Goal-Oriented Reduction for Automata Networks

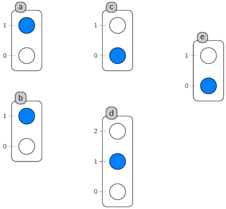
Loïc Paulevé

LRI, CNRS / Université Paris-Sud, France
loic.pauleve@lri.fr
http://loicpauleve.name

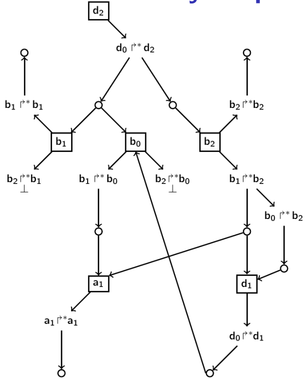
Journées Bioss “Réduction”, Marseille, May 28, 2015

Overview

Automata network



Local Causality Graph



Over-approximation of reachability
Under-approximation of reachability
Under-approximation of cut sets

NEW **Model reduction**

Goal-oriented reduction

Motivations

Local causality analysis in previous work...

- ...decide efficiently reachability properties (and cut sets)
- ...but they can be inconclusive (abstractions).

We may still want to do **exhaustive explorations** of the state space...

- ...to ensure conclusive-ness
- ...to ensure that we are not missing cut-sets for reachability
- ...to do any more precise analysis.

May local causality analysis help exhaustive analysis of the state space?

Model reduction

Model reductions

- merge/remove components/transitions,
- try to preserve some properties.

Goal-oriented reduction of automata networks

- dedicated to a given reachability property (reach a_i, then b_j, ...);
- reduction by removing transitions;
- conserve all minimal traces satisfying a reachability property
- valid for any update schedule.

Minimal traces (sequences of transitions)
A trace $\pi \models P$ is minimal w.r.t. P iff there is no sub-trace $\pi' \subsetneq \pi$ s.t. $\pi' \models P$.

Examples for $P =$ reach a_i:

- $b_0 \xrightarrow{c_0} b_1, c_0 \xrightarrow{b_1} c_1, a_0 \xrightarrow{b_1, c_1} a_i$ (YES)
- $b_0 \xrightarrow{c_0} b_1, c_0 \xrightarrow{b_1} c_1, d_0 \xrightarrow{c_1} d_1, a_0 \xrightarrow{b_1, c_1} a_i$ (NO)
- $b_0 \xrightarrow{c_0} b_1, c_0 \xrightarrow{b_1} c_1, b_1 \xrightarrow{a_0} b_0, d_0 \xrightarrow{c_1} d_1, b_0 \xrightarrow{d_1} b_1, a_0 \xrightarrow{b_1, c_1} a_i$ (NO)

Model reduction

Model reductions

- merge/remove components/transitions,
- try to preserve some properties.

Goal-oriented reduction of automata networks

- dedicated to a given reachability property (reach a_i, then b_j, ...);
- reduction by removing transitions;
- conserve all minimal traces satisfying a reachability property
- valid for any update schedule.

Minimal traces (sequences of transitions)
A trace $\pi \models P$ is minimal w.r.t. P iff there is no sub-trace $\pi' \subsetneq \pi$ s.t. $\pi' \models P$.

Examples for $P =$ reach a_i:

- $b_0 \xrightarrow{c_0} b_1, c_0 \xrightarrow{b_1} c_1, a_0 \xrightarrow{b_1,c_1} a_i$ (YES)
- $b_0 \xrightarrow{c_0} b_1, c_0 \xrightarrow{b_1} c_1, \mathbf{d_0} \xrightarrow{\mathbf{c_1}} \mathbf{d_1}, a_0 \xrightarrow{b_1,c_1} a_i$ (NO)
- $b_0 \xrightarrow{c_0} b_1, c_0 \xrightarrow{b_1} c_1, b_1 \xrightarrow{a_0} b_0, d_0 \xrightarrow{c_1} d_1, b_0 \xrightarrow{d_1} b_1, a_0 \xrightarrow{b_1,c_1} a_i$ (NO)

Model reduction

Model reductions

- merge/remove components/transitions,
- try to preserve some properties.

Goal-oriented reduction of automata networks

- dedicated to a given reachability property (reach a_i, then b_j, ...);
- reduction by removing transitions;
- conserve all minimal traces satisfying a reachability property
- valid for any update schedule.

Minimal traces (sequences of transitions)
A trace $\pi \models P$ is minimal w.r.t. P iff there is no sub-trace $\pi' \subsetneq \pi$ s.t. $\pi' \models P$.

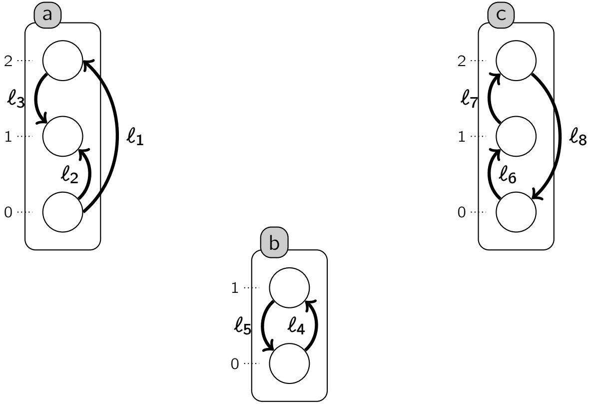
Examples for $P =$ reach a_i:

- $b_0 \xrightarrow{c_0} b_1, c_0 \xrightarrow{b_1} c_1, a_0 \xrightarrow{b_1, c_1} a_i$ (YES)
- $b_0 \xrightarrow{c_0} b_1, c_0 \xrightarrow{b_1} c_1, d_0 \xrightarrow{c_1} d_1, a_0 \xrightarrow{b_1, c_1} a_i$ (NO)
- $b_0 \xrightarrow{c_0} b_1, c_0 \xrightarrow{b_1} c_1, \mathbf{b_1} \xrightarrow{\mathbf{a_0}} \mathbf{b_0}, \mathbf{d_0} \xrightarrow{\mathbf{c_1}} \mathbf{d_1}, \mathbf{b_0} \xrightarrow{\mathbf{d_1}} \mathbf{b_1}, a_0 \xrightarrow{b_1, c_1} a_i$ (NO)

Outline

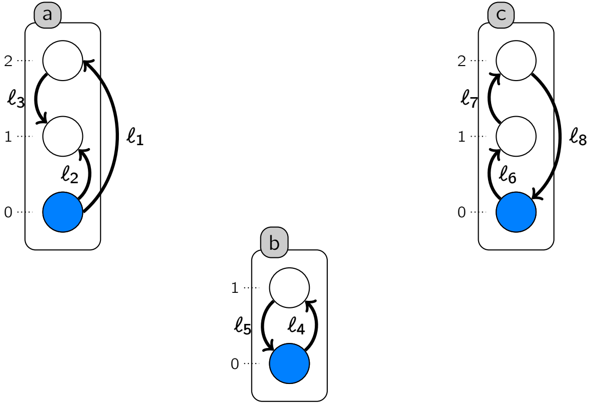
Outline

Asynchronous Finite Automata Networks



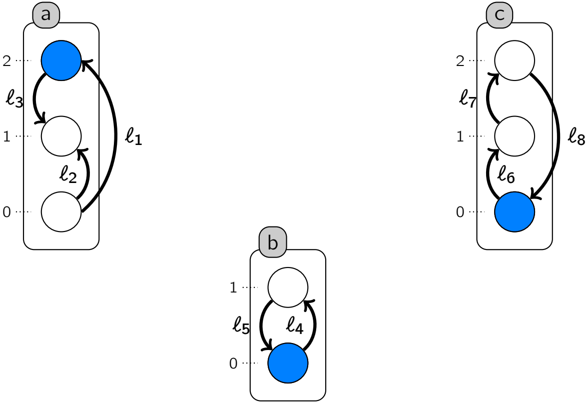
a:	$\ell_1 = \{c_0\}$	$\ell_2 = \{b_0\}$	$\ell_3 = \emptyset$
b:	$\ell_4 = \{a_2, c_1\}$	$\ell_5 = \{a_0\}$	
c:	$\ell_6 = \{b_0\}$	$\ell_7 = \{b_0, a_1\}$	$\ell_8 = \{b_1\}$

Asynchronous Finite Automata Networks



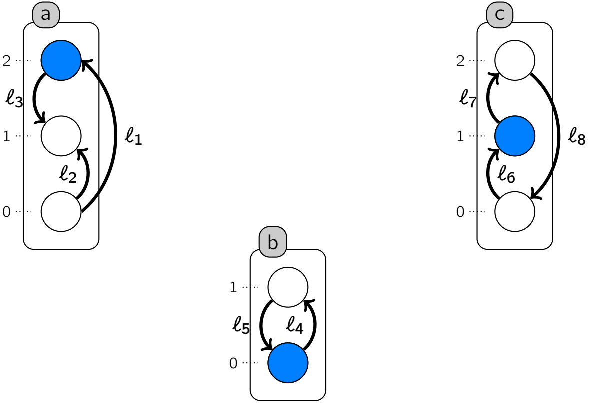
a:	$\ell_1 = \{c_0\}$	$\ell_2 = \{b_0\}$	$\ell_3 = \emptyset$
b:	$\ell_4 = \{a_2, c_1\}$	$\ell_5 = \{a_0\}$	
c:	$\ell_6 = \{b_0\}$	$\ell_7 = \{b_0, a_1\}$	$\ell_8 = \{b_1\}$

Asynchronous Finite Automata Networks



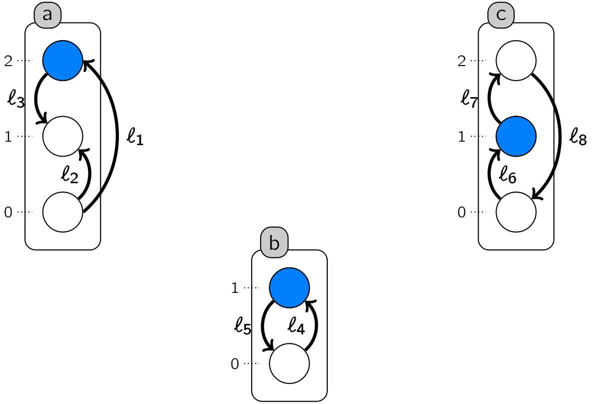
a:	$\ell_1 = \{c_0\}$	$\ell_2 = \{b_0\}$	$\ell_3 = \emptyset$
b:	$\ell_4 = \{a_2, c_1\}$	$\ell_5 = \{a_0\}$	
c:	$\ell_6 = \{b_0\}$	$\ell_7 = \{b_0, a_1\}$	$\ell_8 = \{b_1\}$

Asynchronous Finite Automata Networks



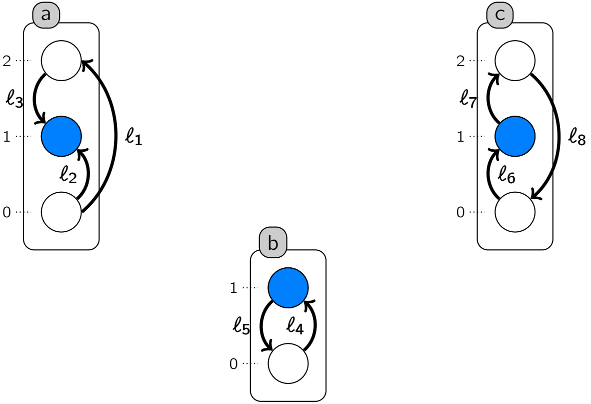
a:	$\ell_1 = \{c_0\}$	$\ell_2 = \{b_0\}$	$\ell_3 = \emptyset$
b:	$\ell_4 = \{a_2, c_1\}$	$\ell_5 = \{a_0\}$	
c:	$\ell_6 = \{b_0\}$	$\ell_7 = \{b_0, a_1\}$	$\ell_8 = \{b_1\}$

Asynchronous Finite Automata Networks



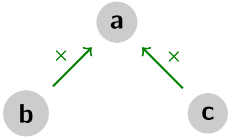
a:	$\ell_1 = \{c_0\}$	$\ell_2 = \{b_0\}$	$\ell_3 = \emptyset$
b:	$\ell_4 = \{a_2, c_1\}$	$\ell_5 = \{a_0\}$	
c:	$\ell_6 = \{b_0\}$	$\ell_7 = \{b_0, a_1\}$	$\ell_8 = \{b_1\}$

Asynchronous Finite Automata Networks



a:	$\ell_1 = \{c_0\}$	$\ell_2 = \{b_0\}$	$\ell_3 = \emptyset$
b:	$\ell_4 = \{a_2, c_1\}$	$\ell_5 = \{a_0\}$	
c:	$\ell_6 = \{b_0\}$	$\ell_7 = \{b_0, a_1\}$	$\ell_8 = \{b_1\}$

Transition-centered specification

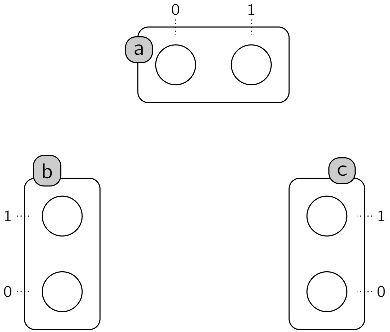


1. $f^a(x) = x[b] \wedge x[c]$

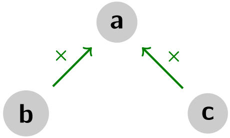
transitions:

$a_0 \to a_1 :\ b_1 \wedge c_1$

$a_1 \to a_0 :\ b_0 \vee c_0$



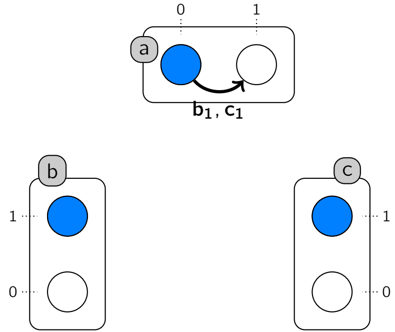
Transition-centered specification



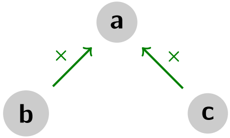
1. $f^a(x) = x[b] \wedge x[c]$
transitions:

$a_0 \rightarrow a_1$: $b_1 \wedge c_1$

$a_1 \rightarrow a_0$: $b_0 \vee c_0$



Transition-centered specification

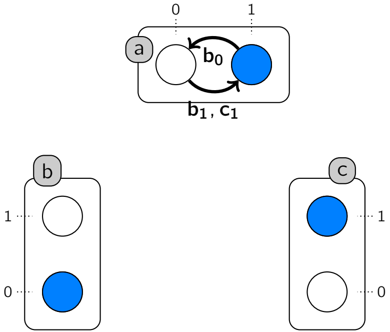


1. $f^a(x) = x[b] \wedge x[c]$

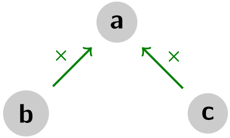
transitions:

$a_0 \rightarrow a_1$: $b_1 \wedge c_1$

$a_1 \rightarrow a_0$: $b_0 \vee c_0$



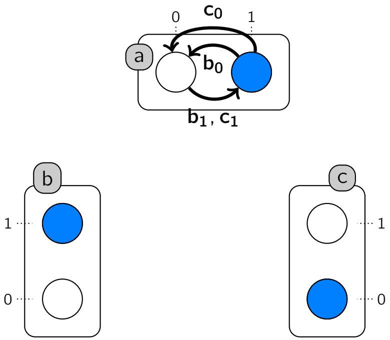
Transition-centered specification



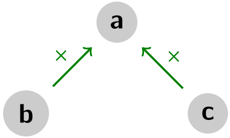
1. $f^a(x) = x[b] \wedge x[c]$
transitions:

$a_0 \to a_1$: $b_1 \wedge c_1$

$a_1 \to a_0$: $b_0 \vee c_0$



Transition-centered specification



1. $f^a(x) = x[b] \wedge x[c]$
transitions:

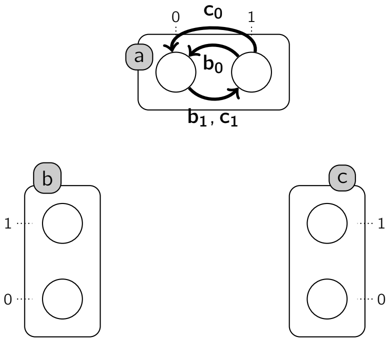
$a_0 \to a_1$: $b_1 \wedge c_1$

$a_1 \to a_0$: $b_0 \vee c_0$

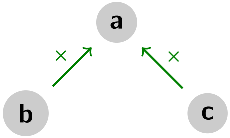
2. Non-deterministic f^a
transitions:

$a_0 \to a_1$: $b_1 \vee c_1$

$a_1 \to a_0$: $b_0 \vee c_0$



Transition-centered specification



1. $f^a(x) = x[b] \wedge x[c]$
transitions:

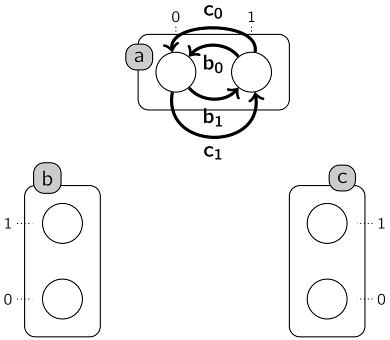
$a_0 \to a_1$: $b_1 \wedge c_1$

$a_1 \to a_0$: $b_0 \vee c_0$

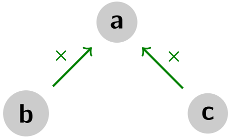
2. Non-deterministic f^a
transitions:

$a_0 \to a_1$: $b_1 \vee c_1$

$a_1 \to a_0$: $b_0 \vee c_0$



Transition-centered specification



1. $f^a(x) = x[b] \wedge x[c]$
transitions:

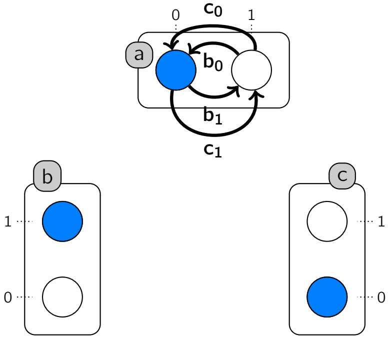
$a_0 \rightarrow a_1\colon\ b_1 \wedge c_1$

$a_1 \rightarrow a_0\colon\ b_0 \vee c_0$

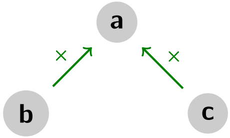
2. Non-deterministic f^a
transitions:

$a_0 \rightarrow a_1\colon\ b_1 \vee c_1$

$a_1 \rightarrow a_0\colon\ b_0 \vee c_0$



Transition-centered specification



1. $f^a(x) = x[b] \wedge x[c]$
transitions:

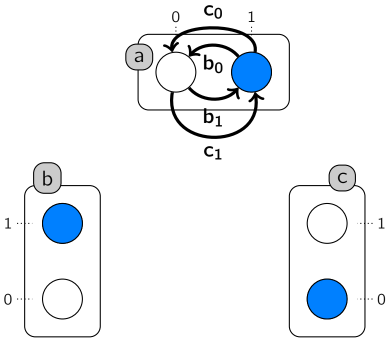
$a_0 \rightarrow a_1$: $b_1 \wedge c_1$

$a_1 \rightarrow a_0$: $b_0 \vee c_0$

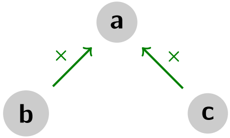
2. Non-deterministic f^a
transitions:

$a_0 \rightarrow a_1$: $b_1 \vee c_1$

$a_1 \rightarrow a_0$: $b_0 \vee c_0$



Transition-centered specification



1. $f^a(x) = x[b] \wedge x[c]$
transitions:

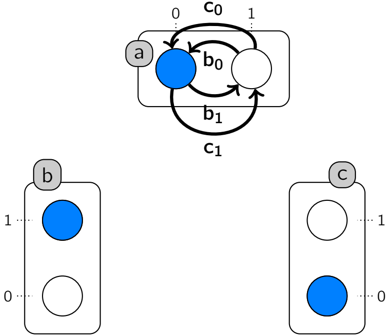
$a_0 \to a_1\colon\ b_1 \wedge c_1$

$a_1 \to a_0\colon\ b_0 \vee c_0$

2. Non-deterministic f^a
transitions:

$a_0 \to a_1\colon\ b_1 \vee c_1$

$a_1 \to a_0\colon\ b_0 \vee c_0$



Traces

Step

- Set of local transitions
- At most one transition per automaton

$$\tau = \{b_0 \xrightarrow{c_0} b_1, a_0 \xrightarrow{b_0, d_1} a_2\}$$
$$^\bullet\tau = \{b_0, c_0, a_0, d_1\}$$

- Is playable in state s iff $^\bullet\tau \subset s$.

Trace for a_i reachability

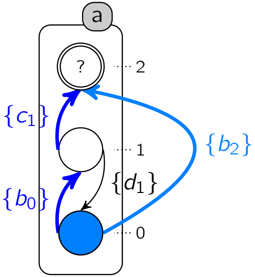
- Sequence π of playable steps with $\pi^{\#\pi} = \{a_j \xrightarrow{\cdots} a_i\}$.
- Minimal iff there is no sub-sequence ϖ for a_i reachability.

In the following:

- We consider only singleton steps (asynchronous update schedule)
- ... but results apply with any steps (any update schedule).

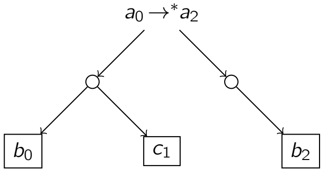
Outline

Local Causality

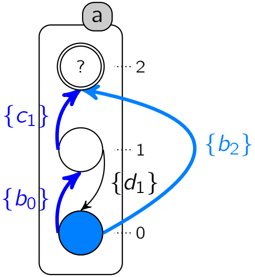


$$\text{local-paths}(a_0 \rightarrow^* a_2) = \{a_0 \xrightarrow{b_0} a_1 \xrightarrow{c_1} a_2,\; a_0 \xrightarrow{b_2} a_2\}$$

$$\text{local-paths}^\#(a_0 \rightarrow^* a_2) = \{\{b_0, c_1\}, \{b_2\}\}$$

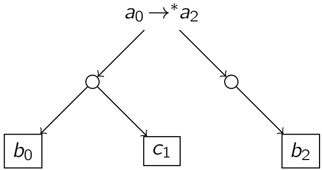


Local Causality



$$\text{local-paths}(a_0 \to^* a_2) = \{a_0 \xrightarrow{b_0} a_1 \xrightarrow{c_1} a_2,\; a_0 \xrightarrow{b_2} a_2\}$$

$$\text{local-paths}^{\#}(a_0 \to^* a_2) = \{\{b_0, c_1\}, \{b_2\}\}$$

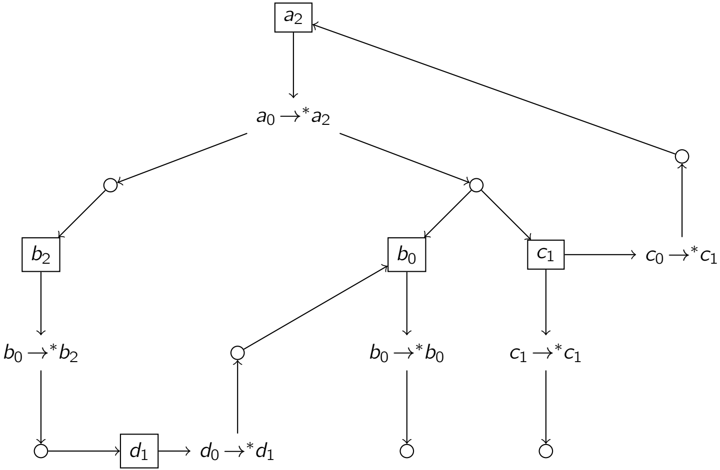


For any trace π starting at some global state s with $a_0 \in s$ and reaching a_2:

- either $a_0 \xrightarrow{b_0} a_1 \xrightarrow{c_1} a_2$ or $a_0 \xrightarrow{b_2} a_2$ is a sub-trace of π;
- either b_1 and c_0, or b_2 are reached before a_2 in π.

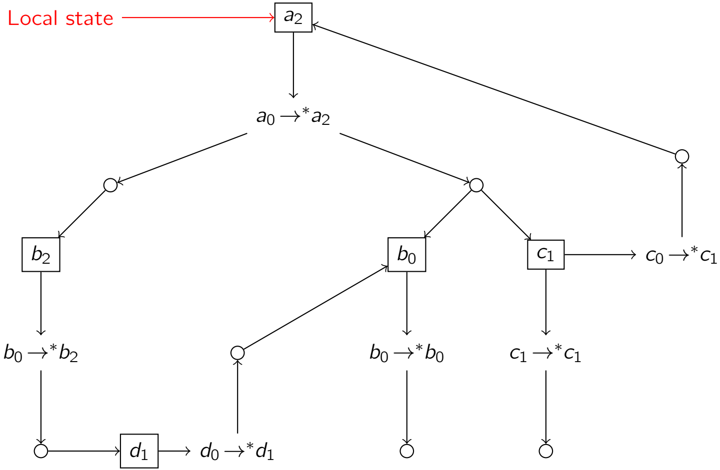
Local Causality Graph

- Causality of a_2.
- Initial context $\varsigma = \{a \mapsto \{0\}; b \mapsto \{0\}; c \mapsto \{0,1\}; d \mapsto \{0\}\}$.



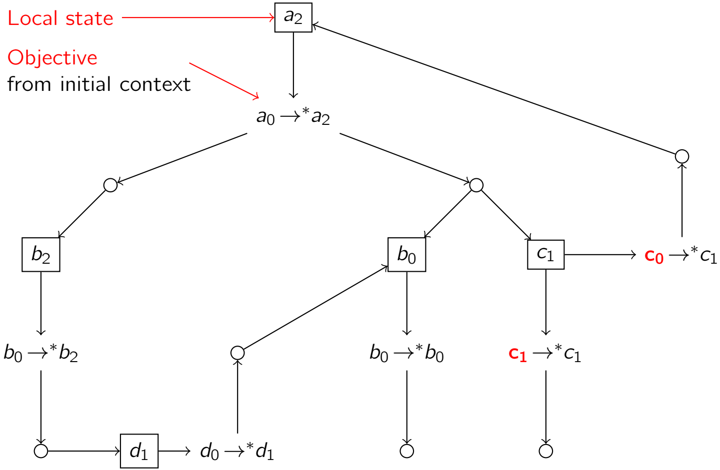
Local Causality Graph

- Causality of a_2.
- Initial context $\varsigma = \{a \mapsto \{0\}; b \mapsto \{0\}; c \mapsto \{0,1\}; d \mapsto \{0\}\}$.



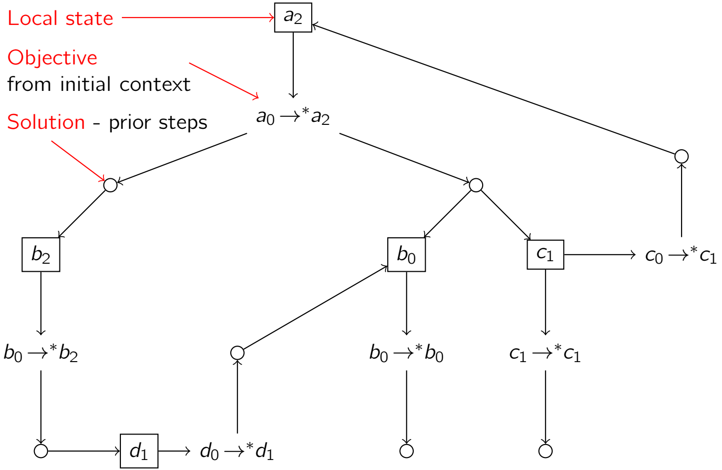
Local Causality Graph

- Causality of a_2.
- Initial context $\varsigma = \{a \mapsto \{0\}; b \mapsto \{0\}; c \mapsto \{\mathbf{0}, \mathbf{1}\}; d \mapsto \{0\}\}$.



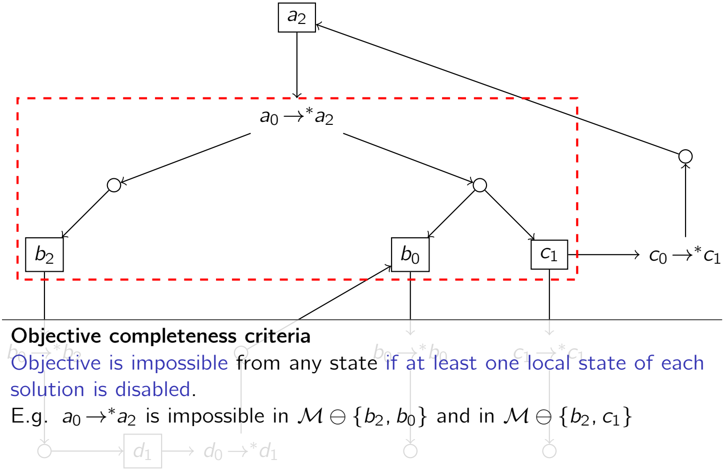
Local Causality Graph

- Causality of a_2.
- Initial context $\varsigma = \{a \mapsto \{0\}; b \mapsto \{0\}; c \mapsto \{0, 1\}; d \mapsto \{0\}\}$.



Local Causality Graph

- Causality of a_2.
- Initial context $\varsigma = \{a \mapsto \{0\}; b \mapsto \{0\}; c \mapsto \{0, 1\}; d \mapsto \{0\}\}$.



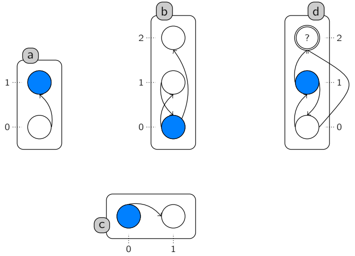
Objective completeness criteria

Objective is impossible from any state if at least one local state of each solution is disabled.

E.g. $a_0 \to^* a_2$ is impossible in $\mathcal{M} \ominus \{b_2, b_0\}$ and in $\mathcal{M} \ominus \{b_2, c_1\}$

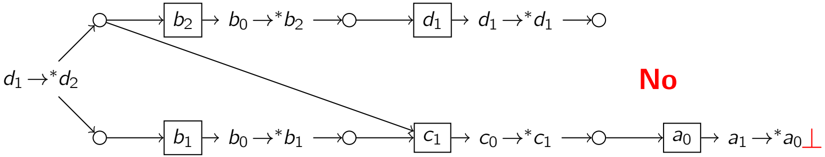
Necessary conditions for reachability

Example



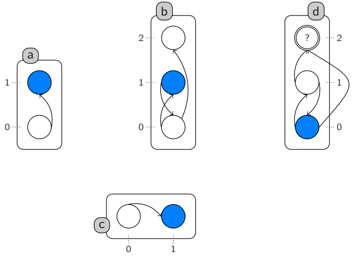
Necessary condition for d_2 reachability from ς:
There exists a traversal of the LCG s.t.:

- objective → follow at least one solution;
- local state → follow all objectives;
- no cycle.



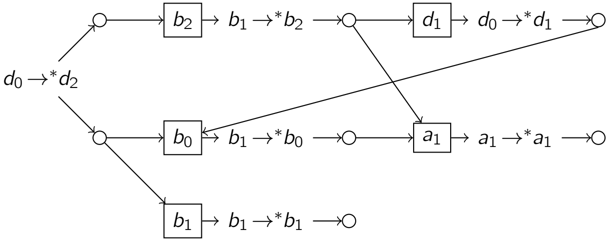
Necessary conditions for reachability

Example



Necessary condition for d_2 reachability from ς:
There exists a traversal of the LCG s.t.:

- objective → follow at least one solution;
- local state → follow all objectives;
- no cycle.



Inconc

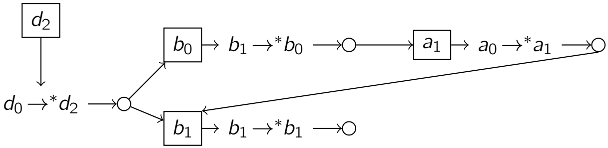
Outline

Reduction for single local reachability

Sketch

1. Compute LCG $\mathcal{G}$ from initial context for given local reachability property
2. Remove impossible objectives
3. Extends its context with local states nodes + intermediates given by local-paths
4. Repeat until fixpoint $\rightarrow \lceil \mathcal{G} \rceil$

$\Rightarrow$ keep only transitions in $\bigcup\{\mathrm{tr}(\text{local-paths}(a_i \rightarrow^* a_j)) \mid a_i \rightarrow^* a_j \in \lceil \mathcal{G} \rceil\}$

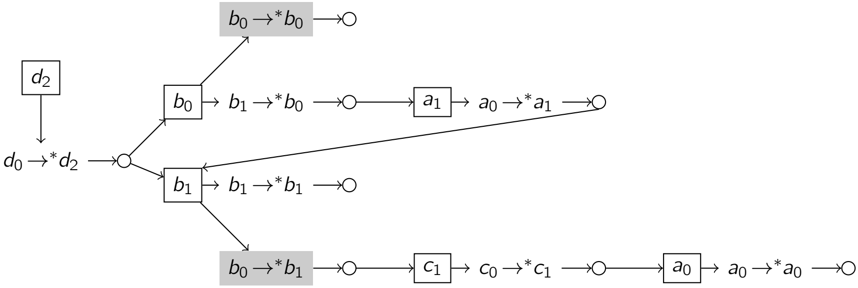


Reduction for single local reachability

Sketch

1. Compute LCG $\mathcal{G}$ from initial context for given local reachability property
2. Remove impossible objectives
3. Extends its context with local states nodes + intermediates given by local-paths
4. Repeat until fixpoint $\rightarrow \lceil \mathcal{G} \rceil$

$\Rightarrow$ keep only transitions in $\bigcup\{\mathrm{tr}(\text{local-paths}(a_i \rightarrow^* a_j)) \mid a_i \rightarrow^* a_j \in \lceil \mathcal{G} \rceil\}$

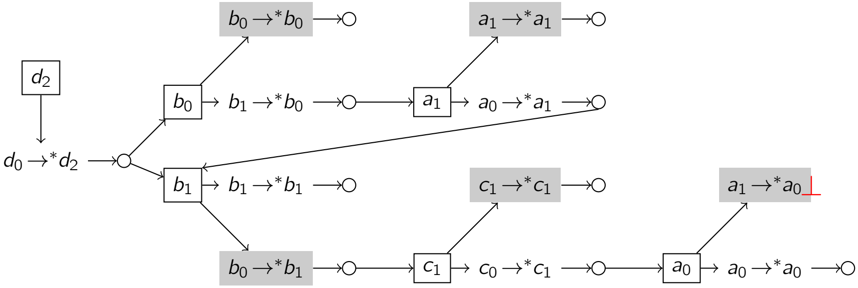


Reduction for single local reachability

Sketch

1. Compute LCG $\mathcal{G}$ from initial context for given local reachability property
2. Remove impossible objectives
3. Extends its context with local states nodes + intermediates given by local-paths
4. Repeat until fixpoint $\rightarrow \lceil\mathcal{G}\rceil$

$\Rightarrow$ keep only transitions in $\bigcup\{\text{tr}(\text{local-paths}(a_i \rightarrow^* a_j)) \mid a_i \rightarrow^* a_j \in \lceil\mathcal{G}\rceil\}$



Goal-oriented reduction

Theorem

Given an AN $\mathcal{A} = (\Sigma, S, T)$, a global state $s \in S$, and one automaton local state a_i, for all minimal trace π from s to a_i, $\mathrm{tr}(\pi) \subset \mathrm{tr}(\lceil\mathcal{G}\rceil)$.

Consequence

The AN $\mathcal{A} = (\Sigma, S, \mathrm{tr}(\lceil\mathcal{G}\rceil))$ conserves all minimal traces for reaching a_i from s.

Goal oriented-reduction

Theorem

Given an AN $\mathcal{A} = (\Sigma, S, T)$, a global state $s \in S$, and one automaton local state a_i, for all minimal trace π from s to a_i, $\text{tr}(\pi) \subset \text{tr}(\lceil \mathcal{G} \rceil)$.

Sketch of proof

We proceed by contradiction.
Let us assume that π is a minimal trace, and $\exists t \in \pi$ such that $t \notin \text{tr}(\lceil \mathcal{G} \rceil)$.
We prove we can build a sub-trace of π that does not contain t but still reaches a_i.
Hence π is not minimal.

Sketch of proof

Let us focus on the last unknown transition π^l of π.

$$\pi : b_0 \xrightarrow{\{c_0\}} b_1, \ldots, d_i \xrightarrow{\cdots} d_j, e_i \xrightarrow{\cdots} e_j, \ldots, a_j \xrightarrow{\{b_i, c_j\}} a_i$$

Remember that local-paths($p_i \rightarrow^* p_j$) returns all *acyclic* sequences of trs between p_i and p_j.

If π^l is the last transition of π $(\pi^l = a_j \rightarrow a_i)$

Lemma: $a_j \rightarrow a_i \notin \text{tr}(a_0 \rightarrow^* a_i) \Rightarrow$ any trace reaching a_j goes first to a_i.

By definition $a_0 \rightarrow^* a_i \in \lceil \mathcal{G} \rceil$.
$\pi^l \notin \text{tr}(\lceil \mathcal{G} \rceil) \Rightarrow \pi^l \notin \text{tr}(\text{local-paths}(a_0 \rightarrow^* a_i)) \Rightarrow \exists m < l : \pi^m = \star \rightarrow a_i$,
therefore π **is not minimal**.

Sketch of proof

Let us focus on the last unknown transition π^l of π.

$$\pi : b_0 \xrightarrow{\{c_0\}} b_1, \ldots, d_i \xrightarrow{\cdots} d_j, e_i \xrightarrow{\cdots} e_j, \ldots, a_j \xrightarrow{\{b_i, c_j\}} a_i$$

Remember that local-paths($p_i \rightarrow^* p_j$) returns all *acyclic* sequences of trs between p_i and p_j.

If π^l is the last transition of π ($\pi^l = a_j \rightarrow a_i$)

Lemma: $a_j \rightarrow a_i \notin \text{tr}(a_0 \rightarrow^* a_i) \Rightarrow$ any trace reaching a_j goes first to a_i.

By definition $a_0 \rightarrow^* a_i \in \lceil \mathcal{G} \rceil$.
$\pi^l \notin \text{tr}(\lceil \mathcal{G} \rceil) \Rightarrow \pi^l \notin \text{tr}(\text{local-paths}(a_0 \rightarrow^* a_i)) \Rightarrow \exists m < l : \pi^m = \star \rightarrow a_i$,
therefore π **is not minimal**.

If $\pi^l = d_i \rightarrow d_j$ and $\nexists d_r \in \pi^{l+1..}$

$\pi^{1..l-1,l+1,..}$ is a valid trace which reaches a_i. Hence, π **is not minimal**.

Sketch of proof

Let us focus on the last unknown transition π^l of π.

$$\pi : b_0 \xrightarrow{\{c_0\}} b_1, \ldots, d_i \xrightarrow{\cdots} d_j, e_i \xrightarrow{\cdots} e_j, \ldots, a_j \xrightarrow{\{b_i, c_j\}} a_i$$

Remember that local-paths($p_i \rightarrow^* p_j$) returns all *acyclic* sequences of trs between p_i and p_j.

If $\pi^l = d_i \rightarrow d_j$ **and** $\exists d_r \in \pi^{l+1..}$

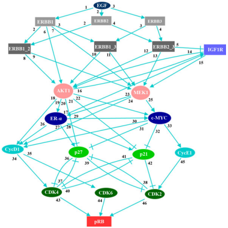
Basically,

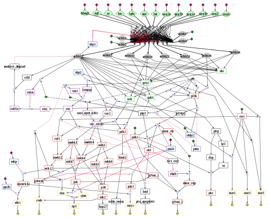
- $d_0 \rightarrow^* d_r \in \lceil \mathcal{G} \rceil$;
- $d_i \rightarrow d_j \notin$ tr(local-paths($d_0 \rightarrow^* d_r$)) implies that $d_i \rightarrow d_j$ is part of a cycle
- we can prove that it is always part of a cycle that can be removed from π (be careful with intertwined cycles)

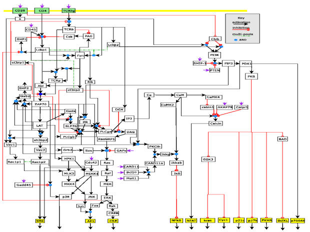
π **is not minimal**.

Results

Preliminary benchmarks with single reachability







Model		# tr	NuSMV time	NuSMV mem	ITS time	ITS mem	# states
Egf-r (20)	normal	68	0.1s	15Mb	0.35s	19Mb	4.200
	reduced	43	0.03s	11Mb	0.13s	8Mb	722
Egf-r (104) profile 1	normal	378	75s	2.1Gb	0.8s	750Mb	$\approx 10^7$
	reduced	0	-	-	-	-	1
Egf-r (104) profile 2	normal	378	KO	KO	540s	1.5Gb	$> 8.10^{14}$
	reduced	211	52s	100Mb	3.4s	100Mb	$\approx 6.10^7$
TCell-r (94)	normal	217	KO	KO	KO	KO	?
	reduced	42	10s	190Mb	0.25s	15Mb	60.000

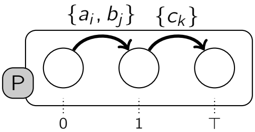
For all cases, reduction step took between 0.01 and 0.1s.

Dealing with sequential reachability properties

We can support more complex reachability properties such as

$$\text{reach } \langle a_i, b_j \rangle \text{ and then } \langle c_k \rangle$$

Classical trick:



…and do reduction for the goal $P_\top$.

Conclusion

Input	- Any automata network (works even with synchronous transitions); - Initial state + seq. reachability prop. (reach a_i, then b_j, etc.)
Output	Automata network with removed local transitions
Complexity	Polynomial in the number of automata, exponential in their size.

Results

- All minimal paths satisfying reachability property are conserved.
- Can lead to drastic reduction of the state space.
- Almost costless. . .

Future work

- Efficient update of the reduction after a transition
- Detect when it is worth to redo a reduction from a new initial state
- Goal-driven unfolding of Petri nets (w/ LSV)
- Combine with model refinement (goal-driven model identification)

SASB'15

6th International Workshop on Static Analysis and Systems Biology

8 September 2015 - Saint-Malo (France)

https://www.lri.fr/sasb2015/

Scope:

- Quantitative and qualitative models
- Topology vs dynamics
- Model reduction
- Abstract interpreration frameworks
- Practical methods for tackling biological models..

Program Co-Chairs

- Loïc Paulevé, CNRS/LRI, Univ. Paris-Sud, France
- Nathalie Théret, INSERM, Rennes, France

Program Committee

- Reka Albert, Pennsylvania State University, USA
- Jérome Feret, Inria/École Normale Sup., France
- Giuditta Franco, University of Verona, Italy
- Johnatan Hayman, University of Cambridge, UK
- Thomas Hinze, University of Jena, Germany
- Cédric Lhoussaine, Université de Lille, France
- Gethin Norman, University of Glasgow, UK
- Tatjana Petrov, IST Austria
- David Safranek, Masaryk Univ., Czech Republic
- Thomas Sauter, University of Luxembourg
- Sylvain Sené, Aix-Marseille Université, France
- Andrei Zinovyev, Institut Curie, France
- Paolo Zuliani, Newcastle University, UK

Papers due on 5th June (ENTCS); Short presentation abstracts due on 23rd June.