Games and Temporal Logics for the Verification of Timed Systems

François Laroussinie

LIAFA / Paris 7 – CNRS

ETR 2009, September 2009
Verification: Model checking

System

Properties

Formalizing step

Automaton, Kripke structure, Petri net, ...

Temporal logic formulas

? |= \( \varphi \)
Why considering games in verification?

- Some properties can be expressed as games (in a natural manner).
- Modeling the interaction of components in a global system.
Games in verification

Why considering games in verification?

- Some properties can be expressed as games (in a natural manner).
- Modeling the interaction of components in a global system. → related to control problems.

Example...
Train crossing example

We have:

- trains and cars that can arrive near the crossing,
- a gate that can be open or closed,
- the gate-keeper who open or close the gate.
Train crossing example

We have:

• trains and cars that can arrive near the crossing,
• a gate that can be open or closed,
• the gate-keeper who open or close the gate.

→ two players game:

• Bob (the gate-keeper), and
• the trains and the cars.

Bob tries (1) to avoid crash and (2) to ensure progress of cars.

Q? Is there a strategy for Bob to avoid a crash and to ensure that nobody will be deadlocked?
Train crossing example

We have:

- trains and cars that can arrive near the crossing,
- a gate that can be open or closed,
- the gate-keeper who open or close the gate.

→ two players game:

- Bob (the gate-keeper), and
- the trains and the cars.

Bob tries (1) to avoid crash and (2) to ensure progress of cars.

Q? Is there a strategy for Bob to avoid a crash and to ensure that nobody will be deadlocked?

Games are a natural model for modeling open systems. A strategy $\rightarrow$ a control policy.
Outline

1. Models for multi-agents systems
2. Alternating-time temporal Logic
3. Model checking
4. Timed extensions
   - Timed ATL
   - Durational CGS
   - Dense-time games
   - Analysis of TCGS
Outline

1. Models for multi-agents systems
2. Alternating-time temporal Logic
3. Model checking
4. Timed extensions
   - Timed ATL
   - Durational CGS
   - Dense-time games
   - Analysis of TCGS
Multi-agent systems

Synchronous multi-agent systems:
(1) in any location, each agent \textit{chooses one move}
(2) the new location of the system results from these choices.
Multi-agent systems

Synchronous multi-agent systems:
(1) in any location, each agent chooses one move
(2) the new location of the system results from these choices.

Here we consider the **Concurrent Game Structures (CGS):**
→ In each location, each one of the $k$ agents has a finite number of possible moves (described with a special alphabet).
→ An $k$-ary transition function gives the location to which the execution goes.

(Other models exist !)
Let AP be a set of atomic propositions.

**Definition [AHK02]**

A CGS $C$ is a 8-tuple $(Q, q_0, \ell, \text{Agt}, M, Mv, \text{Edg})$ s.t:

- $Q$ is a finite set of states, $q_0 \in Q$;
- $\ell : Q \rightarrow 2^{AP}$ is the labeling of atomic prop. ;
- $\text{Agt} = \{A_1, ..., A_k\}$: a set of *agents*;
- $M$ is a finite alphabet of moves;
- $Mv : Q \times \text{Agt} \rightarrow 2^M$ the choice function;
- $\text{Edg} : Q \times M^k \rightarrow Q$: the transition table.

**Size of $C$:** $|\text{Loc}| + |\text{Edg}|$

**Next($\ell, A_i, m$)**: possible successor locations when $A_i$ plays $m$. 

---

**Concurrent Game Structure**
CGS – example

Paper, rock and scissors

\[ \langle p.p \rangle, \langle r.r \rangle, \langle s.s \rangle \]

\[ \text{Start} \]

\[ q_0 \]

2-Win \[ q_2 \]

1-Win \[ q_1 \]

<table>
<thead>
<tr>
<th>( q_0 )</th>
<th>( p )</th>
<th>( r )</th>
<th>( s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>( q_0 )</td>
<td>( q_1 )</td>
<td>( q_2 )</td>
</tr>
<tr>
<td>( r )</td>
<td>( q_2 )</td>
<td>( q_0 )</td>
<td>( q_1 )</td>
</tr>
<tr>
<td>( s )</td>
<td>( q_1 )</td>
<td>( q_2 )</td>
<td>( q_0 )</td>
</tr>
</tbody>
</table>
CGS – example

Paper, rock and scissors

\[ \langle p.p \rangle, \langle r.r \rangle, \langle s.s \rangle \]

Start \hspace{1cm} q_0

\[ \langle s.r \rangle, \langle p.s \rangle, \langle r.p \rangle \]

\[ \langle r.s \rangle, \langle s.p \rangle, \langle p.r \rangle \]

2–Win \hspace{1cm} q_2

1–Win \hspace{1cm} q_1

A_1

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>q_0</td>
<td>q_1</td>
<td>q_2</td>
</tr>
<tr>
<td>r</td>
<td>q_2</td>
<td>q_0</td>
<td>q_1</td>
</tr>
<tr>
<td>s</td>
<td>q_1</td>
<td>q_2</td>
<td>q_0</td>
</tr>
</tbody>
</table>

A_2

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>q_0</td>
<td>q_1</td>
<td>q_2</td>
</tr>
<tr>
<td>r</td>
<td>q_2</td>
<td>q_0</td>
<td>q_1</td>
</tr>
<tr>
<td>s</td>
<td>q_1</td>
<td>q_2</td>
<td>q_0</td>
</tr>
</tbody>
</table>
Paper, rock and scissors

\[ \langle p.p \rangle, \langle r.r \rangle, \langle s.s \rangle \]

\[ A_1 \]

\[
\begin{array}{c|ccc}
q_0 & p & r & s \\
\hline
p & q_0 & q_1 & q_2 \\
r & q_2 & q_0 & q_1 \\
s & q_1 & q_2 & q_0 \\
\end{array}
\]

\[ A_2 \]

\[
\begin{array}{c|ccc}
q_0 & p & r & s \\
\hline
p & q_0 & q_1 & q_2 \\
r & q_2 & q_0 & q_1 \\
s & q_1 & q_2 & q_0 \\
\end{array}
\]
CGS – example

Paper, rock and scissors

The transition table may be exponential in the nb of agents... But succinct encodings exist.
3 players: $A_1, A_2, A_3$. 

$$q_0 \xrightarrow{\langle 1.1.1, 1.1.2, 1.2.1, 1.2.2 \rangle} q_1 \xrightarrow{\langle 2.2.1, 2.2.2 \rangle} q_2 \xrightarrow{\langle 2.1.1, 2.1.2 \rangle}$$
Symbolic CGS – example

3 players: $A_1, A_2, A_3$.

$A_1 \not\rightarrow 1 \land A_2 \rightarrow 2$

$q_0$ \hspace{1cm} $A_1 \rightarrow 1$

$q_1$ \hspace{1cm} $q_2$ \hspace{1cm} "otherwise"

$\text{Mov}(q_0) \overset{\text{def}}{=} (A_1 \rightarrow 1, q_0), (A_2 \rightarrow 2, q_1), (\top, q_2)$

→ adds succinctness.
Following [JD05], one define symbolic CGSs as follows:

**Definition – symbolic CGS**

→ The transition function from each location $q$ is defined by a finite sequence $((\varphi_1, q_1), \ldots, (\varphi_n, q_n))$ s.t.:

- $q_i \in Q$ and $\varphi_i$ is a boolean combination of propositions “$A_j \rightarrow c$” (i.e. “Agent $A_j$ plays $c$”)
- $\varphi_n = \top$
- $\text{Edg}(q, m_1, \ldots, m_k) = q_j$ with $j = \min_i \{m_1 \ldots m_k \models \varphi_i\}$

$|\text{Edg}| = \text{sum of the sizes of the formulas.}$
Comparison of games models

For classical systems, there exist a lot of behavioral equivalences! (And bisimulation is usually the good criterion.)

How can we compare games?
Comparison of games models

For classical systems, there exist a lot of behavioral equivalences!
(And bisimulation is usually the good criterion.)

How can we compare games?
by using alternating bisimulation.
Idea: each move of any coalition has to be simulated...
Translations between models

Time complexity of translations between the three models:

- ATS
- CGS
- Symbolic CGS

"equivalent" = alternating bisimilar
In a **turn-based system**, only one player has several moves in a given location.
In a **turn-based system**, only one player has several moves in a given location.
In a turn-based system, only one player has several moves in a given location.
Turn-based CGSs

And...
Turn-based CGSs

And...
Turn-based CGSs

And...
How can we state properties of the train crossing example like:

There is a strategy for Bob to avoid a crash and to ensure that nobody will be deadlocked?
How can we state properties of the train crossing example like:

There is a strategy for Bob to avoid a crash and to ensure that nobody will be deadlocked?

with a special kind of temporal logic: a temporal logic for games!
Outline

1. Models for multi-agents systems

2. Alternating-time temporal Logic

3. Model checking

4. Timed extensions
   - Timed ATL
   - Durational CGS
   - Dense-time games
   - Analysis of TCGS
Temporal logics for the specification of reactive systems

Classical Temporal Logics (CTL, LTL, . . . ) are very convenient to specify reactive systems

- good expressive power,
- natural semantics,
- succinctness,
- efficient ( . . . ) decision procedures (and tools !),
- nice extensions (timed TL, probabilistic TL, . . . ).
Linear-time Temporal Logics – LTL

Formulas built from \( \text{AP}, \land, \neg, \ U, \ X, \ S, \ S^{-1}, \ldots \)

- \( \varphi \ U \psi \): \( \varphi \) holds for every suffix until \( \psi \) is satisfied.
- \( F \varphi \overset{\text{def}}{=} \top \ U \varphi \): “eventually \( \varphi \)”
- \( G \psi \overset{\text{def}}{=} \neg F \neg \varphi \): “always \( \varphi \)”

Formulas are interpreted over infinite executions: \( q_0 \cdot q_1 \cdot q_2 \cdots \)
\( S \models \Phi \iff \Phi \) holds for any run in \( S \).

- \( G (\text{problem} \Rightarrow F \text{ alarm}) \)
- \( (G F \ P_1) \Rightarrow (G F \ P_2) \)
Formulas contain also **path quantifiers**: \( E \) and \( A \).

Formulas are interpreted over **states** of Kripke structures.

\( S \models \Phi \iff \Phi \text{ holds for } q_{init} \).

- \( E \left( P_1 \text{ U } ( A X \ P_2) \right) \)
- \( AG \left( \text{problem } \Rightarrow \ A F \text{ alarm} \right) \)
- \( AG \left( \text{problem } \Rightarrow \ E F \text{ alarm} \right) \)
Alternating-time Temporal Logics – ATL

ATL is a logic for expressing properties in games or multi-agent systems.

**Multi-agent systems**: several agents can concurrently act upon the behavior of the system.

With ATL, one can express properties such as “there is a strategy for agent $A$ to ensure a temporal property $\Phi$”

$\langle A \rangle \phi$
• Linear-time temporal logics (LTL, ... ) express properties over executions.

\( F \) (winning)

• Branching-time temporal logics (CTL, CTL *, ... ) allow the use of existential and universal path quantifier.

\( EF \) (winning) or \( AF \) (winning)
• Linear-time temporal logics (LTL, ...) express properties over executions.

\[ F(\text{winning}) \]

• Branching-time temporal logics (CTL, CTL *, ...) allow the use of existential and universal path quantifier.

\[ EF(\text{winning}) \text{ or } AF(\text{winning}) \]

• Alternating-time temporal logic allows us to quantify over subsets of paths.

\[ \langle A \rangle F(\text{winning}) : \text{“The player } A \text{ can enforce the system to reach a } \text{winning} \text{ state (whatever the other agents do)”} \]
ATL vs LTL, CTL,...

- Linear-time temporal logics (LTL, ...) express properties over executions.
  \( F \) (winning)

- Branching-time temporal logics (CTL, CTL *,...) allow the use of existential and universal path quantifier.
  \( EF \) (winning) or \( AF \) (winning)

- Alternating-time temporal logic allows us to quantify over subsets of paths.
  \( \langle A \rangle F \) (winning) : “The player \( A \) can enforce the system to reach a winning state (whatever the other agents do)”.

\[
\langle \emptyset \rangle \Phi \overset{\text{def}}{=} A \Phi \quad \langle \text{Agt} \rangle \Phi \overset{\text{def}}{=} E \Phi
\]
For a CGS, we use the following notions for a coalition $A \subseteq \text{Agt}$:

**Definition**

- $\text{Mv}(q, A)$: the set of possible moves for $A$.
  
  $m = (m_a)_{a \in A}$ with $m_a \in \text{Mv}(q, a)$.

- If $m' \in \text{Mv}(q, \text{Agt}\setminus A)$, $m \cdot m'$ denotes the corresponding complete move.

- $\text{Next}(q, A, m)$: the set of possible successor locations from $q$ when every agent in $A$ plays according to $m$.

- $\text{Next}(q)$: the set of possible successor locations from $q$.
For any CGS, we define the following standard notions:

**Definition**

- **computation** \( \rho = q_1 q_2 q_3 \cdots \) s.t. \( \forall i, q_{i+1} \in \text{Next}(q_i) \).

- **strategy** for \( A_i \) = function \( f_{A_i} \) that maps any finite prefix of a computation to a possible move for \( A_i \).
  \[
  f_{A_i}(q_1, \cdots, q_m) \in \text{Next}(q_m, A_i) \quad f_{A_i} \in \text{Strat}(A_i)
  \]

  A strategy for \( A \subseteq \text{Agt} = (f_a)_{a \in A} \) with \( f_a \in \text{Strat}(a) \)

  \( f_A \) is **state-based** iff it only depends on the current state.

- The **outcomes** \( \text{Out}(q, F_A) \) are the set of computations from \( q \) induced by the strategy \( F_A \) for \( A \).
Syntax and semantics of ATL [AHK02]

**Definition**

\[
\text{ATL} \ni \varphi_s, \psi_s ::= P \mid \neg \varphi_s \mid \varphi_s \lor \psi_s \mid \langle A \rangle \varphi_p \\
\varphi_p ::= X \varphi_s \mid \varphi_s U \psi_s \mid \varphi_s W \psi_s
\]

with \( P \in \text{AP} \) and \( A \subseteq \text{Agt} \).
Syntax and semantics of ATL [AHK02]

**Definition**

\[
\text{ATL} \ni \varphi_s, \psi_s ::= P \mid \neg \varphi_s \mid \varphi_s \lor \psi_s \mid \langle A \rangle \varphi_p \\
\varphi_p ::= \mathbf{X} \varphi_s \mid \varphi_s \mathbf{U} \psi_s \mid \varphi_s \mathbf{W} \psi_s
\]

with \( P \in \text{AP} \) and \( A \subseteq \text{Agt.} \)

\[
q \models_s \langle A \rangle \varphi_p \quad \text{iff} \quad \exists F_A \in \text{Strat}(A). \forall \rho \in \text{Out}(q, F_A). \rho \models_s \varphi_p,
\]

\[
\rho \models_s \mathbf{X} \varphi_s \quad \text{iff} \quad \rho[1] \models_s \varphi_s,
\]

\[
\rho \models_s \varphi_s \mathbf{U} \psi_s \quad \text{iff} \quad \exists i. \rho[i] \models_s \psi_s \text{ and } \forall 0 \leq j < i. \rho[j] \models_s \varphi_s,
\]

\[
\rho \models_s \varphi_s \mathbf{W} \psi_s \quad \text{iff} \quad (\rho \models_s \varphi_s \mathbf{U} \psi_s) \text{ or } \forall i. \rho[i] \models_s \varphi_s.
\]
ATL– examples

\[ \langle \text{Controller} \rangle \mathbf{G} (\neg \text{Bad}) \]

\[ \langle A \rangle \mathbf{F} \left( \neg \langle B \rangle \mathbf{F} P \land \neg \langle C \rangle \mathbf{F} P \right) \]

\[ \langle A \rangle \mathbf{F} \left( \neg \langle B \rangle \mathbf{F} P \land \neg \langle C \rangle \mathbf{F} P \land \langle B, C \rangle \mathbf{F} P \right) \]

\[ \langle \text{Sender,Receiver} \rangle \mathbf{F} (\text{msg-ok}) \]
• $\langle \text{Agt} \rangle \varphi \equiv E \varphi$ and $\langle \emptyset \rangle \varphi \equiv A \varphi$
Semantics and expressiveness

- $\langle \text{Agt} \rangle \varphi \equiv E\varphi$ and $\langle \emptyset \rangle \varphi \equiv A\varphi$

- For ATL it is sufficient to consider state-based strategies. [AHK02, Sch04]. ($\neq$ ATL*)
Semantics and expressiveness

• \( \langle \text{Agt} \rangle \varphi \equiv \text{E}\varphi \) and \( \langle \emptyset \rangle \varphi \equiv \text{A}\varphi \)

• For ATL it is sufficient to consider state-based strategies. [AHK02, Sch04]. \((\not= \text{ATL}^*)\)

• \( \neg \langle A \rangle \varphi \not= \langle \text{Agt} \setminus A \rangle \neg \varphi \) [AHK02, GvD06]
Semantics and expressiveness

- $\langle \text{Agt} \rangle \varphi \equiv E\varphi$ and $\langle \emptyset \rangle \varphi \equiv A\varphi$

- For ATL it is sufficient to consider state-based strategies. [AHK02, Sch04].

- $\neg \langle A \rangle \varphi \not\equiv \langle \text{Agt}\backslash A \rangle \neg \varphi$ [AHK02, GvD06]

Ex: $q_0 \models \neg \langle A_1 \rangle X p \land \neg \langle A_2 \rangle \neg X p$

\[\begin{array}{c}
q_0 \\
\langle 1,1 \rangle \\
\langle 2,2 \rangle \\
p \\
q_1 \\
\langle 1,2 \rangle \\
\langle 2,1 \rangle \\
\neg p \\
q'_2 \\
\langle 2,1 \rangle \\
\neg p \\
q'_1
\end{array}\]
From the equivalence $\varphi \mathbf{W} \psi \equiv \mathbf{G} \varphi \lor \varphi \mathbf{U} \psi$, one can deduce:

$$\mathbf{E} \varphi \mathbf{W} \psi \equiv \mathbf{E} \mathbf{G} \varphi \lor \mathbf{E} \varphi \mathbf{U} \psi$$

And ATL?
From the equivalence \( \varphi W \psi \equiv G \varphi \lor \varphi U \psi \), one can deduce:
\[
E \varphi W \psi \equiv EG \varphi \lor E \varphi U \psi
\]

And ATL?
\[
\langle A \rangle a W b \nRightarrow \langle A \rangle G a \lor \langle A \rangle (a U b)
\]

<table>
<thead>
<tr>
<th>1.1</th>
<th>1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
</tr>
</tbody>
</table>

\( \models \langle A_1 \rangle (a W b) \)
From the equivalence $\varphi W \psi \equiv G \varphi \lor \varphi U \psi$, one can deduce:

$$E\varphi W \psi \equiv EG \varphi \lor E\varphi U \psi$$

And ATL?

$$\langle A \rangle a W b \not\Rightarrow \langle A \rangle G a \lor \langle A \rangle (a U b)$$

$$\models \langle A_1 \rangle (a W b)$$

$$\langle A \rangle \_ W \_$$ can not be expressed with $$\langle A' \rangle \_ U \_$$ and $$\langle A' \rangle G \_$$.
In a turn-based system, only one player has several moves in a given location.

In these systems:

$$\langle A \rangle (\varphi \mathbf{W} \psi) \equiv \neg \langle \text{Agt} \setminus A \rangle (\neg \psi) \mathbf{U} (\neg \psi \land \neg \varphi)$$

Such models are determined: given a winning objective $\Phi$, either there is strategy for $A$ to ensure $\Phi$, or there is a strategy for the other players ($\text{Agt} \setminus A$) to enforce $\neg \Phi$. 
Outline

1. Models for multi-agents systems
2. Alternating-time temporal Logic
3. Model checking
4. Timed extensions
   - Timed ATL
   - Durational CGS
   - Dense-time games
   - Analysis of TCGS
Model checking

\( S \models \Phi \) ?

Model checking algorithms are based on the fixpoint computations:

\[
\langle A \rangle p \mathbf{U} q \equiv \mu Z. \left( q \lor (p \land \langle A \rangle X Z) \right)
\]  

(1)

The difference with CTL: \( \langle A \rangle X \) instead of \( EX \).

Model checking

$S \models \Phi$?

Model checking algorithms are based on the fixpoint computations:

$$\langle \langle A \rangle \rangle p \mathbf{U} q \equiv \mu Z. \left( q \lor (p \land \langle \langle A \rangle \rangle X Z) \right)$$

(1)

The difference with CTL: $\langle \langle A \rangle \rangle X$ instead of $\mathbf{EX}$.

$\langle \langle A \rangle \rangle X \varphi$ characterizes the controllable predecessors of $\varphi$ for $A$:

$$\text{CPre}(A, S) \overset{\text{def}}{=} \{ \ell \in \text{Loc} \mid \exists m_A \in \text{Mov}(\ell, A) \text{s.t.} \text{Next}(\ell, A, m_A) \subseteq S \}$$

The crucial point of the model checking algorithm is the computation of CPre.
Fixpoint computation

\[ \langle A_1 \rangle F P \]
Fixpoint computation

\[ \langle A_1 \rangle F P \]
Fixpoint computation

\[ \langle A_1 \rangle F^* P \]
Fixpoint computation

\( \langle A_1 \rangle F P \)
Complexity overview

**ATL model checking**

- PTIME-complete for CGS. [AHK02]
- $\Delta^p_2$-complete for ATS. [LMO07]
- $\Delta^p_3$-complete for symbolic CGS. [LMO07]

**Computing CPre [LMO08]**

- in AC$^0$ for CGS
- NP-complete for ATS
- $\Sigma^p_2$-complete for symbolic CGS
Polynomial-time hierarchy $\text{PH}$

- $\text{PTIME}$
- $\text{NP}$
- $\Sigma_2^P = \text{NP}^{\text{NP}}$
- $\Delta_2^P = \text{PTIME}^{\text{NP}}$
- $\Pi_2^P = \text{co-NP}^{\text{NP}}$
- $\Delta_3^P = \text{PTIME}^{\Sigma_2^P}$
- $\Delta_3^P = \text{co-NP}$
- $\text{PSPACE}$

$\text{PH}$
Outline

1. Models for multi-agents systems
2. Alternating-time temporal Logic
3. Model checking
4. Timed extensions
   - Timed ATL
   - Durational CGS
   - Dense-time games
   - Analysis of TCGS
Timed and multi-agent systems

KS + CTL \( + \text{time} \) \( \rightarrow \) TA + TCTL

\( + \text{agents} \downarrow \)

CGS + ATL \( + \text{time} \) \( \rightarrow \) ? + TATL

\( + \text{agents} \downarrow \)
1 Models for multi-agents systems
2 Alternating-time temporal Logic
3 Model checking
4 Timed extensions
   - Timed ATL
   - Durational CGS
   - Dense-time games
   - Analysis of TCGS
Syntax of TATL

\[
TATL \ni \varphi_s, \psi_s \quad ::= \quad p \mid \neg \varphi_s \mid \varphi_s \lor \psi_s \mid \langle A \rangle \varphi_p
\]

\[
\psi_p \quad ::= \quad \varphi_s U_d \psi_s \mid \varphi_s R_d \psi_s
\]

Where \( d \in \mathbb{N}^* \) and \( \sim \in \{<, \leq, =, \geq, >\} \)

TATL is interpreted over game structures containing a quantitative notion of time.
Timed ATL

Semantics

\[ q \models \langle A \rangle \varphi_p \iff \exists F_A. \forall \rho \in \text{Out}(q, F_A). \rho \models \varphi_p \]

\[ \rho \models \varphi_s \mathbf{U_{\sim d}} \psi_s \iff \exists i \in \mathbb{N}. \rho_s(i) \models \psi_s \]

and \( \rho_s(j) \models \varphi_s \) for any \( 0 \leq j < i \)

and Duration(\( \rho(0) \to \rho(i) \)) \( \sim d \)

\[ \rho \models \varphi_s \mathbf{R_{\sim d}} \psi_s \iff \rho \models \neg (\neg \varphi_s \mathbf{U_{\sim d}} (\neg \psi_s)) \]
1. Models for multi-agents systems
2. Alternating-time temporal Logic
3. Model checking
4. Timed extensions
   - Timed ATL
   - Durational CGS
   - Dense-time games
   - Analysis of TCGS
Discrete time!
(Time delays are controlled by additional agents.)
Model checking TATL over tight DCGSs

\[ \text{Computation of function } v_i(\ell): \text{ the minimal duration that } A \text{ can ensure before reaching } P_2 \text{ (along a path satisfying } P_1). \]

\[
\begin{cases}
  \text{if } \ell \models \varphi_2 : & v_0(\ell) = 0 \\
  \text{if } \ell \models \neg \varphi_2 : & v_0(\ell) = +\infty \\
  \text{if } \ell \models \varphi_2 : & v_{i+1}(\ell) = 0 \\
  \text{if } \ell \models \neg \varphi_1 \land \neg \varphi_2 : & v_{i+1}(\ell) = +\infty \\
  \text{otherwise} : & v_{i+1}(\ell) = \min_{c \in \text{Mov}(\ell, A)} \max_{\bar{c} \in \text{Mov}(\ell, \bar{A})} \left( \text{Edg}_\tau(\ell, c \cdot \bar{c}) + v_i(\text{Edg}_\ell(\ell, c \cdot \bar{c})) \right)
\end{cases}
\]
Model-Checking Complexity

<table>
<thead>
<tr>
<th>Modalities</th>
<th>( U_\leq )</th>
<th>( U_\geq )</th>
<th>( U_= )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>( O(</td>
<td>A</td>
<td>^2) )</td>
</tr>
</tbody>
</table>

Theorem ([LMO06])

- Model-checking TATL over TDCGS is EXPTIME-complete.
- Model-checking of TATL_{\leq,\geq} over TDCGS is PTIME-complete.

Deciding \( S \models \langle A \rangle \mathcal{F}_{=c} P \) is EXPTIME-complete
A countdown game $C$ consists of a weighted graph $(S, T)$ (with $T \subseteq S \times \mathbb{N}_{>0} \times S$).

A configuration of the game is a pair $(s, c)$, with $s \in S$, $c \in \mathbb{N}$.

A move $(s, c)$ is as follows:

- player 1 chooses a duration $d$, s.t. $0 < d \leq c$ and $\exists (s, d, s') \in T$
- then player 2 chooses a state $s'$ s.t. $(s, d, s') \in T$ of duration $d$.

The resulting new configuration is $(s', c - d)$.

Player 1 wins if a configuration $(-, 0)$ is reached.
A countdown game $C$ consists of a weighted graph $(S, T)$ (with $T \subseteq S \times \mathbb{N}_{>0} \times S$).

A configuration of the game is a pair $(s, c)$, with $s \in S, c \in \mathbb{N}$.

A move $(s, c)$ is as follows:

- player 1 chooses a duration $d$, s.t. $0 < d \leq c$ and
  $\exists (s, d, s') \in T$
- then player 2 chooses a state $s'$ s.t. $(s, d, s') \in T$ of duration $d$.

The resulting new configuration is $(s', c - d)$.

Player 1 wins if a configuration $(-, 0)$ is reached.

**Theorem [JLS07]**

Deciding the winner in countdown games is EXPTIME-C.
Outline

1. Models for multi-agents systems
2. Alternating-time temporal Logic
3. Model checking
4. Timed extensions
   - Timed ATL
   - Durational CGS
   - Dense-time games
   - Analysis of TCGS
KS
Kripke structures
How to combine time and game? (I)

KS
Kripke structures

Time

TA
Timed Automata
How to combine time and game? (I)

KS
Kripke structures

Time

TA
Timed Automata

Game

CGS
Concurrent Game Struct.
How to combine time and game? (I)

KS
Kripke structures

Game

CGS
Concurrent Game Struct.

Time

TA
Timed Automata

?T?G?
Timed ? Game ?
How to combine time and game? (I)

KS
Kripke structures

Time

TA
Timed Automata

Game

TGA
Timed Game Automata

CGS
Concurrent Game Struct.

[dAFH+03]
Timed Game Automata [dAFH⁺03]

\[ x := 0 \]

\[ x > 0 \]
Timed Game Automata [dAFH+03]

\[ c_2 \ ; \ x := 0 \]

\[ c_1 \ ; \ x > 0 \]
At state \((q_1, 0)\) both players propose a move:

- Player 1 proposes \((t_1, c_1)\),
- Player 2 proposes \((t_2, c_2)\).
At state $(q_1, 0)$ both players propose a move:

- **Player 1** proposes $(t_1, c_1)$,
- **Player 2** proposes $(t_2, c_2)$.

The next state is given by
At state \((q_1, 0)\) both players propose a move:

- Player 1 proposes \((t_1, c_1)\),
- Player 2 proposes \((t_2, c_2)\).

The next state is given by

\[
(q_1, 0) \xrightarrow{c_1} (q_2, t_1) \quad \text{if } t_1 < t_2
\]
At state \((q_1, 0)\) both players propose a move:

- Player 1 proposes \((t_1, c_1)\),
- Player 2 proposes \((t_2, c_2)\).

If \(t_2 < t_1\), then the next state is given by

\[ (q_1, 0) \rightarrow (q_1, 0) \]
At state \((q_1, 0)\) both players propose a move:

- Player 1 proposes \((t_1, c_1)\),
- Player 2 proposes \((t_2, c_2)\).

The next state is given by

\[
\begin{align*}
\text{If } t_1 &= t_2, \\
(q_1, 0) &\xrightarrow{c_2} (q_1, 0) \\
(q_2, t_1) &\xrightarrow{c_1} (q_2, t_1)
\end{align*}
\]
With classical semantics, we have that:

\[ \mathcal{A} \not\models \langle A_1 \rangle F q_2 \]
With classical semantics, we have that:

\[ A \not\models \langle A_1 \rangle F q_2 \]

Indeed, Player 1 cannot prevent Player 2 to play:

\[ q_1 \xrightarrow{0,c_2} q_1 \xrightarrow{0,c_2} q_1 \xrightarrow{0,c_2} q_1 \xrightarrow{0,c_2} \ldots \]
With classical semantics, we have that:

\[ A \not\models \langle A_1 \rangle F q_2 \]

Indeed, Player 1 can not prevent Player 2 to play:

\[ q_1 \xrightarrow{0,c_2} q_1 \xrightarrow{0,c_2} q_1 \xrightarrow{0,c_2} q_1 \xrightarrow{0,c_2} \ldots \]

Thus Player 2 wins the game by blocking time.
In \[dAFH^+03, \text{HP06}\], an ‘alternative semantics’ has been proposed:

\[ \mathcal{A} \models_{NZ} \langle A_1 \rangle F q_2 \]
In [dAFH⁺03, HP06], an ‘alternative semantics’ has been proposed:

\[ \mathcal{A} \models_{NZ} \langle \langle A_1 \rangle \rangle F q_2 \]

meaning that Player 1 has a strategy such that:

- either the game reaches \( q_2 \) and time diverges,
- or Player 2 is responsible for time convergence.
In [dAFH+03, HP06], an 'alternative semantics' has been proposed:

\[ \mathcal{A} \models_{NZ} \langle A_1 \rangle F q_2 \]

meaning that *Player 1* has a strategy such that:

- either the game reaches \( q_2 \) and time diverges,
- or *Player 2* is responsible for time convergence.

The winning strategy for *Player 1* is \( (\frac{1}{2^n}, c_1) \).
How to combine time and game? (II)

**KS**
Kripke structures

**Time**

**TA**
Timed Automata

**Game**

**TGA**
Timed Game Automata

**CGS**
Concurrent Game Struct.
How to combine time and game? (II)

KS
Kripke structures

Time

TA
Timed Automata

TGA
Timed Game Automata

Game

CGS
Concurrent Game Struct.

DCGS

---
How to combine time and game? (II)

KS
Kripke structures

Time

TA
Timed Automata

TGA
Timed Game Automata

TCGS
Timed Concurrent Game Struct.

CGS
Concurrent Game Struct.

Game

Concurrent Game Struct.
Semantics of TCGS

\[
\langle c_2, c_1' \rangle \\
\langle c_2, c_1' \rangle \quad x \leq 2 \\
\langle c_2, c_2' \rangle \quad x = 1 \\
\langle c_1, c_2' \rangle \\
\langle c_1, c_2' \rangle \\
\langle c_1, c_2' \rangle \\
\langle c_2, c_1' \rangle \\
\langle c_2, c_1' \rangle \quad x \geq 2 \\
\langle c_2, c_2' \rangle \\
\langle c_2, c_2' \rangle \\
\langle c_2, c_1' \rangle \\
\langle c_2, c_1' \rangle \quad x \geq 2
\]
At state \((q_2, 0)\) both players propose a move:

- **Player 1 proposes** \((t_1, f_1)\), where \(f_1 : \mathbb{R}^+ \rightarrow \{c_1, c_2\}\),

- **Player 2 proposes** \((t_2, f_2)\), where \(f_2 : \mathbb{R}^+ \rightarrow \{c_1', c_2'\}\).
At state \((q_2, 0)\) both players propose a move:

- Player 1 proposes \((t_1, f_1)\), where \(f_1 : \mathbb{R}^+ \rightarrow \{c_1, c_2\}\),
- Player 2 proposes \((t_2, f_2)\), where \(f_2 : \mathbb{R}^+ \rightarrow \{c'_1, c'_2\}\).

The next state is determined by:

\[
(q_2, 0, t, \langle f_1(t), f_2(t) \rangle) \xrightarrow{t, \langle f_1(t), f_2(t) \rangle} (q', x') \quad \text{where} \quad t = \min(t_1, t_2).
\]
At state \((q_2, 0)\) both players propose the moves:
Semantics of TCGS

At state \((q_2, 0)\) both players propose the moves:

- Player 1 proposes \(f_1\) with \(c_2\) and \(c_1\)
- Player 2 proposes \(f_2\) with \(c'_2\) and \(c'_1\)

The diagrams show the possible transitions and proposals at each state.
Semantics of TCGS

At state \((q_2, 0)\) both players propose the moves:

\[
\begin{align*}
\langle c_2, c'_1 \rangle \\
x \leq 2
\end{align*}
\]

\[
\begin{align*}
\langle c_1, c'_2 \rangle \\
x \geq 2
\end{align*}
\]

\[
\begin{align*}
\langle c_1, c'_1 \rangle \\
x = 1
\end{align*}
\]
Semantics of TCGS

At state \((q_2, 0)\) both players propose the moves:

\[
(q_2, 0) \rightarrow (q_4, 1)
\]
At state \((q_2, 0)\) both players propose the moves (second try):
Semantics of TCGS

At state \((q_2, 0)\) both players propose the moves (second try):

- For the first player, \(f_1\):
  - \(c_1\) and \(c_2\) are proposed with \(t_1 = 2.5\).

- For the second player, \(f_2\):
  - \(c'_1\) and \(c'_2\) are proposed with \(t_2 = 1\).
At state \((q_2, 0)\) both players propose the moves (second try):

\[ (q_2, 0) \xrightarrow{1,\langle f_1(1), f_2(1)\rangle} \]
At state \((q_2, 0)\) both players propose the moves (second try):

\[ f_1 : c_1, c_2 \rightarrow c_1', c_2', t_1 = 2.5 \]

\[ f_2 : c_1', c_2' \rightarrow c_1, c_2, t_2 = 1 \]
Model checking TATL – overview

**Theorem**

- Model checking TATL over DCGS is EXPTIME-complete.
- Model checking TATL over TGA is EXPTIME-complete.
- Model checking TATL over TCGS is EXPTIME-hard and can be done in EXPSPACE.

See [UPPAAL Tiga](http://www.cs.aau.dk/~adavid/tiga/)!
Outline

1. Models for multi-agents systems
2. Alternating-time temporal Logic
3. Model checking
4. Timed extensions
   a. Timed ATL
   b. Durational CGS
   c. Dense-time games
   d. Analysis of TCGS
A Timed CGS defines an infinite and "alternating" timed transition system.

We want to build a finite abstraction... as the region graph for timed automata.

- in TA: from equivalent states, there exist equivalent runs (i.e. going through the same regions).
A Timed CGS defines an infinite and "alternating" timed transition system.

We want to build a finite abstraction... as the region graph for timed automata.

- in TA: from equivalent states, there exist equivalent runs (i.e. going through the same regions).
- in TCGS: from equivalent states, there exist strategies ensuring equivalent runs.
a Timed Automaton = a Finite Automaton + clocks

a TA $A^t$ defines an infinite timed transition system.
Standard model-checking algorithms cannot be used for $A^t \models \Phi$
Algorithms for timed verification

A Timed Automaton = a Finite Automaton + clocks

A $\mathcal{A}^t$ defines an infinite timed transition system. Standard model-checking algorithms cannot be used for $\mathcal{A}^t \models \Phi$

Region graph [Alur, Courcoubetis, Dill]

- Partition the set of states into a finite number of regions
- Truth value of $\Phi$ is the same for all states in a given region.
- Reduce $\mathcal{A}^t \models \Phi$ to $\mathcal{A}_{finite} \models \Phi'$.

But the number of regions is exponential in the number of clocks and the encoding of constants.
Property Φ: reachability of the control state $q_F$.

Two configurations $(\ell, \nu)$ and $(\ell, \nu')$ have the same behavior (w.r.t. Φ) when

1. Any action transition enabled from $\nu$ is also enabled from $\nu'$; and the target configurations have the same behavior... (and vice versa)

2. For any delay transition $t$ from $\nu$, there is a delay transition $t'$ s.t. $(q, \nu + t)$ an $(q, \nu' + t')$ have the same behavior... (and vice versa)
Property $\Phi$: reachability of the control state $q_F$.

Two configurations $(\ell, v)$ and $(\ell, v')$ have the same behavior (w.r.t. $\Phi$) when

1. Any action transition enabled from $v$ is also enabled from $v'$; and the target configurations have the same behavior... (and vice versa)

2. For any delay transition $t$ from $v$, there is a delay transition $t'$ s.t. $(q, v + t)$ and $(q, v' + t')$ have the same behavior... (and vice versa)

This is a (time abstract) bisimulation!
The region abstraction

\[X = \{x, y\}\]

\[M_x = 3 \text{ and } M_y = 2\]
The region abstraction

$X = \{x, y\}$

$M_x = 3$ and $M_y = 2$

- “compatibility” between regions and constraints $x \sim k$ and $y \sim k$
The region abstraction

• “compatibility” between regions and constraints \( x \sim k \) and \( y \sim k \)

• “compatibility” between regions and time elapsing

\[
X = \{x, y\}
\]

\[
M_x = 3 \text{ and } M_y = 2
\]
The region abstraction

\[ X = \{x, y\} \]
\[ M_x = 3 \text{ and } M_y = 2 \]

- “compatibility” between regions and constraints \( x \sim k \) and \( y \sim k \)
- “compatibility” between regions and time elapsing
The region abstraction

- “compatibility” between regions and constraints $x \sim k$ and $y \sim k$
- “compatibility” between regions and time elapsing

$X = \{x, y\}$
$M_x = 3$ and $M_y = 2$
The region abstraction

\[ X = \{x, y\} \]
\[ M_x = 3 \quad \text{and} \quad M_y = 2 \]

region defined by
\[ l_x = ]1; 2[ , \quad l_y = ]0; 1[ \]
\[ \{x\} < \{y\} \]

- “compatibility” between regions and constraints \( x \sim k \) and \( y \sim k \)
- “compatibility” between regions and time elapsing
The region abstraction

\[ X = \{x, y\} \]
\[ M_x = 3 \text{ and } M_y = 2 \]

region defined by
\[ I_x = ]1; 2[ , \quad I_y = ]0; 1[ \]
\[ \{x\} < \{y\} \]

successor by delay transition:
\[ I_x = ]1; 2[ , \quad I_y = [2; 2] \]

- “compatibility” between regions and constraints \( x \sim k \) and \( y \sim k \)
- “compatibility” between regions and time elapsing
Region graph

\( \mathcal{A} \times R^X_{/\equiv_{\mathcal{A},\Phi}} \) is a standard cartesian product.

Every node contains information on the value of any clock.
An example [AD 90's]
We have to verify that the region abstraction is correct for the strategies in TCGS.

→ considering “regions-based” strategies has to be sufficient.
We have to verify that the region abstraction is correct for the strategies in TCGS.

→ considering “regions-based” strategies has to be sufficient.

What is a “region-based” strategy?

- **region-uniform**: the moves given by the strategy are region-definable, and
- **region-invariant**: the strategy only depends on the projection of the history on regions.
Region-uniform strategies

For any $\rho$, if $\lambda_A(\rho) = (d, f)$, the value of $f$ has to be constant on regions.
Region-uniform strategies

For any $\rho$, if $\lambda_A(\rho) = (d, f)$, the value of $f$ has to be constant on regions.

A region-uniform strategy
Region-uniform strategies

For any $\rho$, if $\lambda_A(\rho) = (d, f)$, the value of $f$ has to be constant on regions.

A NON region-uniform strategy
Region-invariant strategies

From equivalent prefixes, a strategy gives the same sequence of moves...
Region-invariant strategies

From equivalent prefixes, a strategy gives the same sequence of moves...
Region-invariant strategies

From equivalent prefixes, a strategy gives the same sequence of moves...
Region-invariant strategies

From equivalent prefixes, a strategy gives the same sequence of moves. . .
Reduction to ‘simple’ strategies

**Theorem**

Let $\mathcal{T}$ be a TCGS and $\Omega$ be a region-invariant winning objective:

- there exists a winning strategy for $\Omega$
- if and only if
- there exists a *region-uniform and -invariant* winning strategy for $\Omega$. 
Reduction to ‘simple’ strategies

Theorem

Let $\mathcal{T}$ be a TCGS and $\Omega$ be a region-invariant winning objective:

there exists a winning strategy for $\Omega$
if and only if
there exists a *region-uniform and -invariant* winning strategy for $\Omega$. 
Theorem

Let \( \mathcal{T} \) be a TCGS and \( \Omega \) be a region-invariant winning objective:

there exists a winning strategy for \( \Omega \) if and only if
there exists a \textit{region-uniform and -invariant} winning strategy for \( \Omega \).
Reduction to ‘simple’ strategies

Theorem

Let $T$ be a TCGS and $\Omega$ be a region-invariant winning objective:

there exists a winning strategy for $\Omega$
if and only if

there exists a *region-uniform and -invariant* winning strategy for $\Omega$.  

---

$x_2$

$x_1$
Reduction to ‘simple’ strategies

**Theorem**

Let $\mathcal{T}$ be a TCGS and $\Omega$ be a region-invariant winning objective:

there exists a winning strategy for $\Omega$

if and only if

there exists a *region-uniform and -invariant* winning strategy for $\Omega$. 

**Diagram:**

The diagram illustrates a region-invariant strategy $\sigma$. The coordinates $x_1$ and $x_2$ are labeled, and the region-invariant property is indicated by the green and red lines.
Reduction to ‘simple’ strategies

Theorem

Let $T$ be a TCGS and $\Omega$ be a region-invariant winning objective:

there exists a winning strategy for $\Omega$
if and only if
there exists a \textit{region-uniform and -invariant} winning strategy for $\Omega$. 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure.png}
\caption{Graphical representation of region-invariant and region-uniform and -invariant strategies.}
\end{figure}
Region CGS

\[
\langle c_2, c_1' \rangle \\
 x := 0
\]

\[
\langle c_1, c_1' \rangle, \quad \cdots \\
\langle c_1, c_2' \rangle \\
 x \leq 1 \\
\]

\[
\langle c_1, c_2 \rangle \\
 x = 1
\]

\[
q_1 \rightarrow q_2
\]
\begin{align*}
\langle c_2, c'_1 \rangle & \quad \text{\textcopyright} \\
x := 0 & \\
\langle c_1, c'_1 \rangle, & \\
\text{\ldots} & \\
\langle c_1, c'_2 \rangle & \quad \text{\textcopyright} \\
x \leq 1 & \\
x = 1 & \\
\langle q_1, x = 0 \rangle & \quad \text{\textcopyright} \\
\langle q_1, 0 < x < 1 \rangle & \\
\langle q_1, x = 1 \rangle & \\
\langle q_2, x = 1 \rangle & \quad \text{\textcopyright} \\
\langle q_2, x > 1 \rangle & \\
\end{align*}
Region CGS

\[
\langle c_2, c'_1 \rangle \\
x := 0
\]

\[
\langle c_1, c'_1 \rangle, \quad \ldots
\]

\[
x \leq 1
\]

\[
\langle c_1, c'_2 \rangle \\
x = 1
\]

\[
(q_1, x = 0) \quad (q_1, 0 < x < 1) \quad (q_1, x = 1)
\]

\[
(q_2, x = 1) \quad (q_2, x > 1)
\]
Region CGS

A delay = a number of a successor region

\[ \langle c_2, c'_1 \rangle \]
\[ x := 0 \]

\[ \langle c_1, c'_1 \rangle, \ldots \]
\[ x \leq 1 \]

\[ \langle c_1, c'_2 \rangle \]
\[ x = 1 \]

\( (q_1, x = 0) \)
\( (q_1, 0 < x < 1) \)
\( (q_1, x = 1) \)
\( (q_2, x = 1) \)
\( (q_2, x > 1) \)
Region CGS

a delay = a number of a successor region

\[
\langle c_2, c_1' \rangle
\]

\[
x := 0
\]

\[
\langle c_1, c_1' \rangle, \ldots
\]

\[
x \leq 1
\]

\[
\langle c_1, c_2' \rangle
\]

\[
x = 1
\]

\[
(q_1, x = 0)
\]

\[
(q_1, 0 < x < 1)
\]

\[
(q_1, x = 1)
\]

\[
(q_2, x = 1)
\]

\[
(q_2, x > 1)
\]
Region CGS

A delay = a number of a successor region

\[ \langle c_2, c_1' \rangle \]

\[ x := 0 \]

\[ \langle c_1, c_1' \rangle, \langle c_1, c_2' \rangle \]

\[ x \leq 1 \]

\[ x = 1 \]

\[ (q_1, x = 0) \]

\[ (q_1, 0 < x < 1) \]

\[ (q_2, x = 1) \]

\[ (q_1, x = 1) \]

\[ (q_2, x > 1) \]
Region CGS

a delay = a number of a successor region

\[ \langle c_2, c_1' \rangle \]

\[ x := 0 \]

\[ \langle c_1, c_1' \rangle, \langle c_1, c_2' \rangle \]

\[ x \leq 1 \]

\[ x = 1 \]

\[ \langle (2, c_1), (1, c_1') \rangle \]

\[ (q_1, x = 0) \]

\[ (q_1, 0 < x < 1) \]

\[ (q_1, x = 1) \]

\[ \langle (2, c_1), (2, c_2') \rangle, \ldots \]

\[ (q_2, x = 1) \]

\[ (q_2, x > 1) \]
Region CGS

a delay = a number of a successor region

\[ \langle c_2, c'_1 \rangle \]

\[ x := 0 \]

\[ \langle c_1, c'_1 \rangle, \quad \ldots \]

\[ q_1 \]

\[ x \leq 1 \]

\[ \langle c_1, c'_2 \rangle \]

\[ x = 1 \]

\[ q_2 \]

\[ \langle (2, c_1), (1, c'_1) \rangle, \langle (1, c_1), (2, c'_1) \rangle, \ldots \]

\[ (q_1, x = 0) \]

\[ (q_1, 0 < x < 1) \]

\[ (q_1, x = 1) \]

\[ (q_2, x = 1) \]

\[ (q_2, x > 1) \]
Region CGS

a delay = a number of a successor region

\[ \langle c_2, c'_1 \rangle \quad x := 0 \]

\[ \langle c_1, c'_1 \rangle, \ldots \quad x \leq 1 \]

\[ \langle c_1, c'_2 \rangle \quad x = 1 \]

Game bisimulation
Games are useful to describe interaction between processes.

Their analysis is more complex than standard model-checking.

Expressing properties over strategies in game structures is difficult.

ATL is an interesting specification language.

Many extensions are possible (and remain decidable!).

- with real-time constraints,
- with bounds over the resources of the strategies,
- with strategy contexts (to increase the expressive power),
- ...
Alternating-time temporal logic. 

L. de Alfaro, M. Faella, T. Henzinger, R. Majumdar, and M. Stoelinga. 
The element of surprise in timed games. 
In *CONCUR’03*, volume 2761 of *LNCS*, pages 144–158. 

V. Goranko and G. van Drimmelen. 
Complete axiomatization and decidability of alternating-time temporal logic. 

T. Henzinger and V. Prabhu. 
Timed alternating-time temporal logic. 
In *FORMATS’06*, volume 4202 of *LNCS*, pages 1–17. 

W. Jamroga and J. Dix.
Do agents make model checking explode (computationally)?


M. Jurdziński, F. Laroussinie, and J. Sproston.
Model checking probabilistic timed automata with one or two clocks.


Model checking timed ATL for durational concurrent game structures.

In E. Asarin and P. Bouyer, editors, *Proceedings of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *Lecture
On the expressiveness and complexity of ATL.

On the expressiveness and complexity of ATL.

P.-Y. Schobbens.
Alternating-time logic with imperfect recall.