Reasoning about Strategies

Aniello Murano

University of Naples Federico II

SPIRIT 2023

Aim

Idea

Looking for a powerful logic in which one can talk explicitly about the strategic behavior of agents in generic multi-player concurrent games.

Application

It can be used as a specification language for the formal verification and synthesis of modular and interactive systems.

From monolithic to multi-agent systems

Historical development(1)

- Model checking: analyzes systems monolithically (system components plus environment) [Clarke & Emerson, Queille & Sifakis, '81].

$$M \models \varphi$$

From monolithic to multi-agent systems

Historical development(2)

- Module checking: separates the environment from the system components, i.e., two-player game between system and environment [Kupferman & Vardi,'96-01].

$$M \models_r \varphi$$

Investigated under perfect/imperfect information, hierarchical, infinite-state systems (pushdown, real-time), backwards modalities, graded modalities....

From monolithic to multi-agent systems

Historical development(3)

- Alternating temporal reasoning: multi-agent systems (components individually considered), playing strategically [Alur et al.,'97-02].

Alternating-time Temporal Logic [Alur et al., ’02]

ATL*

Branching-time Temporal Logic with the strategic modalities $\langle\langle A \rangle\rangle$ and $[[A]]$.

ATL

Fragment of ATL where temporal operators immediately follow strategic modality.

$\langle\langle A \rangle\rangle \psi$: There is a strategy for the agents in A enforcing the property ψ, independently of what the agents not in A can do.

Example

$\langle\langle \{\alpha, \beta\} \rangle\rangle \mathrm{G} \neg \mathit{fail}$: “Agents α and β cooperate to ensure that a system (having possibly more than two processes (agents)) never enters a fail state”.

- Strategies are treated only implicitly.
- Quantifier alternation fixed to 1.

Strategic Logic

Strategy Logic (SL), was introduced as a more general framework (both in its syntax and semantics), for explicit reasoning about strategies in multi-player concurrent games.

Outline

1. Strategy Logic
   - Syntax and semantics
   - Interesting examples
2. Fragments of Strategy Logic
   - A high level picture
   - Semi-prenex fragments
3. Behavioral games
   - Strategy quantification
4. Imperfect Information
5. At the end ...

Concurrent game model

CGS

A *concurrent game structure* is a tuple $\mathcal{G} = \langle \mathrm{AP}, \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \lambda, \tau, s_0 \rangle$.

Intuitively

$\mathcal{G}$ is a Graph whose States St are labeled with Atomic Propositions *AP* and Transitions τ are Agents' Decision, i.e., Actions Ac taken by Agents Ag.

Strategy and Play

A *perfect recall strategy* is a function that maps each *history* of the game to an *action*.
A *memoryless strategy* is a function that maps each *state* of the game to an *action*.
A play is a path of the game determined by the history of strategies.

Syntax and semantics of SL

SL syntactically extends LTL by means of *strategy quantifiers*, the existential $\langle\langle x \rangle\rangle$ and the universal $[[x]]$, and *agent binding* (a, x).

Sintax of SL

SL *formulas* are built as follows way, where x is a variable and a an agent.

$$\varphi ::= \text{LTL} \mid \langle\langle x \rangle\rangle \varphi \mid [[x]] \varphi \mid (a, x) \varphi.$$

Semantics of SL

- $\langle\langle x \rangle\rangle \varphi$ (also write $\exists x.\varphi$): *"there exists a strategy x for which φ is true"*.
- $[[x]] \varphi$ (also write $\forall x.\varphi$): *"for all strategies x, it holds that φ is true"*.
- $(a, x)\varphi$: *"φ holds, when the agent a uses the strategy x"*.

Failure is not an option

Example (No failure property)

"In a system $\mathcal{S}$ built on three processes, α, β, and γ, the first two have to cooperate in order to ensure that $\mathcal{S}$ never enters a failure state".

Three different formalization in SL.

- $\langle\langle x\rangle\rangle\langle\langle y\rangle\rangle[[z]](\alpha, x)(\beta, y)(\gamma, z)(\mathsf{G}\neg \mathit{fail})$: α and β have two strategies, x and y, respectively, that, independently of what γ decides, ensure that a failure state is never reached.
- $\langle\langle x\rangle\rangle[[z]]\langle\langle y\rangle\rangle(\alpha, x)(\beta, y)(\gamma, z)(\mathsf{G}\neg \mathit{fail})$: β can choose his strategy y dependently of that one chosen by γ.
- $\langle\langle x\rangle\rangle[[z]](\alpha, x)(\beta, x)(\gamma, z)(\mathsf{G}\neg \mathit{fail})$: α and β have a common strategy x to ensure the required property.

Multi-player Nash equilibrium

Example (Nash equilibrium)

Let $\mathcal{G}$ be a game with the n agents $\alpha_1, \ldots, \alpha_n$, each one having its own LTL goal $\psi_1, \ldots, \psi_n$. We want to know if $\mathcal{G}$ admits a Nash equilibrium, i.e., if there is a "best" strategy x_i w.r.t. the goal ψ_i, for each agent α_i, once all other strategies are fixed.

$$\varphi_{NE} \triangleq \langle\langle x_1 \rangle\rangle \cdots \langle\langle x_n \rangle\rangle (\alpha_1, x_1) \cdots (\alpha_n, x_n) (\textstyle\bigwedge_{i=1}^{n} (\langle\langle y \rangle\rangle (\alpha_i, y) \psi_i) \rightarrow \psi_i).$$

Intuitively, if $\mathcal{G} \models \varphi_{NE}$ then $x_1, \ldots, x_n$ form a Nash equilibrium, since, when an agent α_i has a strategy y that allows the satisfaction of ψ_i, he can use x_i instead of y, assuming that the remaining agents $\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n$ use $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$.

Expressiveness

Theorem

SL *is* *strictly more expressive* *than* ATL*.

Explanation

- Unbounded quantifier alternation.
- More than one temporal goal at a time.
- Agents can be forced to share the same strategy.

A comparison

Expressiveness

SL is more expressive than ATL*.

Computational complexities

	ATL*	SL
Model checking	2EXPTIME-COMPLETE	"NONELEMENTARY-COMPLETE"
Satisfiability	2EXPTIME-COMPLETE	Undecidable

A comparison

Expressiveness

SL is more expressive than ATL*.

Computational complexities

	ATL*	SL
Model checking	2EXPTIME-COMPLETE	"NONELEMENTARY-COMPLETE"
Satisfiability	2EXPTIME-COMPLETE	Undecidable

How to get tractable fragments of SL?

A possible solution

The idea

Introduce syntactic restrictions of SL in order to characterize its "degree of freedom" with respect to ATL*.

Fragments

A chain of fragments was introduced, including SL[BG] and SL[1G].

Goals

Intuitively, a goal is a sequence of bindings $\flat$ followed by an LTL formula.
So, SL[1G] contains formulas of the kind $\wp\flat\varphi$ where $\wp$ is a prefix of quantified strategies.

Expressiveness

The expressiveness chain

$$\text{ATL}^* < \text{SL[1G]} < \text{SL[BG]} \leq \text{SL}$$

An overview

	Model checking	Satisfiability
SL	NONELEMENTARY-COMPLETE	Undecidable
SL[BG] SL[1G]	NONELEMENTARY-COMPLETE 2EXPTIME-COMPLETE	Undecidable 2EXPTIME-COMPLETE
ATL*	2EXPTIME-COMPLETE	2EXPTIME-COMPLETE

Why is SL hard?

The choice of an action made by an agent in a strategy, for a given history of the game, may depend on other strategies, i.e., on the actions for each possible history of the game.

Model checking SL

Some good news

- The model checking for SL[BG] is non-elementary in the alternation depth + 1 to deal with LTL and the SL[BG] formula expressing Nash Equilibrium has alternation depth equal to 1. Therefore Nash Equilibrium can be checked in 2EXPTIME.
- For a fixed size LTL formula, Nash Equilibrium can be checked in EXPTIME.
- The complexity of SL[BG] model checking w.r.t. the size of the model is in PTIME.

Imperfect Information

Problem

- A CGS describes a *perfect information* game.
- That is not the case in many strategic scenarios (e.g. battleship).

Imperfect Information (II)

- Indistinguishable states: $q \sim_a s$
- Knowledge operators: $K_a \varphi$
- Example
  - q represents a state in which it is raining
  - p represents a state in which it *not* is raining
  - $q \sim_{Ann} s$ means that Ann *sees* the same information in both states
  - Formula K_{Ann}rains: does Ann know that it is raining?
  - Formula $K_{Bob}K_{Ann}$rains: does Bob know that Ann knows it is raining?

Uniform strategies

- Agents take the same action in states that they cannot distinguish
- Example
  - Ann can have a strategy of taking the umbrella whenever she considers it possible that it is raining
  - Bob can have a strategy to warn Ann whenever he knows she does not know that it is raining

Imperfect Information and Knowledge with SL

- CGS with II: CGS augmented with indistinguishable relations for each agent
- SLK: SL extended with knowledge operators
- Semantics based on uniform strategies

Model checking

Model checking SLK formulas is undecidable in general

Known decidable cases

- Memoryless strategies
- Bounded memory strategies
- Hierarchical information
- Public actions

Theorem

Model checking SLK with memoryless strategies is PSPACE-complete.

Conclusion

- We have introduced SL as a logic for the temporal description of multi-player concurrent games, in which strategies are treated as first order objects.
- SL model checking has a NONELEMENTARYTIME-COMPLETE formula complexity.
- SL satisfiability is undecidable.
- Known SL tractable fragments that maintain expressivity
- SL has also been extended to deal with knowledge and imperfect information

Main references

- F. Mogavero, A. Murano, G. Perelli, M. Vardi: Reasoning About Strategies: On the Model-Checking Problem. TOCL 2014
- F. Mogavero, A. Murano, G. Perelli, M. Vardi: Reasoning about Strategies: on the Satisfiability Problem. LMCS 2017
- P. Bouyer, O. Kupferman, N. Markey, B. Maubert, A. Murano, G. Perelli: Reasoning about Quality and Fuzziness of Strategic Behaviors. TOCL 2023
- B. Aminof, M. Kwiatkowska, B. Maubert, A. Murano, S. Rubin: Probabilistic Strategy Logic. IJCAI 2019.
- P. CermÃ¡k, A. Lomuscio, A. Murano: Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications. AAAI 2015
- P. Cermak, A. Lomuscio, F. Mogavero, A. Murano: MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. CAV 2014
- F. Belardinelli, A. Lomuscio, A. Murano, S. Rubin: Verification of multi-agent systems with public actions against strategy logic. Artif. Intell. 2020
- R. Berthon, B. Maubert, A. Murano, S. Rubin, M. Vardi: Strategy Logic with Imperfect Information. TOCL 2021