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

1. Introduction and Related Work
2. Standard and New Strategies
3. Timed ATL - syntax and semantics
4. Comparing Satisfaction Relations based on Strategies
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
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 (T)ATL or Strategy Logic

...and verify them by model checking
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.
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 (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]
A **Tight Durational Concurrent Game Structure** is a 7–tuple $\mathcal{A} = (\text{Agents}, \Sigma, Q, \mathcal{AP}, L, \text{protocol}, \text{trans})$, where:

- **Agents** is a finite set of all the agents,
- $\Sigma$ is a finite set of actions,
- $Q$ is a finite set of locations,
- $\mathcal{AP}$ is a set of atomic propositions,
- $L : Q \rightarrow \mathcal{P}(\mathcal{AP})$ is a location labeling function,
- **protocol**: $\text{Agents} \times Q \rightarrow \mathcal{P}(\Sigma) \setminus \{\emptyset\}$ is a protocol function,
- **trans**: $Q \times \Sigma^{\lvert \text{Agents} \rvert} \rightarrow Q \times \mathbb{N}_+$ is a transition function.
Runs are modeled in the space (locations x time) $S := Q \times \mathbb{N}$:

$$(q_0, 0) \xrightarrow{(a,y)} (q_0, 2) \xrightarrow{(a,x)} (q_0, 3) \xrightarrow{(c,y)} (q_2, 5)$$
Runs are modeled in the space \( S := Q \times \mathbb{N} \):

\[
(q_0, 0) \xrightarrow{(a,y)} (q_0, 2) \xrightarrow{(a,x)} (q_0, 3) \xrightarrow{(c,y)} (q_2, 5)
\]
Let \( q \in Q, s \in S \) and \( \pi \in S^+ \cup S^\omega \).

- \( \text{loc}(s) \) and \( \text{time}(s) \): the location and time of \( s \), resp.,
- \( \pi(i) \): the \( i \)-th state of \( \pi \),
- \( \pi_i \): the prefix of \( \pi \) of length \( i \),
- \( \pi^i \): the suffix of \( \pi \) starting from \( \pi(i) \),
Notations

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

Example. Count how many times the final location appears along $\pi$, e.g.:

$$
\pi = ((q_0, 0), (q_0, 2)),
\pi' = ((q_0, 0), (q_0, 2), (q_0, 3)),
\pi'' = ((q_0, 0), (q_0, 2), (q_0, 3), (q_2, 5)),
$$

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

Wojciech Penczek and al.  Timed ATL: Forget memory, Just count 10/28
Let $a \in \text{Agents}$:

**Timed perfect recall strategies ($\Sigma_T$)**

Functions $\sigma_a : S^+ \to \Sigma$ s.t., $\forall \pi \in S^+ \sigma_a(\pi) \in protocol_a(loc(\pi_F))$.

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

**Timed memoryless strategies ($\Sigma_t$)**

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

(Intuition: agent $a$ selects an action based on the final state)
Let $a \in Agents$:

**Timed perfect recall strategies ($\Sigma_T$)**

Functions $\sigma_a : S^+ \rightarrow \Sigma$, s.t., $\forall \pi \in S^+, \sigma_a(\pi) \in protocol_a(loc(\pi_F))$.

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

**Timed memoryless strategies ($\Sigma_t$)**

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

(Intuition: agent $a$ selects an action based on the final state)
**Timeless perfect recall strategies** ($\Sigma_R$)

Strategies $\sigma_a \in \Sigma_T$ s.t., for each $n \in \mathbb{N}$ and $\pi, \pi' \in S^n$, if $\text{loc}(\pi(i)) = \text{loc}(\pi'(i))$ 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** ($\Sigma_r$)

Strategies $\sigma_a \in \Sigma_T$ s.t., for each $\pi, \pi' \in S^+$, if $\text{loc}(\pi_F) = \text{loc}(\pi'_F)$, 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 \(\pi, \pi' \in S^n\), 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 \(\pi, \pi' \in S^+\), 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)
Timeless perfect recall strategies ($\Sigma_R$)

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

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

Strategies \(\sigma_a \in \Sigma_T\) s.t., for each \(n \in \mathbb{N}\) and \(\pi, \pi' \in S^n\), if \(\text{loc}(\pi(i)) = \text{loc}(\pi'(i))\) 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** \((\Sigma_r)\)

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

(Intuition: agent \(a\) selects an action based on the final location)
Counting strategies ($\Sigma\#$)

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

Alternative notation

A counting strategy is a function $\sigma_a^\# : Q \times \mathbb{N} \rightarrow \Sigma$ s.t.

$\sigma_a^\#(q, k) := \sigma_a(\pi)$ if $q = \text{loc}(\pi_F)$ and $k = \#_F(\pi)$. 
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 $q \in Q$ there exist:

- actions $act_1, \ldots, act_{n+1} \in \Sigma$, and
- integer intervals $l_1 = [1, i_1), l_2 = [i_1, i_2), \ldots, l_{n+1} = [i_n, \infty)$

s.t. for all $1 \leq j \leq n + 1$: $\sigma^\#_a(q, k) = act_j$ if $k \in l_j$.

Example: a counting strategy is 2–threshold if for any location $q \in Q$ there are three actions $act_1, act_2, act_3$ s.t. first only $act_1$ is used when $q$ is visited, then only $act_2$, and finally only $act_3$, ad infinitum.
Threshold strategies ($\Sigma_{\# n}$)

A counting strategy $\sigma_{\#}^a \in \Sigma_{\#}$ is called $n$–threshold for some $n \in \mathbb{N}_+$ iff for each location $q \in Q$ 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 $q \in Q$ there are three actions $\text{act}_1, \text{act}_2, \text{act}_3$ s.t. first only $\text{act}_1$ is used when $q$ is visited, then only $\text{act}_2$, and finally only $\text{act}_3$, ad infinitum.
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 $q \in Q$ 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 $q \in Q$ there are three actions $\text{act}_1, \text{act}_2, \text{act}_3$ s.t. first only $\text{act}_1$ is used when $q$ is visited, then only $\text{act}_2$, and finally only $\text{act}_3$, ad infinitum.
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 $q \in Q$ there exist:

- actions $\text{act}_1, \ldots, \text{act}_{n+1} \in \Sigma$, and
- integer intervals $l_1 = [1, i_1), l_2 = [i_1, i_2), \ldots, l_{n+1} = [i_n, \infty)$

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

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

- actions $act_{1}, \ldots, act_{n+1} \in \Sigma$, 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 $1 \leq j \leq n+1$: $\sigma^{\#}_{a}(q, k) = act_j$ if $k \in I_j$.

Example: a counting strategy is 2–threshold if for any location $q \in Q$ there are three actions $act_1, act_2, act_3$ 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** $\sigma_A$ 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 $\pi \in S^\omega$ denote $\sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i))$.

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

Intuition: when coalition $A$ follows $\sigma_A$, 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 $\sigma_A$ 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 $\pi \in S^\omega$ denote $\sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i))$.

- The outcome of $\sigma_A$ in state $s \in S$ is the set $out(s, \sigma_A) \subseteq S^\omega$ 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 $act' \in protocol_A(loc(\pi(i)))$.

Intuition: when coalition $A$ follows $\sigma_A$, 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** $\sigma_A$ 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 $\pi \in S^\omega$ denote $\sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i))$.

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

  **Intuition:** when coalition $A$ follows $\sigma_A$, 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** \( \sigma_A \) 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 \( \pi \in S^\omega \) denote \( \sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i)) \).

- **The outcome** of \( \sigma_A \) in state \( s \in S \) is the set \( out(s, \sigma_A) \subseteq S^\omega \) s.t. \( \pi \in out(s, \sigma_A) \) iff \( \pi(0) = s \) and for each \( i \in \mathbb{N} \)
  \[
  \pi(i) \xrightarrow{(\sigma_A(\pi_i),act')} \pi(i + 1) \text{ for some } act' \in protocol_A(loc(\pi(i))).
  \]

  **Intuition:** when coalition \( A \) follows \( \sigma_A \), then in every state it selects actions according to the joint strategy while the remaining agents can choose any actions.
A joint strategy $\sigma_A$ 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 $\pi \in S^\omega$ denote $\sigma_A(\pi_i) := (\sigma_{a_1}(\pi_i), \ldots, \sigma_{a_k}(\pi_i))$.

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

Intuition: when coalition $A$ follows $\sigma_A$, 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:

\[ \phi ::= p \mid \neg \phi \mid \phi \lor \phi \mid \langle A \rangle X \phi \mid \langle A \rangle \phi U_{\sim \eta} \phi \mid \langle A \rangle \phi R_{\sim \eta} \phi, \]

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

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

Derived modalities: \( F \) (“in the future”) and \( G \) (“globally”):

\[ \langle A \rangle F_{\sim \eta} \phi := \langle A \rangle \top U_{\sim \eta} \phi, \quad \langle A \rangle G_{\sim \eta} \phi := \langle A \rangle \bot R_{\sim \eta} \phi. \]
Timed Alternating-Time Temporal Logic (TATL)

The language of TATL is defined by the following grammar:

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

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

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

Derived modalities: $F$ (“in the future”) and $G$ (“globally”):

$$\langle A \rangle F_{\sim \eta} \phi ::= \langle A \rangle \top U_{\sim \eta} \phi, \quad \langle A \rangle G_{\sim \eta} \phi ::= \langle A \rangle \bot R_{\sim \eta} \phi.$$
Syntax of TATL

Timed Alternating-Time Temporal Logic (TATL)

The language of TATL is defined by the following grammar:

\[ \phi ::= p \mid \neg \phi \mid \phi \lor \phi \mid \langle\langle A\rangle\rangle X \phi \mid \langle\langle A\rangle\rangle \phi U_{\sim \eta} \phi \mid \langle\langle A\rangle\rangle \phi R_{\sim \eta} \phi, \]

where \( p \in AP \), \( A \subseteq \text{Agents} \), \( \sim \in \{\leq, =, \geq\} \), and \( \eta \in \mathbb{N} \).

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

Derived modalities: \( F \) (“in the future”) and \( G \) (“globally”):

\[ \langle\langle A\rangle\rangle F_{\sim \eta} \phi := \langle\langle A\rangle\rangle \top U_{\sim \eta} \phi, \quad \langle\langle A\rangle\rangle G_{\sim \eta} \phi := \langle\langle A\rangle\rangle \bot R_{\sim \eta} \phi. \]
The language of TATL is defined by the following grammar:

\[ \phi ::= p \mid \neg \phi \mid \phi \lor \phi \mid \langle A \rangle X \phi \mid \langle A \rangle \phi U_{\sim \eta} \phi \mid \langle A \rangle \phi R_{\sim \eta} \phi, \]

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

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

Derived modalities: \( F \) (“in the future”) and \( G \) (“globally”):

\[ \langle A \rangle F_{\sim \eta} \phi := \langle A \rangle \top U_{\sim \eta} \phi, \langle A \rangle G_{\sim \eta} \phi := \langle A \rangle \bot R_{\sim \eta} \phi. \]
Syntax of TATL

Timed Alternating-Time Temporal Logic (TATL)

The language of TATL is defined by the following grammar:

\[ \phi ::= p \mid \neg \phi \mid \phi \lor \phi \mid \langle A \rangle X \phi \mid \langle A \rangle \phi U_{\sim} \eta \phi \mid \langle A \rangle \phi R_{\sim} \eta \phi, \]

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

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

Derived modalities: \( F \) (“in the future”) and \( G \) (“globally”):

\[ \langle A \rangle F_{\sim} \eta \phi ::= \langle A \rangle \top U_{\sim} \eta \phi, \langle A \rangle G_{\sim} \eta \phi ::= \langle A \rangle \bot R_{\sim} \eta \phi. \]
The language of TATL is defined by the following grammar:

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

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

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

Derived modalities: $F$ (“in the future”) and $G$ (“globally”):

$$\langle A \rangle F_{\sim} \eta \phi := \langle A \rangle T U_{\sim} \eta \phi, \langle A \rangle G_{\sim} \eta \phi := \langle A \rangle \bot R_{\sim} \eta \phi.$$
Syntax of TATL

Timed Alternating-Time Temporal Logic (TATL)

The language of TATL is defined by the following grammar:

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

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

We interpret $\langle\langle A\rangle\rangle \psi$ as “the coalition $A$ has a strategy to enforce $\psi$”, $X$ 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:

\[ \phi ::= p \mid \neg \phi \mid \phi \lor \phi \mid \langle A \rangle X \phi \mid \langle A \rangle \phi U \sim \eta \phi \mid \langle A \rangle \phi R \sim \eta \phi, \]

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

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

Derived modalities: \( F \) (“in the future”) and \( G \) (“globally”):

\[ \langle A \rangle F \sim \eta \phi ::= \langle A \rangle \top U \sim \eta \phi, \langle A \rangle G \sim \eta \phi ::= \langle A \rangle \bot R \sim \eta \phi. \]
Timed Alternating-Time Temporal Logic (TATL)

The language of TATL is defined by the following grammar:

\[ \phi ::= p \mid \neg \phi \mid \phi \lor \phi \mid \langle\langle A\rangle\rangle X \phi \mid \langle\langle A\rangle\rangle \phi U_{\sim \eta} \phi \mid \langle\langle A\rangle\rangle \phi R_{\sim \eta} \phi, \]

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

We interpret \( \langle\langle A\rangle\rangle \psi \) as “the coalition \( A \) has a strategy to enforce \( \psi \)”, \( X \) 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

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

Examples of properties:

- \(\langle A \rangle G_{\geq 42}\text{safe}\): “Coalition A has a strategy to enforce that safe holds always after reaching 42 time units”.

- \(\langle A \rangle F_{= 13}\text{finish}\): “Coalition A has a strategy to enforce that finish is reached in exactly 13 time units”.
TATL\textsubscript{$\leq$,\textgreater{}}: a subset of TATL with only $\leq$, $\geq$ allowed,
e.g., $\langle A \rangle G_{\geq 42}\text{safe} \in TATL_{\leq,\geq}$, $\langle A \rangle F_{= 13}\text{finish} \notin TATL_{\leq,\geq}$.

Examples of properties:

- $\langle A \rangle G_{\geq 42}\text{safe}$: “Coalition A has a strategy to enforce that safe holds always after reaching 42 time units”.

- $\langle A \rangle F_{= 13}\text{finish}$: “Coalition A has a strategy to enforce that finish is reached in exactly 13 time units”.

Wojciech Penczek and al.  Timed ATL: Forget memory, Just count 17/28
For each type of strategy define the corresponding satisfaction relation, i.e., $|=Y$ corresponds to $\Sigma_Y$, for $Y \in \{T, t, R, r, \#, \#n\}$.

**Satisfaction relation**

$M, q |= Y \langle A \rangle \psi$ iff there exists a strategy $\sigma_A \in \Sigma_Y$ for $A$ s.t. $\psi$ holds along each outcome $\pi \in \text{out}((q, 0), \sigma_A)$.

**Satisfaction relation over outcomes**

- $\pi |= X\phi$ iff $\text{loc}(\pi(1)) |= \phi$,
- $\pi |= \phi U_{\sim \eta} \psi$ iff $\text{loc}(\pi(i)) |= \psi$ for some $i$ s.t. $\text{time}(\pi(i)) \sim \eta$ and $\text{loc}(\pi(j)) |= \phi$ for all $j < i$,
- $\pi |= \phi R_{\sim \eta} \psi$ iff for all $i$: $\text{time}(\pi(i)) \sim \eta \implies \text{loc}(\pi(i)) |= \psi$ or $\text{loc}(\pi(j)) |= \phi$ for some $j < i$. 
Hierarchy of satisfaction relations

\[ \models T \quad \models t \quad \models R \quad \models \# \quad \models \#_1 \quad \models \#_0 = \models r \]

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 \( q \in Q \) and \( \phi \in \text{TATL} \) 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 \( \langle A \rangle \psi \in \text{TATL} \) and \( c \in \mathbb{N} \) be the greatest integer in \( \psi \).

If \( \sigma_A \in \Sigma_T \) implements \( \langle A \rangle \psi \), then there exists its reduction \( \sigma'_A \) s.t. \( \forall q \in Q \forall t \geq c \) \( \sigma'_A(q, t) = \sigma'_A(q, c + 1) \), which also implements \( \langle A \rangle \psi \).

Intuitively, there is no need to track time after it exceeds \( c \).
Theorem (1) Timed strategies do not need memory

For each $q \in Q$ and $\phi \in \text{TATL}$ 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 $\langle A \rangle \psi \in \text{TATL}$ and $c \in \mathbb{N}$ be the greatest integer in $\psi$.
If $\sigma_A \in \Sigma_T$ implements $\langle A \rangle \psi$, then there exists its reduction $\sigma'_A$ s.t.

\[
\forall q \in Q \forall t \geq c \quad \sigma'_A(q, t) = \sigma'_A(q, c + 1),
\]

which also implements $\langle A \rangle \psi$.

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 $q \in Q$ and $\phi \in \text{TATL}$ 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 $\ll A \rr \psi \in \text{TATL}$ and $c \in \mathbb{N}$ be the greatest integer in $\psi$. If $\sigma_A \in \Sigma_T$ implements $\ll A \rr \psi$, then there exists its reduction $\sigma'_A$ s.t. $\forall q \in Q \forall t \geq c \sigma'_A(q, t) = \sigma'_A(q, c + 1)$, which also implements $\ll A \rr \psi$.

Intuitively, there is no need to track time after it exceeds $c$. 

Wojciech Penczek and al.  Timed ATL: Forget memory, Just count 20/28
Key implications: time versus order

Theorem (2) $\models \# \implies \models$

For each $q \in Q$ and $\phi \in \text{TATL}$, if $q \models \# \phi$, then $q \models \phi$. 
Key implications: time versus order, cont’d

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

**Theorem (3) |\(= \implies |\#\)**

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

(3) cannot be extended to TATL, see next slide.
Key implications: time versus order, cont’d

\[ \text{TATL}_{\leq, \geq} : \text{a subset of TATL with only } \leq, \geq \text{ allowed, e.g., } \langle \langle A \rangle \rangle G_{\geq 42} \text{safe } \in \text{TATL}_{\leq, \geq}, \langle \langle A \rangle \rangle F_{=13} \text{finish } \not\in \text{TATL}_{\leq, \geq}. \]

Theorem (3) \( \models \iff \models \# \)

For each \( q \in Q \) 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, cot’d

$q_0 \models_T \langle 1 \rangle F=5 p$, but $q_0 \nvDash \# \langle 1 \rangle F=5 p$, 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

$q_0 \models T \langle 1 \rangle F_{=5} p$, but $q_0 \not\models \# \langle 1 \rangle F_{=5} p$, 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

$q_0 \models \tau \langle 1 \rangle F_{=5} p$, but $q_0 \not\models \# \langle 1 \rangle F_{=5} p$, 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: counting up to...

Theorem (4) Threshold for $\text{TATL}_{\leq,\geq}$ is 2

For each $q \in Q$ and $\phi \in \text{TATL}_{\leq,\geq}$, 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

$q_0 \models \#_1 \langle 1 \rangle F_{\geq 5} p$: loops four times and jumps ahead

$q_0 \not\models \#_0 \langle 1 \rangle F_{\geq 5} p$: loops forever, or jumps too early
Key implications: counting up to..., ct’d

Theorem (5)

There is no threshold for TATL.

\[ F_{\geq 17} p: \textbf{three distinct actions} \text{ needed to sum up to exactly 17 time units.} \]

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

\[
\begin{align*}
\models T & \quad \models t & \quad \models R \\
\models # & \quad \models #_1 & \quad \models #_0 = \models r \\
\end{align*}
\]

The Red implications hold only for 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 $\text{TATL}^*$ and **parametric** versions,
- **Incomplete** knowledge semantics,
- **Model checking** algorithms.
Thank you!