

Reasoning about Natural Strategic Ability

Vadim Malvone

Université d'Evry

FMAI 2019

Joint work with Wojtek Jamroga and Aniello Murano

PREFACE (1)

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. Arienne 5).
 - Business: failure can cost in loss of money (e.g. Denver airport).
- In such systems failure is not an option.

PREFACE (1)

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. Ariane 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;
- φ : specifying a property;
- \models : verifying that the model M satisfies the property φ .

PREFACE (2)

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.

PREFACE (3)

Model

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

- 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;
- π is a labelling function;
- tr is a transition function.

PREFACE (3)

Model

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

- 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;
- π is a labelling function;
- tr is a transition function.

Specification

Logics for the strategic reasoning such as ATL and Strategic Logic.

PREFACE (4)

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.

BETWEEN MATHEMATICS AND REAL LIFE

- 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*

$(cond, act)$

such that:

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



[JMM17] W. Jamroga, V. Malvone, and A. Murano.

Reasoning about natural strategic ability.

In *AAMAS*, pages 714–722, 2017.

NATURAL STRATEGIES: EXAMPLE

Consider the following strategy for *buying a train ticket*:

- ① $(\neg \text{ticket} \wedge \neg \text{selected}, \text{select});$
- ② $(\neg \text{ticket} \wedge \text{selected}, \text{pay});$
- 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:

$$\varphi ::= \mathbf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle^{\leq k} X\varphi \mid \langle\langle A \rangle\rangle^{\leq k} \varphi U\varphi \mid \langle\langle A \rangle\rangle^{\leq k} \varphi W\varphi.$$

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

REASONING ABOUT NATURAL ABILITY: NATATL

Syntax

A formula in **NatATL** is defined as:

$$\varphi ::= \mathbf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle^{\leq k} X\varphi \mid \langle\langle A \rangle\rangle^{\leq k} \varphi U \varphi \mid \langle\langle A \rangle\rangle^{\leq k} \varphi W \varphi.$$

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

Semantics

$M, q \models \langle\langle A \rangle\rangle^{\leq k} \gamma$ 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 $M, \lambda \models \gamma$.

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 ($\langle\langle c \rangle\rangle F\text{buy}$).
- If the strategy is too complex, people won't use it anyway.
- Instead, we should require $\langle\langle c \rangle\rangle^{\leq k} F\text{buy}$ 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 ($\langle\langle c \rangle\rangle F\text{buy}$).
- If the strategy is too complex, people won't use it anyway.
- Instead, we should require $\langle\langle c \rangle\rangle^{\leq k} F\text{buy}$ 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 $\langle\langle a \rangle\rangle^{\leq k} F\text{win} \wedge \neg \langle\langle a \rangle\rangle^{\leq k-1} F\text{win}$.

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 $(\text{safe}^*, \text{digGold});$
 - ② $(\text{safe}^* \cdot (\neg \text{safe} \wedge \text{haveGun}), \text{shoot});$
 - ③ $(\text{safe}^* \cdot (\neg \text{safe} \wedge \neg \text{haveGun}), \text{run});$
 - ④ $(\top^* \cdot (\neg \text{safe}) \cdot (\neg \text{safe}), \text{hide});$
 - ⑤ $(\top^*, \text{idle}).$

RELATIONSHIPS BETWEEN TYPES OF NATURAL STRATEGIES

Theorem

The following results hold in *NatATL*:

- ❶ For all M, q , and all formulas $\varphi = \langle\langle A \rangle\rangle^{\leq k}_{\gamma}$, it holds that:

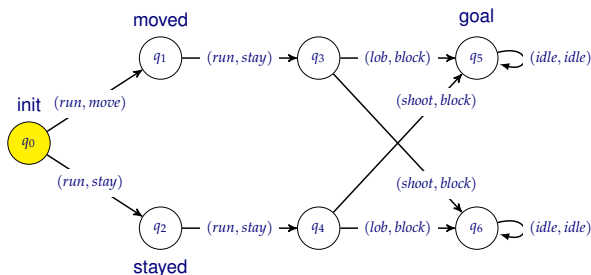
$$M, q \models_r \varphi \text{ implies } M, q \models_R \varphi$$

- ❷ There exist M, q , and a formula $\varphi = \langle\langle A \rangle\rangle^{\leq k}_{\gamma}$, such that:

$$M, q \models_R \varphi \text{ and not } M, q \models_r \varphi$$

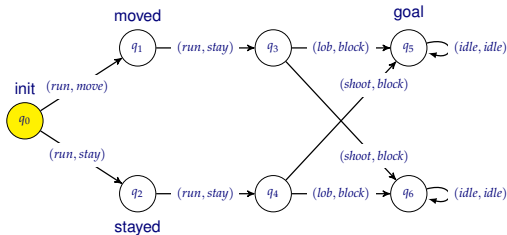
**r* = strategies without recall (memoryless) and *R* = strategies with recall.

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)

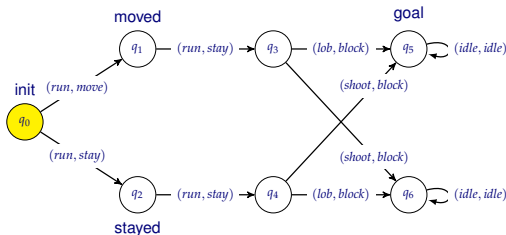


A strategy with recall for the attacker to score the goal can be:

- ❶ $(\text{init}, \text{run});$
- ❷ $(\text{init} \cdot (\text{moved} \vee \text{stayed}), \text{run});$
- ❸ $(\top^* \cdot \text{moved} \cdot \top, \text{lob});$
- ❹ $(\top^* \cdot \text{stayed} \cdot \top, \text{shoot});$
- ❺ $(\top^*, \text{idle}).$

The complexity of the strategy is 22.

EXAMPLE: SOCCER SCENARIO (3)



- Then, $\varphi = \langle\langle 1 \rangle\rangle^{\leq 22} F \text{goal}$ is true for strategies with recall.
- On the other hand, φ 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.

VERIFICATION OF NATURAL STRATEGIES

Model checking $NatATL_r$

- \mathbf{P} for fixed or bounded k ;
- $\mathbf{P}^{\mathbf{NP}} = \Delta_2^{\mathbf{P}}$ -complete when k is a parameter of the problem.

VERIFICATION OF NATURAL STRATEGIES

Model checking $NatATL_r$

- P for fixed or bounded k ;
- $P^{NP} = \Delta_2^P$ -complete when k is a parameter of the problem.

Model checking $NatATL_R$

- Δ_2^P for fixed or bounded k ;
- $PSPACE$ 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.



[JMM19] W. Jamroga, V. Malvone, and A. Murano.

Natural strategic ability.

Artificial Intelligence (AIJ), (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 is *surely winning* in $G \Leftrightarrow \forall \lambda \in out(q_0, s_A)$ 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 .

DECISION PROBLEMS: SURELY WINNING (2)

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

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

Prune M according to s_A , obtaining model M' ;

return $mCheck_{CTL^*}(M', q_0, \mathbf{A} \bigwedge_{i \in A} \Phi_i);$

DECISION PROBLEMS: SURELY WINNING (2)

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

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

Prune M according to s_A , obtaining model M' ;

return $mCheck_{CTL^*}(M', q_0, \mathbf{A} \bigwedge_{i \in A} \Phi_i);$

Hint for lower bound

We show a reduction from model checking LTL.

DECISION PROBLEMS: SURELY WINNING (2)

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

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

Prune M according to s_A , obtaining model M' ;

return $mCheck_{CTL^*}(M', q_0, \mathbf{A} \bigwedge_{i \in A} \Phi_i);$

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 $s_{Ag} = (s_1, \dots, s_i, \dots, s_{|Ag|})$ of natural strategies under bound $k \in \mathbb{N}$:

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

DECISION PROBLEMS: NASH EQUILIBRIUM (1)

Definition

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

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

Best response

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

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

for all $s'_i \in \Sigma_i^r$ such that $compl(s'_i) \leq k$.

DECISION PROBLEMS: NASH EQUILIBRIUM (2)

Algorithm *IsNotNash*(G, s_{Ag}, k):

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

DECISION PROBLEMS: NASH EQUILIBRIUM (2)

Algorithm *IsNotNash*(G, s_{Ag}, k):

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

Hint for lower bound

We use a reduction from *SAT*.

DECISION PROBLEMS: NASH EQUILIBRIUM (2)

Algorithm $IsNotNash(G, s_{Ag}, k)$:

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

Hint for lower bound

We use a reduction from *SAT*.

Complexity

$IsNotNash$ is **NP-complete** \Rightarrow $IsNash$ is **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.