Satisfiability Checking of ATL and SL with Simple Goals

M. Kacprzak, A. Niewiadomski, W. Penczek

Bialystok University of Technology, Faculty of Computer Science, Poland
Siedlce University, Faculty of Exact and Natural Sciences, Poland
Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Seminar of ZTSRiO, October 14, 2021

Introduction

1. Why Satisfiability Checking?
2. How to Specify Strategic Abilities?
   1. Alternating-time Temporal Logic
   2. Strategy Logic with Simple Goals
3. Satisfiability Algorithm
4. MsATL and SGSAT tools

Why Satisfiability Checking?

- Many important properties are based on strategic ability
- Functionality ≈ ability of authorized users to complete tasks
- Security ≈ inability of unauthorized users to complete tasks
- One can try to formalize such properties in modal logics of strategic ability, such as ATL or Strategy Logic
- Model synthesis: find a model satisfying a given property, provided that such a model exists
- Model checking: by **checking satisfiability** of $\varphi_{system} \wedge \neg\varphi_{correct}$

Satisfiability (SAT) versus Model Checking (MC)

PL : SAT: NPTIME-complete, MC: PTIME-complete

LTL : SAT: PSPACE-complete, MC: PSPACE-complete,
from SAT to MC using automata approach,

CTL : SAT: EXPTIME-complete, MC: PTIME-complete

CTL* : SAT: 2EXPTIME-complete, MC: PSPACE-complete,

ATL_I : SAT: EXPTIME-complete, MC: PTIME-complete,

ATL_i : SAT: ??, MC: $\Delta_2^P - complete$,

$SL[SG]_I$: SAT: $\leq$ 2EXPTIME-complete, MC: PTIME-complete,

TCTL : SAT: undecidable, MC: PSPACE-complete,

Assumption: model is represented by an LTS or CGS and the size is given by the number of its transitions

Outline

1. Alternating-time Temporal Logic
2. Satisfiability Algorithm for ATL and MsATL tool
3. Strategy Logic with Simple Goals
4. Satisfiability Algorithm for SL[SG] and SGSAT tool

ATL: What Agents Can Achieve?

- ATL: Alternating-time Temporal Logic [Alur et al. 1997-2002]
- Temporal logic meets game theory
- Main idea: cooperation modalities

$\langle\langle A \rangle\rangle$: coalition A has a collective strategy to enforce Φ

$\rightsquigarrow$ Φ can include temporal operators: X (next), F (sometime in the future), G (always in the future), U (strong until)

Example formulae

"Agent 1 can ensure that the safe will never be opened."

$$\langle\langle 1 \rangle\rangle G(\neg\ open)$$

"Agent 1 cannot open the safe by himself."

$$\neg \langle\langle 1 \rangle\rangle F(open)$$

"Agents 1 and 2 can cooperate to open the safe."

$$\langle\langle 1, 2 \rangle\rangle F(open)$$

Multi-agent system and model

MAS

- $\mathcal{A}$ - a finite set of **agents**
  - L_i - a set of local states of $i \in \mathcal{A}$
  - Act_i - a set of local actions of $i \in \mathcal{A}$
  - P_i, T_i - a local protocol and a transition function of $i \in \mathcal{A}$
- $Act = Act_1 \times \cdots \times Act_n$ - a set of **global actions**
- $\mathcal{PV}$ - a finite set of **global propositions**

Model $M = (\mathcal{S}t, \iota, T, V)$

• $\mathcal{S}t = L_1 \times \cdots \times L_n$ - the **global states**
• $\iota \in \mathcal{S}t$ - an **initial state**
• $T : \mathcal{S}t \times Act \rightharpoonup \mathcal{S}t$ - the **global transition (partial) function** based on local protocols
• $V : \mathcal{S}t \to 2^{\mathcal{PV}}$ - the **valuation** of the states

Example model

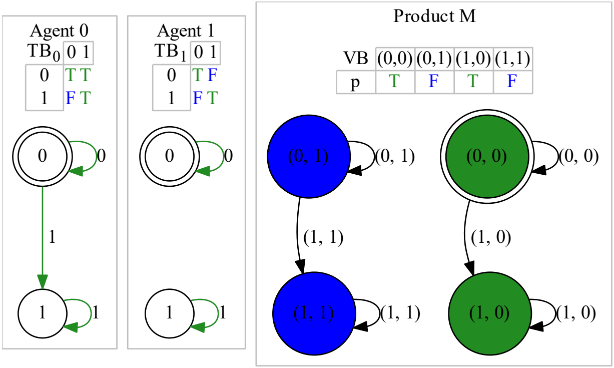


Figure: Example model M

Strategies and Abilities

Strategy

A **strategy** is a conditional plan.

We represent strategies by functions $\sigma_i : St \to Act_i$.

$\rightsquigarrow$ memoryless, perfect information strategies

Semantics of ATL

$M, s \models \langle\langle A \rangle\rangle \Phi$ iff there is a collective strategy σ_A such that, for every path λ that may result from execution of σ_A from s, we have that $M, \lambda \models \Phi$.

Semantic Variants of ATL

Available information:

- Perfect information (I) vs. imperfect information strategies (i)

Memory of agents:

- Perfect recall (R) vs. memoryless strategies (r)

MsATL uses memoryless (r) strategies, with and without perfect information, and is the only one dedicated to checking satisfiability of imperfect information.

Problem and computational complexity

Decision problem ATL_YSAT, for $Y \in \{i, I\}$

Decide whether an ATL_Y formula is satisfiable.

Computational complexity

- ATL_ISAT - EXPTIME-complete (for a fixed number of agents)

  V. Goranko, G. Van Drimmelen
  Complete axiomatization and decidability of ATL.
  Theoretical Computer Science, 2006.

- ATL_iSAT - unknown

Related work

- Inspiration: SAT Modulo Monotonic Theories Solver for CTL

T. Klenze, S. Bayless, A.J. Hu
Fast, Flexible, and Minimal CTL Synthesis via SMT.
Computer Aided Verification, 2016.

- External MC tools:
  - **MCMAS** - for ATL_IMC

A. Lomuscio, H. Qu, F. Raimondi
MCMAS: an open-source model checker for the ver. of MAS.
International Journal on Software Tools for Technology Transfer, 2017.

  - **STV** - for ATL_iMC

D. Kurpiewski, M. Knapik, W. Jamroga
On Domination and Control in Strategic Ability.
AAMAS, 2019.

- Comparison: **TATL** - for ATL_ISAT

A. David
Deciding ATL Satisfiability by Tableaux.*
Int. Conf. on Aut. Deduction, 2015.

Outline

1. Alternating-time Temporal Logic
2. Satisfiability Algorithm for ATL and MsATL tool
3. Strategy Logic with Simple Goals
4. Satisfiability Algorithm for SL[SG] and SGSAT tool

Boolean encoding of ATL model

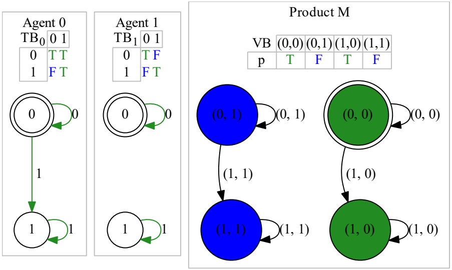


Figure: Example model M

$$v_M = (\underbrace{T, T, F, T}_{TB_0:\ \text{trans. of Ag.0}}, \underbrace{T, F, F, T}_{TB_1:\ \text{rans. of Ag.1}}, \underbrace{T, F, T, F}_{VB:\ \text{global props.}})$$

Partial model

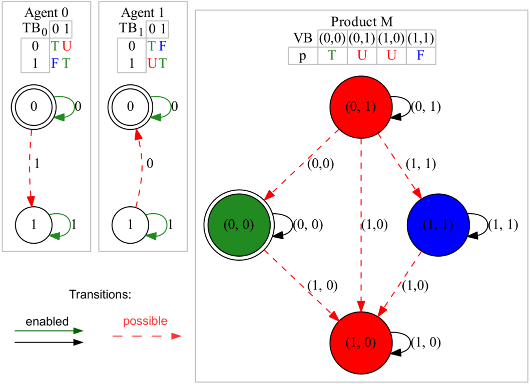


Figure: Partial model M_{par}

$$v_M = \Big(\underbrace{T, U, F, T}_{TB_0:\ \text{trans. of Ag.0}} , \underbrace{T, F, U, T}_{TB_1:\ \text{trans. of Ag.1}} , \underbrace{T, U, U, F}_{VB:\ \text{global props.}} \Big)$$

Approximations

For fixed $Y \in \{i, I\}$, partial model M_{par}, and a formula ϕ, we compute

approximation of models:

- $CLASS^{Y}_{M_{par}}(\phi)$ - a set of models s.t.:

  if M is a total extension of M_{par} and $M, s \models_Y \phi$

  for some $s \in \mathcal{S}t$, then $M \in CLASS^{Y}_{M_{par}}(\phi)$

approximation of states:

- $\| \phi \|^{Y}_{M_{par}} = \{s \in \mathcal{S}t : \exists_{M \in CLASS^{Y}_{M_{par}}(\phi)} \; M, s \models_Y \phi\}$

How to use these approximations?

For fixed $Y \in \{i, I\}$, M_{par}, ϕ, and an initial state ι,

if M_{par} can be extended to a total model M s.t. $M, \iota \models_Y \phi$, then

$$M \in CLASS^Y_{M_{par}}(\phi) \text{ and } \iota \in \| \phi \|^Y_{M_{par}}$$

Consequently, if

$$\iota \notin \| \phi \|^Y_{M_{par}}$$

then M_{par} cannot be extended to any total model satisfying ϕ.

Monotonicity property

The implication

$$\text{if } M, s \models^Y \phi \text{ then } M', s \models^Y \phi$$

holds if the following conditions are satisfied

CASE: $\phi \in \{\langle\langle\Gamma\rangle\rangle Xp, \langle\langle\Gamma\rangle\rangle Gp, \langle\langle\Gamma\rangle\rangle pUp\}$

- positive monotonicity wrt. local transitions of agents $i \in \Gamma$: $v_M[TB_i] \leq v_{M'}[TB_i]$
- negative monotonicity wrt. local transitions of agents $i \notin \Gamma$: $v_M[TB_i] \geq v_{M'}[TB_i]$
- positive monotonicity wrt. propositions: $v_M[VB] \leq v_{M'}[VB]$
  $\rightsquigarrow$ assuming that $F < T$

Construction of M_{over} and M_{under}

Partial model M_{par}:

$$v_{M_{par}} = (\underbrace{T, U, U, T}_{TB_0:\text{ trans. of Ag.0}} , \underbrace{T, U, U, T}_{TB_1:\text{ trans. of Ag.1}} , \underbrace{U, U, U, U}_{VB:\text{ global prop.}})$$

M^{Γ}_{over} for the given M_{par} and a set of agents $\Gamma = \{0\}$:

$$v_{M^{\Gamma}_{over}} = (\underbrace{T, T, T, T}_{TB_0:\text{ trans. of Ag.0}} , \underbrace{T, F, F, T}_{TB_1:\text{ trans. of Ag.1}} , \underbrace{T, T, T, T}_{VB:\text{ global prop.}})$$

M^{Γ}_{under} for the given M_{par} and a set of agents $\Gamma = \{0\}$:

$$v_{M^{\Gamma}_{under}} = (\underbrace{T, F, F, T}_{TB_0:\text{ trans. of Ag.0}} , \underbrace{T, T, T, T}_{TB_1:\text{ trans. of Ag.1}} , \underbrace{F, F, F, F}_{VB:\text{ global prop.}})$$

Notice that M_{over} and M_{under} meet the conditions of monotonicity.

Where do we use M_{over} and M_{under} ?

- At the beginning we have a partial model which is gradually extended to a total model.
- In the next steps of our procedure, we use a model checker, which must have a total model as input.
- Therefore, the partial model must be extended to a total model.

How to compute $\| \phi \|^{Y}_{M_{par}}$?

We use the algorithm computing $\| \phi \|^{Y}_{M_{par},\lambda}$ for $\lambda \in \{over, under\}$ and assume that $\| \phi \|^{Y}_{M_{par}} = \| \phi \|^{Y}_{M_{par},over}$

CASE: $\phi \in \{\neg p\}$

$$\| \phi \|^{Y}_{M_{par},\lambda} = MC(\phi, M^{\mathcal{A}}_{under}) \;\; for \;\; \lambda = over$$

$$\| \phi \|^{Y}_{M_{par},\lambda} = MC(\phi, M^{\mathcal{A}}_{over}) \;\; for \;\; \lambda = under$$

CASE: $\phi \in \{\langle\langle\Gamma\rangle\rangle Xp, \langle\langle\Gamma\rangle\rangle Gp, \langle\langle\Gamma\rangle\rangle pUp\}$

$$\| \phi \|^{Y}_{M_{par},\lambda} = MC(\phi, M^{\Gamma}_{\lambda})$$

$\rightsquigarrow$ $MC(\phi, M)$ is a **model checking** algorithm that returns the set of states satisfying ϕ in M

What about nested formulae ?

CASE: $\phi \in \{\neg\psi, \langle\langle\Gamma\rangle\rangle X\psi, \langle\langle\Gamma\rangle\rangle G\psi\}$

- calculate $\| \psi \|^{Y}_{M_{par},\lambda}$ (recursively)
- replace ψ in ϕ with a new proposition p_ψ which holds in the states of
$$s \in \| \psi \|^{Y}_{M_{par},\lambda}$$
- compute $\| \phi' \|^{Y}_{M'_{par}}$ for $\phi' \in \{\neg p_\psi, \langle\langle\Gamma\rangle\rangle X p_\psi, \langle\langle\Gamma\rangle\rangle G p_\psi\}$, resp.

$\rightsquigarrow$ M' is an extension of M by adding a new proposition p_ψ

Satisfiability procedure

Model requirements

- number of agents
- number of local states of agents
- number of global propositions

Input and output

Input: $Y \in \{i, I\}$, ϕ, ι, and model requirements determining M_{par}.

Output: M s.t. $M, \iota \models_Y \phi$, meeting the requirements of M_{par} or

the answer that such a model does not exist.

Satisfiability procedure

1. $depth := 0$; set $v_{M_{par}}$
2. compute $\parallel \phi \parallel^{Y}_{M_{par}}$
3. if $\iota \in \parallel \phi \parallel^{Y}_{M_{par}}$, then
   (a) if M_{par} is total; SAT; return M_{par}
   (b) otherwise $depth := depth + 1$;
   assign value to an unsigned variable; go to step 2
4. if $\iota \notin \parallel \phi \parallel^{Y}_{M_{par}}$, then
   (a) if $depth > 0$, analyse the conflict, undo decisions up to the conflict depth c and assign the opposite value to the conflicting variable; $depth := c$; go to step 2
   (b) if $depth = 0$ return UNSAT.

MsATL tool architecture

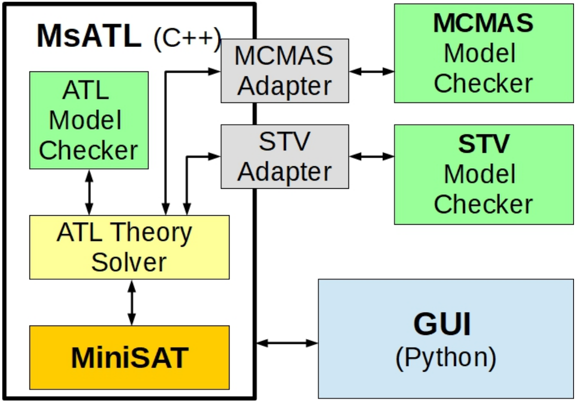


Figure: http://monosatatl.epizy.com

Experimental results for perfect information

Form. 1: $\langle\langle 0\rangle\rangle \mathrm{X}(\neg p_0 \vee \langle\langle 1\rangle\rangle \mathrm{G}(\neg p_1 \vee \langle\langle 0,1\rangle\rangle \mathrm{F}(\neg p_1 \vee \langle\langle 0,1\rangle\rangle \mathrm{F}(\neg p_0 \vee \langle\langle 2\rangle\rangle \mathrm{F}\langle\langle 0\rangle\rangle \mathrm{X}(\neg p_0 \vee \langle\langle 1\rangle\rangle \mathrm{G}(\neg p_1 \vee \langle\langle 0,1\rangle\rangle \mathrm{G}(\langle\langle 0\rangle\rangle \mathrm{F}\neg p_0)))))))$

Table: Experimental results for perfect information.

Id	1	2	3	4	5	6	7	8
Depth	9	13	17	20	23	26	30	33
Connectives	13	19	25	31	35	41	49	55
MsAtl (sec.)	0.22	0.23	0.24	0.31	0.32	0.34	0.38	0.43
TATL (sec.)	0.58	6.2	29.7	74.6	229	552	1382	3948

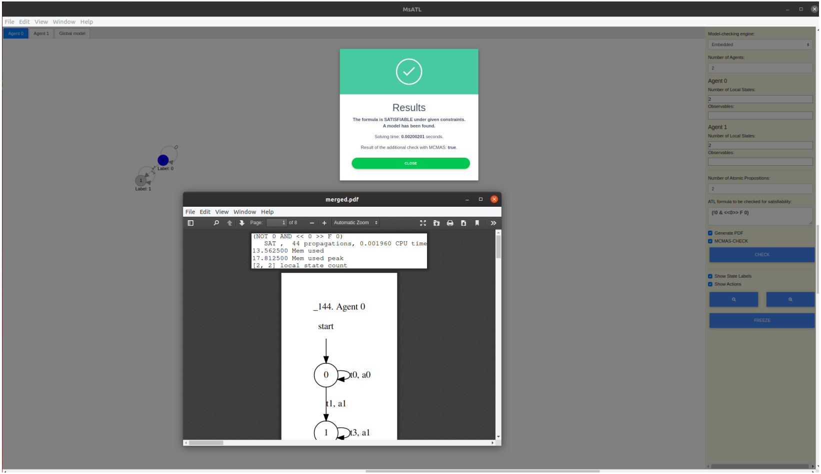
Experimental results for imperfect information

Form. 2: $\langle\langle 2\rangle\rangle \mathrm{F}(\neg p_3 \wedge p_1 \vee p_3 \wedge \langle\langle 0,2\rangle\rangle \mathrm{F}(\neg p_3 \wedge \langle\langle 0,2\rangle\rangle \mathrm{F} p_1 \wedge \neg p_3))$

Table: Experimental results for imperfect information. The 'L' parameter is the number of local states of the agents.

Id	Coalitions	Depth	Connectives	L=2	L=3	L=4	L=5
1	1	2	4	12.1	37.2	88.8	226
2	2	3	9	16.4	52.7	167	542
3	3	3	6	15.8	56.6	163	559
4	3	4	6	22.9	68.1	194	746
5	4	7	6	35.8	124	285	795
6	5	13	13	70.9	265	647	2480
7	5	17	15	88.2	314	744	2365
8	5	21	18	106	383	1110	3470

Live demonstration



Outline

1. Alternating-time Temporal Logic
2. Satisfiability Algorithm for ATL and MsATL tool
3. Strategy Logic with Simple Goals
4. Satisfiability Algorithm for SL[SG] and SGSAT tool

Strategy Logic: shared strategies

- SL: Strategy Logic [F. Mogavero, A. Murano, L. Sauro: Reasoning About Strategies, 2010]
- Strategic plans over temporal goals.
- Nash equilibrium, Stackelberg equilibrium.
- Main idea: strategy quantifiers and agent bindings

$\exists_x$: for some strategy x

$\forall_x$: every strategy x

(x, i): strategy x is used by agent i

Strategy Logic with Simple Goals

A variant of SL: Simple Goal Strategy Logic, SL[SG]

Belardinelli, F.; Jamroga, W.; Kurpiewski, D.; Malvone, V.; and Murano, A.
Strategy logic with simple goals: Tractable reasoning about strategies.
IJCAI 2019

Strategy Logic with Simple Goals

SL[SG]: main syntax constructions

$$\wp\flat\, \mathsf{X}\, \varphi, \quad \wp\flat(\varphi\, \mathsf{U}\, \varphi)$$

$\wp$ - quantification prefix, e.g. $\forall_x \exists_y \forall_z$

$\flat$ - binding prefix, e.g. $(x, 1)(x, 2)(y, 3)$

Assumptions:

- each binding prefix $\flat$ contains every agent of $\mathcal{A}$ (the set of all agents),
- every agent is bound to exactly one variable,
- every variable appearing in $\flat$ is quantified in $\wp$.

As a result: every formula is a **sentence**.

Examples

What is the difference between ATL and SL?

Properties not expressible in ATL

- the quantifiers over variables referring to strategies can be of different types and can appear in any order, e.g. for $\mathcal{A} = \{1, 2, 3, 4\}$

$$\exists_{x_1} \forall_{x_2} \forall_{x_3} \exists_{x_4} (1, x_1)(2, x_2)(3, x_3)(4, x_4)(p \, \mathrm{U} \, q)$$

- two or more agents can be assigned to the same strategy, e.g. for $\mathcal{A} = \{1, 2\}$

$$\exists_x (1, x)(2, x) \, \mathrm{X} \, p$$

Multi-agent system

MAS consists of n **agents** $\mathcal{A} = \{1, 2, \ldots, n\}$.
Every agent $i \in \mathcal{A}$ is associated with:

- L_i - a finite non-empty set of **local states**
- $\iota_i \in L_i$ - an **initial local state**
- Act_i - a finite non-empty set of **local actions**
- $P_i : L_i \to 2^{Act_i} \setminus \{\emptyset\}$ - a local **protocol**
- $T_i : L_i \times Act \to L_i$ - a (partial) **local transition function** such that $T_i(l_i, \alpha)$ is defined iff $\alpha^i \in P_i(l_i)$
- $\mathcal{PV}_i$ - a finite non-empty set of **local propositions**
- $V_i : L_i \to 2^{\mathcal{PV}_i}$ - a local **valuation function**

- $Act = Act_1 \times \cdots \times Act_n$ - the set of **joint global actions**
- $\mathcal{PV} = \bigcup_{i=1}^{n} \mathcal{PV}_i$ - the union of the local propositions

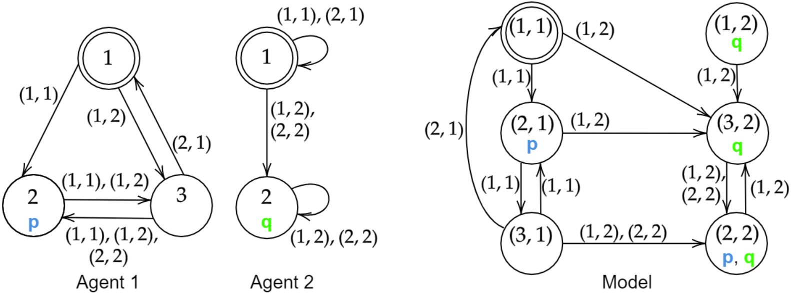
Model

We consider **synchronous** MAS, where each global action is a n-tuple $\langle a_i \rangle_{i \in \mathcal{A}}$ with $a_i \in Act_i$.

Model

The *model* for MAS is a 4-tuple $M = (\mathcal{S}t, \iota, T, V)$ where
- $\mathcal{S}t = L_1 \times \cdots \times L_n$ is a set of the **global states**,
- $\iota = (\iota_1, \ldots, \iota_n) \in \mathcal{S}t$ is the *initial global state*,
- $T : \mathcal{S}t \times Act \to \mathcal{S}t$ is the partial *global transition function*, such that $T(s, \alpha) = s'$ iff $T_i(s_i, \alpha) = s'_i$ for all $i \in \mathcal{A}$
- $V : \mathcal{S}t \to 2^{\mathcal{PV}}$ is the valuation function such that $V((l_1, \ldots, l_n)) = \bigcup_{i=1}^{n} V_i(l_i)$.

Example MAS and model



Strategies

Strategy

A **strategy** is a conditional plan.

We represent strategies by functions $\sigma_i : St^+ \to Act_i$.

$\rightsquigarrow$ memoryfull, perfect information strategies

Semantic Variants of SL

Available information:

- Perfect information (I) vs. imperfect information strategies (i)

Memory of agents:

- Perfect recall (R) vs. memoryless strategies (r)

SGSAT uses memoryfull strategies with perfect information and is the only one dedicated to checking SL[SG].

Shared actions

Shared actions

For every non-empty set $\Gamma \subseteq \mathbb{A}$, the set of **shared actions** of agents Γ is determined $Act_\Gamma \triangleq \cap_{i \in \Gamma} Act_i$.

Such a set must be non-empty if we want to ensure the existence of a non-empty set of **shared strategies** for Γ (i.e. a set of strategies that can be used by every $i \in \Gamma$).

Shared strategies

Shared strategies

If the range of a strategy σ is a subset of $Act_i \cap Act_j$, then we say that the strategy is **shared** by agents i and j

Notation:

$shr(x, \varphi)$ - the set of **all agents** bound to the variable x within the formula φ

$\Sigma(M)_{shr(x,\varphi)}$ - the set of **all strategies** shared by agents from $shr(x, \varphi)$

Assignment

An assignment gives a valuation of variables with strategies, where the latter are used to determine the behavior of agents.

Assignment

An **assignment** is a function

$$\chi : Var \cup \mathcal{A} \to \Sigma(M)$$

such that for every agent $i \in \mathcal{A}$, $\chi(i)$ is a strategy for i.

For $z \in Var \cup \mathcal{A}$ and $\sigma \in \Sigma(M)$, the *variant* χ^z_σ is the assignment that maps z to σ and coincides with χ on all other variables and agents.

Semantics of SL

Semantics of SL

$(M, h, \chi) \models \exists x \psi$ iff there is a strategy $\sigma \in \Sigma(M)_{shr(x,\psi)}$, such that $(M, h, \chi^x_\sigma) \models \psi$

there is a strategy shared by all agents bound to x, s.t. if the agents perform the strategy then ψ holds

$(M, h, \chi) \models (x, i)\psi$ iff $(M, h, \chi^i_{\chi(x)}) \models \psi$

if agent i performs the strategy assigned to x by χ, then ψ holds

Decision problem

Bounded satisfiability problem, SLSG SAT

Decide whether a SL[SG] formula is satisfiable under some fixed initial restrictions on MAS.

Restrictions concern:

- the number of agents,
- local actions,
- local states, and
- local propositions of every agent.

Computational complexity

- SL SAT - highly undecidable - Σ_1^1-HARD, does not have the bounded-tree model property

  Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. Y.
  Reasoning about strategies: on the satisfiability problem.
  Log. Methods Comput. Sci. 13(1), 2017.

- SL[1G] SAT - 2EXPTIME (One-Goal Strategy Logic)

  Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. Y.
  Reasoning about strategies: On the model-checking problem.
  ACM Trans. Comput. Logic 15(4), 2014.

- SL[SG] SAT - no worse than 2EXPTIME

  (since SL[SG] is a sublogic of SL[1G])

- SL[SG] MC - P-Time-complete

  Belardinelli, F.; Jamroga, W.; Kurpiewski, D.; Malvone, V.; and Murano, A.
  Strategy logic with simple goals: Tractable reasoning about strategies.
  IJCAI 2019

Related work

- Inspiration: SAT Modulo Monotonic Theories Solver for CTL

T. Klenze, S. Bayless, A.J. Hu
Fast, Flexible, and Minimal CTL Synthesis via SMT.
Computer Aided Verification, 2016.

- Previous work:

Niewiadomski, A.; Kacprzak, M.; Kurpiewski, D.; Knapik, M.; Penczek, W.; and Jamroga, W.
MsATL: A tool for SAT- based ATL satisfiability checking.
Proc. of AAMAS, 2020

Kacprzak, M.; Niewiadomski, A.; and Penczek, W.
Sat-based ATL satisfiability checking.
Proc. of KR, 2020

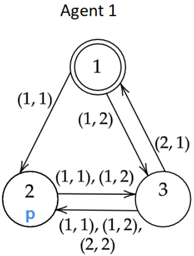
- External MC tools:
  - **MCMAS** - for SL[1G]

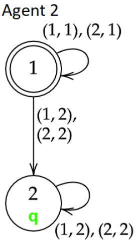
Cermák, P.; Lomuscio, A.; and Murano, A.
Verifying and synthesising multi-agent systems against one-goal strategy logic specifications.
Proc. of AAAI Conference on Artificial Intelligence, 2015.

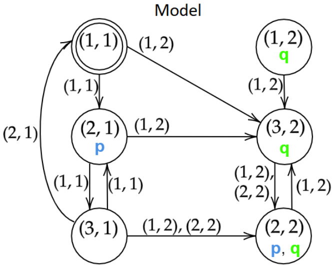
Outline

1. Alternating-time Temporal Logic
2. Satisfiability Algorithm for ATL and MsATL tool
3. Strategy Logic with Simple Goals
4. Satisfiability Algorithm for SL[SG] and SGSAT tool

Boolean encoding of SL[SG] model







PB1

State/Act	1	2
1	T	F
2	T	F
3	T	T

PB2

State/Act	1	2
1	T	T
2	F	T

PB_i - local protocol

TB1

Src/Tgt	1	2	3
1	(F,F,F,F)	(T,F,F,F)	(F,T,F,F)
2	(F,F,F,F)	(F,F,F,F)	(T,T,F,F)
3	(F,F,T,F)	(T,T,F,T)	(F,F,F,F)

TB2

Src/Tgt	1	2
1	(T,F,T,F)	(F,T,F,T)
2	(F,F,F,F)	(F,T,F,T)

TB_i - local transition function

VB1

St/Prop	p
1	F
2	T
3	F

VB2

St/Prop	q
1	F
2	T

VB_i - local valuation

Boolean encoding of SL[SG] model - assumptions

A1 In each local state there is a legal action:

$$\varphi_1 = \bigwedge_{i \leq n; k \leq n_i} \bigvee_{t \leq m_i} pb_i(k, t)$$

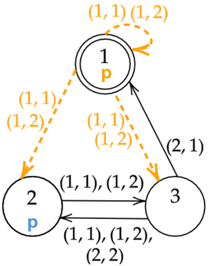
A2 Transition (l_i, α, l'_i) is defined iff α is legal in l_i:

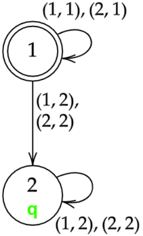
$$\varphi_2 = \bigwedge_{i \leq n; t \leq m_i; k \leq n_i; j \in gl_i(t)} \Big(\big(\bigvee_{k' \leq n_i} tb_i(k, j, k') \big) \leftrightarrow pb_i(k, t) \Big)$$

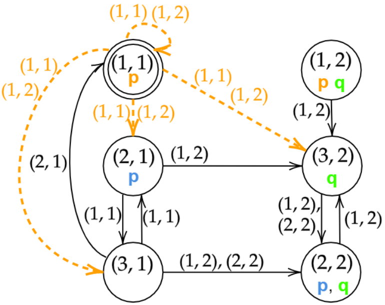
A3 Transition relation is a function:

$$\varphi_3 = \bigwedge_{i \leq n; k, k' \leq n_i; j \leq |Act|} \Big(tb_i(k, j, k') \rightarrow \bigwedge_{k'' \leq n_i, k'' \neq k'} \neg tb_i(k, j, k'') \Big).$$

Partial model







PB1

State/Act	1	2
1	U	F
2	T	F
3	T	T

PB2

State/Act	1	2
1	T	T
2	F	T

TB1

Src/Tgt	1	2	3
1	(U,U,F,F)	(U,U,F,F)	(U,U,F,F)
2	(F,F,F,F)	(F,F,F,F)	(T,T,F,F)
3	(F,F,T,F)	(T,T,F,T)	(F,F,F,F)

TB2

Src/Tgt	1	2
1	(T,F,T,F)	(F,T,F,T)
2	(F,F,F,F)	(F,T,F,T)

VB1

St/Prop	p
1	U
2	T
3	F

VB2

St/Prop	q
1	F
2	T

PB_i- local protocol

TB_i- local transition function

VB_i- local valuation

possible transition

p possible valuation

Monotonicity property

The implication

$$\text{if } M, h, \chi \models \phi \text{ then } M', h, \chi \models \phi$$

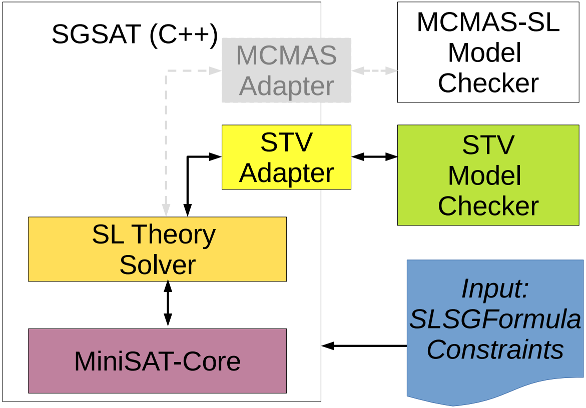
holds if the following conditions are satisfied

CASE: $\phi \in \{\wp\flat\, \mathsf{X}\, p, \wp\flat(p_1\, \mathsf{U}\, p_2)\}$ for $p, p_1, p_2 \in \mathcal{PV}$

- positive monotonicity wrt. local transitions and local protocol of agents $i \in E(\wp\flat)$: $v_M(b_i) \leq v_{M'}(b_i)$ for each $b_i \in TB_i \cup PB_i$
- negative monotonicity wrt. local transitions and local protocol of agents $i \in A(\wp\flat)$: $v_M(b_i) \geq v_{M'}(b_i)$ for each $b_i \in TB_i \cup PB_i$
- positive monotonicity wrt. propositions: $v_M(vb) \leq v_{M'}(vb)$ for each $vb \in VB$

  $\rightsquigarrow$ assuming that $F < T$

SGSAT tool architecture



Preliminary experimental results for SL[SG]

$\varphi_1 = \exists x_1 \ldots \exists x_n (x_1, 1) \ldots (x_n, n) \mathrm{F}(p_1^1 \wedge \ldots \wedge p_n^1)$, and
$\varphi_2 = \forall x_1 \exists x_2 \ldots \exists x_n (x_1, 1) \ldots (x_n, n) \mathrm{F}(p_1^1 \wedge \ldots \wedge p_n^1)$.

					φ_1		φ_2	
n	ls	la	lp	vars	satT	runT	satT	runT
2	2	2	2	48	0.05	1.69	0.05	1.14
2	3	2	2	96	0.42	4.19	0.39	4.37
2	4	2	2	160	2.60	17.5	2.85	20.0
2	5	2	2	240	19.8	125	21.5	132
2	2	5	2	228	0.43	6.08	0.4	6.26
3	2	2	2	120	0.56	5.22	0.48	5.24
3	3	2	2	252	37.4	237	41.1	262
3	2	3	2	354	3.88	29.9	3.83	29.6
3	2	4	2	804	18.9	147	19.9	155
3	2	5	2	1542	66.9	607	79.8	708
4	2	2	2	288	13.6	96.4	13.9	101
4	2	3	2	1336	257	1700	270	2462
5	2	2	2	680	501	4121	423	3397

Table: The number of: agents, local states, local actions, local propositions, and variables encoding MAS. Next, the time consumed by SAT-solver, and the total runtime (in seconds).


Thank You!