Games and Automata: From Boolean to Quantitative Verification

- Habilitation thesis defense -

Laurent Doyen
CNRS

ENS Cachan, March 13th, 2012

Outline

- Antichain Algorithms

  Finite automata, Büchi automata, alternating automata, partial-observation games, QBF

- Quanti

  Energ
  obser

- Quantitative Languages

  Automata-based model, complexity, expressiveness, closure properties, mean-payoff automaton expression

Context and perspective of a selection of results

Model-checking

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

Check if a Model satisfies a Property ?

...in an automated way

[Clarke, Emerson, Pnueli, Sifakis,...]

Model-checking

What kind of properties ?

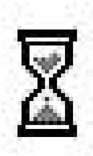
Model-checking

What kind of properties ?

Avoid failures !

Model-checking

What kind of properties ?



Ensure responsiveness !

Model-checking

What kind of models ?

Model-checking

What kind of models ?

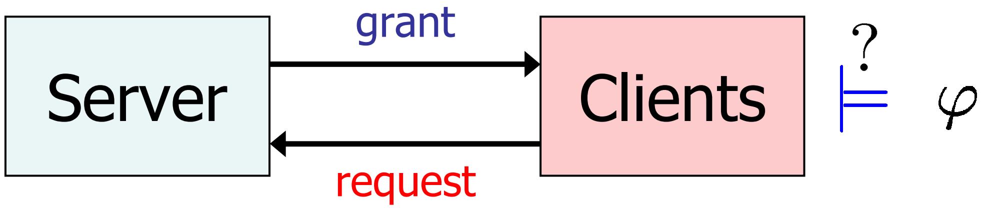
Reactive systems:

- Non-terminating
- Safety-critical
- Data abstraction

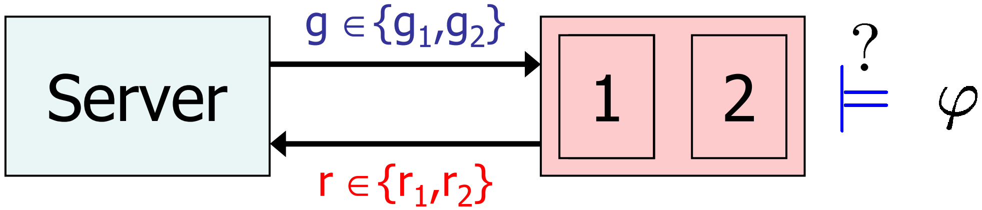
Model-checking

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

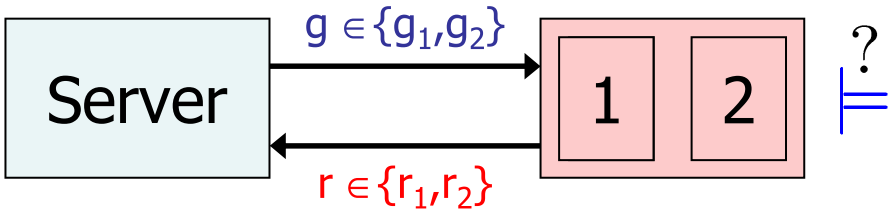
Example



Example

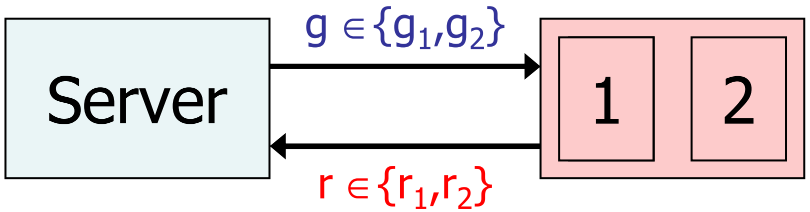


Example



« Every request is eventually granted, no simultaneous grants »

Example

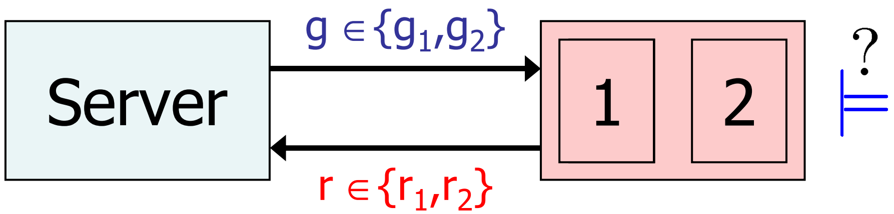


$\stackrel{?}{\models}$ « Every request is eventually granted, no simultaneous grants »

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

ω-automaton

Example



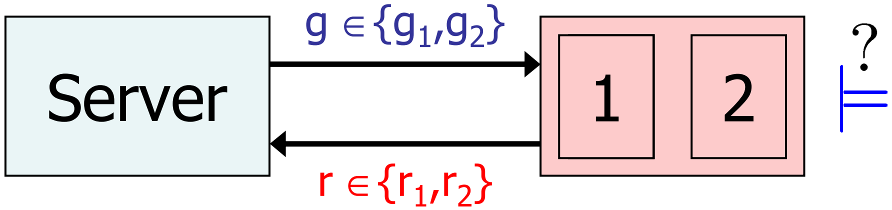
« Every request is eventually granted, no simultaneous grants »

- Closure properties
- Expressiveness
- Decidability

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

ω-automaton

Example



« Every request is eventually granted, no simultaneous grants »

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

Closure properties

Expressiveness

Decidability

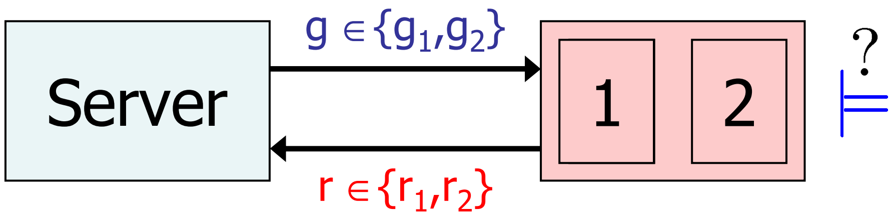
ω-automaton

$\Box(r_i \rightarrow \Diamond g_i) \wedge \Box\neg(g_1 \wedge g_2)$

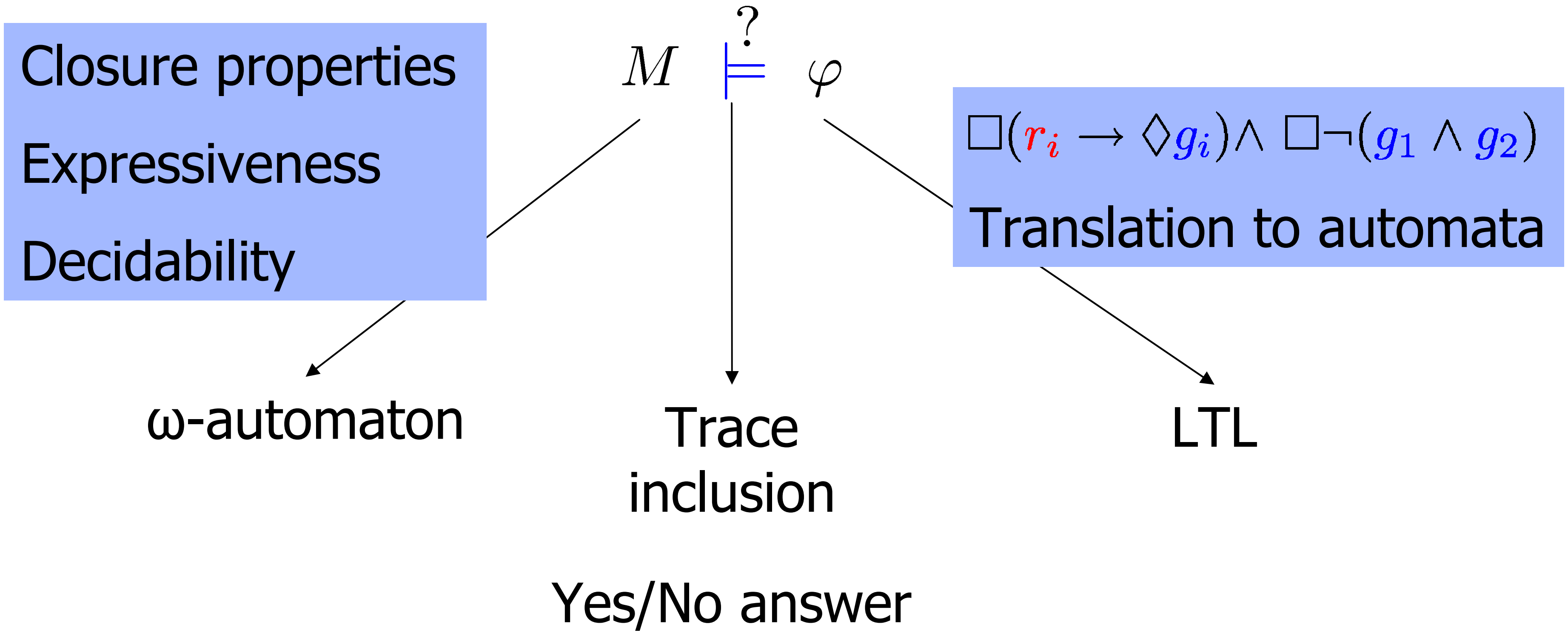
Translation to automata

LTL

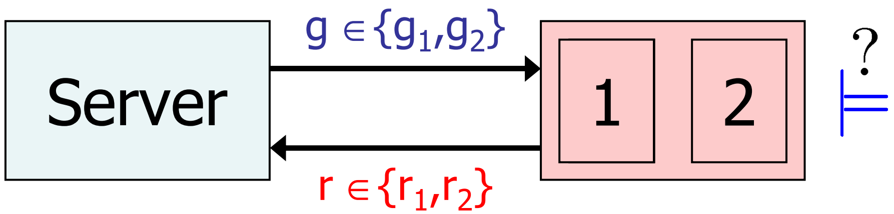
Example



« Every request is eventually granted, no simultaneous grants »



Example



« Every request is eventually granted, no simultaneous grants »

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

- Closure properties
- Expressiveness
- Decidability

$\Box(r_i \rightarrow \Diamond g_i) \wedge \Box\neg(g_1 \wedge g_2)$

Translation to automata

ω-automaton

Trace inclusion

LTL

Automata-based approach to model-checking

[Vardi, Wolper,...]

Outline

From Boolean to quantitative verification

Outline

From Boolean to quantitative verification

- Boolean automata-based Verification

  1. Techniques to speed up well-known verification algorithms by orders of magnitude

- Quantitative Verification

  2. A surprising complexity result in game theory

  3. A robust and decidable class of quantitative languages

  -

Algorithm ?

$$M \models \varphi \qquad L(M) \subseteq L(\varphi)$$

Algorithm ?

$M \models \varphi$ $\quad$ $L(M) \subseteq L(\varphi)$

Translation to automata $\quad$ $L(M) \subseteq L(A_\varphi)$

Algorithm ?

$M \models \varphi$		$L(M) \subseteq L(\varphi)$
Translation to automata		$L(M) \subseteq L(A_\varphi)$
		$L(M) \cap L(A_\varphi)^c = \varnothing$
Closure properties		$L(M \times A_\varphi^c) = \varnothing$

Algorithm ?

$M \models \varphi$ $\quad L(M) \subseteq L(\varphi)$

Translation to automata $\quad L(M) \subseteq L(A_\varphi)$

$$L(M) \cap L(A_\varphi)^c = \varnothing$$

Closure properties $\quad L(M \times A_\varphi^c) = \varnothing$

This problem is PSPACE-complete

Algorithm ?

$M \models \varphi$	$L(M) \subseteq L(\varphi)$
Translation to automata	$L(M) \subseteq L(A_\varphi)$
	$L(M) \cap L(A_\varphi)^c = \varnothing$
Closure properties	$L(M \times A_\varphi^c) = \varnothing$

This problem is PSPACE-complete

even if A_φ is given explicitly, even over finite words, and even if $L(M) = \Sigma^*$ $\quad L(A^c) = \varnothing$

Efficient Algorithm ?

(over finite words) $L(A^c) = \varnothing$

iff there is no path from inital to accepting states in A^c.

Efficient Algorithm ?

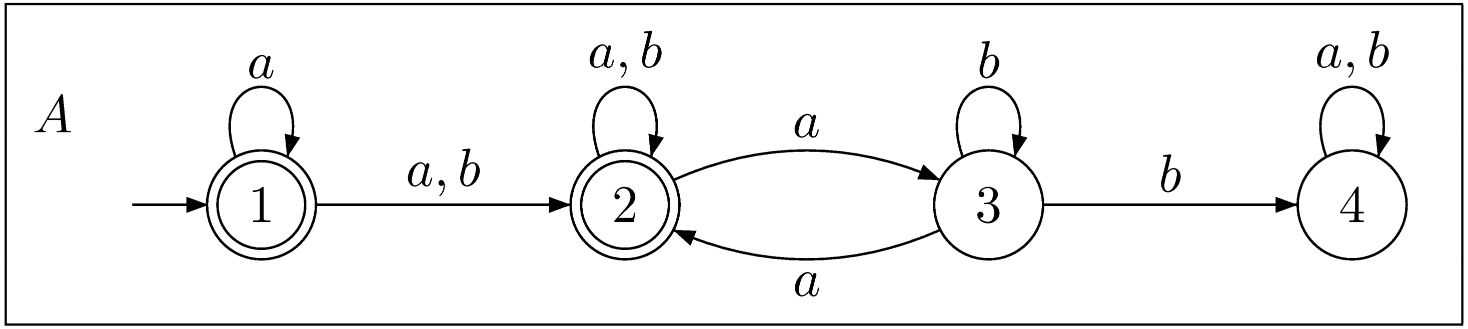
(over finite words) $L(A^c) = \varnothing$

iff there is no path from inital to accepting states in A^c.

Subset construction

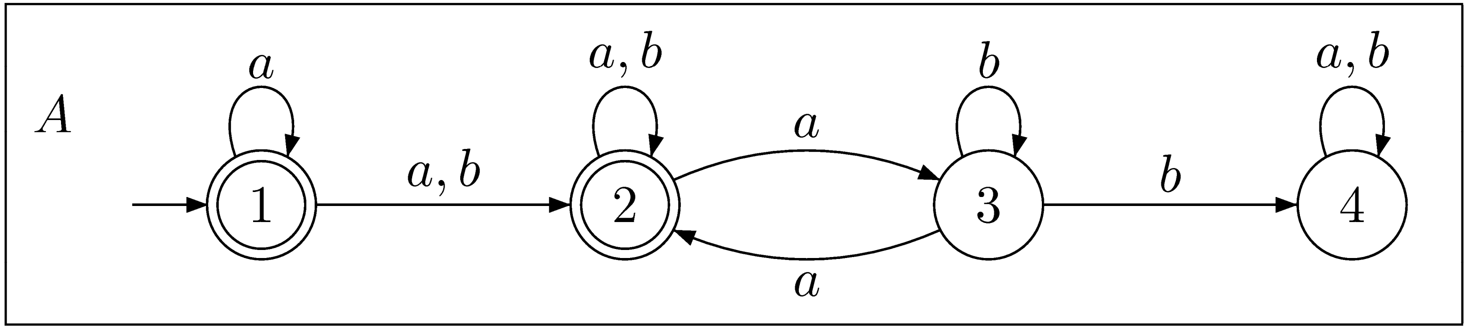
(state-explosion problem)

Subset Construction



$$L(A^c) \stackrel{?}{=} \varnothing$$

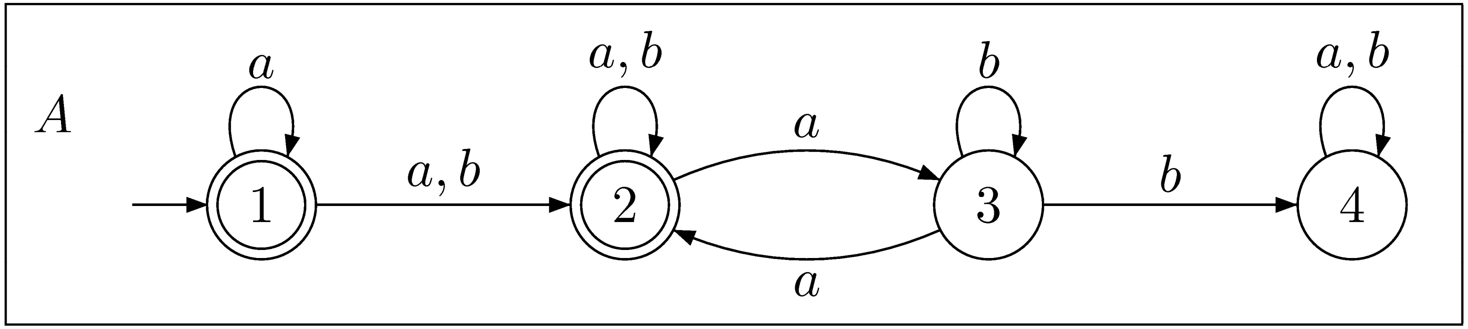
Subset Construction



$L(A^c) \stackrel{?}{=} \varnothing$

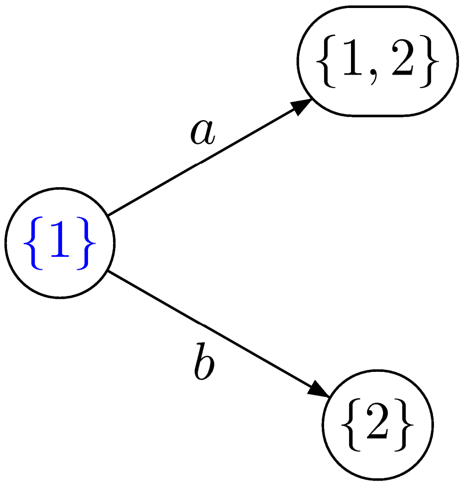
$\{1\} \xrightarrow{?} \subseteq \{3, 4\}$

Subset Construction

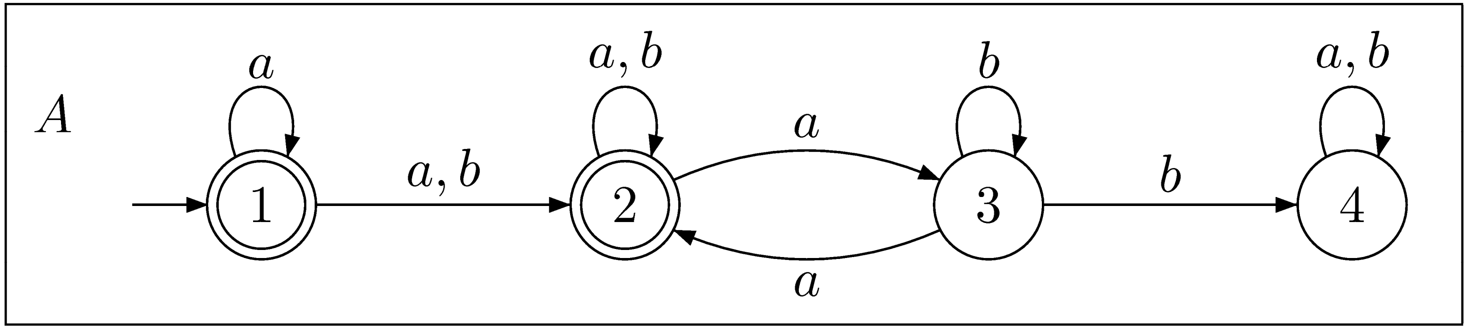


$$L(A^c) \stackrel{?}{=} \emptyset$$

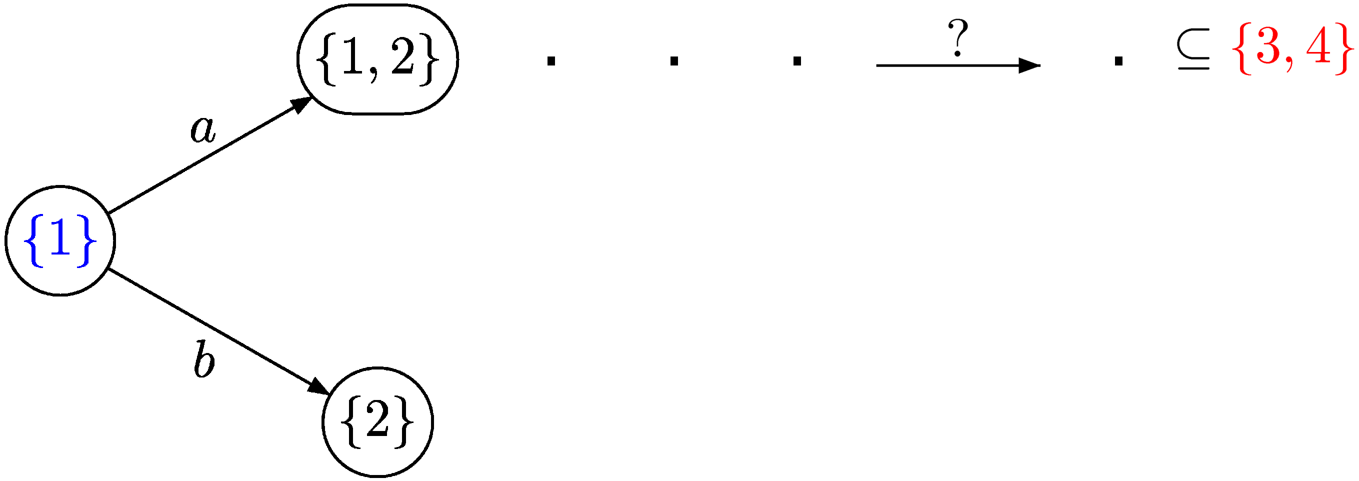
$$\{1\} \xrightarrow{\quad ? \quad} \subseteq \{3, 4\}$$



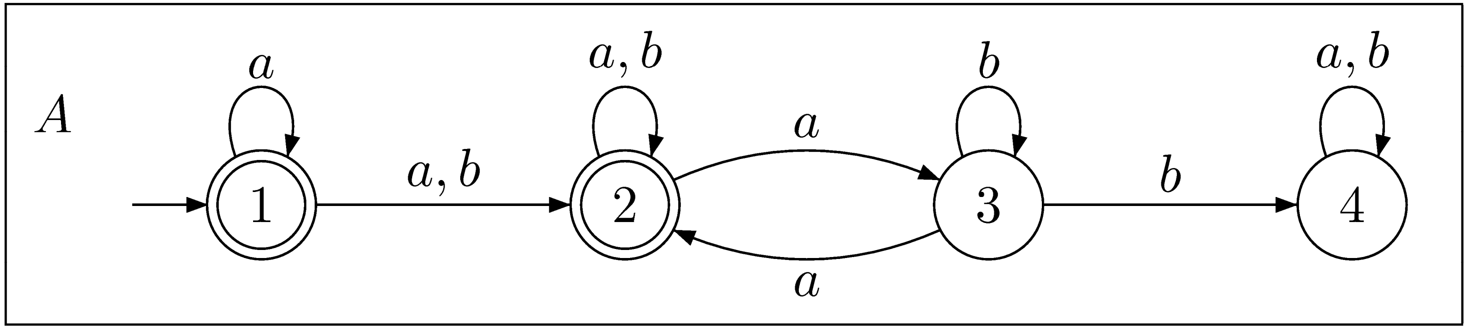
Subset Construction



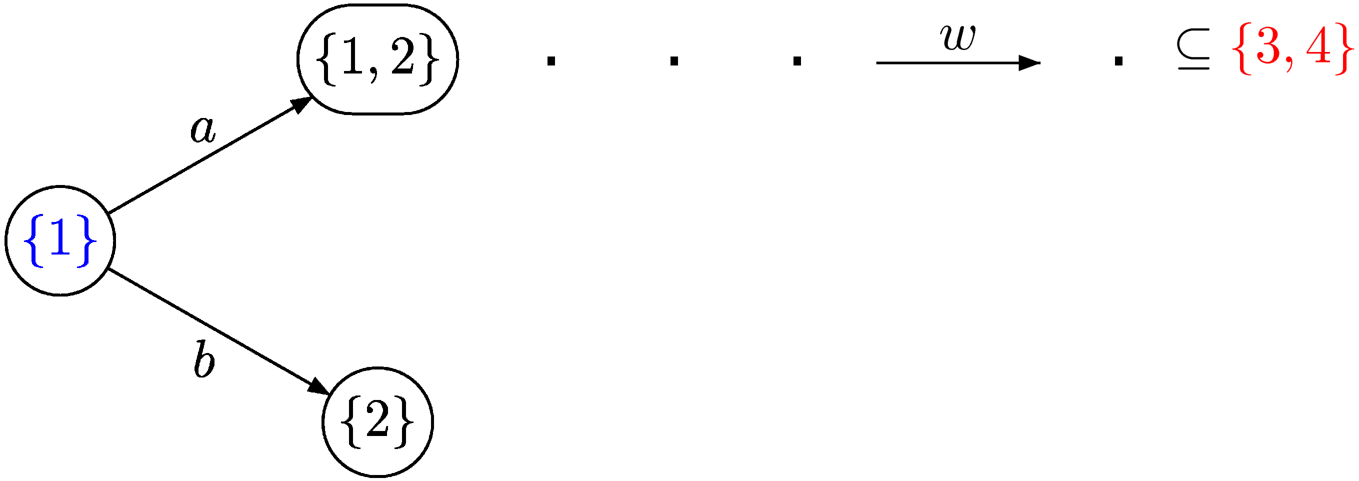
$L(A^c) \stackrel{?}{=} \emptyset$



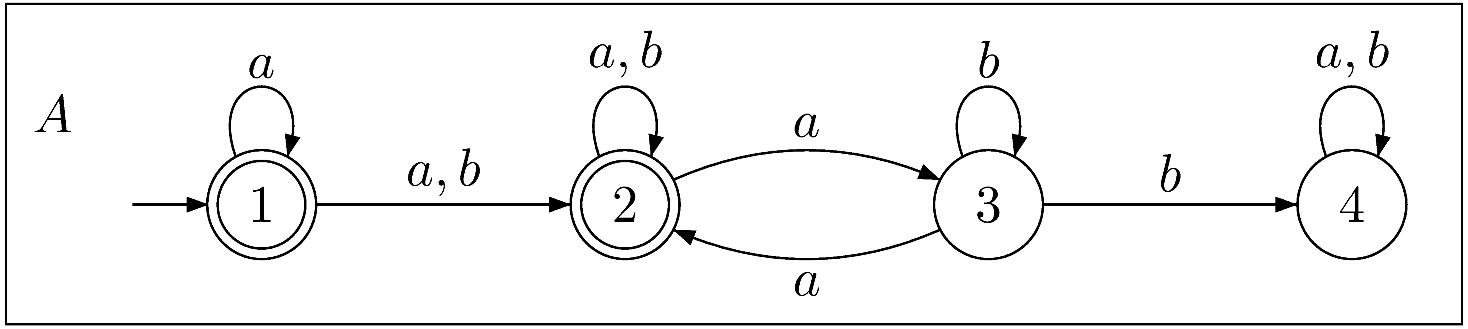
Subset Construction



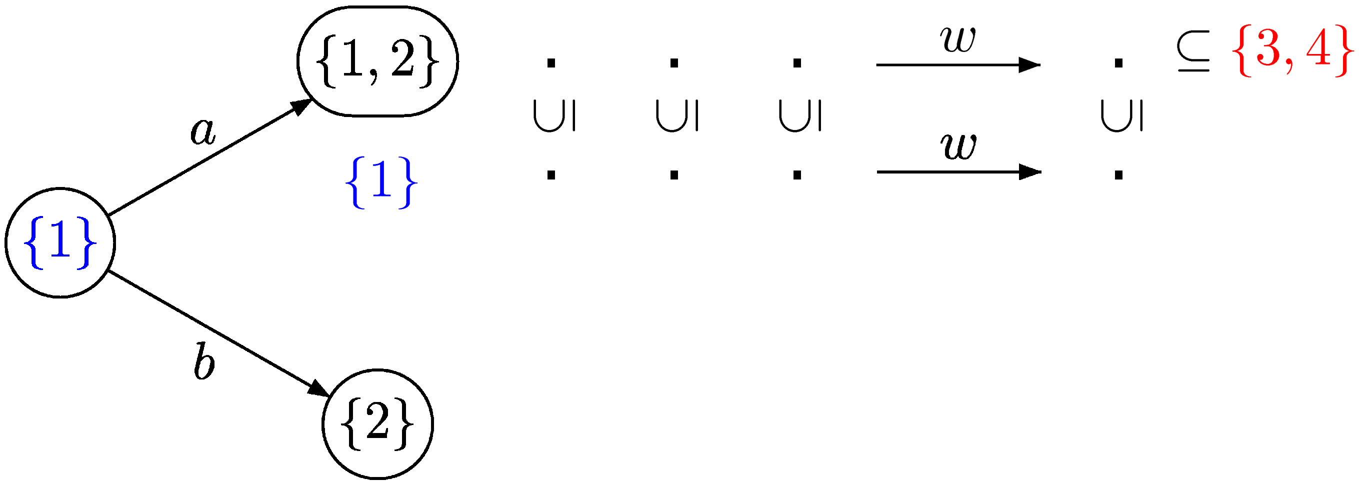
$L(A^c) \stackrel{?}{=} \emptyset$



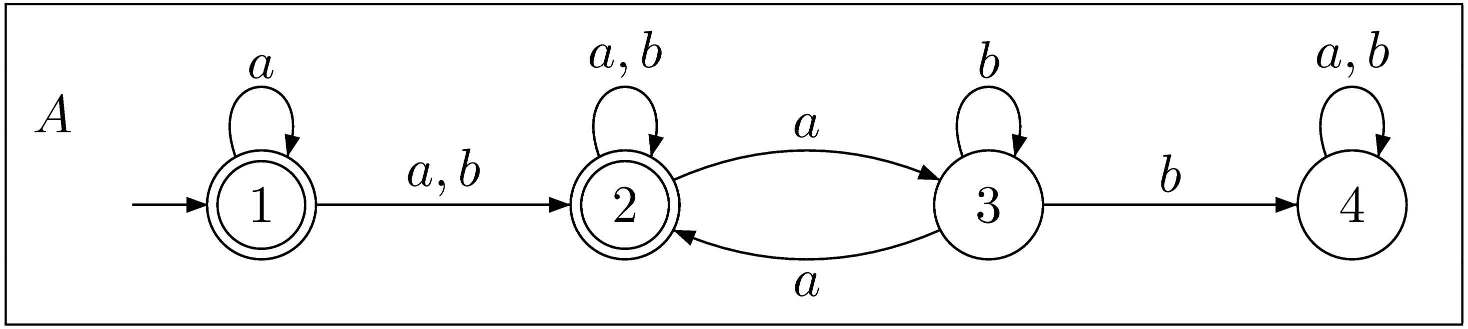
Subset Construction



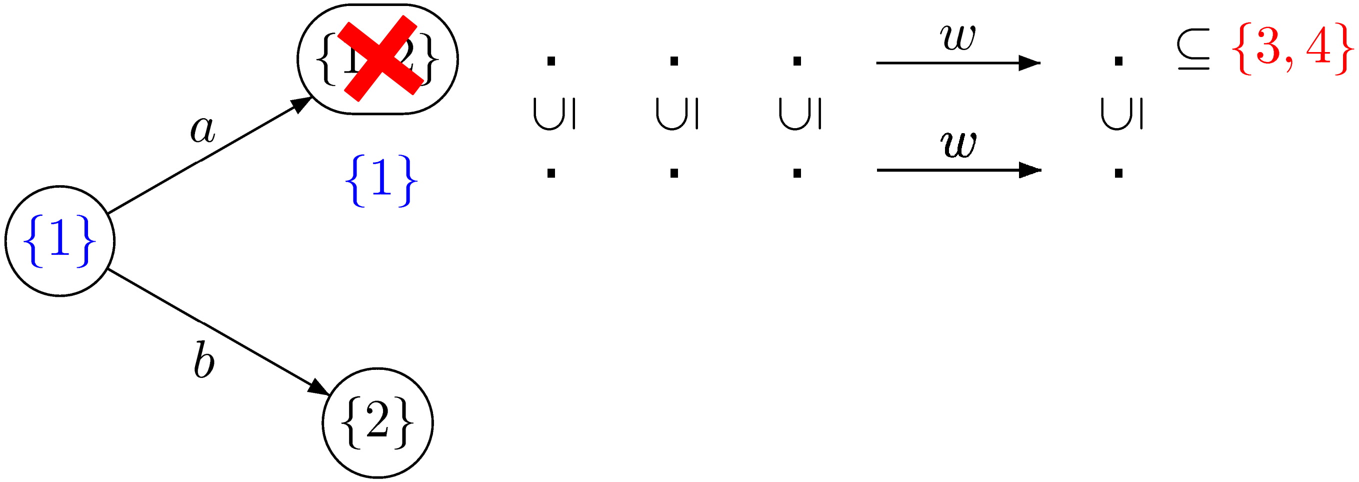
$L(A^c) \stackrel{?}{=} \emptyset$



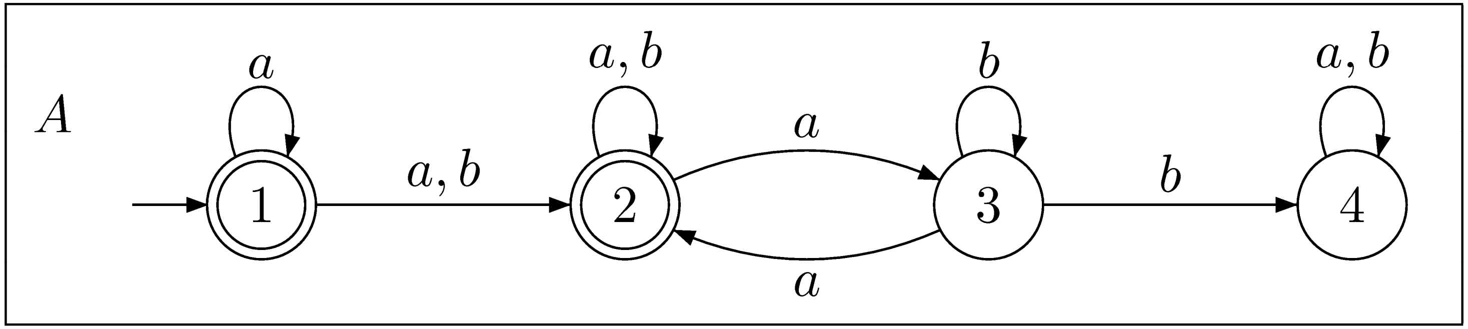
Subset Construction



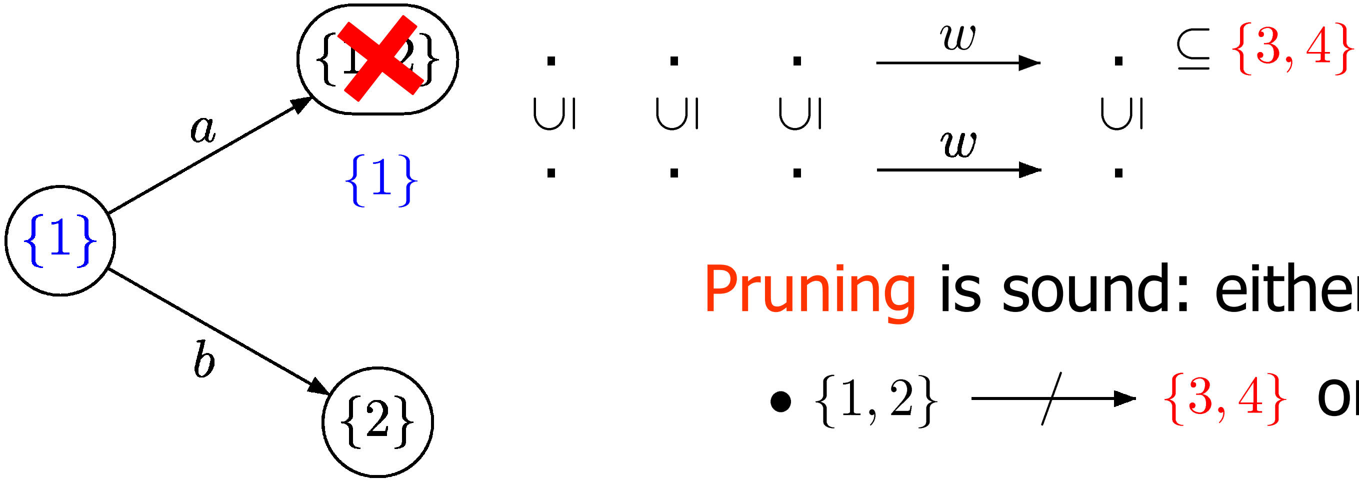
$L(A^c) \stackrel{?}{=} \varnothing$



Subset Construction



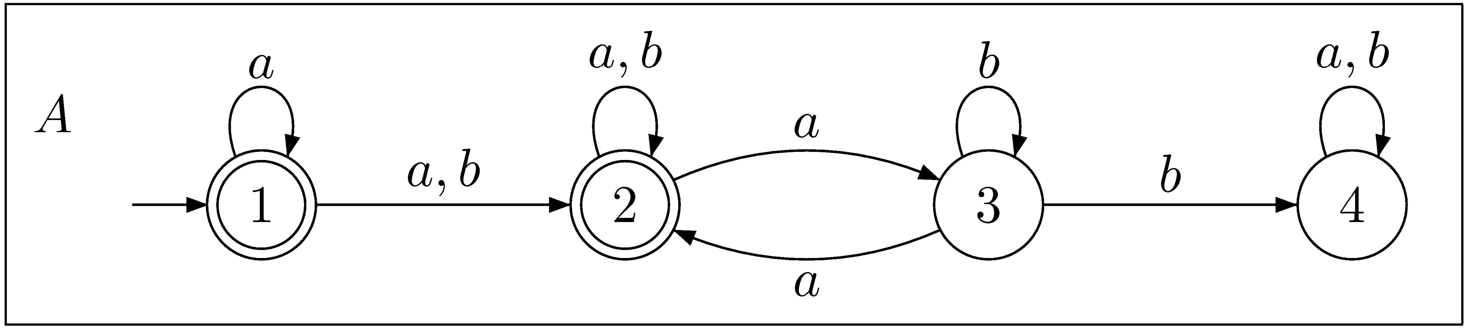
$L(A^c) \stackrel{?}{=} \emptyset$



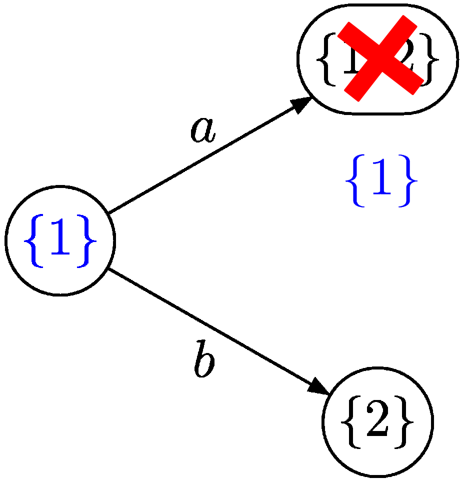
Pruning is sound: either

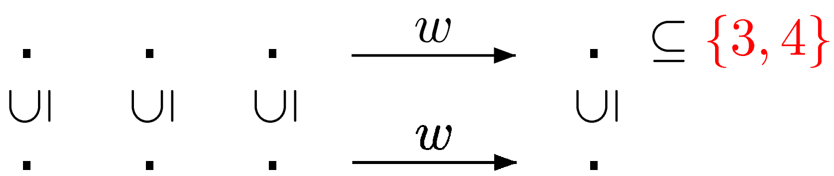
- $\{1,2\} \not\xrightarrow{\quad} \{3,4\}$ or

Subset Construction



$L(A^c) \stackrel{?}{=} \emptyset$



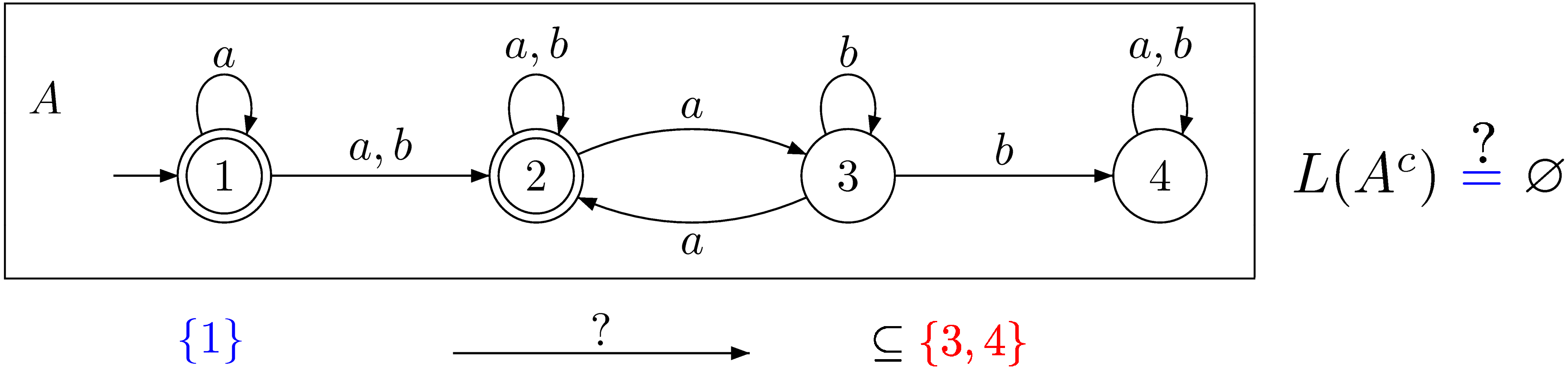


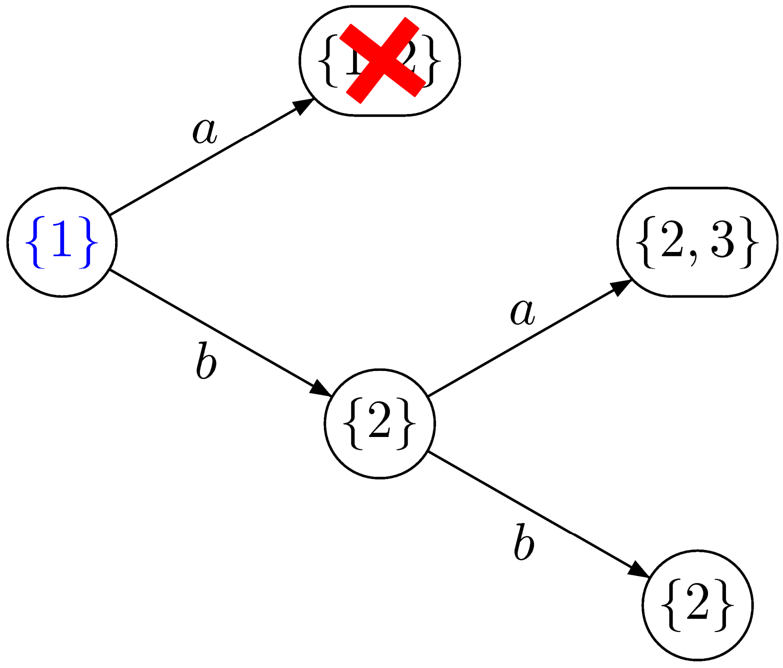
Pruning is sound: either

- $\{1,2\} \not\longrightarrow \{3,4\}$ or
- $\{1,2\} \xrightarrow{\exists w} \{3,4\}$

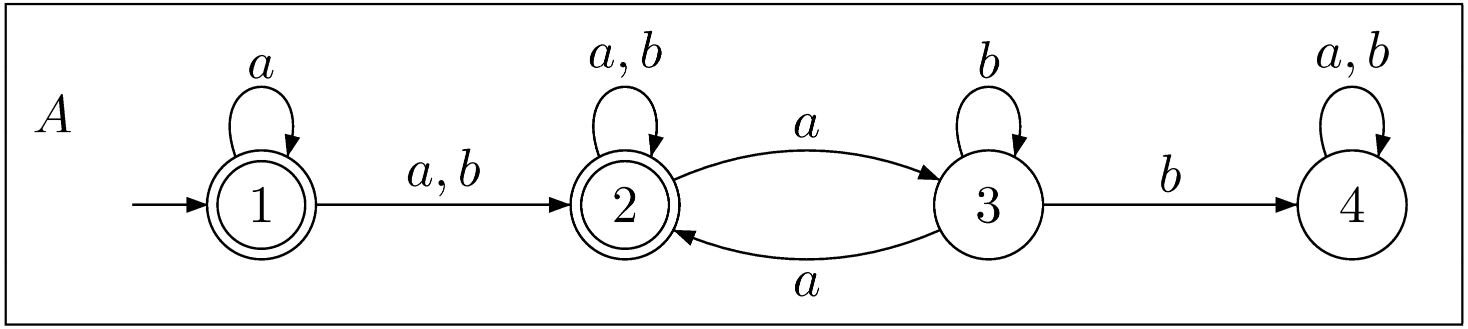
  $\{1\} \xrightarrow{\exists w} \{3,4\}$

Subset Construction



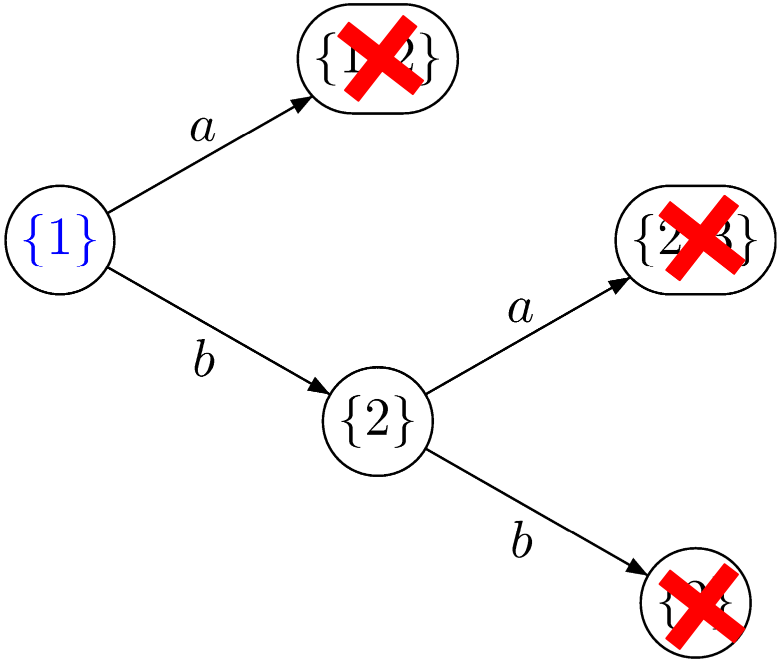


Subset Construction

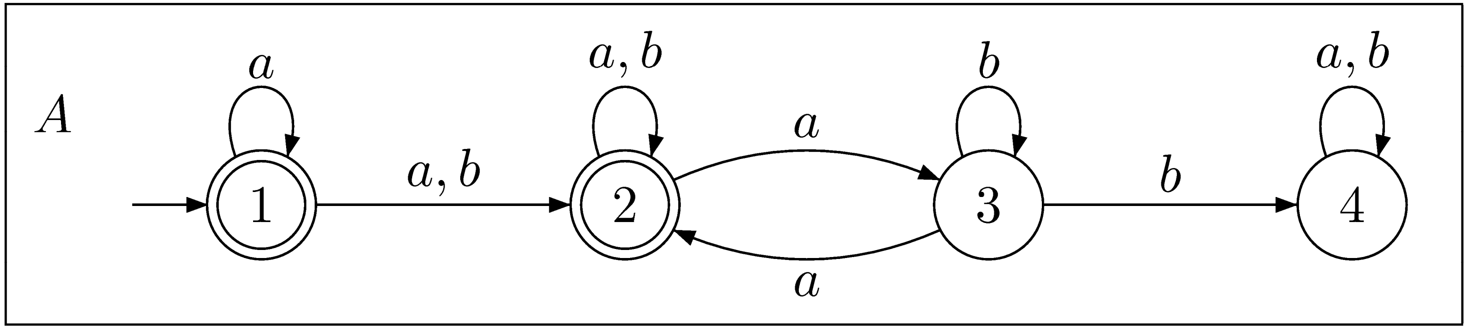


$L(A^c) \stackrel{?}{=} \emptyset$

$\{1\} \xrightarrow{?} \subseteq \{3, 4\}$

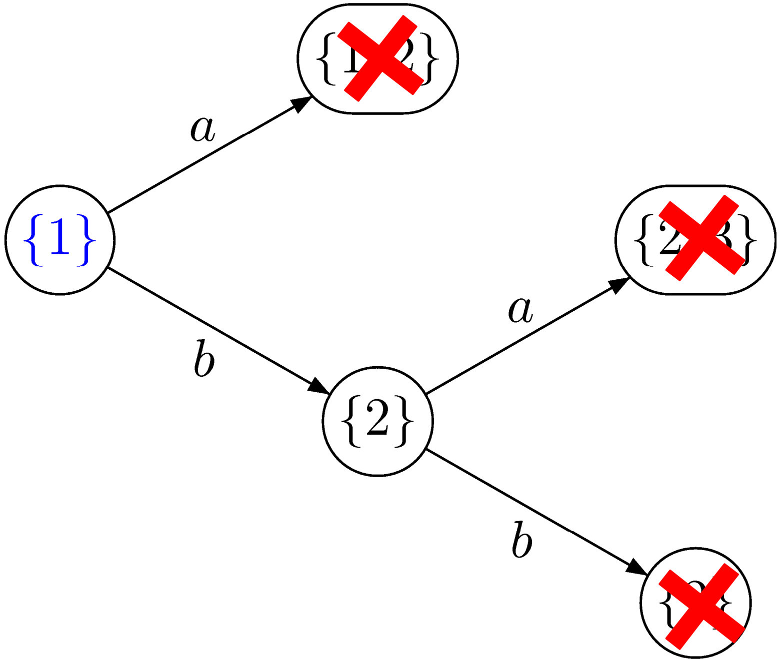


Subset Construction



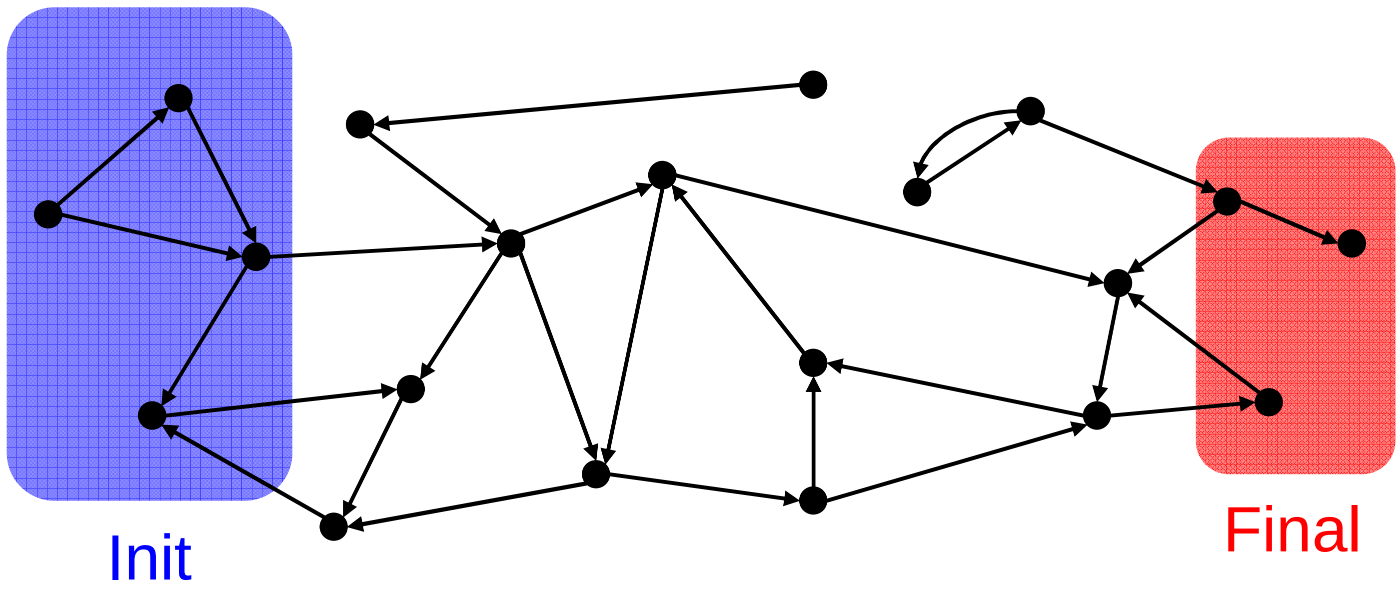
$L(A^c) \stackrel{?}{=} \varnothing$

$\{1\} \xrightarrow{?} \subseteq \{3, 4\}$



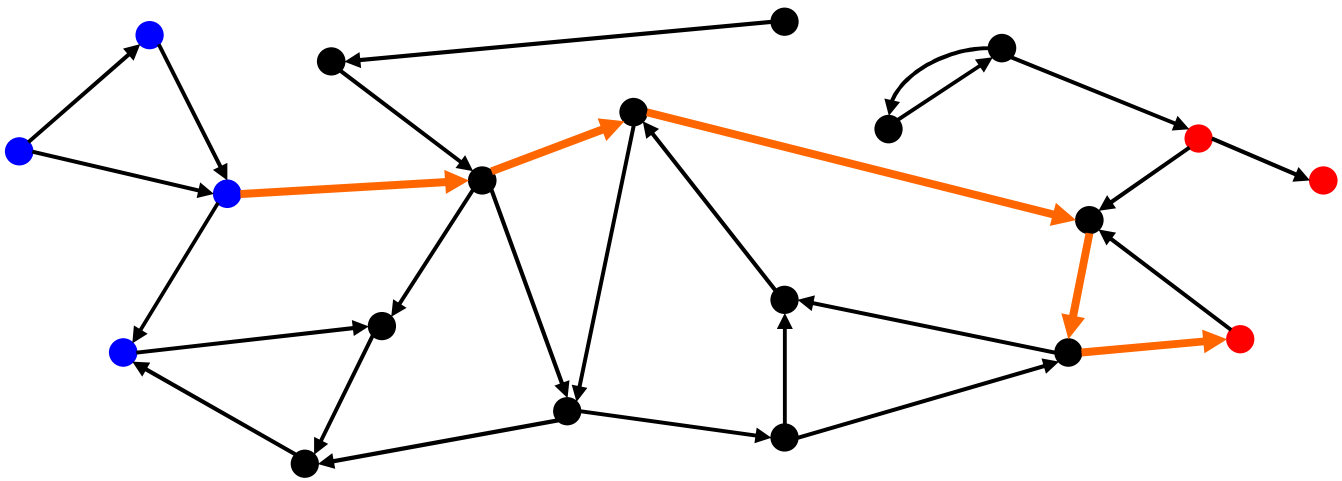
Hence, $L(A^c) = \varnothing$

Reachability



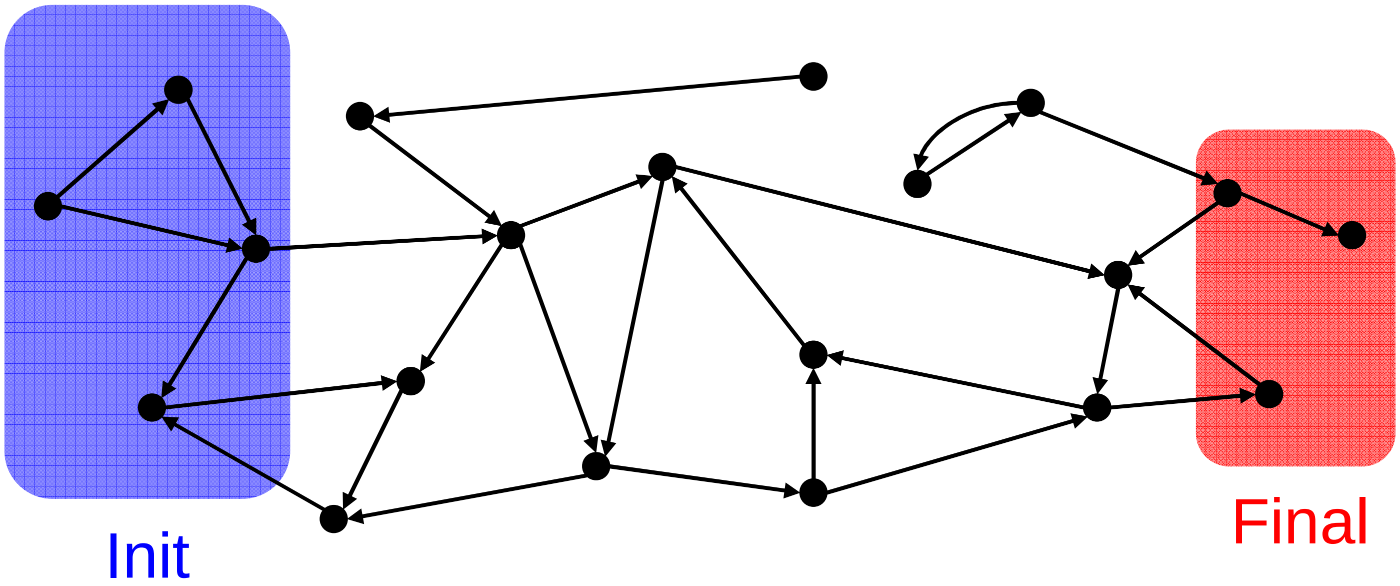
Is there a (finite) path from Init to Final ?

Reachability

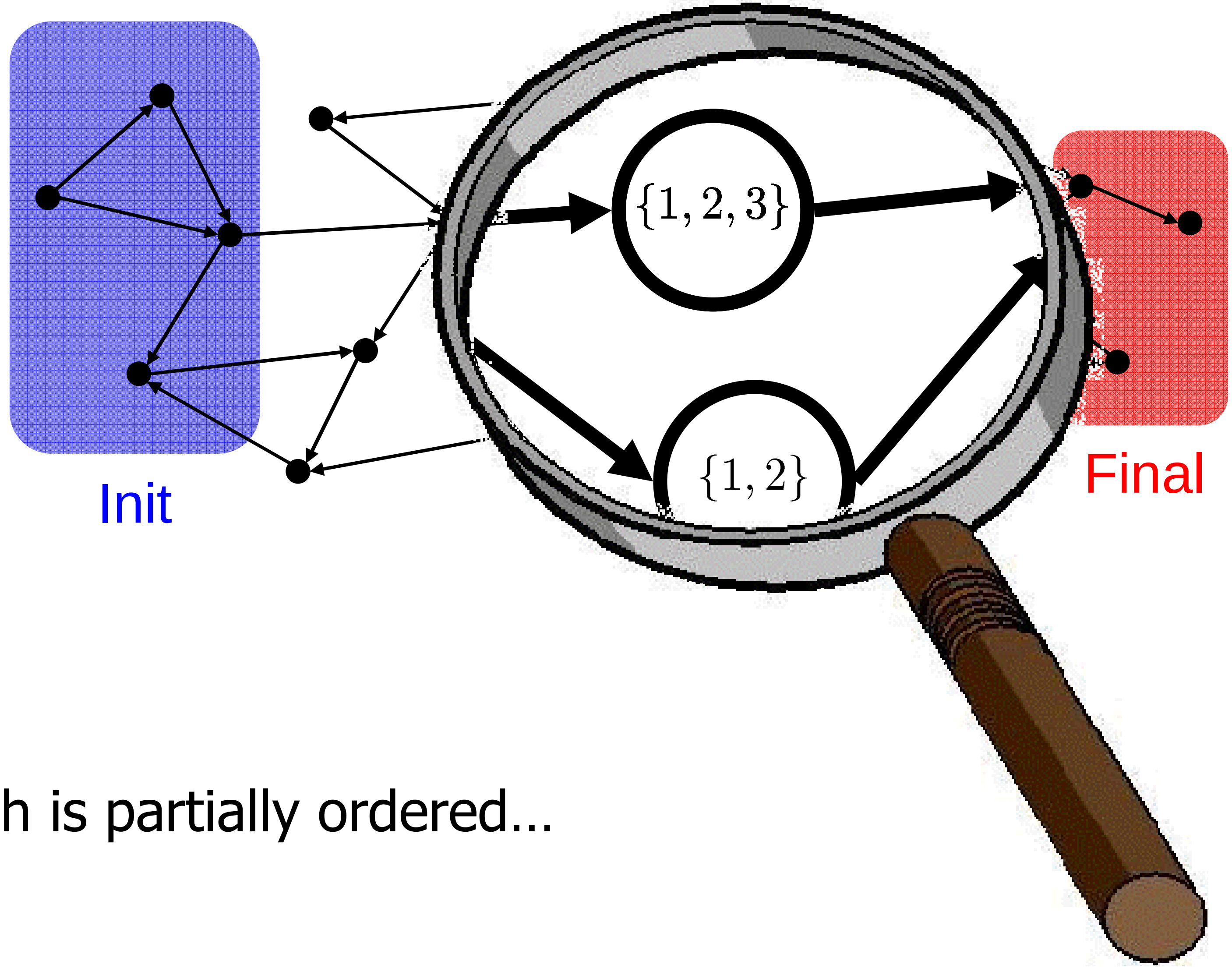


Is there a (finite) path from Init to Final ?

Structure in graphs

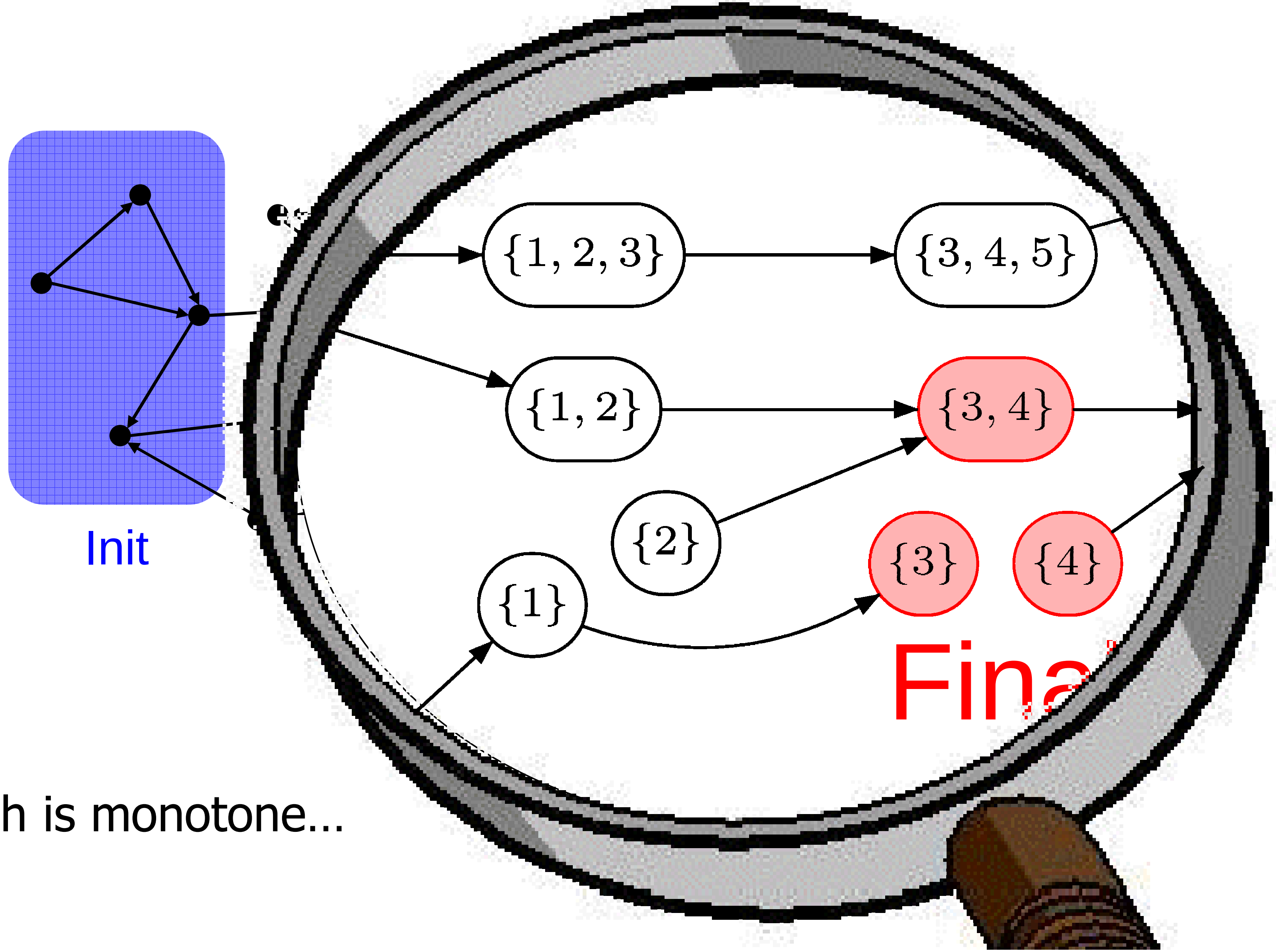


Structure in graphs



Graph is partially ordered…

Structure in graphs



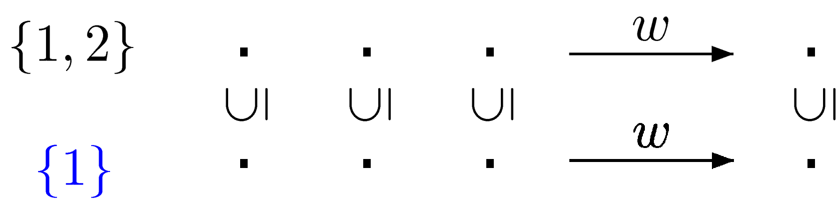
Graph is monotone…

Structure in graphs

$$
\begin{array}{cccccc}
\{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{\;w\;} & \cdot \\
 & \cup\!\vert & \cup\!\vert & \cup\!\vert & & \cup\!\vert \\
\color{blue}{\{1\}} & \cdot & \cdot & \cdot & \xrightarrow{\;w\;} & \cdot
\end{array}
$$

Key property

Structure in graphs



Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A[c]

$\subseteq$ is a backward simulation relation in A[c]

Structure in graphs

$$
\begin{array}{ccccccc}
\{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{\;w\;} & \cdot \\
 & \rotatebox[origin=c]{90}{$\subseteq$} & \rotatebox[origin=c]{90}{$\subseteq$} & \rotatebox[origin=c]{90}{$\subseteq$} & & \rotatebox[origin=c]{90}{$\subseteq$} \\
\{1\} & \cdot & \cdot & \cdot & \xrightarrow{\;w\;} & \cdot
\end{array}
$$

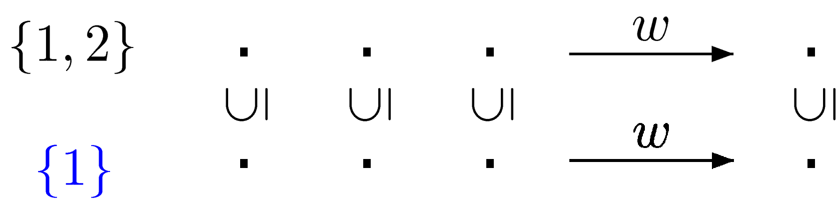
Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c^

Use $\subseteq$ to prune the search

Structure in graphs



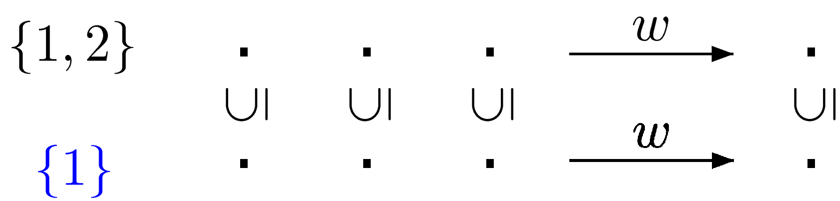
Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

Use $\subseteq$ to prune the search

Structure in graphs

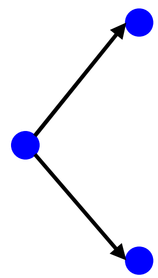


Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

Use $\subseteq$ to prune the search



Structure in graphs

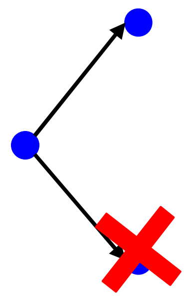
$$\begin{array}{cccccc} \{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \\ & \cup\!| & \cup\!| & \cup\!| & & \cup\!| \\ \{1\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \end{array}$$

Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

Use $\subseteq$ to prune the search



Structure in graphs

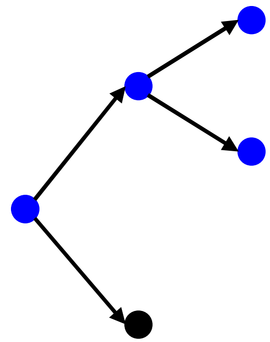
$$\begin{array}{ccccccc} \{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \\ & \cup\!\! \mid & \cup\!\! \mid & \cup\!\! \mid & & \cup\!\! \mid \\ \{1\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \end{array}$$

Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

Use $\subseteq$ to prune the search



Structure in graphs

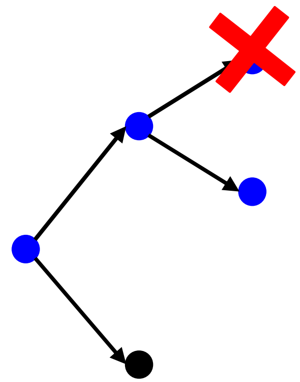
$$\{1,2\} \quad \cdot \quad \cdot \quad \cdot \quad \xrightarrow{w} \quad \cdot$$
$$\quad\quad\;\; \cup\!| \quad \cup\!| \quad \cup\!| \quad\quad\quad\;\; \cup\!|$$
$$\{1\} \quad \cdot \quad \cdot \quad \cdot \quad \xrightarrow{w} \quad \cdot$$

Key property

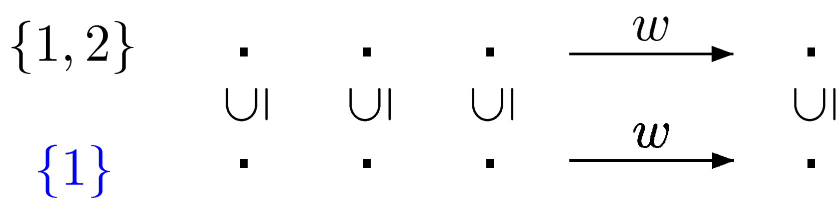
Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

Use $\subseteq$ to prune the search



Structure in graphs

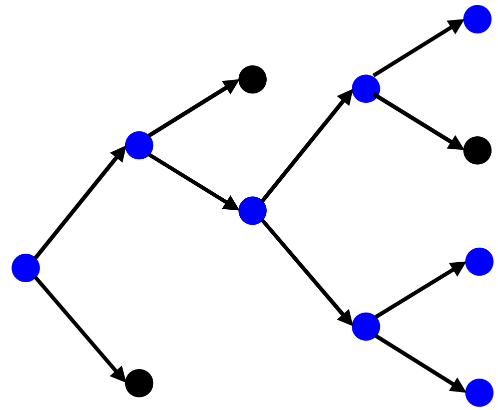


Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

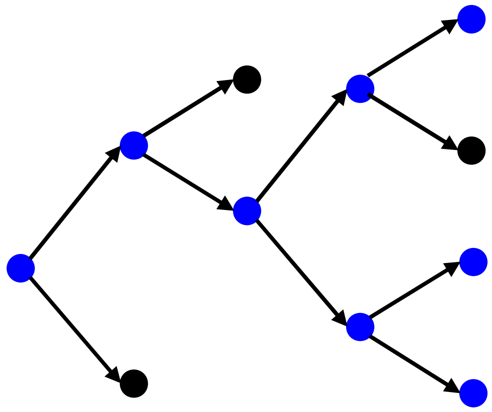
Use $\subseteq$ to prune the search



Structure in graphs

$$\begin{array}{cccccc} \{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \\ & \cup\!| & \cup\!| & \cup\!| & & \cup\!| \\ \{1\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \end{array}$$

Key property



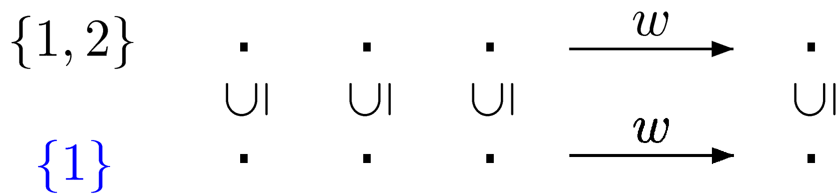
Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

Use $\subseteq$ to prune the search

Antichain of promising states

Structure in graphs



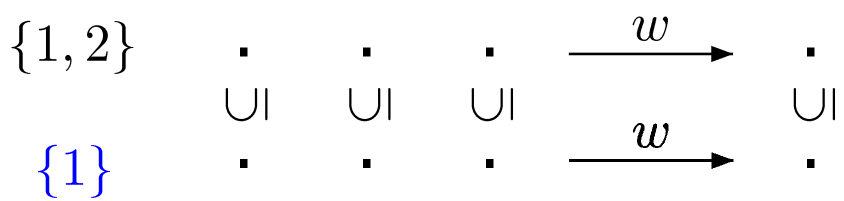
Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

$\subseteq$ is a backward simulation relation in A^c

Structure in graphs



Key property

Two interpretations:

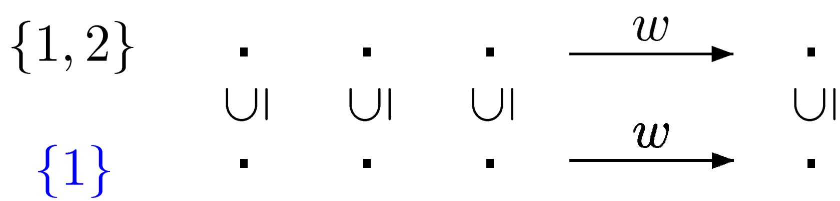
$\subseteq$ is a forward simulation relation in A^c

$\subseteq$ is a backward simulation relation in A^c

iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets

Structure in graphs



Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

$\subseteq$ is a backward simulation relation in A^c

iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets

Structure in graphs

$$\begin{array}{ccccccc} \{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \\ & \supseteq & \supseteq & \supseteq & & \supseteq \\ \{1\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \end{array}$$

Key property

Two interpretations:

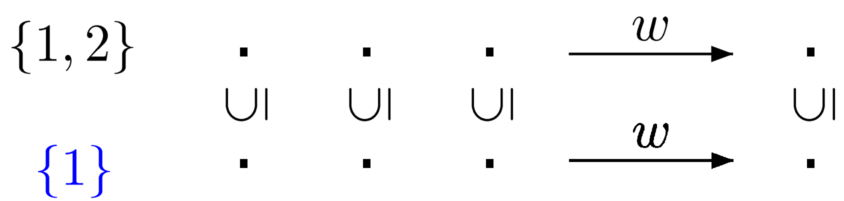
$\subseteq$ is a forward simulation relation in A^c

$\subseteq$ is a backward simulation relation in A^c

iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets

Structure in graphs



Key property

Two interpretations:

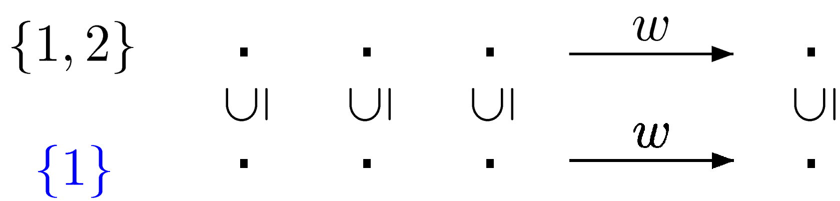
$\subseteq$ is a forward simulation relation in A^c

$\subseteq$ is a backward simulation relation in A^c

iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets

Structure in graphs



Key property

Two interpretations:

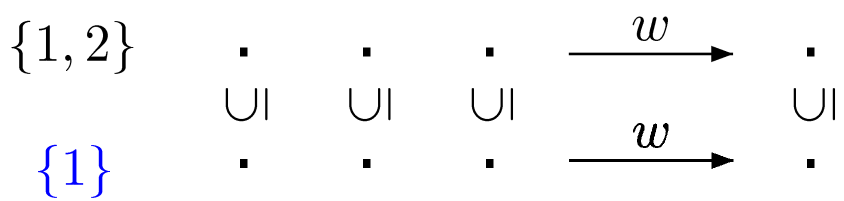
$\subseteq$ is a forward simulation relation in A^c

$\subseteq$ is a backward simulation relation in A^c

iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets

Structure in graphs



Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

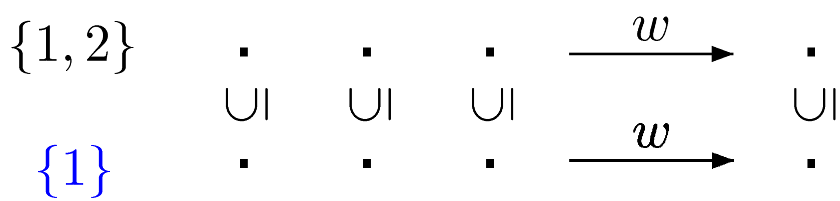
$\subseteq$ is a backward simulation relation in A^c

iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets

Antichains as a symbolic representation
(minimal elements)

Structure in graphs



Key property

Two interpretations:

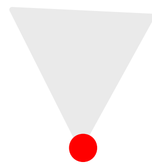
$\subseteq$ is a forward simulation relation in A^c

$\subseteq$ is a backward simulation relation in A^c

iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets

Antichains as a symbolic representation (minimal elements)



Structure in graphs

$$\begin{array}{cccccc} \{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \\ & \cup\!| & \cup\!| & \cup\!| & & \cup\!| \\ \{1\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \end{array}$$

Key property

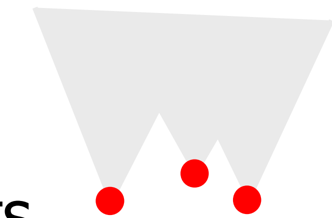
Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

$\subseteq$ is a backward simulation relation in A^c

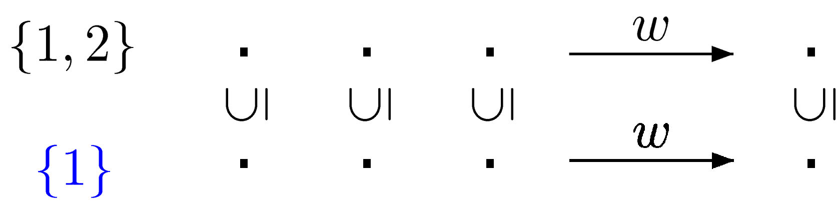
iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets



Antichains as a symbolic representation (minimal elements)

Structure in graphs



Key property

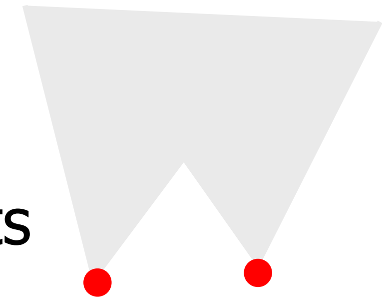
Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

$\subseteq$ is a backward simulation relation in A^c

iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets



Antichains as a symbolic representation (minimal elements)

Structure in graphs

$$
\begin{array}{ccccccc}
\{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \\
 & \cup\!| & \cup\!| & \cup\!| & & \cup\!| \\
\{1\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot
\end{array}
$$

Key property

Two interpretations:

$\subseteq$ is a forward simulation relation in A^c

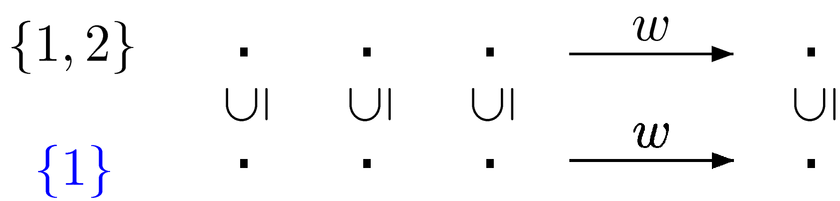
$\subseteq$ is a backward simulation relation in A^c

iff post(.) preserves $\subseteq$-upward closure

post*(.) computes a sequence of $\subseteq$-upward sets

Antichains as a symbolic representation (minimal elements)

Structure in graphs

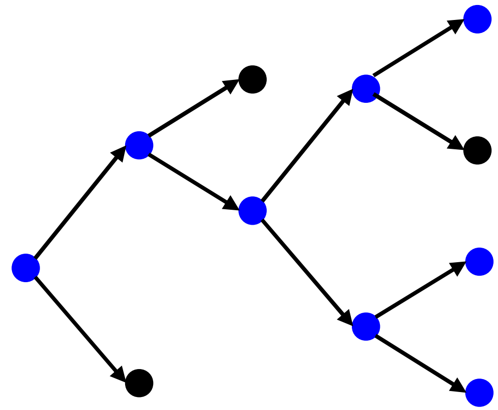


Key property

Two interpretations:

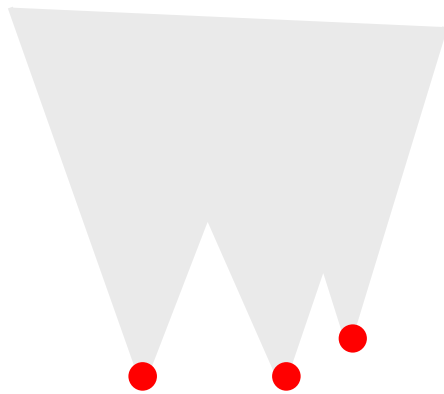
$\subseteq$ is a forward simulation relation in A^c

Promising states



$\subseteq$ is a backward simulation relation in A^c

Symbolic representation

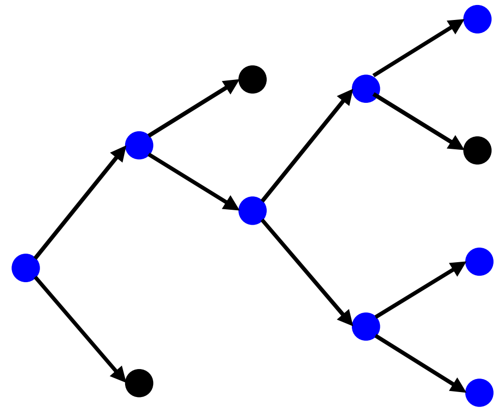


Structure in graphs

$$\begin{array}{cccccc} \{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \\ & \cup\!| & \cup\!| & \cup\!| & & \cup\!| \\ \color{blue}{\{1\}} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \end{array}$$

Key property

Two interpretations:

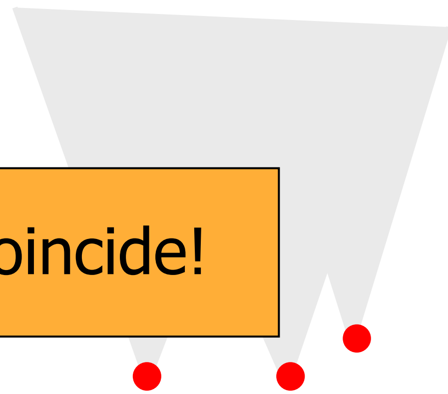


$\sqsubseteq$ is a forward simulation relation in A^c

Promising states

$\sqsubseteq$ is a backward simulation relation in A^c

Symbolic representation



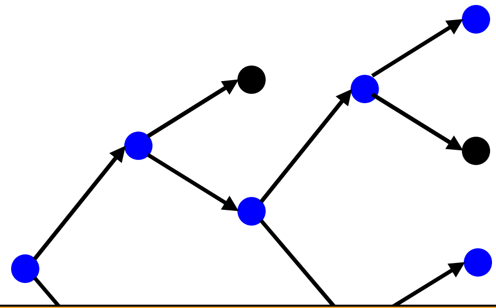
Here the two interpretations coincide!

Structure in graphs

$$\begin{array}{ccccccc} \{1,2\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \\ & \cup\!\mid & \cup\!\mid & \cup\!\mid & & \cup\!\mid \\ \{1\} & \cdot & \cdot & \cdot & \xrightarrow{w} & \cdot \end{array}$$

Key property

Two interpretations:

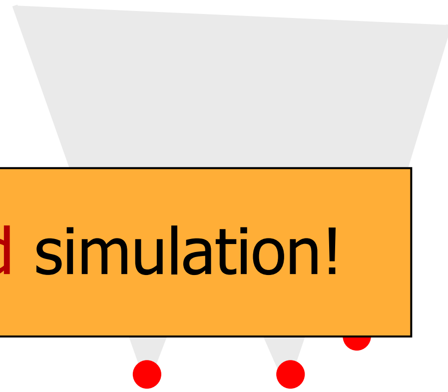


$\subseteq$ is a forward simulation relation in A^c

Promising states

Works with ANY forward simulation!

$\subseteq$ is a backward simulation relation in A^c



Symbolic representation

Works with ANY backward simulation!

Antichains everywhere!

Partial-observation Reachability/Parity games	HSCC'06, CSL'06, CONCUR'08, Inf&Comp'10
Finite automata (language inclusion, universality)	CAV'06
Büchi automata (language inclusion, universality)	TACAS'07, LMCS'09
LTL satisfiability and model-checking	TACAS'08
QBF	ATVA'11

...

Antichains everywhere!

Partial-observation Reachability/Parity games	HSCC'06, CSL'06, CONCUR'08, Inf&Comp'10
Finite automata (language inclusion, universality)	CAV'06
Büchi automata (language inclusion, universality)	TACAS'07, LMCS'09
LTL satisfiability and model-checking	TACAS'08
QBF	ATVA'11

...


J-F. Raskin


M. De Wulf


N. Maquet


T. Henzinger


D. Berwanger

...

Antichains everywhere!

Partial-observation Reachability/Parity games	HSCC'06, CSL'06, CONCUR'08, Inf&Comp'10
Finite automata (language inclusion, universality)	CAV'06
Büchi automata (language inclusion, universality)	TACAS'07, LMCS'09
LTL satisfiability and model-checking	TACAS'08
QBF	ATVA'11

...

Finite Tree Automata [Bouajjani et al. 08]

Program Termination [Vardi et al. 09]

Minimizing Alternating Büchi [Abdulla et al. 09]

LTL synthesis [Raskin et al. 09]

Büchi universality [Vardi et al. 10]

Simulation Subsumption [Abdulla et al. 10,11]

Tools

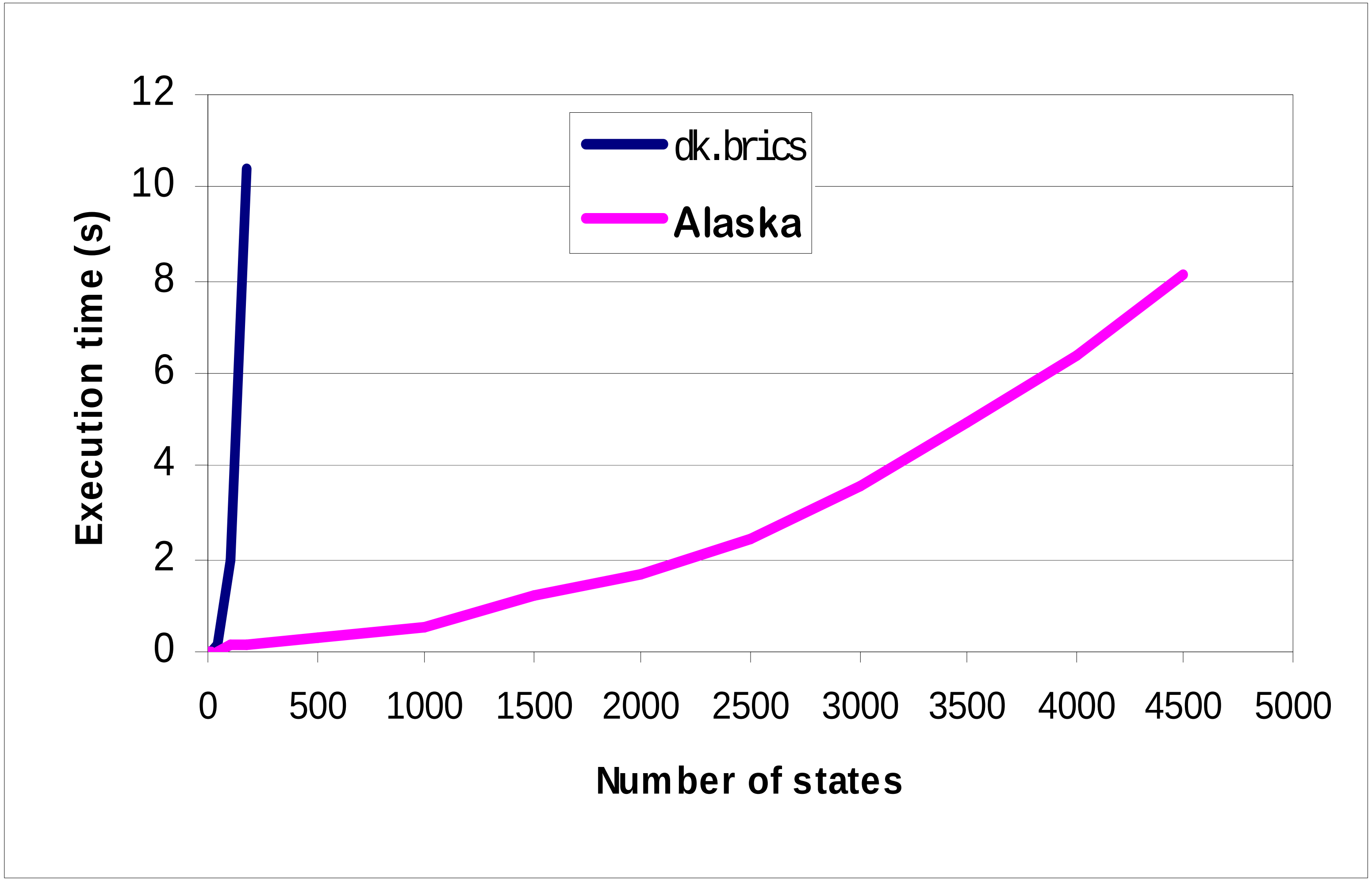
ATVA'08

TACAS'09

Raskin et al.

http://www.antichains.be

Tools



NFA universality

Tools

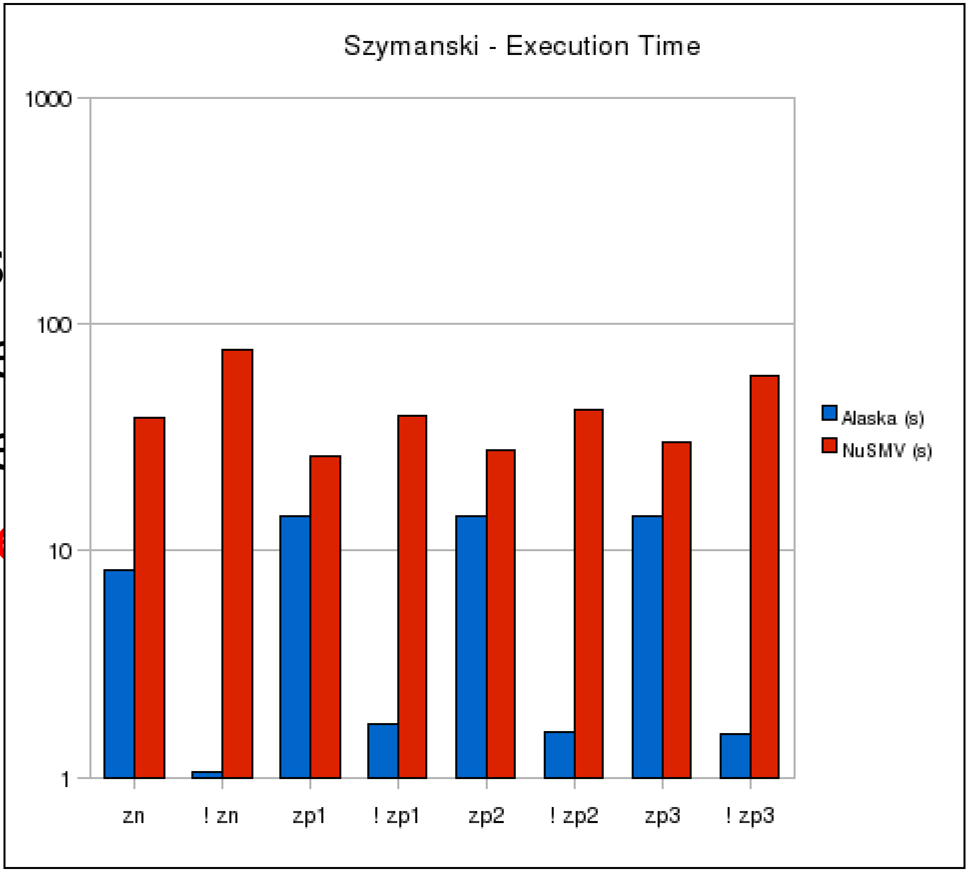
Reachability/Parity games

Finite automata (language

Büchi automata (language

LTL satisfiability and mode

LTL synthesis



50 times faster than nuSMV…

LTL model-checking

Outline

From Boolean to quantitative verification

- Boolean Verification

  1. Techniques to speed up well-known verification algorithms by orders of magnitude

- Quantitative Verification

  2. A surprising complexity result in game theory

  3. A robust and decidable class of quantitative languages

Model-checking

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

Check if a Model satisfies a Property ?

...in an automated way

[Clarke, Emerson, Sifakis,...]

Model-checking

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

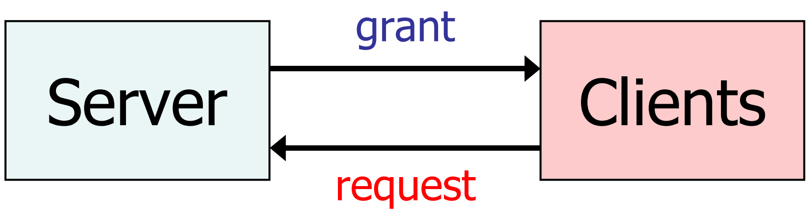
Check if a Model satisfi… operty ?

…in an … lated way

Generalisation ?

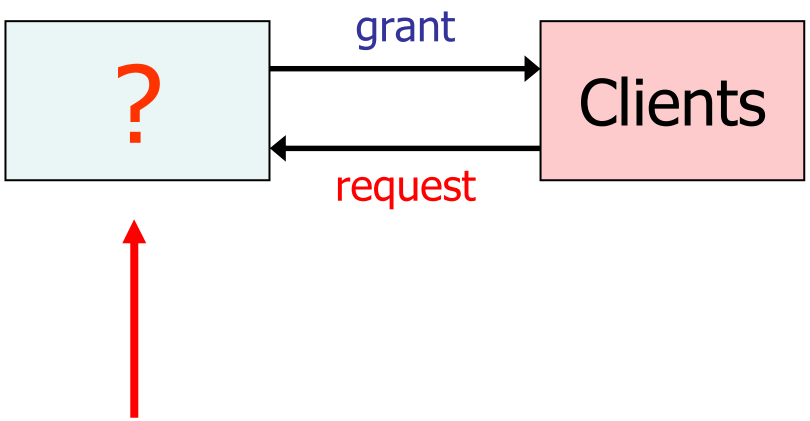
[Clarke, Emerson, Sifakis,...]

From graphs to games



« Every request is eventually granted, no simultaneous grants »

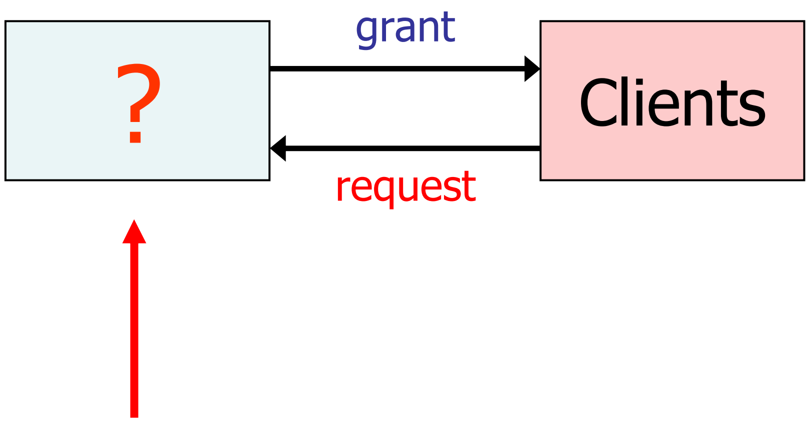
From graphs to games



(Part of) the Model
is not given

« Every request is eventually granted, no simultaneous grants »

From graphs to games



« Every request is eventually granted, no simultaneous grants »

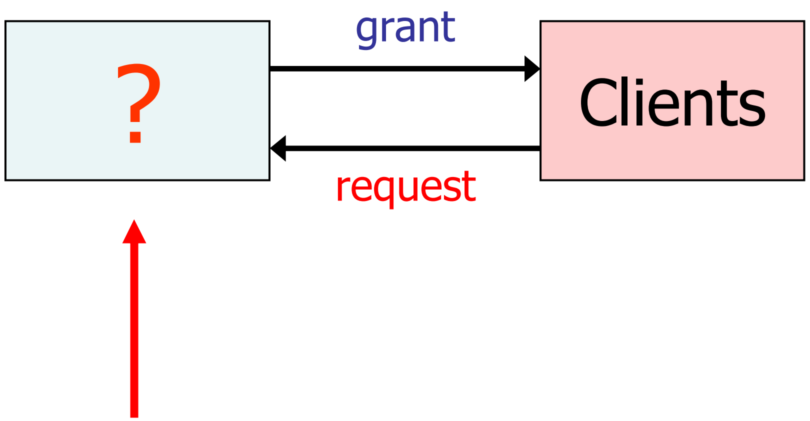
(Part of) the Model is not given

➔ Construct a correct system

(typically reduces to game solving)

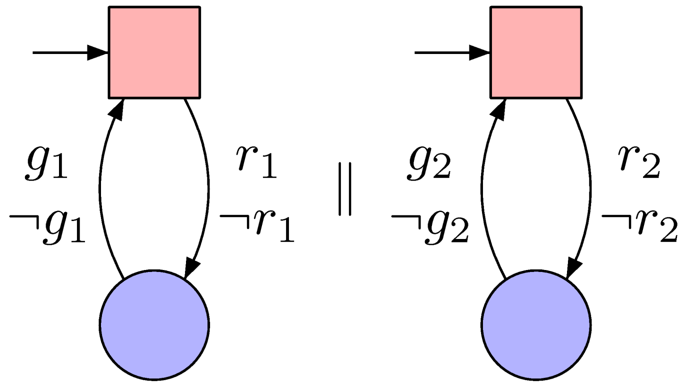
[Church, Büchi, Landweber, Rabin, Pnueli,...]

From graphs to games



(Part of) the Model
is not given

« Every request is eventually granted, no simultaneous grants »

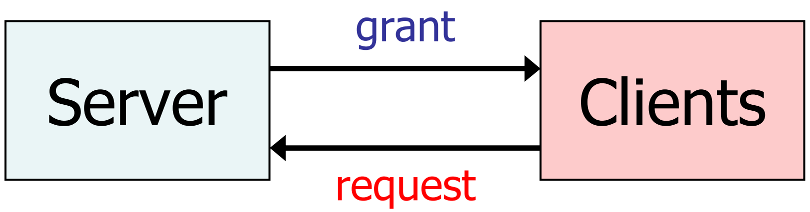


➔ Construct a correct system

(typically reduces to game solving)

[Church, Büchi, Landweber, Rabin, Pnueli,...]

From Boolean to Quantitative spec

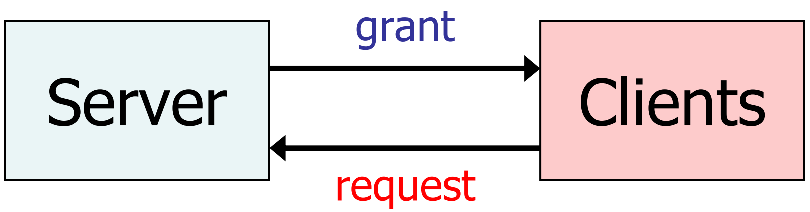


« Every request is eventually granted, no simultaneous grants »

Solution 1: grant within 10^6 years

Solution 2: grant even if no request

From Boolean to Quantitative spec



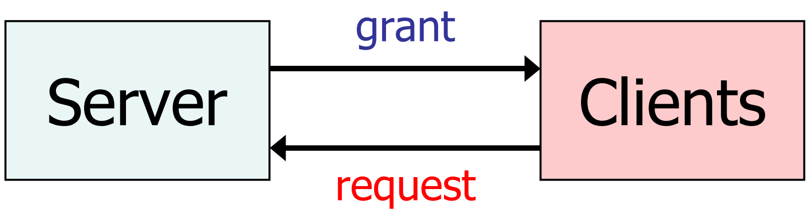
« Every request is eventually granted, no simultaneous grants »

Solution 1: grant within 10^6 years

Solution 2: grant even if no request

Boolean specs do not distinguish correct systems

From Boolean to Quantitative spec



« Every request is eventually granted, no simultaneous grants »

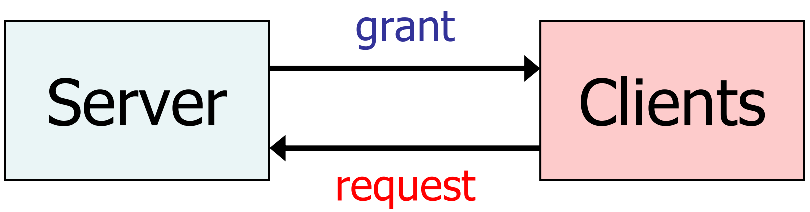
Solution 1: grant within 10^6 years

Solution 2: grant even if no request

Switch to Quantitative Spec

« Minimize delays for pending requests, minimize number of grants »

From Boolean to Quantitative spec



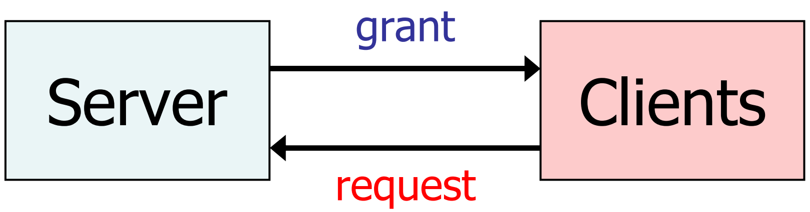
« Every request is eventually granted, no simultaneous grants »

Wrong solution 1: no grant at all

Wrong solution 2: 99% request granted

Boolean specs do not distinguish wrong systems either!

From Boolean to Quantitative spec



« Every request is eventually granted, no simultaneous grants »

Wrong solution 1: no grant at all

Wrong solution 2: 99% request granted

Switch to Quantitative Spec

« Maximize average number of granted requests »

From Boolean to…

Boolean acceptance conditions separate good and bad runs:

$$\{0,1\}^{\omega} \to \{0,1\}$$

E.g., (co)Büchi, Muller, parity, etc.

From Boolean to…

Boolean acceptance conditions separate good and bad runs:

$$\{0,1\}^\omega \to \{0,1\}$$

E.g., (co)Büchi, Muller, parity, etc.

Quantitative value functions assign value to runs:

$$\mathbb{R}^\omega \to \mathbb{R}$$

Some value functions

For $v = v_0 v_1 \dots$ ($v_i \in \mathbb{R}$), let

($v_i \in \{0,1\}$)

- $\mathsf{Sup}(v) = \sup\{v_n \mid n \geq 0\}$; (reachability)
- $\mathsf{LimSup}(v) = \limsup_{n \to \infty} v_n$; (Büchi)
- $\mathsf{LimInf}(v) = \liminf_{n \to \infty} v_n$; (coBüchi)

Some value functions

For $v = v_0 v_1 \dots$ ($v_i \in \mathbb{R}$), let ($v_i \in \{0,1\}$)

- $\mathsf{Sup}(v) = \sup\{v_n \mid n \geq 0\}$; (reachability)
- $\mathsf{LimSup}(v) = \limsup_{n\to\infty} v_n$; (Büchi)
- $\mathsf{LimInf}(v) = \liminf_{n\to\infty} v_n$; (coBüchi)
- $\mathsf{LimAvg}(v) = \limsup_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} v_i$; aka $\mathsf{MeanPayoff}(v)$
- given $0 < \lambda < 1$, $\mathsf{Disc}_\lambda(v) = \sum_{i=0}^{\infty} \lambda^i \cdot v_i$.

Outline

From Boolean to quantitative verification

- Boolean Verification
  1. Techniques to speed up well-known verification algorithms by orders of magnitude

- Quantitative Verification
  2. Mean-payoff parity games are in NP $\cap$ coNP
  3. A robust and decidable class of quantitative languages

  -

Example

Mean-payoff parity games

Example

Mean-payoff parity games

ω-regular specifications
(reactivity, liveness,…)

Example

Mean-payoff parity games

ω-regular specifications
(reactivity, liveness,…)

- Memoryless strategies
- NP $\cap$ coNP

Example

Mean-payoff parity games

Quantitative specification
(cost optimization,…)

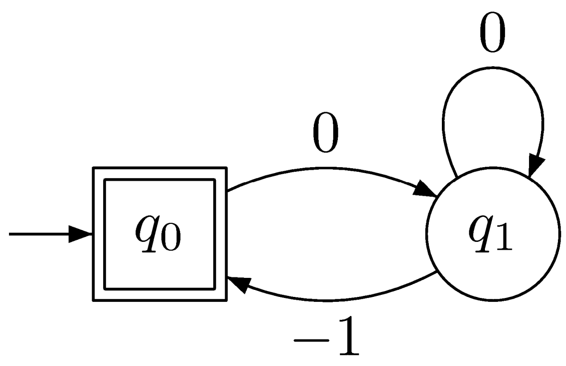
- Memoryless strategies
- NP $\cap$ coNP

ω-regular specifications
(reactivity, liveness,…)

- Memoryless strategies
- NP $\cap$ coNP

Example

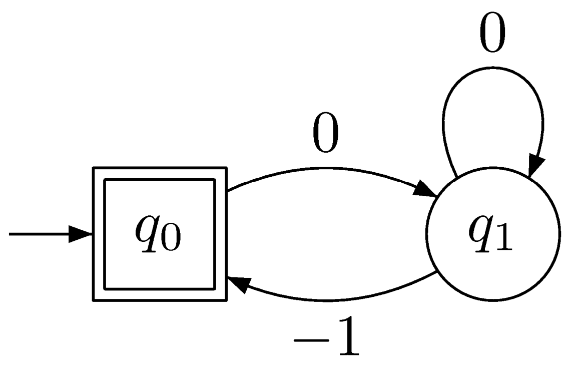
Mean-payoff Büchi games



Visit q_0 infinitely often,
and maximize mean-payoff

Example

Mean-payoff Büchi games



Visit q_0 infinitely often,
and maximize mean-payoff

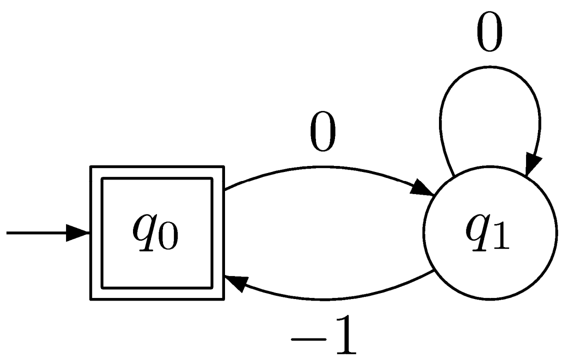
Optimal strategy: spend more and more time in q_1

$0, -1, 0, 0, -1, 0, 0, 0, -1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, -1, 0, \ldots$

Requires infinite memory...

Example

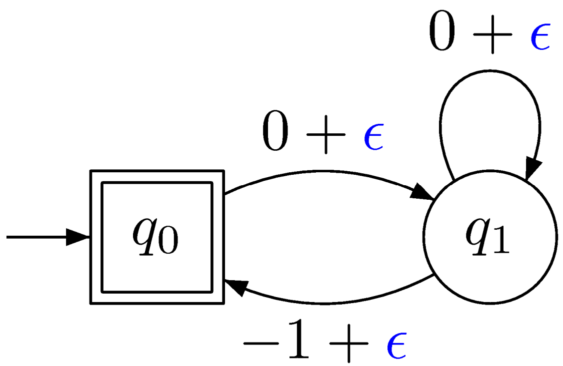
Mean-payoff parity games



- ~~Memoryless strategies~~
- still in NP $\cap$ coNP

Example

Mean-payoff parity games

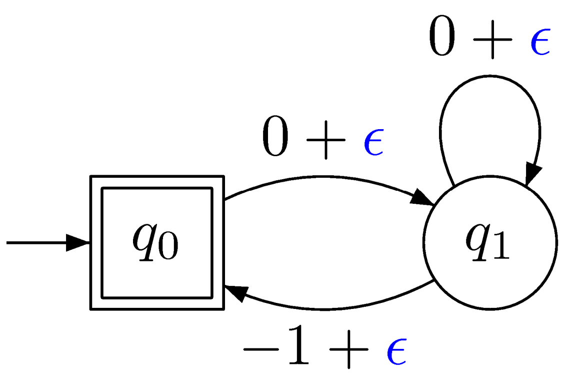


- ~~Memoryless strategies~~
- still in NP $\cap$ coNP

1. Reduction to parity games with positive counter
2. Finite-memory strategies suffice

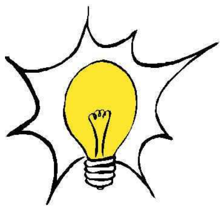
Example

Mean-payoff parity games



- ~~Memoryless strategies~~
- still in NP $\cap$ coNP

1. Reduction to parity games with positive counter
2. Finite-memory strategies suffice



3. Winning strategies can be decomposed into memoryless strategies, and combined using counters.
4. Decomposition can be guessed in NP

Example

Mean-payoff parity games

- ~~Memoryless strategies~~
- still in NP $\cap$ coNP

K. Chatterjee

ICALP'10

Outline

From Boolean to quantitative verification

- Boolean Verification
  1. Techniques to speed up well-known verification algorithms by orders of magnitude
- Quantitative Verification
  2. Mean-payoff parity games are in NP $\cap$ coNP
  3. A robust and decidable class of quantitative languages

-

Quantitative Languages

Long-term goal

Is there a Quantitative Framework with

- an appealing mathematical formulation,
- useful expressive power, robustness and
- good algorithmic properties ?

(Like the boolean theory of ω-regularity.)

Note: “Quantitative” is more than “timed” and “probabilistic”

[Henzinger,...]

Quantitative languages

A quantitative language is a function:

$$L : \Sigma^{\omega} \longrightarrow \mathbb{R}$$

L(w) can be interpreted as:

- the amount of some resource needed by the system to produce w (power, energy, time consumption),
- a reliability measure (the average number of “faults” in w).

Quantitative languages

A quantitative language is a function:

$$L : \Sigma^\omega \longrightarrow \mathbb{R}$$

Classical Boolean languages are the special case where

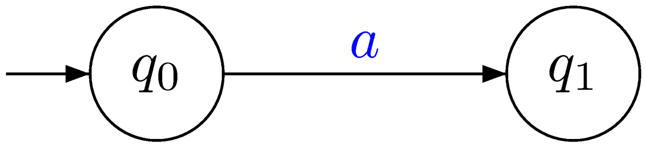
$$L : \Sigma^\omega \longrightarrow \{0, 1\}$$

L(w) can be interpreted as:

- the amount of some resource needed by the system to produce w (power, energy, time consumption),
- a reliability measure (the average number of "faults" in w).

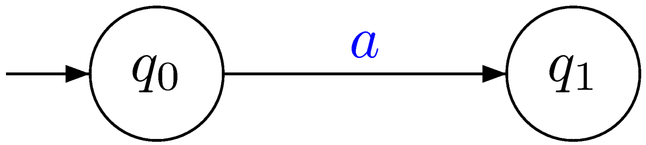
Languages & Automata

Boolean languages are generated by finite automata.



Languages & Automata

Boolean languages are generated by finite automata.

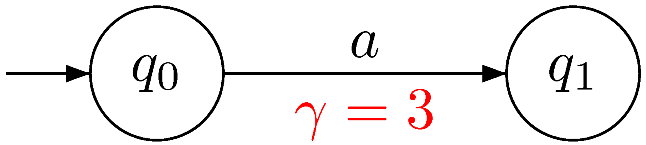


Quantitative languages are generated by weighted automata,

$L_A(w)$ =

A is deterministic:	value of (unique) run
A is non-deterministic:	sup of run values
A is universal:	inf of run values
A is alternating:	value of game-outcome run (sup inf)

...



Quantitative Languages

	det.	nondet.	univ.	alt.
Sup				
LimSup				
LimInf				
LimAvg				
Disc_λ				

20 classes of quantitative languages...

Quantitative Languages

1. Decision problems
2. Expressiveness
3. Closure properties

Decision problems

Given weighted automata A, B and $\nu \in \mathbb{Q}$

decide

Quant. emptiness $\exists w : L_A(w) \geq \nu$

Quant. universality $\forall w : L_A(w) \geq \nu$

Decision problems

Given weighted automata A, B and $\nu \in \mathbb{Q}$

decide

Quant. emptiness	$\exists w : L_A(w) \geq \nu$
Quant. universality	$\forall w : L_A(w) \geq \nu$
Quant. inclusion	$\forall w : L_A(w) \leq L_B(w)$
Quant. equivalence	$\forall w : L_A(w) = L_B(w)$

Decision problems

Given weighted automata A, B and $\nu \in \mathbb{Q}$

decide

Quant. emptiness	$\exists w : L_A(w) \geq \nu$
Quant. universality	$\forall w : L_A(w) \geq \nu$
Quant. inclusion	$\forall w : L_A(w) \leq L_B(w)$
Quant. equivalence	$\forall w : L_A(w) = L_B(w)$

	det.	nondet.	univ.	alt.
Sup	P	P	PSpace	PSpace
LimSup	P	P	PSpace	PSpace
LimInf	P	P	PSpace	PSpace
LimAvg	P	P	undec.	undec.
Disc_λ	P	P	?	?

	det.	nondet.	univ.	alt.
Sup	P	PSpace	PSpace	PSpace
LimSup	P	PSpace	PSpace	PSpace
LimInf	P	PSpace	PSpace	PSpace
LimAvg	P	undec.	undec.	undec.
Disc_λ	P	?	?	?

	det.	nondet.	univ.	alt.
Sup	P	PSpace	PSpace	PSpace
LimSup	P	PSpace	PSpace	PSpace
LimInf	P	PSpace	PSpace	PSpace
LimAvg	P	undec.	undec.	undec.
Disc_λ	P	?	?	?

	det.	nondet.	univ.	alt.
Sup	P	PSpace	PSpace	PSpace
LimSup	P	PSpace	PSpace	PSpace
LimInf	P	PSpace	PSpace	PSpace
LimAvg	P	undec.	undec.	undec.
Disc_λ	P	?	?	?

CSL'08, CSL'10, ToCL'10

Decision problems

Given weighted automata A, B and $\nu \in \mathbb{Q}$

decide

Quant. emptiness $\exists w : L_A(w) \geq \nu$

Quant. universality $\forall w : L_A(w) \geq \nu$

Quant. inclusion $\forall w : L_A(w) \leq L_B(w)$

Quant. equivalence $\forall w : L_A(w) = L_B(w)$

Undecidable for LimAvg.

Open question for Disc.

	det.	nondet.	univ.	alt.
Sup	P	P	PSpace	PSpace
LimSup	P	P	PSpace	PSpace
LimInf	P	P	PSpace	PSpace
LimAvg	P	P	undec.	undec.
Disc_λ	P	P	?	?

	det.	nondet.	univ.	alt.
Sup	P	PSpace	PSpace	PSpace
LimSup	P	PSpace	PSpace	PSpace
LimInf	P	PSpace	PSpace	PSpace
LimAvg	P	undec.	undec.	undec.
Disc_λ	P	?	?	?

	det.	nondet.	univ.	alt.
Sup	P	PSpace	PSpace	PSpace
LimSup	P	PSpace	PSpace	PSpace
LimInf	P	PSpace	PSpace	PSpace
LimAvg	P	undec.	undec.	undec.
Disc_λ	P	?	?	?

	det.	nondet.	univ.	alt.
Sup	P	PSpace	PSpace	PSpace
LimSup	P	PSpace	PSpace	PSpace
LimInf	P	PSpace	PSpace	PSpace
LimAvg	P	undec.	undec.	undec.
Disc_λ	P	?	?	?

CSL'08, CSL'10, ToCL'10

Quantitative Languages

1. Decision problems
2. Expressiveness
3. Closure properties

Expressiveness

Compare classes of quantitative languages defined by weighted automata

O(20 x 20) comparisons…

Expressiveness

Compare classes of quantitative languages defined by weighted automata

O(20 x 20) comparisons…

LimAvg and Disc_λ cannot be determinized.

LICS'09, LMCS'10

Quantitative Languages

1. Decision problems
2. Expressiveness
3. Closure properties

Operations

$$L_1, L_2 : \Sigma^\omega \to \mathbb{R}$$

Operations on quantitative languages:

- $\max(L_1, L_2)$ — $L_1 \cup L_2$
- $\min(L_1, L_2)$ — $L_1 \cap L_2$
- complement(L_1) = $1 - L_1$ — $\Sigma^\omega \setminus L_1$
- $L_1 + L_2$

Operations

$$L_1, L_2 : \Sigma^\omega \to \mathbb{R}$$

Operations on quantitative languages:

- $\max(L_1,L_2)$ — $L_1 \cup L_2$
- $\min(L_1,L_2)$ — $L_1 \cap L_2$
- complement(L_1) = $1-L_1$ — $\Sigma^\omega \setminus L_1$
- $L_1 + L_2$

Note $L_1 \leq L_2$ iff $L_1 + (1-L_2) \leq 1$

LimAvg Automata

LimAvg	Closure properties			
	max	min	Sum	comp.
Deterministic	×	×	×	√
Nondeterministic	√	×	×	×
Alternating	√	√	×	√

LICS'09, FCT'09

LimAvg Automata

LimAvg	Closure properties				Decision problems			
	max	min	Sum	comp.	empt.	univ.	incl.	equiv.
Deterministic	×	×	×	√	√	√	√	√
Nondeterministic	√	×	×	×	√	×	×	×
Alternating	√	√	×	√	×	×	×	×

LICS'09, FCT'09

Beyond Weighted Automata

LimAvg Automata

LimAvg	Closure properties				Decision problems			
	max	min	Sum	comp.	empt.	univ.	incl.	equiv.
Deterministic	×	×	×	√	√	√	√	√
Nondeterministic	√	×	×	×	√	×	×	×
Alternating	√	√	×	√	×	×	×	×
Expressions	√	√	√	√	√	√	√	√

LimAvg Automaton Expressions

LimAvg-automaton expressions are defined by:

E ::= A | max(E,E) | min(E,E) | Sum(E,E)

where A is a deterministic LimAvg-automaton.

LimAvg Automaton Expressions

LimAvg-automaton expressions are defined by:

E ::= A | max(E,E) | min(E,E) | Sum(E,E)

where A is a deterministic LimAvg-automaton.

E.g.: max(A_1 + A_2, min(A_3, A_4))

LimAvg Automaton Expressions

LimAvg-automaton expressions are defined by:

E ::= A | max(E,E) | min(E,E) | Sum(E,E)

where A is a deterministic LimAvg-automaton.

Closure properties:

LimAvg	Closure properties			
	max	min	Sum	comp.
Deterministic	×	×	×	√
Nondeterministic	√	×	×	×
Alternating	√	√	×	√
Expressions	√	√	√	√

LimAvg Automaton Expressions

LimAvg-automaton expressions are defined by:

E ::= A | max(E,E) | min(E,E) | Sum(E,E)

where A is a deterministic LimAvg-automaton.

Decision problems: all questions reduce to quant. emptiness

$$\exists w : \mathsf{E}(w) \geq \nu$$

Value set

Solve decision problems using the value set:

E.g.: $E = \max(A_1 + A_2, \min(A_3, A_4))$

Value Set = $\{ (L_{A_1}(w), L_{A_2}(w), L_{A_3}(w), L_{A_4}(w)) \mid w \in \Sigma^\omega \} \subseteq \mathbb{R}^4$

How to compute this set ?

Value set

Solve decision problems using the value set:

E.g.: $E = \max(A_1 + A_2, \min(A_3, A_4))$

Value Set = $\{ (L_{A_1}(w), L_{A_2}(w), L_{A_3}(w), L_{A_4}(w)) \mid w \in \Sigma^\omega \} \subseteq \mathbb{R}^4$

How to compute this set ?

Uses arguments in computational geometry, yields 4EXPTIME complexity for emptiness.

Value set

Solve decision problems using the value set:

E.g.: $E = \max(A_1 + A_2, \min(A_3, A_4))$

Value Set = $\{ (L_{A_1}(w), L_{A_2}(w), L_{A_3}(w), L_{A_4}(w)) \mid w \in \Sigma^\omega \} \subseteq \mathbb{R}^4$

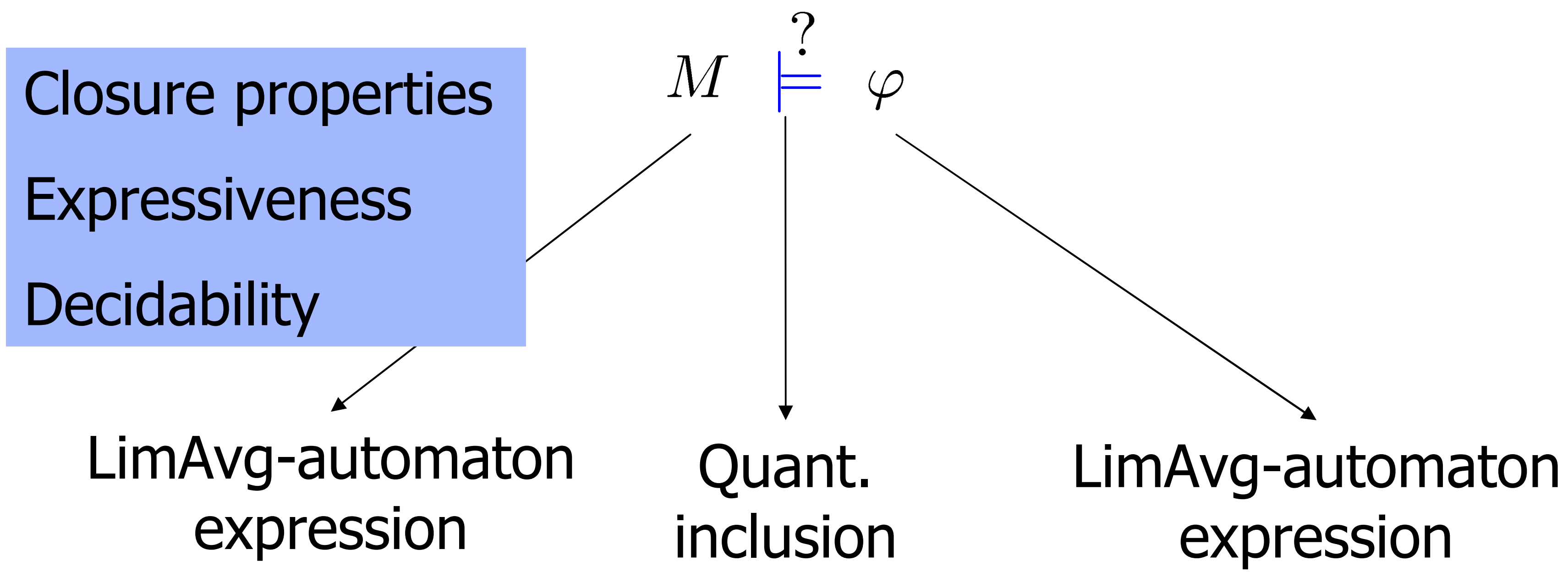
$E(\Sigma^\omega) = \{ \max(x+y, \min(z,t)) \mid (x,y,z,t) \in \text{Value Set} \}$

is a finite union of intervals.

Find maximum of $E(\Sigma^\omega)$ to solve emptiness

LimAvg Automaton Expressions

LimAvg	Closure properties				Decision problems			
	max	min	Sum	comp.	empt.	univ.	incl.	equiv.
Deterministic	✗	✗	✗	✓	✓	✓	✓	✓
Nondeterministic	✓	✗	✗	✗	✓	✗	✗	✗
Alternating	✓	✓	✗	✓	✗	✗	✗	✗
Expressions	✓	✓	✓	✓	✓	✓	✓	✓



LimAvg Automaton Expressions

LimAvg	Closure properties				Decision problems			
	max	min	Sum	comp.	empt.	univ.	incl.	equiv.
Deterministic	×	×	×	√	√	√	√	√
Nondeterministic	√	×	×	×	√	×	×	×
Alternating	√	√	×	√	×	×	×	×
Expressions	√	√	√	√	√	√	√	√

Closure properties

ssiveness

al

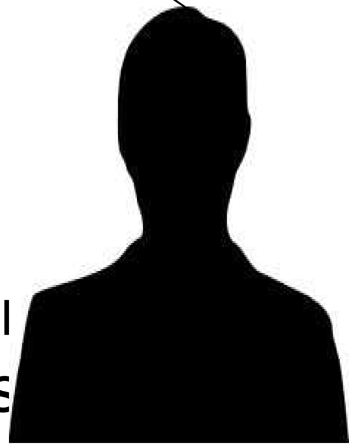
$M \overset{?}{\models} \varphi$

CONCUR'10

K. Chatterjee

H. Edelsbrunner

T. Henzinger



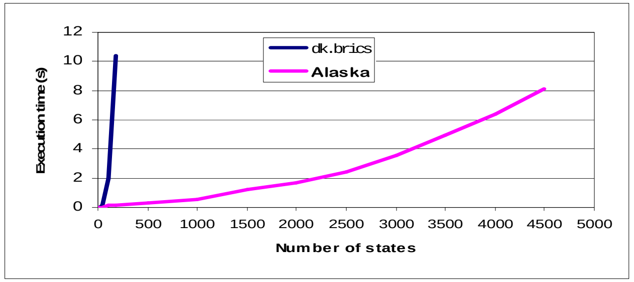
P. Rannou

mAvg-automaton
expression

Conclusion

Conclusion – Key results

1. Efficient antichain algorithms



2. Quantitative games

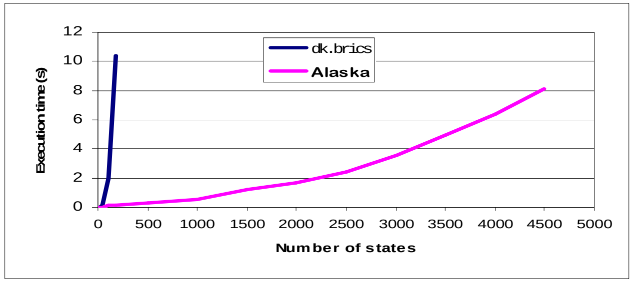
   Mean-payoff parity games in NP $\cap$ coNP

3. Quantitative generalization of languages

   LimAvg automaton expressions: robust and decidable

Perspectives

1. Efficient antichain algorithms



Can we predict the performance of antichain algorithms ?

Complexity theory beyond worst-case…

Perspectives

2. Quantitative games

Mean-payoff parity games in NP $\cap$ coNP

- Multi-dimensional mean-payoff games – complexity
- New classes of quantitative stochastic games

  in progress, PhD thesis of Mahsa Shirmohammadi

- New classes of games on counter systems

  in progress, PhD thesis of Julien Reichert

Perspectives

3. Quantitative generalization of languages

LimAvg automaton expressions: robust and decidable

- Discounted-sum “expressions” ?
- Incorporate Boolean conditions
- Theory of quantitative regularity
  - analogous of Borel hierarchy
  - safety vs. liveness
  - logical characterization

Acknowledgments

The work in this thesis has been carried out in the following teams:

- Tom Henzinger (EPFL, 2006-2008)
- Jean-François Raskin (ULB, 2009)
- Alain Finkel (LSV, 2009-now)

T. Henzinger

J-F. Raskin

A. Finkel

Credits

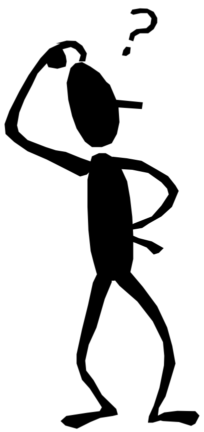
With the following co-authors (students in blue):

- Dietmar Berwanger
- Thomas Brihaye
- Lubos Brim
- Véronique Bruyère
- Jakub Chaloupka
- Krishnendu Chatterjee
- Aldric Degorre
- Martin De Wulf
- Marc Ducobu
- Herbert Edelsbrunner
- Gilles Geeraerts
- Raffaella Gentilini
- Hugo Gimbert
- Tom Henzinger
- Barbara Jobstmann
- Axel Legay
- Nicolas Maquet
- Nicolas Markey
- Thierry Massart
- Dejan Nickovic
- Joël Ouaknine
- Tatjana Petrov
- Sangram Raje
- Philippe Rannou
- Jean-François Raskin
- Julien Reichert
- Mahsa Shirmohammadi
- Rohit Singh
- Szymon Torunczyk
- James Worrell

edits

The end

Thank you !



Questions ?