Squeezing State Spaces of (Attack-Defence) Trees

Michał Knapik[1] Wojciech Penczek[1]
Laure Petrucci[2] **Teofil Sidoruk**[1]

[1]Institute of Computer Science, Polish Academy of Sciences

[2]LIPN, CNRS UMR 7030, Université Sorbonne Paris Nord

LAMAS
May 10, 2020

Motivation

Attack-Defence Trees [Kordy et al., 2011, Aslanyan and Nielson, 2015]

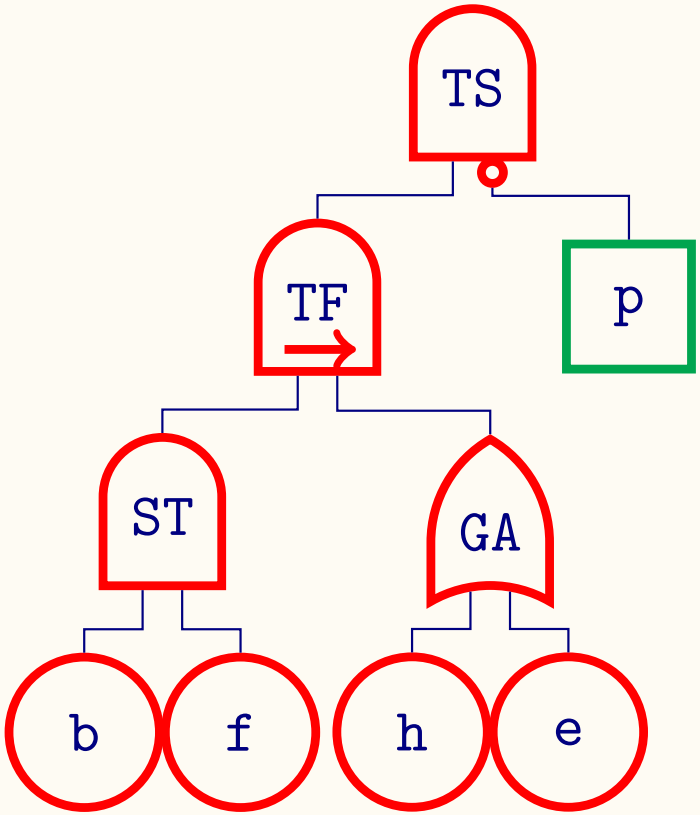
allow for studying interactions between attacker and defender parties:

- performance
- feasibility

An agent-aware model

- asynchronous multi-agent systems, an automata-based formalism [Jamroga et al., 2018]
- extended with attributes and functions
- quantitative and qualitative analysis

Attack-Defence Trees



Name	Cost	Time
TS (treasure stolen)		
p (police)	€100	10 min
TF (thieves fleeing)		
ST (steal treasure)		2 min
b (bribe gatekeeper)	€500	1 h
f (force arm. door)	€100	2 h
GA (get away)		
h (helicopter)	€500	3 min
e (emergency exit)		10 min

Condition for TS:

$$init\_time(\mathtt{p}) > init\_time(\mathtt{ST}) + time(\mathtt{GA})$$

Exhaustive analysis

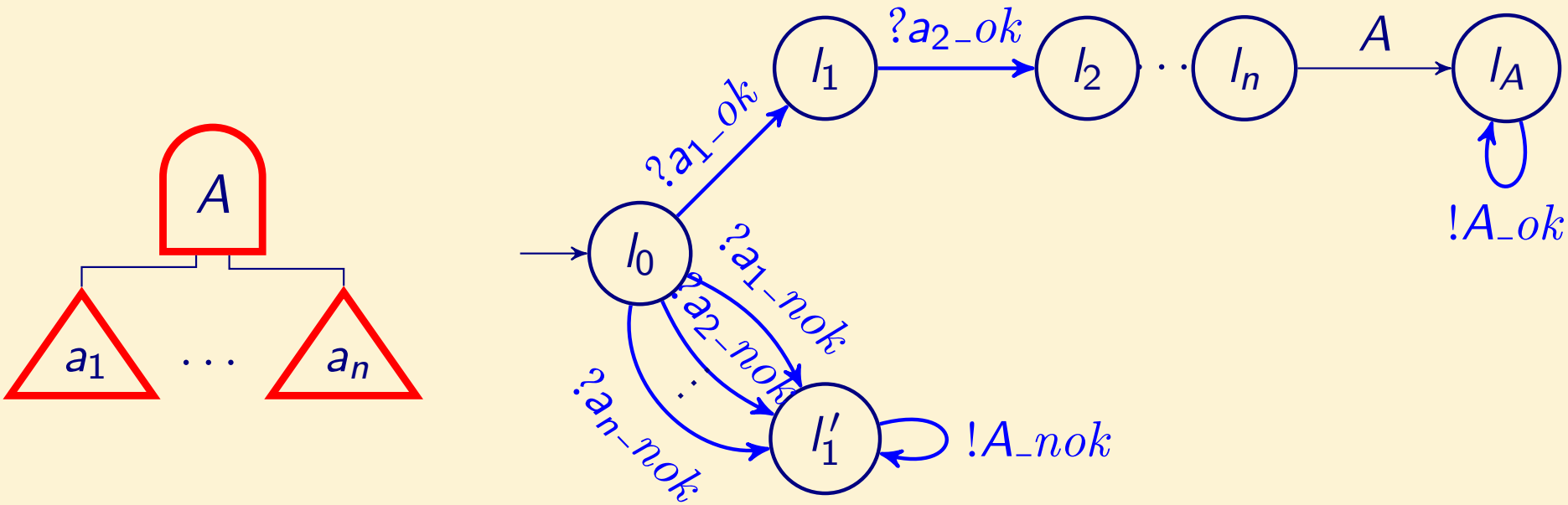
- Build the EAMAS by replacing each ADTree node by an automaton
- State space explosion

Modelling with reduced patterns [Arias et al., 2019]

Exhaustive analysis

- Build the EAMAS by replacing each ADTree node by an automaton
- State space explosion

Modelling with reduced patterns [Arias et al., 2019]

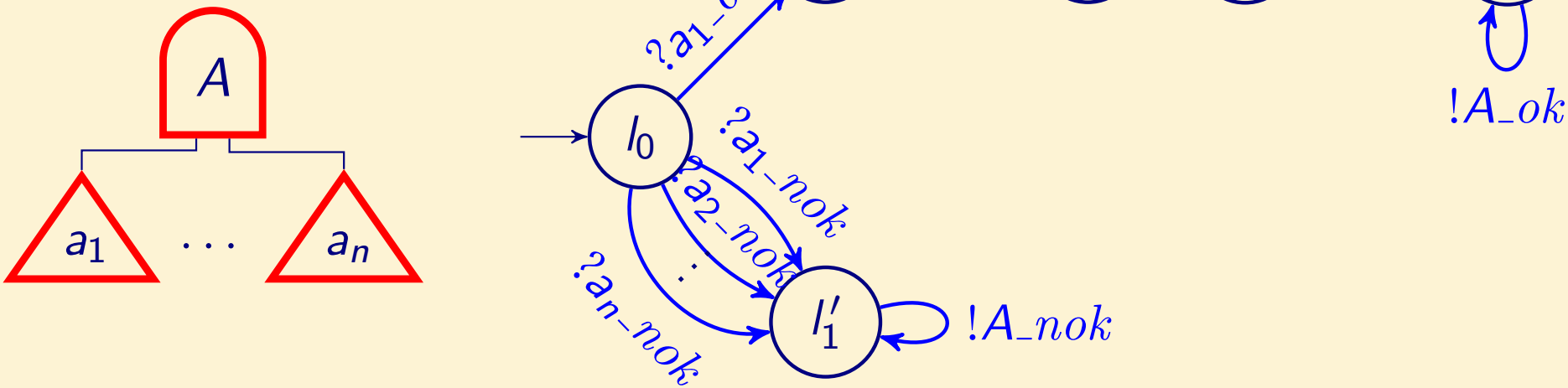


Exhaustive analysis

- Build the EAMAS by replacing each ADTree node by an automaton
- State space explosion

Modelling with reduced patterns [Arias et al., 2019]

Goes beyond POR!

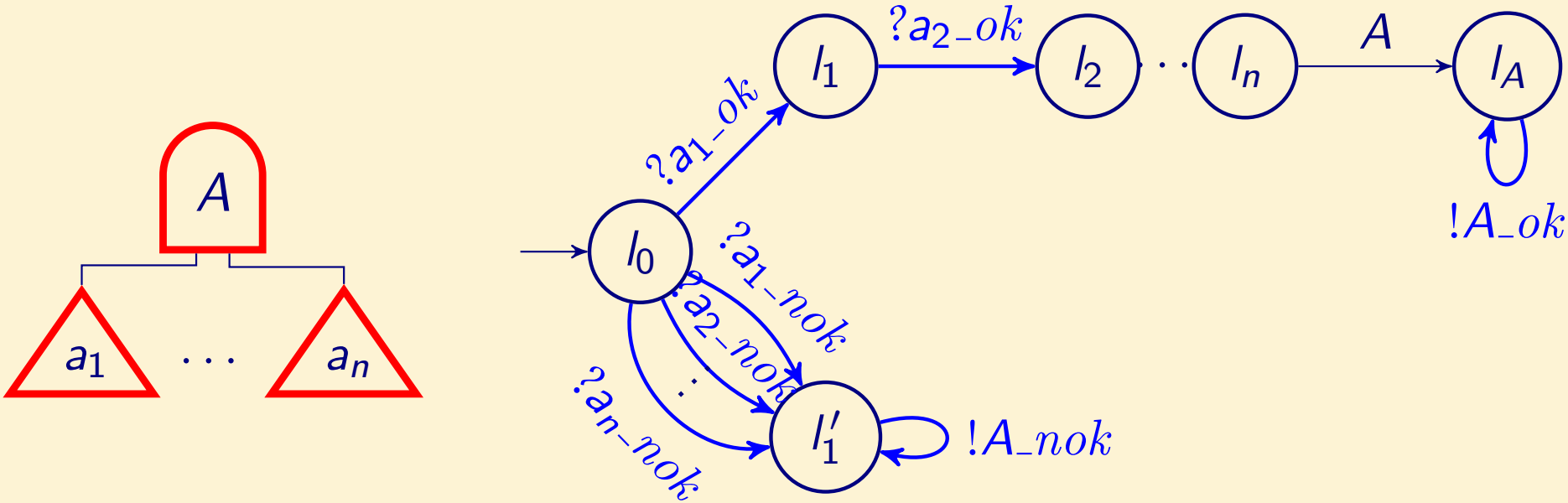


Exhaustive analysis

- Build the EAMAS by replacing each ADTree node by an automaton
- State space explosion

Modelling with reduced patterns [Arias et al., 2019]

Goes beyond POR!



Patterns state space reduction is not enough!

Outline

Guarded Update Systems

Asynchronous product of automata equipped with:

- integer variables
- guards: boolean formulae over linear terms on variables
- updates: assignments obtained by functions over variables
- in synchronised transitions, a variable should not be updated more than once

$\mathcal{GUS}$ synchronisation topology

- nodes: individual automata
- edges: connect nodes that share a synchronised transition

Guarded Update Systems

Asynchronous product of automata equipped with:

- integer variables
- guards: boolean formulae over linear terms on variables
- updates: assignments obtained by functions over variables
- in synchronised transitions, a variable should not be updated more than once

$\mathcal{GUS}$ synchronisation topology

- nodes: individual automata
- edges: connect nodes that share a synchronised transition

Properties for Tree Topologies

Precedence

Actions synchronised with children actions occur before the other ones

Root-directed Synchronisation Tree

Precedence is satisfied for the whole tree

Update separability

- a variable is updated in at most one component
- it is tested only in the ancestors of this component in the tree

ADTrees topologies are root-directed and update-separable

Properties for Tree Topologies

Precedence

Actions synchronised with children actions occur before the other ones

Root-directed Synchronisation Tree

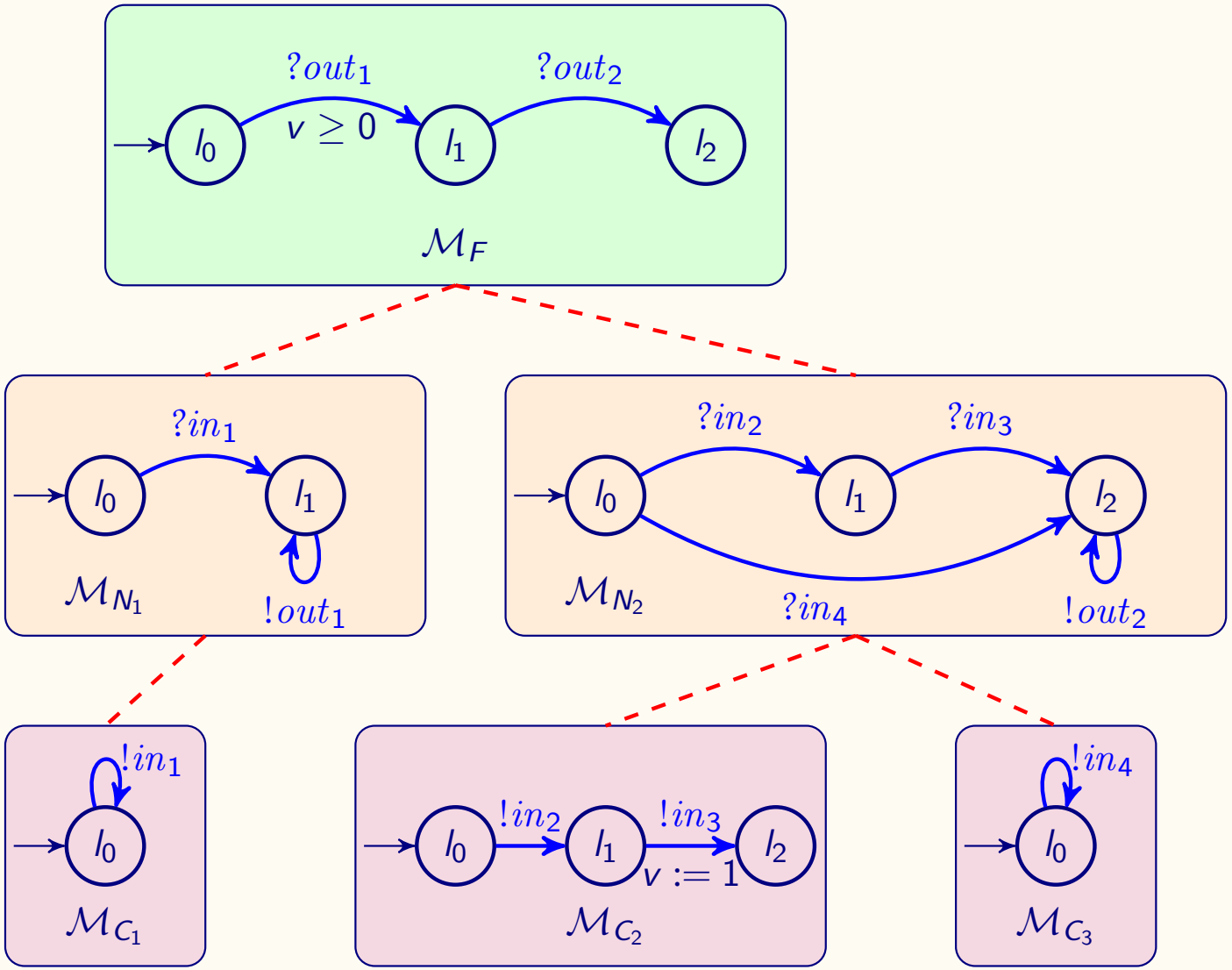
Precedence is satisfied for the whole tree

Update separability

- a variable is updated in at most one component
- it is tested only in the ancestors of this component in the tree

ADTrees topologies are root-directed and update-separable

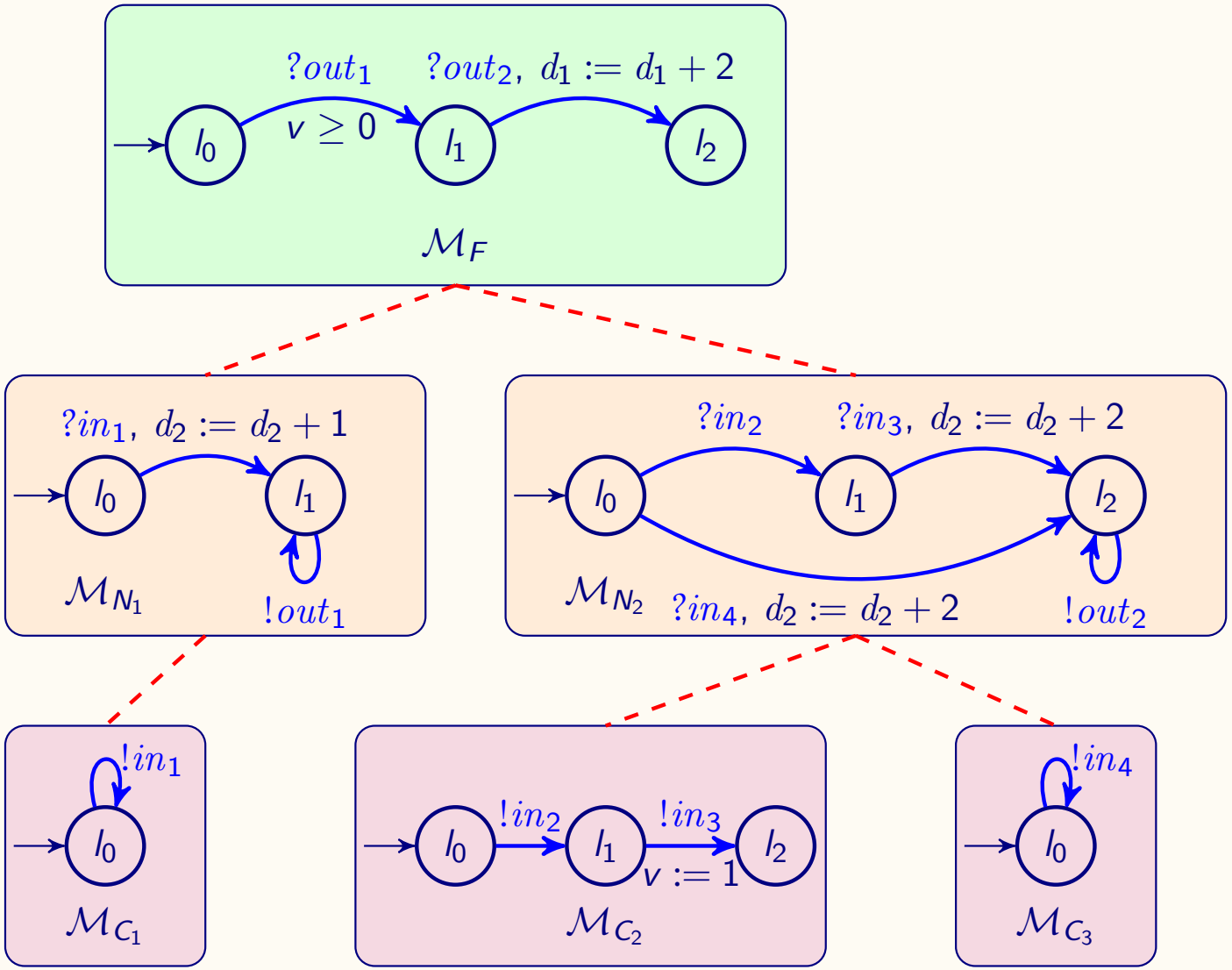
Layered Reduction for Trees



State space size

Full: 14 states, 19 edges Reduced: 12 states, 14 edges

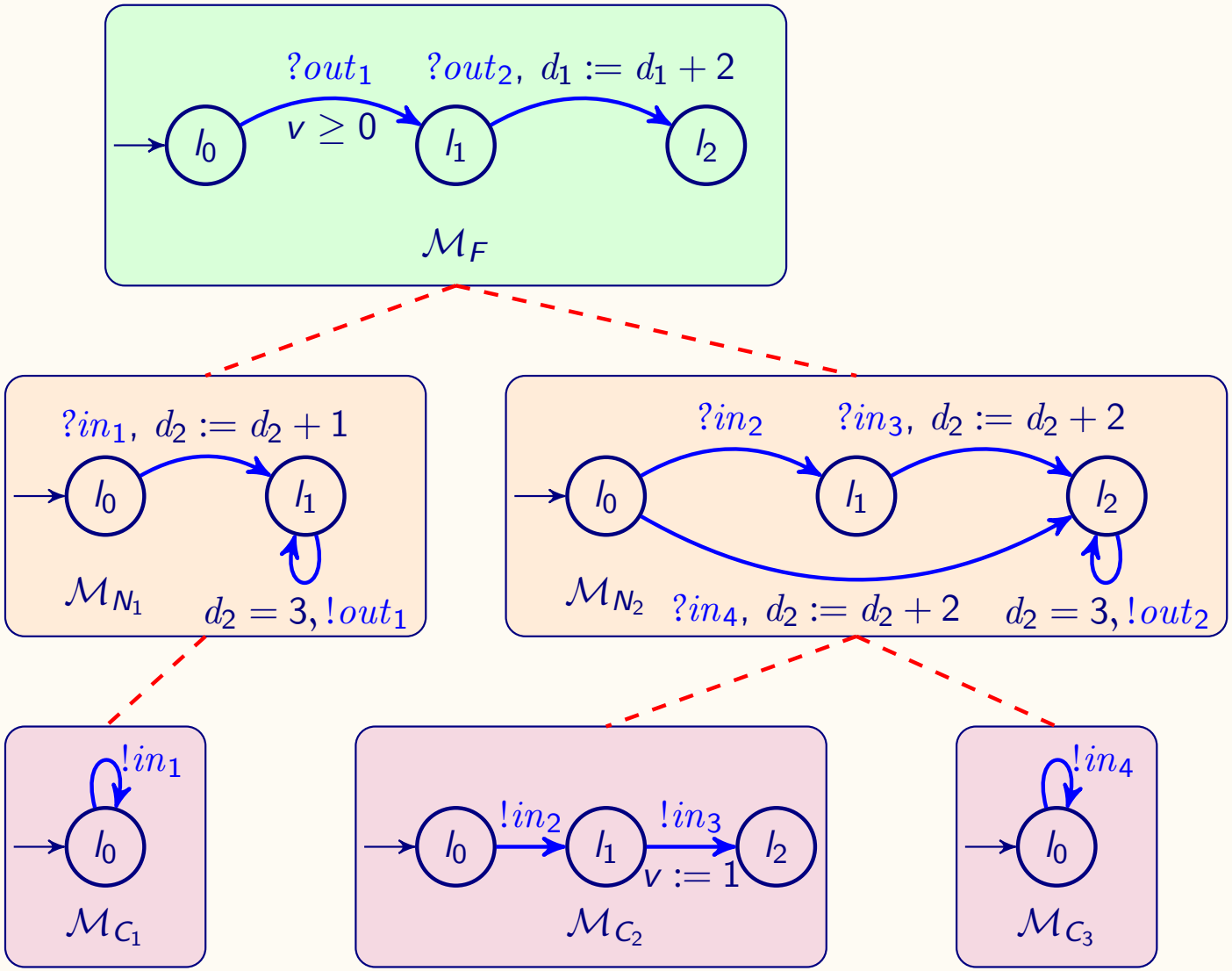
Layered Reduction for Trees



State space size

Full: 14 states, 19 edges Reduced: 12 states, 14 edges

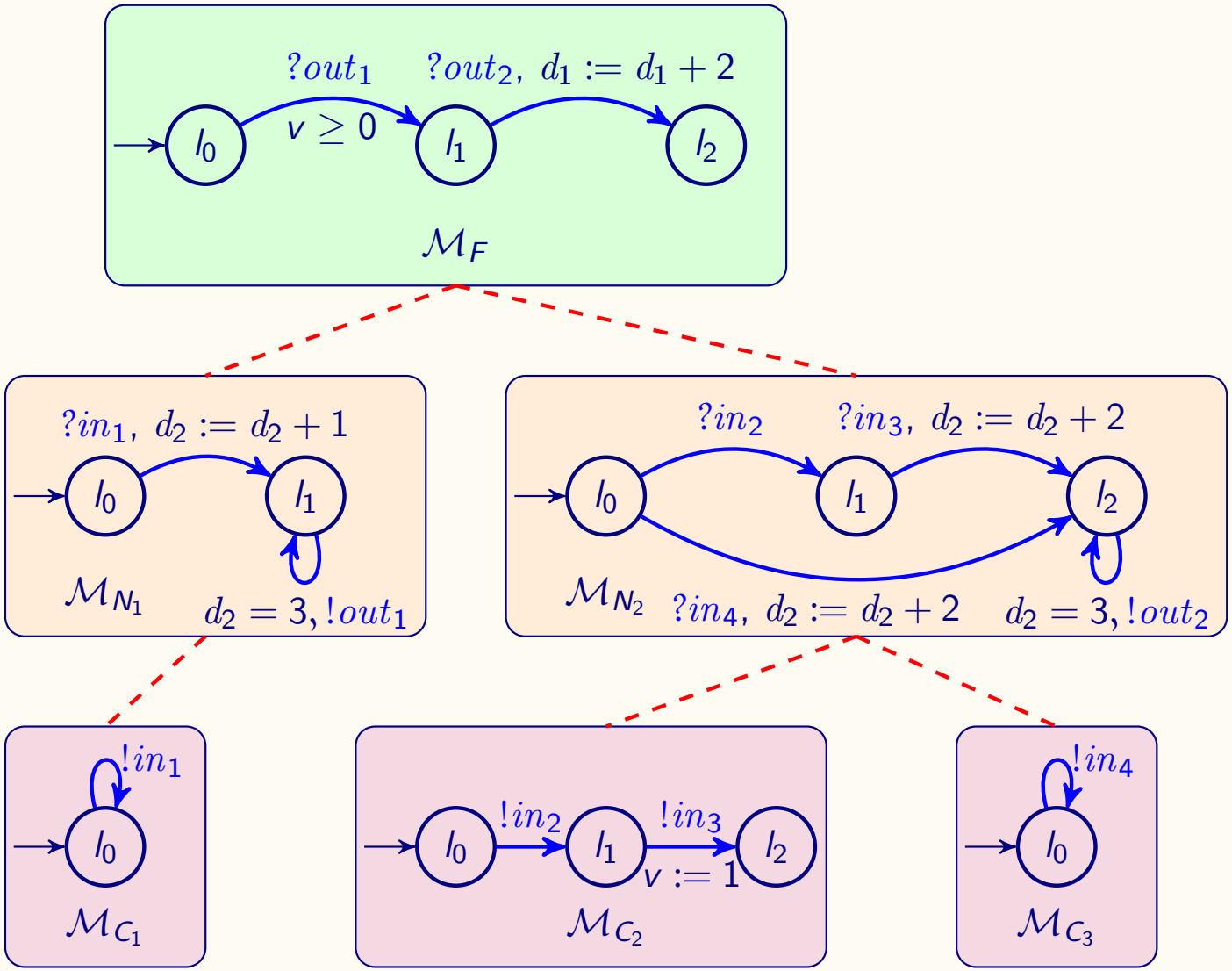
Layered Reduction for Trees



State space size

Full: 14 states, 19 edges Reduced: 12 states, 14 edges

Layered Reduction for Trees



State space size

Full: 14 states, 19 edges Reduced: 12 states, 14 edges

Experiments

Experimental setup

- model-checker IMITATOR (http://imitator.fr)
- 2.7 GHz Intel Core i7, with 16 GB of memory
- timeout of 30 minutes

Some case studies applying both reductions

Case study	vs. patterns		vs. full	
	% size	% reduction	% size	% reduction
treasure-hunters	47.44 %	52.56 %	13.26 %	86.74 %
forestall	24.97 %	75.03 %	2.37 %	97.63 %
iot-dev	40.90 %	59.10 %	8.53 %	91.47 %
gain-admin	No Timeout!			

Experiments

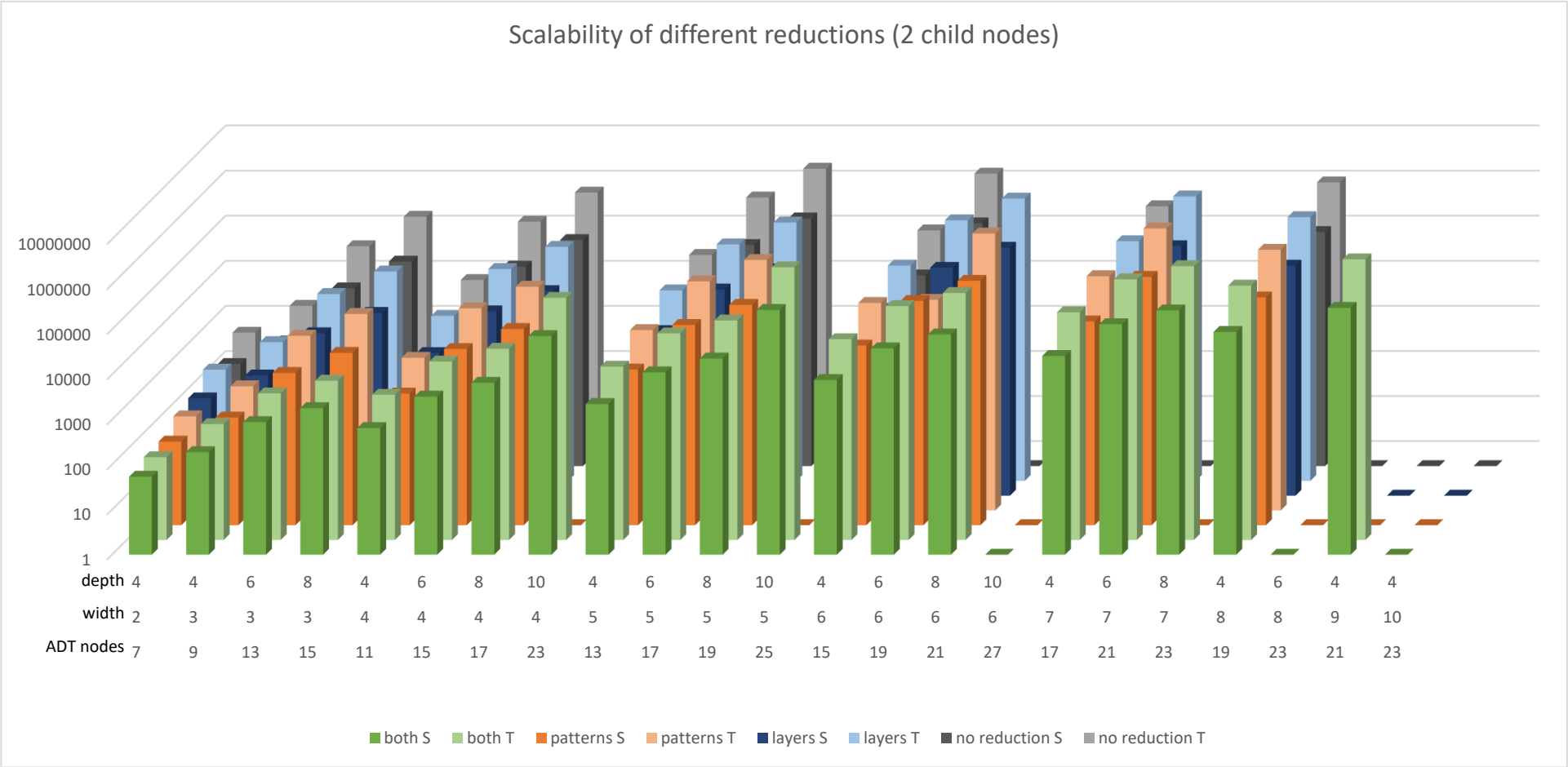
Experimental setup

- model-checker IMITATOR (http://imitator.fr)
- 2.7 GHz Intel Core i7, with 16 GB of memory
- timeout of 30 minutes

Some case studies applying both reductions

Case study	vs. patterns		vs. full	
	% size	% reduction	% size	% reduction
treasure-hunters	47.44 %	52.56 %	13.26 %	86.74 %
forestall	24.97 %	75.03 %	2.37 %	97.63 %
iot-dev	40.90 %	59.10 %	8.53 %	91.47 %
gain-admin	No Timeout!			

Scaling up Experiments



Conclusion & Perspectives

Summary

- A framework for model-checking systems that manipulate data
- Layered reduction approach to harness state space explosion
- Gains confirmed by extensive experiments on Attack-Defence Trees

Future work

- Extend the approach to DAGs
- Take into account the assignment of agents to ADTree nodes
- Study other application domains exhibiting such topologies (e.g. workflows)
- Use as a basis for a compositional and parallel analysis

Conclusion & Perspectives

Summary

- A framework for model-checking systems that manipulate data
- Layered reduction approach to harness state space explosion
- Gains confirmed by extensive experiments on Attack-Defence Trees

Future work

- Extend the approach to DAGs
- Take into account the assignment of agents to ADTree nodes
- Study other application domains exhibiting such topologies (e.g. workflows)
- Use as a basis for a compositional and parallel analysis

Bibliography

References I

Arias, J., Budde, C., Penczek, W., Petrucci, L., and Stoelinga, M. (2019).
Hackers vs. Security: Attack-Defence Trees as Asynchronous Multi-Agent Systems.
https://arxiv.org/abs/1906.05283.

Aslanyan, Z. and Nielson, F. (2015).
Pareto Efficient Solutions of Attack-Defence Trees.
In *Principles of Security and Trust*, volume 9036, pages 95–114. Springer Berlin Heidelberg.

Jamroga, W., Penczek, W., Dembinski, P., and Mazurkiewicz, A. W. (2018).
Towards partial order reductions for strategic ability.
In *AAMAS 2018*, pages 156–165. ACM.

Kordy, B., Mauw, S., Radomirović, S., and Schweitzer, P. (2011).
Foundations of attack-defense trees.
In *FAST 2010*, volume 6561 of *LNCS*, pages 80–95. Springer.