Reasoning about Natural Strategic Ability

Vadim Malvone

Université d’Evry

FMAI 2019

Joint work with Wojtek Jamroga and Aniello Murano
System Correctness

- A very important problem in critical systems:
  - Safety: errors can cost lives (e.g. Therac-25).
  - Mission: errors can cost in terms of objectives (e.g. Arianne 5).
  - Business: failure can cost in loss of money (e.g. Denver airport).
- In such systems failure is not an option.
System Correctness

- A very important problem in critical systems:
  - Safety: errors can cost lives (e.g. Therac-25).
  - Mission: errors can cost in terms of objectives (e.g. Arianne 5).
  - Business: failure can cost in loss of money (e.g. Denver airport).
- In such systems failure is not an option.

Model checking: $M \models \varphi$

There are three fundamental parts:

- $M$: modeling a system;
- $\varphi$: specifying a property;
- $\models$: verifying that the model $M$ satisfies the property $\varphi$. 
Multi-agent systems

- There are many agents (players) interacting among them.
- Each agent has a set of strategies.
- A strategy is a conditional plan that at each step of the game prescribes an action.
- The composition of strategies, one for each player, induces an unique computation.
Model

A concurrent game structure is a tuple $M =< Ag, AP, St, s_I, Ac, \pi, tr >$:

- $Ag$ is a set of agents (or players);
- $AP$ is a set of atomic propositions;
- $St$ is a set of states;
- $s_I \in S$ is a designated initial state;
- $Ac$ is a set of actions;
- $\pi$ is a labelling function;
- $tr$ is a transition function.
Preface (3)

<table>
<thead>
<tr>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>A concurrent game structure is a tuple $M = \langle Ag, AP, St, s_I, Ac, \pi, tr \rangle$:</td>
</tr>
<tr>
<td>• $Ag$ is a set of agents (or players);</td>
</tr>
<tr>
<td>• $AP$ is a set of atomic propositions;</td>
</tr>
<tr>
<td>• $St$ is a set of states;</td>
</tr>
<tr>
<td>• $s_I \in S$ is a designated initial state;</td>
</tr>
<tr>
<td>• $Ac$ is a set of actions;</td>
</tr>
<tr>
<td>• $\pi$ is a labelling function;</td>
</tr>
<tr>
<td>• $tr$ is a transition function.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logics for the strategic reasoning such as ATL and Strategic Logic.</td>
</tr>
</tbody>
</table>
Strategies

- Depending on the memory, we distinguish between:
  - *memoryless strategies* $\Rightarrow \sigma : St \rightarrow Ac$;
  - *bounded strategies* $\Rightarrow \sigma : St^{<g} \rightarrow Ac$;
  - *memoryfull strategies* $\Rightarrow \sigma : St^{+} \rightarrow Ac$.

- In the memoryless case, the players take a decision by considering the actual state of the game.
- In the bounded case, the players take a decision by considering a partial history of the game.
- In the memoryfull case, the players take a decision by considering the full history of the game.
Strategies are mathematical creatures $\implies$ functions from system states to actions.

- This makes sense for robots or programs, but not for humans!

- Strategies for humans should be simple in order for the person to understand it, memorize it, and execute it.
Natural Strategies [JMM17]

A natural memoryless strategy $s_a$ for agent $a$ is a list of condition-action rules

$$(\text{cond}, \text{act})$$

such that:

- $\text{cond}$ is a boolean combination of propositions,
- $\text{act}$ is an available action in every state $q \models \text{cond}$,
- the last pair on the list is $(\top, \text{idle})$.

Reasoning about natural strategic ability.
Consider the following strategy for *buying a train ticket*:

1. \((\neg \text{ticket} \land \neg \text{selected}, \text{select})\);
2. \((\neg \text{ticket} \land \text{selected}, \text{pay})\);
3. \((\top, \text{idle})\).
**Natural Strategies: Complexity**

The complexity of strategy $s_a$ ($\text{compl}(s_a)$) can be defined by:

- Number of used propositions $\Rightarrow |\text{dom}(s_a)|$;
- Largest condition $\Rightarrow \max\{|\phi| \mid (\phi, \alpha) \in s_a\}$;
- Total size of the representation $\Rightarrow \sum_{(\phi, \alpha) \in s_a} |\phi|$.
**Reasoning about Natural Ability: NatATL**

### Syntax

A formula in NatATL is defined as:

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle\langle A \rangle\rangle \leq^k X \varphi \mid \langle\langle A \rangle\rangle \leq^k U \varphi \mid \langle\langle A \rangle\rangle \leq^k W \varphi. \]

where \( p \in AP \), \( k \in \mathbb{N} \), and \( A \) is a set of agents.
REASONING ABOUT NATURAL ABILITY: NatATL

Syntax

A formula in NatATL is defined as:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle \leq^k X \varphi \mid \langle A \rangle \leq^k \varphi U \varphi \mid \langle A \rangle \leq^k \varphi W \varphi.$$  

where $p \in AP$, $k \in \mathbb{N}$, and $A$ is a set of agents.

Semantics

$M, q \models \langle A \rangle \leq^k \gamma$ iff there is a natural strategy $s_A$ such that $\text{compl}(s_A) \leq k$, and for each path $\lambda \in \text{out}(q, s_A)$ we have $M, \lambda \models \gamma$. 
Reasoning about usability, example: *ticket vending machine*

- It is not enough that a customer has a strategy to buy the ticket \( \langle c \rangle F\text{buy} \).
- If the strategy is too complex, people won’t use it anyway.
- Instead, we should require \( \langle c \rangle \leq^k F\text{buy} \) for a reasonably low \( k \).
**What’s the Use?**

Reasoning about usability, example: *ticket vending machine*

- It is not enough that a customer has a strategy to buy the ticket ($\langle\langle c \rangle\rangle F_{buy}$).
- If the strategy is too complex, people won’t use it anyway.
- Instead, we should require $\langle\langle c \rangle\rangle \leq^k F_{buy}$ for a reasonably low $k$.

Gaming

- The designer can define the *game level* by the *complexity of the smallest winning strategy* for the player.
- Formally, the level $k$ iff $\langle\langle a \rangle\rangle \leq^k F_{win} \land \neg \langle\langle a \rangle\rangle \leq^{k-1} F_{win}$.
NATURAL STRATEGIES WITH RECALL

• Similar to memoryless strategies, but the conditions are given by regular expressions over Boolean formulas.

• Example: a strategy for a Wild West explorer:

1. (safe*, digGold);
2. (safe* · (¬safe ∧ haveGun), shoot);
3. (safe* · (¬safe ∧ ¬haveGun), run);
4. (⊤* · (¬safe) · (¬safe), hide);
5. (⊤*, idle).
RELATIONSHIPS BETWEEN TYPES OF NATURAL STRATEGIES

Theorem

The following results hold in NatATL:

1. For all $M, q$, and all formulas $\varphi = \langle A \rangle \leq^k \gamma$, it holds that:
   
   $M, q \models r \varphi$ implies $M, q \models R \varphi$

2. There exist $M, q$, and a formula $\varphi = \langle A \rangle \leq^k \gamma$, such that:
   
   $M, q \models R \varphi$ and not $M, q \models r \varphi$

$r = \text{strategies without recall (memoryless) and } R = \text{strategies with recall.}$
Example: Soccer scenario (1)

- The attacker is running towards the goal with the ball.
- The goalkeeper can either stay close to the goal line or move towards the attacker.
- Then, after one more step, the attacker can either shoot straight or lob the ball over the goalkeeper.
**Example: Soccer Scenario (2)**

A strategy with recall for the attacker to score the goal can be:

1. \( (\text{init}, \ run) \);
2. \( (\text{init} \cdot (\text{moved} \lor \text{stayed}), \ run) \);
3. \( (\top^* \cdot \text{moved} \cdot \top, \ lob) \);
4. \( (\top^* \cdot \text{stayed} \cdot \top, \ shoot) \);
5. \( (\top^*, \ idle) \).

The complexity of the strategy is 22.
**Example: Soccer scenario (3)**

- Then, $\varphi = \langle 1 \rangle \leq 22 F_{\text{goal}}$ is true for strategies with recall.
- On the other hand, $\varphi$ is false for memoryless strategies.
- In fact, the formula is false for any bound $k$.
- To see that, recall that conditions in natural memoryless strategies can only refer to boolean properties of the current state.
- Then, it is impossible to define two different behaviors in states $q_3$ and $q_4$ within a natural memoryless strategy.
Model checking $NatATL_r$

- $P$ for fixed or bounded $k$;
- $P^{NP} = \Delta^P_2$-complete when $k$ is a parameter of the problem.
**Verification of Natural Strategies**

**Model checking $\text{NatATL}_r$**
- $\text{P}$ for fixed or bounded $k$;
- $\text{P}^{\text{NP}} = \Delta^\text{P}_2$-complete when $k$ is a parameter of the problem.

**Model checking $\text{NatATL}_R$**
- $\Delta^\text{P}_2$ for fixed or bounded $k$;
- $\text{PSPACE}$ when $k$ is a parameter of the problem.
CONCURRENT GAME WITH OBJECTIVES [JMM19]

A concurrent game is a tuple $G = (M, q_0, \Phi)$, where:

- $M$ is a concurrent game structure,
- $q_0 \in St$ is a state in $M$,
- $\Phi : Ag \rightarrow \mathcal{L}_{LTL}$ assigns each agent with an $LTL$ formula.

Natural strategic ability.
*Artificial Intelligence (AI)*, (to appear).
**Decision problems: Surely Winning (1)**

**Definition**

Given a concurrent game $G$, a subset of agents $A \subseteq Ag$, a natural number $k \in \mathbb{N}$, and a natural collective strategy $s_A$ of $A$, we say that:

$$s_A \text{ is surely winning in } G \iff \forall \lambda \in out(q_0, s_A) \text{ and } a \in A: \lambda \models \Phi_a$$

Moreover, coalition $A$ surely wins in $G$ under bound $k$ iff it has a sure winning strategy of size at most $k$. 

Vadim Malvone
Reasoning about Natural Strategic Ability
**Decision problems: Surely Winning (2)**

**Algorithm** *SureWin*(G, A, k):

1. $s_A = \text{GuessStrat}(G, A, k)$;
2. Prune $M$ according to $s_A$, obtaining model $M'$;
3. return $m\text{Check}_{\text{CTL}^*}(M', q_0, A \land_{i \in A} \Phi_i)$;

Hint for lower bound

We show a reduction from model checking LTL.

Complexity *SureWin* is $\text{PSPACE}$-complete.
Algorithm $\text{SureWin}(G, A, k)$:

\[
\begin{align*}
  s_A &= \text{GuessStrat}(G, A, k); \\
  \text{Prune } M \text{ according to } s_A, \text{ obtaining model } M'; \\
  \text{return } m\text{Check}_{\text{CTL}^*}(M', q_0, A \land_{i \in A} \Phi_i);
\end{align*}
\]

Hint for lower bound

We show a reduction from model checking LTL.
**Algorithm** *SureWin*(G, A, k):

\[ s_A = \text{GuessStrat}(G, A, k); \]

Prune \( M \) according to \( s_A \), obtaining model \( M' \);

return \( mCheck_{\text{CTL}^*}(M', q_0, A \land_{i \in A} \Phi_i) \);

**Hint for lower bound**

We show a reduction from model checking LTL.

**Complexity**

*SureWin* is *PSPACE*-complete.
## Decision Problems: Nash Equilibrium (1)

**Definition**

Given a concurrent game $G$ and a profile $s_{Ag} = (s_1, \ldots, s_i, \ldots, s_{|Ag|})$ of natural strategies under bound $k \in \mathbb{N}$:

$s_{Ag}$ is a *Nash Equilibrium* in $G \iff \forall i \in Ag, s_i$ is a best response.
**Decision problems: Nash Equilibrium (1)**

**Definition**

Given a concurrent game $G$ and a profile $s_{Ag} = (s_1, \ldots, s_i, \ldots, s_{|Ag|})$ of natural strategies under bound $k \in \mathbb{N}$:

$s_{Ag}$ is a Nash Equilibrium in $G$ $\iff \forall i \in Ag$, $s_i$ is a best response.

**Best response**

Given $G$, a player $i$, and a profile $s_{Ag} = (s_1, \ldots, s_i, \ldots, s_{|Ag|})$ under bound $k \in \mathbb{N}$, $s_i$ is a best response in $s_{Ag}$ if and only if:

$$\text{path}(s_{Ag}) \not\models \Phi_i \Rightarrow \text{path}((s_1, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_{|Ag|})) \not\models \Phi_i$$

for all $s'_i \in \Sigma'_i$ such that $\text{compl}(s'_i) \leq k$. 
**Decision problems: Nash Equilibrium (2)**

**Algorithm** `IsNotNash(G, s_Ag, k):

for every $i \in Ag$ do
  if $\text{path}(s_Ag) \not\models \Phi_i$ then
    Guess $s'_i$ with $\text{compl}(s'_i) \leq k$;
    if $\text{path}((s_1, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_{|Asg|})) \models \Phi_i$ then return (true);
return (false);
**Algorithm** $\text{IsNotNash}(G, s_{Ag}, k)$:

for every $i \in Ag$ do
  if $\text{path}(s_{Ag}) \not\models \Phi_i$ then
    Guess $s_i'$ with $\text{compl}(s_i') \leq k$;
    if $\text{path}((s_1, \ldots, s_{i-1}, s_i', s_{i+1}, \ldots, s_{|Ag|})) \models \Phi_i$ then return (true);
return (false);

**Hint for lower bound**

We use a reduction from $\text{SAT}$.
Algorithm $\text{IsNotNash}(G, s_{Ag}, k)$:

for every $i \in Ag$ do
  if $\text{path}(s_{Ag}) \not\models \Phi_i$ then
    Guess $s'_i$ with $\text{compl}(s'_i) \leq k$;
    if $\text{path}((s_1, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_{|Ag|})) \models \Phi_i$ then return (true);
  return (false);

Hint for lower bound
We use a reduction from $\text{SAT}$.

Complexity
$\text{IsNotNash}$ is $\textbf{NP}$-complete $\Rightarrow \text{IsNash}$ is $\textbf{coNP}$-complete.
CONCLUSIONS

• We proposed the concept of natural strategies, based on an intuitive representation of conditional plans.
• We proposed how to measure the complexity of such strategies.
• We defined NatATL, a variant of alternating-time temporal logic to reason about natural strategic ability.
• We studied the complexity of NatATL model checking.
• We considered two main cases here: memoryless strategies and strategies with recall of the past.
• We showed that the relationship between natural strategies with recall and memoryless is more intricate than normally in ATL.
• We investigated some decision problems for natural abilities of agents in concurrent games with LTL winning conditions.