Partial Order Reductions for Temporal, Epistemic, and Strategy Logics

Everything you always wanted to know about POR ....
but were afraid to ask for

Wojciech Penczek

Institute of Computer Sciences, PAS, Warsaw, Poland

WG2.2 Meeting, Vienna, the 24th of September 2019

Outline

- Methods of state space reductions
- Some history of Partial Order Reductions (POR)
- POR for temporal logics: LTL-X, CTL*-X
- POR for epistemic logics: LTLK-X, CTL*K-X
- POR for strategy logics: $sATL^*_{ir}$ and $sATL^*_{iR}$

Model checking for modal logics

Model checking problem

$$M, s \overset{?}{\models} \varphi$$

a Kripke model — a modal formula

Complexity

From P-Time to undecidable.
But, $|M|$ is typically exponential in the size of a system !!!

Model checking for modal logics

Model checking problem

$$M, s \stackrel{?}{\models} \varphi$$

a Kripke model a modal formula

Complexity

From P-Time to undecidable.
But, $|M|$ is typically exponential in the size of a system !!!

Possible solutions

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - multi-valued model checking over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Symmetry reductions - model checking over smaller models for CTLK (see Cohen, Dams, Lomuscio, Qu)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- Simpler strategies - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible solutions

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - multi-valued model checking over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Symmetry reductions - model checking over smaller models for CTLK (see Cohen, Dams, Lomuscio, Qu)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- Simpler strategies - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible solutions

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - multi-valued model checking over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Symmetry reductions - model checking over smaller models for CTLK (see Cohen, Dams, Lomuscio, Qu)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- Simpler strategies - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible solutions

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - multi-valued model checking over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Symmetry reductions - model checking over smaller models for CTLK (see Cohen, Dams, Lomuscio, Qu)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- Simpler strategies - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible solutions

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - multi-valued model checking over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Symmetry reductions - model checking over smaller models for CTLK (see Cohen, Dams, Lomuscio, Qu)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- Simpler strategies - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible solutions

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - multi-valued model checking over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Symmetry reductions - model checking over smaller models for CTLK (see Cohen, Dams, Lomuscio, Qu)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- Simpler strategies - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Possible solutions

- Symbolic model checking - BDD-based (Lomuscio, Raimondi), SAT-based Unbounded Model Checking for ATL (Kacprzak, Lomuscio, Penczek)
- Abstractions - multi-valued model checking over abstract models for variants of ATL(K) (Belardinelli, Lomuscio, Michaliszyn)
- Bisimulation-based reductions - for ATL_{ir} (Belardinelli, Condurache, Dima, ...)
- Symmetry reductions - model checking over smaller models for CTLK (see Cohen, Dams, Lomuscio, Qu)
- Upper and lower approximations - for ATL_{ir} (Jamroga, Knapik, Kurpiewski)
- **Partial order reductions** - model checking over smaller models for LTLK-X, CTLK-X, sATL* (Lomuscio, Penczek, Qu, Jamroga, ...)
- Simpler strategies - counting strategies for TATL (Andre, Jamroga, Knapik, Penczek, Petrucci)

Partial Order Reductions

Idea

- This is a method of generating reduced state spaces of distributed systems which preserve properties of our interest.
- The reduction exploits the idea that when a property does not distinguish between the interleavings of the same (Mazurkiewicz) trace, then it is sufficient to generate a reduced state space which contains only one interleaving for each trace.
- In practice one generates more than one interleaving per trace, but as few as possible.

History of Partial Order Reductions

Three Big Names

- Antti Valmari, ICATPN 1989 - stubborn sets
- Patrice Godefroid, CAV 1990, CAV 1991 - sleep sets
- Doron Peled, CONCUR 1992 - ample sets

Syntax

I assume that you are familiar with LTL, CTL*, and epistemic logics ...

Syntax of ATL*:

$$\phi ::= \mathsf{p} \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \langle\langle A \rangle\rangle\gamma,$$

$$\gamma ::= \phi \mid \gamma \wedge \gamma \mid \gamma \vee \gamma \mid \mathsf{X}\,\gamma \mid \gamma\,\mathsf{U}\,\gamma \mid \gamma R \gamma,$$

where $p \in \mathcal{AP}$ and A - a set o agents.

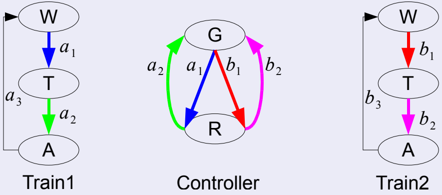


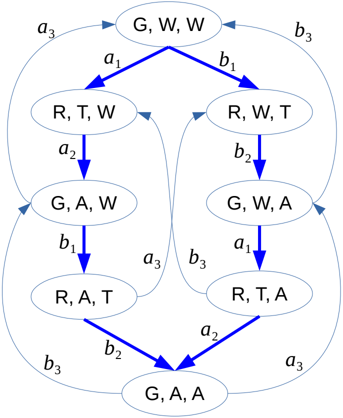
Figure: TC composed of two trains and the controler

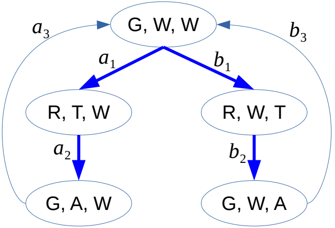
Interleaved Interpreted Systems

A Model is tuple
$\mathcal{A} = (Agents, Act, \mathcal{Q}, \mathcal{AP}, \mathcal{V}, prot, trans, \{\sim_i | \ i \in Agents\})$, s.t.:

- $Agents$ is a finite set of all the agents,
- $Act = A_1 \cup \ldots \cup A_n$ is a finite set of actions,
- $\mathcal{Q} = L_1 \times \ldots \times L_n$ is a finite set of global locations (states),
- $\mathcal{V}\colon \mathcal{Q} \to 2^{\mathcal{AP}}$ is a valuation function,
- $prot_i\colon L_i \to 2^{A_i}$ - a protocol function of agent i,
- $t_i : L_i \times A_i \to L_i$ - an i-local evolution partial function,
- $trans : \mathcal{Q} \times Act \to \mathcal{Q}$ - an interleaved evolution partial function: $trans((g_1, \ldots, g_n), \mathrm{act}) = (g'_1, \ldots, g'_n)$ iff $t_i(g_i, \mathrm{act}) = g'_i$ if $\mathrm{act} \in A_i$ and $g_i = g'_i$ if $\mathrm{act} \notin A_i$,
- $g \sim_i g'$ iff $g_i = g'_i$ for each $i \in Agents$ - the indistinguishabilty relations.

the full model	a reduced model





Semantics of ATL*: ($Y \in \{IR, iR, Ir, ir\}$).

$M, g \models_Y \langle\langle A \rangle\rangle \gamma$ iff
there is a joint Y-strategy σ_A for agents A such that,
for each path $\pi \in out_M(g, \sigma_A)$, we have $M, \pi \models_Y \gamma$, where

- I - complete information, i - incomplete information,
- R - perfect recall, r - imperfect recall.

Properties of TGC in ATL*:

- $\langle\langle c \rangle\rangle G(\neg \text{in\_tunnel}_1)$ - the controller can keep Train 1 out,
- $\langle\langle c \rangle\rangle F(\text{in\_tunnel}_1 \wedge F \neg \text{in\_tunnel}_1)$ - the controller can let Train 1 through,

Explaining the idea behind POR

POR aims at generating reduced models, preserving some temporal formula ψ.

- Independency of actions
  $Ind = \{(a, b) \mid Agents(a) \cap Agents(b) = \emptyset\}$, restricted such that either a or b is invisible, i.e., does not change the valuations of the atomic propositions used in ψ,
- Two infinite sequences of global locations and actions: $g_0 a_0 g_1 a_1 \ldots$ and $g_0 a'_0 g'_1 a'_1 \ldots$ that differ in the ordering of independent actions only are called trace equivalent,
- ψ does not distinguish between trace-equivalent sequences.

Algorithm DFS-POR

DFS-POR is used to compute paths of the reduced model M'.
A stack represents the path $\pi = g_0 a_0 g_1 a_1 \cdots g_n$ currently being visited.
For g_n, the following three operations are computed in a loop:

1. The set $en(g_n) \subseteq Act$ of enabled actions is identified and a subset $E(g_n) \subseteq en(g_n)$ of necessary actions is heuristically selected.
2. For any action $a \in E(g_n)$ compute the successor state g' of g_n such that $g_n \stackrel{a}{\to} g'$, and add g' to the stack.
   Recursively proceed to explore the submodel originating at g'.
3. Remove g_n from the stack.

Catch

The problem of computing a minimal $E(g)$ is NP-complete.

Algorithm DFS-POR

DFS-POR is used to compute paths of the reduced model M'.
A stack represents the path $\pi = g_0 a_0 g_1 a_1 \cdots g_n$ currently being visited.
For g_n, the following three operations are computed in a loop:

1. The set $en(g_n) \subseteq Act$ of enabled actions is identified and a subset $E(g_n) \subseteq en(g_n)$ of necessary actions is heuristically selected.
2. For any action $a \in E(g_n)$ compute the successor state g' of g_n such that $g_n \xrightarrow{a} g'$, and add g' to the stack. Recursively proceed to explore the submodel originating at g'.
3. Remove g_n from the stack.

Catch

The problem of computing a minimal $E(g)$ is NP-complete.

Algorithm DFS-POR

DFS-POR is used to compute paths of the reduced model M'.
A stack represents the path $\pi = g_0 a_0 g_1 a_1 \cdots g_n$ currently being visited.
For g_n, the following three operations are computed in a loop:

1. The set $en(g_n) \subseteq Act$ of enabled actions is identified and a subset $E(g_n) \subseteq en(g_n)$ of necessary actions is heuristically selected.
2. For any action $a \in E(g_n)$ compute the successor state g' of g_n such that $g_n \stackrel{a}{\to} g'$, and add g' to the stack.
   Recursively proceed to explore the submodel originating at g'.
3. Remove g_n from the stack.

Catch

The problem of computing a minimal $E(g)$ is NP-complete.

Algorithm DFS-POR

DFS-POR is used to compute paths of the reduced model M'.
A stack represents the path $\pi = g_0 a_0 g_1 a_1 \cdots g_n$ currently being visited.
For g_n, the following three operations are computed in a loop:

1. The set $en(g_n) \subseteq Act$ of enabled actions is identified and a subset $E(g_n) \subseteq en(g_n)$ of necessary actions is heuristically selected.
2. For any action $a \in E(g_n)$ compute the successor state g' of g_n such that $g_n \stackrel{a}{\to} g'$, and add g' to the stack.
   Recursively proceed to explore the submodel originating at g'.
3. Remove g_n from the stack.

Catch

The problem of computing a minimal $E(g)$ is NP-complete.

Conditions for selection of $E(g)$

Basic Conditions

C1 Along each path π in M that starts at g, each action $a \in Act \setminus E(g)$ that is dependent on an action in $E(g)$ cannot be executed in π without an action in $E(g)$ is executed first.

C2 If $E(g) \neq en(g)$, then each action in $E(g)$ is invisible,

C3 For every cycle in M' there is at least one node g in that cycle for which $E(g) = en(g)$.

Conditions for selection of $E(g)$

Basic Conditions

- **C1** Along each path π in M that starts at g, each action $a \in Act \setminus E(g)$ that is dependent on an action in $E(g)$ cannot be executed in π without an action in $E(g)$ is executed first.
- **C2** If $E(g) \neq en(g)$, then each action in $E(g)$ is invisible,
- **C3** For every cycle in M' there is at least one node g in that cycle for which $E(g) = en(g)$.

Conditions for selection of $E(g)$

Basic Conditions

- **C1** Along each path π in M that starts at g, each action $a \in Act \setminus E(g)$ that is dependent on an action in $E(g)$ cannot be executed in π without an action in $E(g)$ is executed first.
- **C2** If $E(g) \neq en(g)$, then each action in $E(g)$ is invisible,
- **C3** For every cycle in M' there is at least one node g in that cycle for which $E(g) = en(g)$.

Equivalence on states and paths

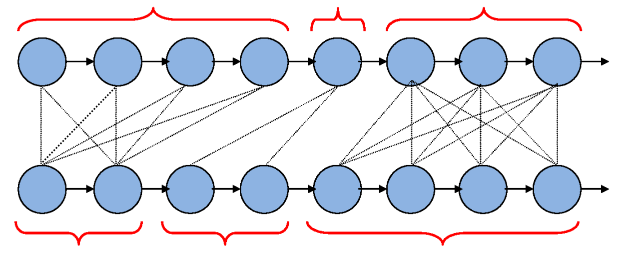


Figure: Two stuttering equivalent paths π and π'

A dotted line between two states g and g' means that $V(g) = V(g')$.

POR for LTL-X

[Peled 1992]

- Logic: LTL-X
- Equivalence induced on models: stuttering trace equivalence,
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any LTL-X formula φ,
- If $E(g)$ satisfies **C1,C3**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any LTL-X formula φ.

CF Concurrency Fairness - no action can be eventually always enabled in a path and be independent of the executed actions.

POR for LTL-X

[Peled 1992]

- Logic: LTL-X
- Equivalence induced on models: stuttering trace equivalence,
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any LTL-X formula φ,
- If $E(g)$ satisfies **C1,C3**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any LTL-X formula φ.

CF Concurrency Fairness - no action can be eventually always enabled in a path and be independent of the executed actions.

POR for CTL*-X

[Gerth, Kuiper, Peled, Penczek 1995]

- Logic: CTL*-X
- Equivalence induced on models: stuttering bisimulation,
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3, C4**, then
  $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any CTL*-X formula φ,
- If $E(g)$ satisfies **C1, C3, C4**, then
  $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any CTL*-X formula φ.

C4 If $E(g) \neq en(g)$, then $E(g)$ is a singleton.

[Gerth, Kuiper, Peled, Penczek 1995]

- Logic: CTL*-X
- Equivalence induced on models: stuttering bisimulation,
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3, C4**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any CTL*-X formula φ,
- If $E(g)$ satisfies **C1, C3, C4**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any CTL*-X formula φ.

C4 If $E(g) \neq en(g)$, then $E(g)$ is a singleton.

Equivalence on states and paths

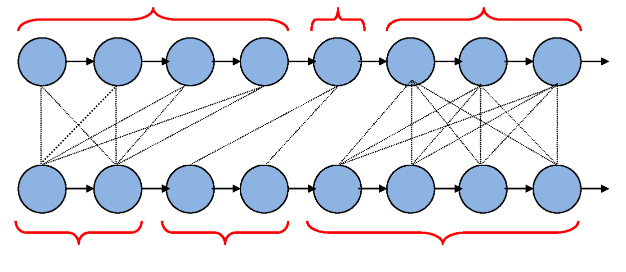


Figure: Two J-stuttering equivalent paths π and π'

$J \subseteq Agents$. A dotted line between two states g and g' means that $V(g) = V(g')$ and $g \sim_J g'$.

$$M, g \models K_i\gamma \text{ iff for all } g' \in Q \text{ if } g \sim_i g' \text{ we have } M, g' \models \gamma.$$

POR for LTLK^J-X (only K_i with $i \in J$)

[Lomuscio, Penczek, Qu, AAMAS 2010]

- Logic: LTLK^J-X
- Equivalence induced on models: J-stuttering trace equivalence,
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3, CJ**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any LTLK^J-X formula φ,
- If $E(g)$ satisfies **C1, C3, CJ**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any LTLK^J-X formula φ.

CJ No action in E(g) changes local states of the agents in J.

[Lomuscio, Penczek, Qu, AAMAS 2010]

- Logic: LTLK^J-X
- Equivalence induced on models: J-stuttering trace equivalence,
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3, CJ**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any LTLK^J-X formula φ,
- If $E(g)$ satisfies **C1, C3, CJ**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any LTLK^J-X formula φ.

CJ No action in E(g) changes local states of the agents in J.

POR for CTL*K^J-X (only K_i with $i \in J$)

[Lomuscio, Penczek, Qu, FI 2010]

- Logic: CTL*K^J-X
- Equivalence induced on models: J-stuttering bisimulation,
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3, C4, CJ**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any CTL*K^J-X formula φ,
- If $E(g)$ satisfies **C1, C3, C4, CJ**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any CTL*K^J-X formula φ.

C4 If $E(g) \neq en(g)$, then $E(g)$ is a singleton.

CJ No action in E(g) changes local states of the agents in J.

POR for CTL*K^J-X (only K_i with $i \in J$)

[Lomuscio, Penczek, Qu, FI 2010]

- Logic: CTL*K^J-X
- Equivalence induced on models: J-stuttering bisimulation,
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3, C4, CJ**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any CTL*K^J-X formula φ,
- If $E(g)$ satisfies **C1, C3, C4, CJ**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any CTL*K^J-X formula φ.

C4 If $E(g) \neq en(g)$, then $E(g)$ is a singleton.

CJ No action in E(g) changes local states of the agents in J.

sATL* over interleaved models

Restrictions of ATL*

- sATL* (simple ATL*) - ATL* without the next state operator and without nested strategic operators,
- sATL_{ir}, sATL^{*}_{ir}
- Model checking $sATL_{ir}$ and sATL^{*}_{ir} is PSPACE-complete in the size of the model representation and the length of a formula.
- sATL_{iR}, sATL^{*}_{iR}
- Model checking $sATL_{iR}$ and sATL^{*}_{iR} is undecidable.

POR for sATL$^*_{ir}$

[Dembiński, Jamroga, Mazurkiewicz, Penczek, AAMAS 2018, Best Paper Award Nomination]

- Logic: sATL$^*_{ir}$
- Equivalence induced on models: ?!?
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any sATL$^*_{ir}$ formula φ that refers only to coalitions A, where the actions of A are visible,
- If $E(g)$ satisfies **C1,C3**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any sATL$^*_{ir}$ formula φ.

Remark: the above theorem does not hold for sATL$^*_{Ir}$.

POR for $sATL^*_{ir}$

[Dembiński, Jamroga, Mazurkiewicz, Penczek, AAMAS 2018, Best Paper Award Nomination]

- Logic: $sATL^*_{ir}$
- Equivalence induced on models: ?!?
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any $sATL^*_{ir}$ formula φ that refers only to coalitions A, where the actions of A are visible,
- If $E(g)$ satisfies **C1,C3**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any $sATL^*_{ir}$ formula φ.

Remark: the above theorem does not hold for $sATL^*_{Ir}$.

POR for sATL$^*_{iR}$

[Jamroga, Penczek, Sidoruk, 2019]

- Logic: sATL$^*_{iR}$
- Equivalence induced on models: ?!?
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any sATL$^*_{iR}$ formula φ that refers only to coalitions A, where the actions of A are visible,
- If $E(g)$ satisfies **C1,C3**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any sATL$^*_{iR}$ formula φ.

Remark: the above theorem does not hold for sATL$^*_{iR}$.

[Jamroga, Penczek, Sidoruk, 2019]

- Logic: $sATL^*_{iR}$
- Equivalence induced on models: ?!?
- $M' \subseteq M$ - the reduced model generated by DFS-POR
- If $E(g)$ satisfies **C1, C2, C3**, then $M, g^0 \models \varphi$ iff $M', g^0 \models \varphi$, for any $sATL^*_{iR}$ formula φ that refers only to coalitions A, where the actions of A are visible,
- If $E(g)$ satisfies **C1,C3**, then $M, g^0 \models_{CF} \varphi$ iff $M', g^0 \models_{CF} \varphi$, for any $sATL^*_{iR}$ formula φ.

Remark: the above theorem does not hold for $sATL^*_{IR}$.

Experimental Results - Trains and Controller (TC)

Modified partial order reduction algorithms for LTL-X can be used for $sATL^*_{ir}$ and $sATL^*_{iR}$.

Property: Controller has a strategy to keep Train 1 out of the tunnel:

$$\langle\langle c \rangle\rangle G(\neg \text{in\_tunnel}_1)$$

Models for n trains

$F(n) \geq 2^{n+1}$ - the size of the full model.
$R(n) = 2n + 1$ - the size of the reduced model.
The reduced model is *exponentially smaller* than the full one.

More benchmarks

We have experimental results for Faulty TGC, Simple Voting Protocol, and Bridge Endplays with n cards, amounting to $40\% - 90\%$ reductions of the state spaces.

Experimental Results - Trains and Controller (TC)

Modified partial order reduction algorithms for LTL-X can be used for $sATL^*_{ir}$ and $sATL^*_{iR}$.

Property: Controller has a strategy to keep Train 1 out of the tunnel:

$$\langle\langle c \rangle\rangle G(\neg \text{in\_tunnel}_1)$$

Models for *n* trains

$F(n) \geq 2^{n+1}$ - the size of the full model.
$R(n) = 2n + 1$ - the size of the reduced model.
The reduced model is *exponentially smaller* than the full one.

More benchmarks

We have experimental results for Faulty TGC, Simple Voting Protocol, and Bridge Endplays with n cards, amounting to $40\% - 90\%$ reductions of the state spaces.

Experimental Results - Trains and Controller (TC)

Modified partial order reduction algorithms for LTL-X can be used for sATL^*_{ir} and sATL^*_{iR}.

Property: Controller has a strategy to keep Train 1 out of the tunnel:

$$\langle\langle c \rangle\rangle G(\neg \mathrm{in\_tunnel}_1)$$

Models for n trains

$F(n) \geq 2^{n+1}$ - the size of the full model.
$R(n) = 2n + 1$ - the size of the reduced model.
The reduced model is *exponentially smaller* than the full one.

More benchmarks

We have experimental results for Faulty TGC, Simple Voting Protocol, and Bridge Endplays with n cards, amounting to $40\% - 90\%$ reductions of the state spaces.

Future work

- Combining POR with model checking methods for sATL*$_{ir}$
- Symbolic on-the-fly model checking for sATL*$_{ir}$
- Application to e-voting protocols

Thank You !