On the Boundary of Behavioral Strategies

Fabio Mogavero, **Aniello Murano**, and Luigi Sauro

Università degli Studi di Napoli Federico II

Highlights of Logic, Games and Automata

Paris, September 19-21, 2013

Strategy Logic[1]

Strategy Logic (SL) has been recently introduced as a powerful formalism to reason about the strategic behavior of agents in multi-player concurrent games. In SL one can reason explicitly about strategies as *first order objects*.

SL strictly extends the well known Alternating-time Temporal Logic ATL*. In SL, it is possible to express several *solution concepts* like Nash, resilient, secure equilibria, dominant strategies, etc.

[1] Mogavero, Murano, Vardi. *Reasoning about Strategies.* FSTTCS 2010

Non-Behavioral Strategies

There is a price to pay for this high expressiveness: SL semantics admits non behavioral strategies, i.e., a choice of an agent, at a given moment of a play, may depend on the choices another agent can make in another counterfactual play.

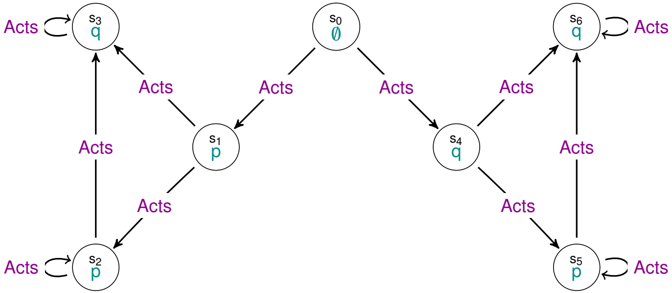
Consequently, strategies cannot be synthesized in practice, since the adversary moves may be unpredictable.

Our Contribution

We introduce and study two maximal fragments of Strategy Logic having a *behavioral semantics*, i.e., all strategies involved in the reasonings are *synthesizable*.

Strategy Logic

Underlying Framework: Concurrent Game Structures



A Concurrent Game Structure is a graph in which states are labeled by Atomic Propositions and edges are labeled by Actions that Agents can take (i.e., Decisions).

A strategy maps histories of the game into actions. Plays are paths determined by strategies.

The logic ATL*

Alternating-time Temporal Logic [Alur, Henzinger and Kupferman, 2002]

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

Observe

In ATL* we have

- Implicit strategies.
- One alternation of quantification.

Strategy Logic [Mogavero, Murano, Vardi, 2010]

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).

Syntax

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

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

Informal semantics

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

Example: Failure is not an option

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, which ensure that a failure state is never reached, no matter what γ decides.
- $\langle\langle x\rangle\rangle[[z]]\langle\langle y\rangle\rangle(\alpha, x)(\beta, y)(\gamma, z)(\mathsf{G}\,\neg \mathit{fail})$: 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.

Expressiveness and Model Checking results

Theorem

SL *is strictly more expressive than* ATL*.

- Unbounded quantifier alternation.
- Reuse of a strategy in different contexts.
- Agents can share strategies.

Theorem

SL *model-checking problem has a "NonElementary-Complete" formula complexity and a* PTIME-COMPLETE *data complexity.*

On the contrary, the subsumed ATL* has a model-checking problem with a 2EXPTIME-COMPLETE formula complexity.

A Natural Question

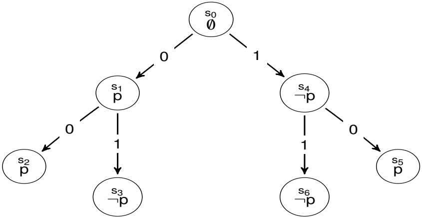
The Question

Why is SL so hard?

The Answer

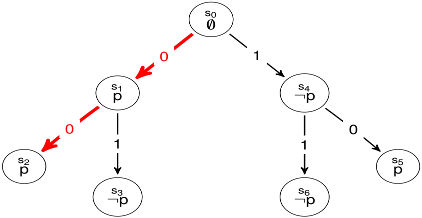
The semantics of SL admits non-behavioral strategies: The choice of an action made by an agent in a strategy, for a given history of the game, may depend on choices over counterfactual possible histories.

Counterfactual Dependence



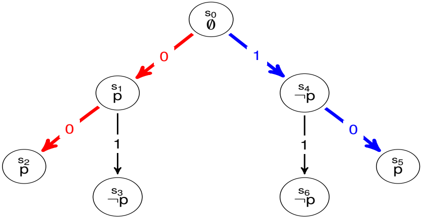
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x)\mathsf{X}\,p \leftrightarrow (\alpha, y)\mathsf{X}\,\neg p$
- $\psi_2 = (\alpha, x)\mathsf{X}\,\mathsf{X}\,p \leftrightarrow (\alpha, y)\mathsf{X}\,\mathsf{X}\,p$

Counterfactual Dependence



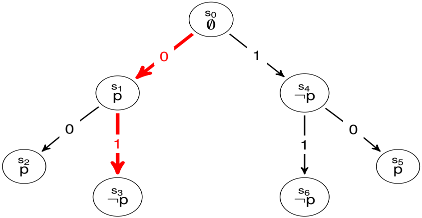
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x)\mathsf{X}\, p \leftrightarrow (\alpha, y)\mathsf{X}\, \neg p$
- $\psi_2 = (\alpha, x)\mathsf{X}\,\mathsf{X}\, p \leftrightarrow (\alpha, y)\mathsf{X}\,\mathsf{X}\, p$

Counterfactual Dependence



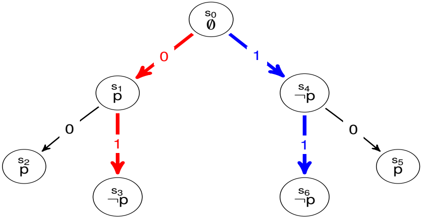
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) X\, p \leftrightarrow (\alpha, y) X \neg p$
- $\psi_2 = (\alpha, x) X X\, p \leftrightarrow (\alpha, y) X X\, p$

Counterfactual Dependence



- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) \mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X}\, \neg p$
- $\psi_2 = (\alpha, x) \mathsf{X}\,\mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X}\,\mathsf{X}\, p$

Counterfactual Dependence



- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) \mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X} \neg p$
- $\psi_2 = (\alpha, x) \mathsf{X}\mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X}\mathsf{X}\, p$

Behavioral Semantics

Behavioral Satisfiability

A formula φ is *behaviorally satisfiable* iff all strategy quantifications required to satisfy φ are solved locally (i.e, on the same play history).

SL Behavioral Fragments

By constraining the use of bindings, we can obtain syntactic fragments of SL having a behavioral semantics.

A maximal behavioral fragment

We propose two behavioral fragments whose syntactic union is not anymore behavioral.

Strategy Logic Fragments

Quantification and bining prefixes

A *quantification prefix* is a sequence $\wp$ of quantifications in which each variable occurs once: $\wp = [[x]][[y]]\langle\langle z\rangle\rangle[[w]]$.

A *binding prefix* is a sequence $\flat$ of bindings such that each agent occurs once: $\flat = (\alpha, x)(\beta, y)(\gamma, y)$.

Quantification and bining prefixes

A *quantification prefix* is a sequence $\wp$ of quantifications in which each variable occurs once: $\wp = [[x]][[y]]\langle\langle z\rangle\rangle[[w]]$.

A *binding prefix* is a sequence $\flat$ of bindings such that each agent occurs once: $\flat = (\alpha, x)(\beta, y)(\gamma, y)$.

A goal is a binding prefix $\flat$ followed by an LTL formula.

Quantification and bining prefixes

A *quantification prefix* is a sequence $\wp$ of quantifications in which each variable occurs once: $\wp = [[x]][[y]]\langle\langle z\rangle\rangle[[w]]$.

A *binding prefix* is a sequence $\flat$ of bindings such that each agent occurs once: $\flat = (\alpha, x)(\beta, y)(\gamma, y)$.

A goal is a binding prefix $\flat$ followed by an LTL formula.

By using a prenex normal form of a combination of goals, we identify a chain of fragments, which we name SL[BG], SL[DG / CG], and SL[1G].

Boolean-Goal Strategy Logic (SL[BG])

Definition

SL[BG] *formulas* are built inductively in the following way, where $\wp$ is a quantification prefix and $\flat$ a binding prefix:

$$\varphi ::= \text{LTL} \mid \wp\psi,$$
$$\psi ::= \flat\varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi,$$

where $\wp$ quantifies over all free variables of ψ.

- For SL[CG], we set $\psi ::= \flat\varphi \mid \psi \wedge \psi$.
- For SL[DG], we set $\psi ::= \flat\varphi \mid \psi \vee \psi$.
- For SL[1G], we set $\psi ::= \flat\varphi$.

The expressiveness chain

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

The behavioral results

Theorem

- SL[CG] *and* SL[DG] *have behavioral semantics.*
- SL[BG] *does not have the behavioral semantics.*

Theorem

Both SL[CG] *and* SL[DG] *model-checking problems have a* 2EXPTIME-COMPLETE *formula complexity and a* PTIME-COMPLETE *data complexity.*

Conclusion

The expressiveness chain

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

Behavioral

SL[CG] and SL[DG] are the maximal fragments having a behavioral semantics.

	Model checking	Satisfiability
SL	"NONELEMENTARY-COMPLETE"	Σ^1_1-HARD
SL[BG]	?	Σ^1_1-HARD
SL[CG / DG]	2EXPTIME-COMPLETE	?
SL[1G]	2EXPTIME-COMPLETE	2EXPTIME-COMPLETE
ATL*	2EXPTIME-COMPLETE	2EXPTIME-COMPLETE

References

- Mogavero, M., & Vardi. Reasoning About Strategies. FSTTCS'10.
- Mogavero, Murano, Perelli, Vardi. What Makes ATL* Decidable? A Decidable Fragment of Strategy Logic. CONCUR'12.
- Mogavero, Murano, Perelli, Vardi. Reasoning About Strategies: On the Model-Checking Problem. TR, arXiv.
- Mogavero, Murano, Sauro. On the Boundary of Behavioral Strategies. LICS'13.