jumps ahead
- $q_0 \not\models_{\#_0} \langle\langle 1 \rangle\rangle F_{\geq 5}$p: loops forever, or jumps too early

Key implications: counting up to. . . , ct'd

Theorem (5)

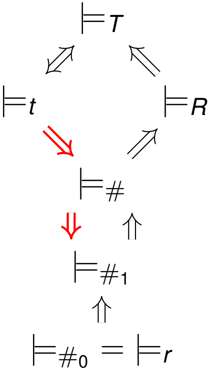
There is no threshold for TATL.



$\langle\langle 1 \rangle\rangle F_{=17}$p: **three distinct actions** needed to sum up to exactly 17 time units.

This can be extended to an arbitrary number (n) of actions using L. Mikulski's sequence: $(10)^n + 1, \ldots, (10)^n + 2^i, \ldots, (10)^n + 2^n)$ for the times of the actions.

Hierarchy of satisfaction relations



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

Conclusions and Future Work

Conclusions

- Hierarchy of strategies for TATL,
- Unexpectedly, $\models$ is equivalent to $\models_{\#}$ for $\text{TATL}_{\leq,\geq}$,
- Threshold for $\text{TATL}_{\leq,\geq}$ is 2.

Future Work

- Extensions to TATL^* and parametric versions,
- Incomplete knowledge semantics,
- Model checking algorithms.

Thank you!