Reasoning about Strategies: From module checking to strategy logic

Aniello Murano

based on joint works with
Fabio Mogavero, Giuseppe Perelli, Luigi Sauro, and Moshe Y. Vardi

Università degli Studi di Napoli "Federico II"

Luxembourg
September 23, 2013

Strategic Reasoning

Game Theory is a fruitful metaphor in the verification and synthesis of multi-agent systems, where agent behaviors are modeled by strategies in a game.

Plenty of modal logics for the specification of strategic reasonings have been introduced, but with a very limited power and no unifying framework.

Our aim

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

An historical introduction to the framework

From monolithic to multi-agent systems

1. Closed systems verification: Model Checking
2. (System vs. Environment) open systems verification: Module Checking
3. Concurrent multi-agent system verification: ATL*
4. A multi-agent logic in which strategies are treated explicitly: Strategy Logic

Outline

Model checking

Historical development(1)

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

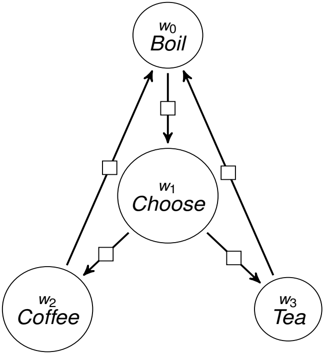
$$M \models \varphi$$

Inputs

- The model M is a Kripke structure, i.e., a labeled-state transition graph.
- The specification φ is a temporal logic formula such as LTL, CTL or CTL*.

A closed system example: A drink dispenser-machine

M :



- $M = \langle \mathrm{AP}, \mathrm{W}, R, \mathsf{L}, w_0 \rangle$
- $\varphi = \exists\, F\ \mathsf{Tea}$
- M only makes internal non-deterministic choices
- $M \models \varphi$

Remark

The system behavior can be represented by the unique three unwinding T_M of M.

Module Checking

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].

$$M \models_r \varphi$$

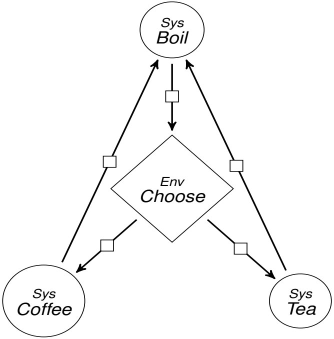
Inputs

- A module M is a Kripke structure with states partitioned in Sys and Env states.
- The specification φ is a temporal logic formula.

The problem

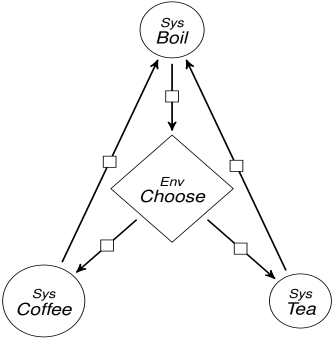
- Checking whether M is correct w.r.t. any possible behavior of the environment.

An open system example



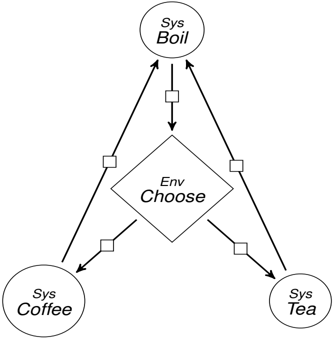
- $M = \langle \mathrm{AP}, \mathrm{Sys}, \mathrm{Env}, R, \mathrm{L}, w_0 \rangle$
- $W = \mathrm{Sys} \cup \mathrm{Env}$
- $\mathrm{Sys} \cap \mathrm{Env} = \emptyset$

An open system example



- $M = \langle \mathrm{AP}, \mathrm{Sys}, \mathrm{Env}, R, \mathrm{L}, w_0 \rangle$
- $W = \mathrm{Sys} \cup \mathrm{Env}$
- $\mathrm{Sys} \cap \mathrm{Env} = \emptyset$
- Always, at the *Choose* state, the environment makes a choice

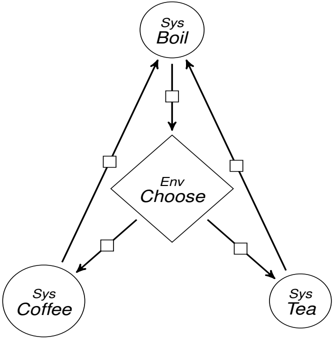
An open system example



- $M = \langle \mathrm{AP}, \mathrm{Sys}, \mathrm{Env}, R, \mathrm{L}, w_0 \rangle$
- $W = \mathrm{Sys} \cup \mathrm{Env}$
- $\mathrm{Sys} \cap \mathrm{Env} = \emptyset$
- Always, at the *Choose* state, the environment makes a choice

- $\varphi = \exists\, F\, \mathrm{Tea}$
- $M \not\models_r \varphi$

An open system example



- $M = \langle \mathrm{AP}, \mathrm{Sys}, \mathrm{Env}, R, \mathrm{L}, w_0 \rangle$
- $W = \mathrm{Sys} \cup \mathrm{Env}$
- $\mathrm{Sys} \cap \mathrm{Env} = \emptyset$
- Always, at the *Choose* state, the environment makes a choice

- $\varphi = \exists F\ \mathrm{Tea}$
- $M \not\models_r \varphi$

Remark

- Everytime an Env state is met, the environment can disable some (but one) of its successors.
- Any possible behavior of the environment induces a different tree (i.e., a partial tree unwinding of M).
- T_M is a particular environment behavior.

Pro vs. Cons

Applications

- Module checking is very useful in open system verification. It allows to check whether a system is correct no matter how the environment behaves.
- It has been studied under perfect/imperfect information, hierarchical, infinite-state systems (pushdown, real-time), backwards modalities, graded modalities....

Pro vs. Cons

Applications

- Module checking is very useful in open system verification. It allows to check whether a system is correct no matter how the environment behaves.
- It has been studied under perfect/imperfect information, hierarchical, infinite-state systems (pushdown, real-time), backwards modalities, graded modalities....

Limitations

- Two-player game between system and environment.
- It is not powerful enough to be used in multi-player strategic reasoning.

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

Historical development(3)

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

ATL*

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

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

Example

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

Underlying framework: the concurrent game structure

CGS

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

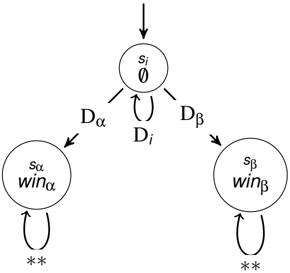
Intuitively

$\mathcal{G}$ 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 *strategy* is a function that maps each *history* of the game to an *action*.
A play is a path of the game determined by the history of strategies.

The paper, rock, and scissor game



- $\mathrm{Ag} = \{\alpha : \mathit{Alice}, \beta : \mathit{Bob}\}$
- $\mathrm{St} = \{s_i, s_\alpha, s_\beta\}$
- s_i initial state
- $\mathrm{AP} = \{win_\alpha, win_\beta\}$
- $\mathrm{Ac} = \{P : \mathit{Paper}, R : \mathit{Rock}, S : \mathit{Scissor}\}$
- $D_i = \{(P,P),(R,R),(S,S)\}$
- $D_\alpha = \{(P,R),(R,S),(S,P)\}$
- $D_\beta = \{(R,P),(S,R),(P,S)\}$

Pro vs. Cons

Pro

ATL* allows multi-agent strategic reasoning.

Limitations

- Strategies are treated only implicitly.
- Quantifier alternation fixed to 1: either $\langle\langle\ \rangle\rangle[[\ ]]$ or $[[\ ]]\langle\langle\ \rangle\rangle$.

Our contribution

Strategy Logic

We introduce *Strategy Logic* (SL), as a more general framework (both in its syntax and semantics), for explicit reasoning about strategies in multi-player concurrent games, where strategies are treated as first order objects.

Some useful fragments

We also consider a chain of syntactic fragments of SL that are strictly more expressive than ATL*, but more tractable than SL.

As for ATL*, the underlying model is a CGS

Recall what is a CGS

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

...and its intuitive explanation

$\mathcal{G}$ 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.

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

- $\langle\langle x\rangle\rangle\varphi$: *"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"*.
- LTL operators are classically interpreted on the resulting play.

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, ensuring that a failure state is never reached, independently of 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.

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.

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

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 $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$.

ATL* model-theoretic properties

Positive model-theoretic properties

- Invariance under bisimulation.
- Invariance under decision-unwinding.
- Bounded decision-tree model property.

SL model-theoretic properties

Negative model-theoretic properties

- Non-invariance under bisimulation.
- Non-invariance under decision-unwinding.
- Unbounded model property.

Positive model-theoretic properties

- Invariance under state-unwinding.
- State-tree model property.

Expressiveness

Theorem

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

Explanation

- Unbounded quantifier alternation.
- 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 natural question

The question

Why is SL hard?

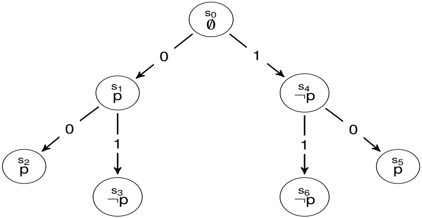
The answer

The choice of an action made by an agent in a strategy, for a given history of the game, may depend on the entire strategy of another agent, i.e., on its actions over all possible histories of the game.

An observation

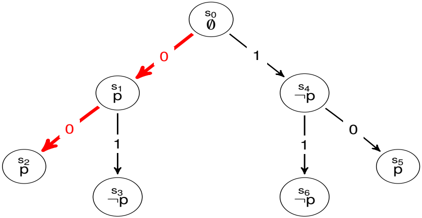
Strategies are not synthesizable, since an agent, to have a chance to win, may need to forecast a possibly infinite amount of information about the behavior of an opponent.

Conterfactual dependence



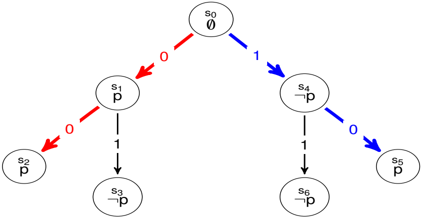
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) X\, p \leftrightarrow (\alpha, y) X\, \neg p$
- $\psi_2 = (\alpha, x) X X\, p \leftrightarrow (\alpha, y) X X\, p$

Conterfactual dependence



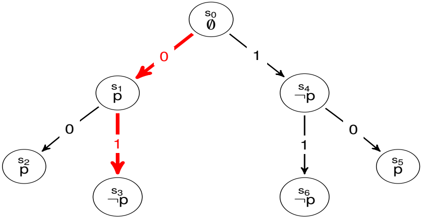
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) \mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X} \neg p$
- $\psi_2 = (\alpha, x) \mathsf{X}\,\mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X}\,\mathsf{X}\, p$

Conterfactual dependence



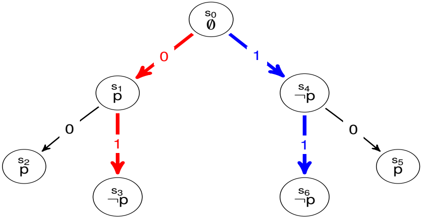
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) X\, p \leftrightarrow (\alpha, y) X\, \neg p$
- $\psi_2 = (\alpha, x) X\, X\, p \leftrightarrow (\alpha, y) X\, X\, p$

Conterfactual dependence



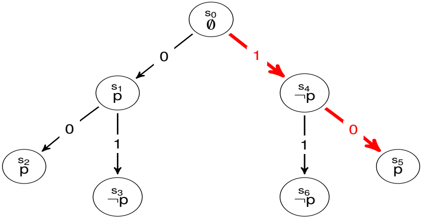
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) \mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X} \neg p$
- $\psi_2 = (\alpha, x) \mathsf{X}\mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X}\mathsf{X}\, p$

Conterfactual dependence



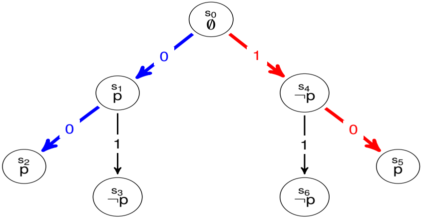
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) X\, p \leftrightarrow (\alpha, y) X\, \neg p$
- $\psi_2 = (\alpha, x) X\, X\, p \leftrightarrow (\alpha, y) X\, X\, p$

Conterfactual dependence



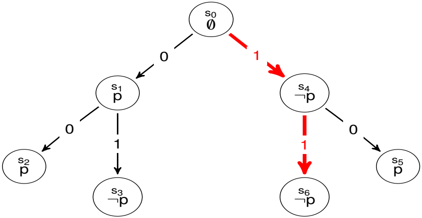
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) \mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X}\, \neg p$
- $\psi_2 = (\alpha, x) \mathsf{X}\,\mathsf{X}\, p \leftrightarrow (\alpha, y) \mathsf{X}\,\mathsf{X}\, p$

Conterfactual dependence



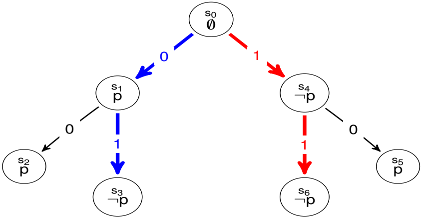
- $\varphi = [[x]]\langle\langle y\rangle\rangle \psi_1 \wedge \psi_2$
- $\psi_1 = (\alpha, x) X\, p \leftrightarrow (\alpha, y) X \neg p$
- $\psi_2 = (\alpha, x) X X\, p \leftrightarrow (\alpha, y) X X\, p$

Conterfactual dependence



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

Conterfactual dependence



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

Elementariness in strategies

Behavioral property

The quantification of a strategy is behavioral if the actions in a given history depend only on the actions of all other strategies on the same history.

Behavioral semantics

A formula is behaviorally satisfiable if it only needs behavioral strategies to be satisfied.

Fact

ATL* is behaviorally satisfiable.

Another question

The question

Is there any other syntactic fragment of SL (strictly subsuming ATL*) having a behavioral semantics?.

Our answer

Yes! We obtain several fragments by using a prenex normal form for SL and by putting different constraints on the use of bindings.

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[1G], we set $\psi ::= \flat\varphi$.

The expressiveness chain

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

The behavioral results

The question

Which fragments of SL have behavioral semantics?

Theorem

- SL[BG] *does not have behavioral semantics.*
- SL[CG] *and* SL[1G] *have behavioral semantics.*

An overview

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

Model-theoretic properties

SL[BG] negative model-theoretic property

Unbounded model property.

SL[CG] negative model-theoretic properties

- Non-invariance under bisimulation.
- Non-invariance under decision-unwinding.

SL[1G] positive model-theoretic properties

- Invariance under bisimulation.
- Invariance under decision-unwinding.
- Bounded decision-tree model property.

In this talk

- 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 and a PTIME-COMPLETE data complexity.
- SL satisfiability is highly undecidable, i.e., Σ_1^1-HARD.

- We have also introduced some fragments of SL, named SL[BG], SL[CG], and SL[1G], all strictly more expressive than ATL*.
- We have studied their model-theoretic properties. In particular, model-checking and satisfiability for SL[1G] are no more complex than those for ATL*, i.e., they are both 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.

ATL* References

- Alur, Henzinger, Kupferman. Alternating-time temporal logic. J.ACM. 2002

Module Checking References

- Kupferman, Vardi, Wolper: Module Checking. I.&C. 2001
- Aminof, Legay, Murano, Serre, Vardi. Pushdown Module Checking with imperfect information I.&C. 2013.