

Relentful Strategic Reasoning in Alternating-Time Temporal Logic

Fabio Mogavero¹ Aniello Murano¹ Moshe Y. Vardi²

¹Università degli Studi di Napoli "Federico II"

<http://people.na.infn.it/~{mogavero,murano}>

²Rice University

<http://www.cs.rice.edu/~vardi/>

3rd Workshop of the ESF Networking Programme on
Games for Design and Verification
Oxford, England, UK, September 20-23, 2010

System correctness (1)

Let \mathbf{S} be a system and \mathbf{P} a desired behavior (specification).

Two very important problems:

- **Model Checking**: Is \mathbf{S} correct w.r.t. \mathbf{P} ?
- **Satisfiability**: Is \mathbf{P} a correct specification?

To answer to these questions, formal methods are used.

- \mathbf{S} can be modeled by a **labeled transition graph** \mathcal{K} (Kripke structure).
- \mathbf{P} can be expressed as a **temporal logic formula** φ .

Then,

- **Model Checking**: Is \mathcal{K} a model of φ ($\mathcal{K} \models \varphi$)?
- **Satisfiability**: Is there a \mathcal{K} such that $\mathcal{K} \models \varphi$?

System correctness (2)

Verification as debugging: failure of verification identifies bugs.

- Both specifications and programs formalize informal requirements.
- Verification contrasts two independent formalizations.
- Failure of verification reveals inconsistency between formalizations.

Model checking: uncommonly effective debugging tool.

- Systematic exploration of the design state space.
- Good at catching difficult “limit cases”.

Satisfiability: useful to verify...

- the realizability of a specification;
- the non-triviality of a specification.

Classical system model

Key Idea: Systems can be represented as Kripke structures!

A Kripke structure $\mathcal{K} = \langle \text{AP}, \mathbb{W}, R, L, w_0 \rangle$ is a labeled transition graph used to model system behaviors:

- 1 AP : *atomic propositions*;
- 2 \mathbb{W} : *worlds represent system states*;
- 3 $w_0 \in \mathbb{W}$: *designated initial world*;
- 4 $R \subseteq \mathbb{W} \times \mathbb{W}$: *edges represent system transitions*;
- 5 $L : \mathbb{W} \rightarrow 2^{\text{AP}}$: *labels represent state properties*;
- 6 $\pi \in \text{Pth}(\mathcal{K})$: *paths represent system executions*.

Classical system specifications

Key Idea: Temporal logic allows the description of the ordering of events!

Two main families of temporal logics:

- **Linear-Time Temporal Logics (LTL)**
 - Each moment in time has a unique possible future.
 - LTL expresses path properties based on the paths state labels.
 - Useful for hardware specification.
- **Branching-Time Temporal Logics (CTL, CTL*, and μ CALCULUS)**
 - Each moment in time may split into various possible future.
 - CTL* expresses state properties based on the existence or universality of paths exiting from that state and satisfying LTL-like properties.
 - Useful for software specification.

From monolithic to multi-agent systems

Historical development:

- **Model checking**: analyzes systems viewed monolithically (system components plus environment).
- **Module checking**: separates out the environment from the system components, but still views the system monolithically [KV96].
- **Alternating temporal reasoning**: multi-agent systems (components individually considered), playing strategically (**ATL**, **ATL***) [AHK02].

Motivation: Reasoning about strategic behavior.

Example

“Agents 1 and 2 cooperate to ensure that a system (having more than two processes (agents)) never enters a fail state.”

Concurrent game structures

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

- AP : set of *atomic propositions*;
- Ag : set of *agents*;
- Ac : set of *actions*;
- St : set of *states*;
- $s_0 \in St$: *designated initial state*;
- $\lambda : St \rightarrow 2^{AP}$: *labeling function*;
- $\tau : St \times Ac^{Ag} \rightarrow St$: *transition function* that mapping a state and a *decision* (i.e., a function from Ag to Ac) to a new state;
- $\rho \in \text{Trc}(\mathcal{G})$: *traces represent partial system executions*;
- $\pi \in \text{Pth}(\mathcal{G})$: *paths represent full system executions*.

Strategies, plays, and ATL* quantifications

Strategy: a strategy for a set $A \subseteq \text{Ag}$ of agents is a function $f_A : \text{Trc}(\mathcal{G}) \rightarrow \text{Ac}^A$ (choice of actions as function of history).

Play: a strategy f_{Ag} for Ag yields a play $\pi = \pi_0 \cdot \pi_1 \cdot \dots$, where $\pi_{i+1} = \tau(\pi_i, f_{\text{Ag}}(\pi_{\leq i}))$ (a play is a path coherent with the strategy).

Strategic quantification in ATL*:

- $\langle\langle A \rangle\rangle \psi$: there exists a strategy of agents in A such that, for all strategies of agents in $\text{Ag} \setminus A$, the resulting plays satisfy ψ ;
- $\llbracket A \rrbracket \psi$: for every strategy of agents in A , there is a strategy of agents in $\text{Ag} \setminus A$ such that the resulting play satisfy ψ .

Relentful reasoning

Example

- “I plan to study and become a professor of CS.”
 $\langle\langle I \rangle\rangle \text{StudyCS} \text{ U } \text{ProfessorCS}$.
- “I plan to study and become a professor of CS, but if I win the lottery, I would rather become a Jazz player.”
 $\langle\langle I \rangle\rangle \text{StudyCS} \text{ U } (\text{ProfessorCS} \vee (\text{Lottery} \wedge \text{Jazz}))$.
- How can I become a Jazz player without studying?
 $\langle\langle I \rangle\rangle (\text{StudyCS} \wedge \text{PreJazz}) \text{ U } (\text{ProfessorCS} \vee \text{Lottery})$, where
 $\text{PreJazz} = \text{Lottery} \rightarrow \langle\langle I \rangle\rangle (\text{StudyJazz} \text{ U } \text{JazzPlayer})$.
- “Should I not study Jazz, in parallel to studying CS, rather than wait for the lottery?”

In modern English “to relent” is used only in the combination of “relentless”.

ATL* memorylessness

Crux: the strategic quantifiers in ATL* are memoryless, i.e., they start a new plan from the current state, as path quantifiers in CTL* start new paths!

$\langle\langle A \rangle\rangle\psi$: the plays starting at the current state and coherent with a strategy of A satisfy ψ .

ATL* memoryfulness

To easily model relentless reasoning we need memoryful quantifications!

- $\langle\langle A \rangle\rangle\psi$: the plays starting at the initial state s_0 , passing through the current state, and coherent with a strategy of A satisfy ψ .
- Special proposition present to refer to the “present”.

Example

“GAMES issues grants only after a proposal has been submitted and approved.”

$grant \rightarrow \langle\langle GAMES \rangle\rangle(F(proposal \wedge F(approved \wedge F present)))$.

Prior works

History

- “Possibly ψ ” (it is always possible to extend the computation of the system to one that satisfies ψ) [Lamport, 1998];
- Memoryfulness in the context of planning [Pistore & Vardi, 2003];
- mCTL* [Kupferman & Vardi, 2006].

mCTL*

- **Expressiveness**: CTL* is as expressive as mCTL*.
- **Succinctness**: mCTL* is at least exponentially more succinct than CTL*.
- **Satisfiability**: 2EXPTIME-COMPLETE.
- **Model checking**: EXPSPACE-COMPLETE.

Outline

1 Memoryful Alternating-Time Temporal Logic

2 Decision Problems

3 Conclusion

Syntax of mATL*

Definition

mATL* *state* (φ) and *path* (ψ) *formulas* are built inductively as follows:

- 1 $\varphi ::= \text{present} \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle\langle A \rangle\rangle\psi \mid \llbracket A \rrbracket\psi,$
- 2 $\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid X\psi \mid \psi U\psi \mid \psi R\psi.$

pATL* is ATL* augmented with past-time operators.

mpATL* is mATL* augmented with past-time operators.

Semantics of mATL*

Definition

Given a concurrent game structure $\mathcal{G} = \langle AP, Ag, Ac, St, \lambda, \tau, s_0 \rangle$, two traces $\rho, \rho_\rho \in \text{Trc}(\mathcal{G})$, a path $\pi \in \text{Pth}(\mathcal{G})$, and a number $k \in \mathbb{N}$, it holds that:

- 1 $\mathcal{G}, \rho, \rho_\rho \models \text{present}$ iff $\rho = \rho_\rho$;
- 2 $\mathcal{G}, \rho, \rho_\rho \models \langle\langle A \rangle\rangle \psi$ iff there exists a strategy f_A of agents in A such that for all f_A -plays π it holds that $\mathcal{G}, \rho \cdot \pi, 0, \rho \models \psi$;
- 3 $\mathcal{G}, \rho, \rho_\rho \models \llbracket A \rrbracket \psi$ iff for all strategies f_A of agents in A there exists an f_A -play π such that $\mathcal{G}, \rho \cdot \pi, 0, \rho \models \psi$;

Expressiveness and Succinctness (1)

mATL* vs ATL*

- ATL* is as expressive as mATL*.
- mATL* is at least exponentially more succinct than ATL*.

Open problem

- What is the precise succinctness relationship?
- Gap: exponential lower bound vs nonelementary upper bound.

Key Idea: memoryful quantification \Leftrightarrow past-time temporal operators

Referring to the **start of a computation** is referring to the **past (linear)**.

Expressiveness and Succinctness (2)

mpATL*, pATL*, and ATL*

- ATL* is a fragment of mpATL*.
- Linear reduction of mpATL* to pATL*.
- Nonelementary reduction of pATL* to ATL*.

Hence, mpATL* is as expressive as ATL*.

Key Idea: Gabbay's Separation Lemma (pLTL separation of past from future).

Open problem

- Precise succinctness relationship between mpATL* and pATL*.

Expressiveness and Succinctness (3)

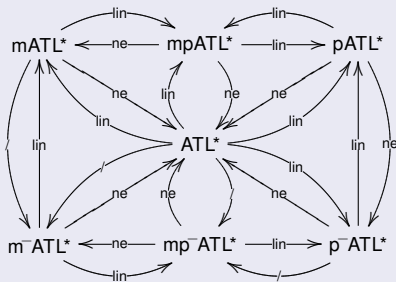


Figure: Hierarchy of expressive power and succinctness.

Outline

1 Memoryful Alternating-Time Temporal Logic

2 Decision Problems

3 Conclusion

Principal question

Does the additional succinctness of mATL^* implies an increasing of the computational-complexity cost of deciding its problems?

	M.C.	Sat.
LTL	PSPACE-COMPLETE	PSPACE-COMPLETE
CTL	PTIME-COMPLETE	EXPTIME-COMPLETE
CTL*	PSPACE-COMPLETE	2EXPTIME-COMPLETE
mCTL*	EXPSpace-COMPLETE	2EXPTIME-COMPLETE

Table: Computational complexity of Model Checking and Satisfiability.

Note: mCTL^* as a worst Model Checking complexity than that of CTL^* !

Problems and solutions (1)

For ATL*

- **Model Checking:** 2EXPTIME-COMPLETE [Alur et al., 2002]
- **Satisfiability:** 2EXPTIME-COMPLETE [Schewe, 2008]

ATL* Model Checking

- Just like for CTL*, perform model checking on state formulas from the inside out.
- However, model checking $\langle\langle A \rangle\rangle\psi$ amounts to LTL temporal synthesis, which can be done in 2EXPTIME [Pnueli & Rosner, 1989].
- We cannot use the same ad-hoc automaton model of CTL* (hesitant automata).

Problems and solutions (2)

Difficulty I:

- Automata are memoryless, unless states carry history.
- Adding history to states blows up the state space exponentially.

Possible solution: 2-way tree automata.

Difficulty II: How to mark the present?

Challenge: Need to add memory to alternating automata, without blowing up the size of the state space.

Solution: Separate history from alternating automata.

- **Satellite:** a deterministic safety automaton that carry history.
- Alternating automaton reads history from satellite.

Exponentials add rather than compose: $2^{2^n} \times 2^{2^n}$ vs. $2^{2^{2^n}}$!

Problems and solutions (3)

Challenge Here: Do it in the framework of agent-action tree automata [Schewe, 2008], rather of the more standard alternating tree automata [Kupferman & Vardi, 2006].

Computational complexity

	M.C.	Sat.
LTL	PSPACE-COMPLETE	PSPACE-COMPLETE
CTL	PTIME-COMPLETE	EXPTIME-COMPLETE
CTL*	PSPACE-COMPLETE	2EXPTIME-COMPLETE
mCTL*	EXPSpace-COMPLETE	2EXPTIME-COMPLETE
ATL*	2EXPTIME-COMPLETE	2EXPTIME-COMPLETE
mATL*	2EXPTIME-COMPLETE	2EXPTIME-COMPLETE
pATL*	2EXPTIME-COMPLETE	2EXPTIME-COMPLETE
mpATL*	2EXPTIME-COMPLETE	2EXPTIME-COMPLETE

Table: Computational complexity of Model Checking and Satisfiability.

Outline

1 Memoryful Alternating-Time Temporal Logic

2 Decision Problems

3 Conclusion

Conclusion

In this work...

- we observe that ATL^* is deficient because strategic quantifications are memoryless;
- we introduce $mATL^*$ together with related logics that add memoryfulness to quantifications;
- we show that these logics add no expressiveness, but succinctness and naturalness w.r.t. ATL^* ;
- finally, we show that there is no computational penalty to reasoning about memoryful quantifications.

Thank you very much for your attention!
I hope my talk was enough strategic for you! :P