Under-approximating Cut Sets for Reachability in Large-Scale Automata Networks

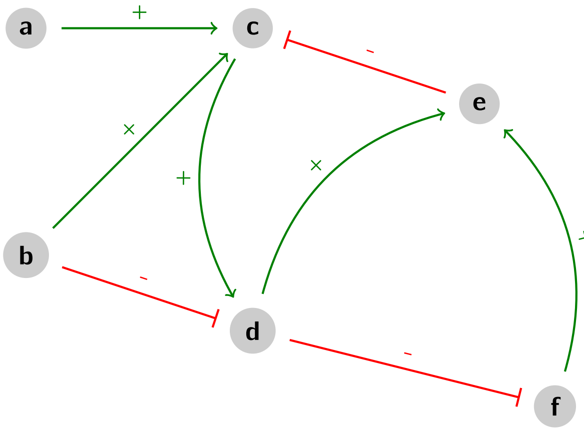
Loïc Paulevé[1], Geoffroy Andrieux[2], Heinz Koeppl[1,3]

[1]ETH Zürich [2]IRISA Rennes, France [3]IBM Research Zürich

25th International Conference on Computer Aided Verification
July 13–19, 2013 - Saint Petersburg, Russia

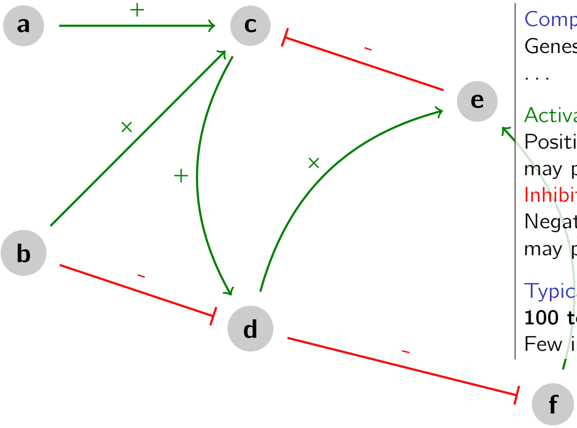
Biological Networks

E.g., Signalling Networks



Biological Networks

E.g., Signalling Networks



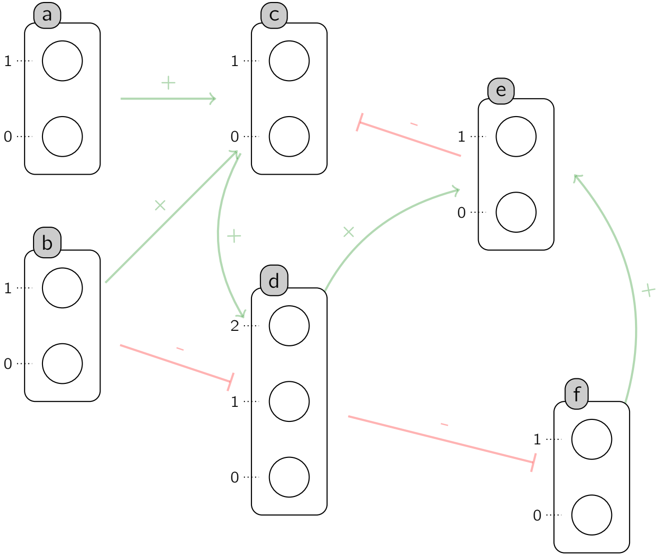
Components
Genes, proteins, complexes, ...

Activations
Positive influence (*a* increase may promote *c* increase).

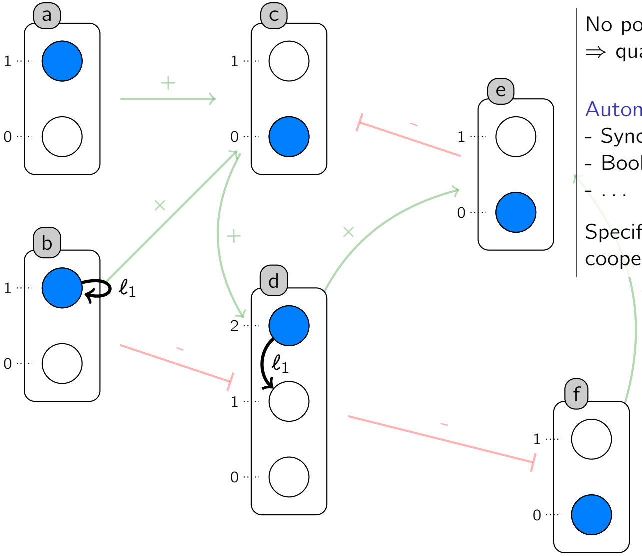
Inhibitions
Negative influence (*b* increase may promote *d* decrease).

Typical settings
100 to +10,000 components
Few information on kinetics

Qualitative Models for Biological Networks



Qualitative Models for Biological Networks

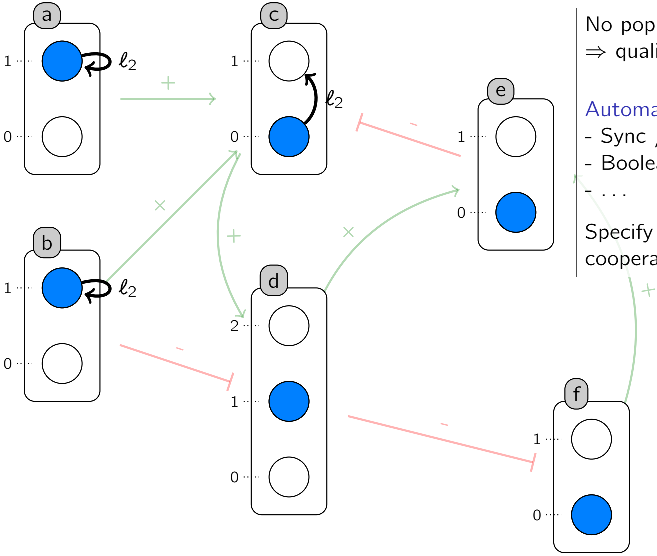


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- ...

Specify partial or complete cooperations.

Qualitative Models for Biological Networks

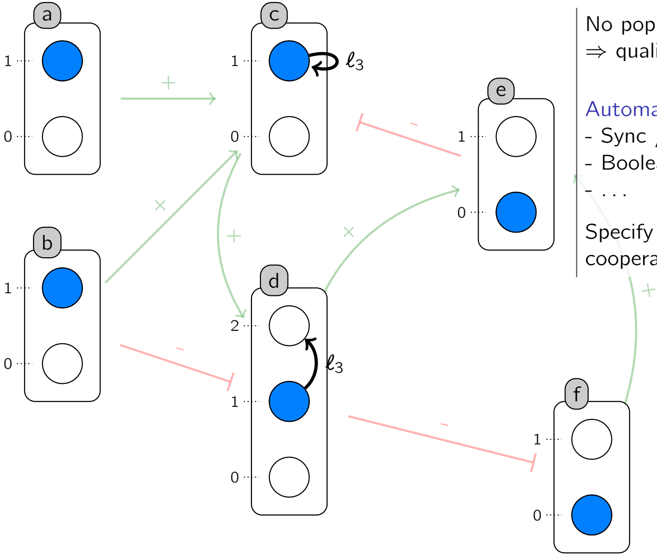


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- ...

Specify partial or complete cooperations.

Qualitative Models for Biological Networks

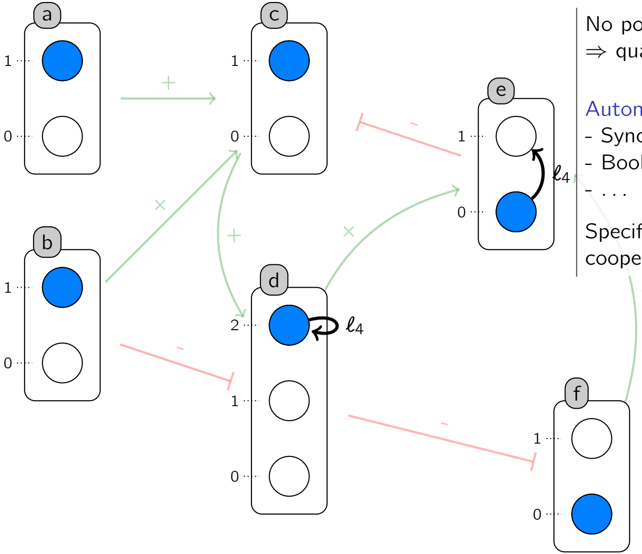


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- . . .

Specify partial or complete cooperations.

Qualitative Models for Biological Networks

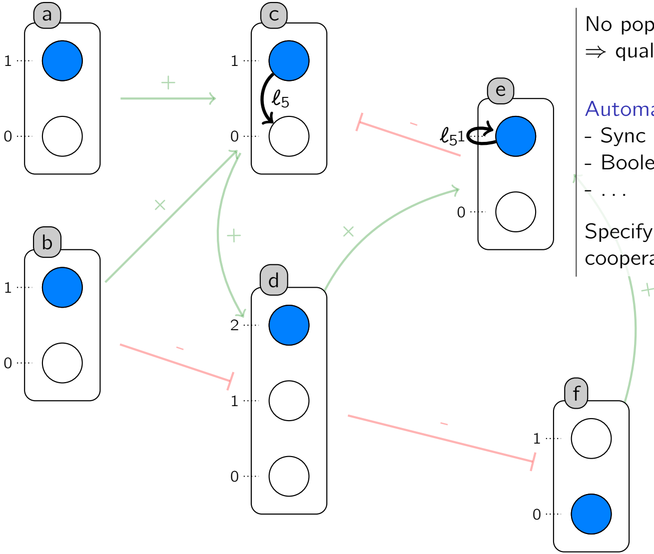


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- . . .

Specify partial or complete cooperations.

Qualitative Models for Biological Networks

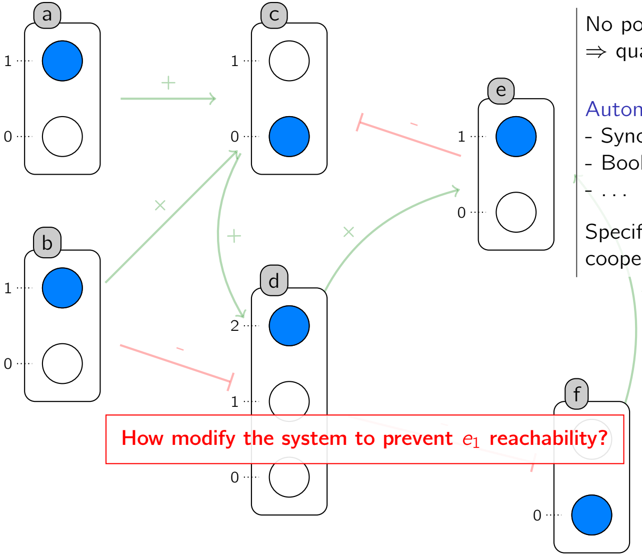


No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- . . .

Specify partial or complete cooperations.

Qualitative Models for Biological Networks



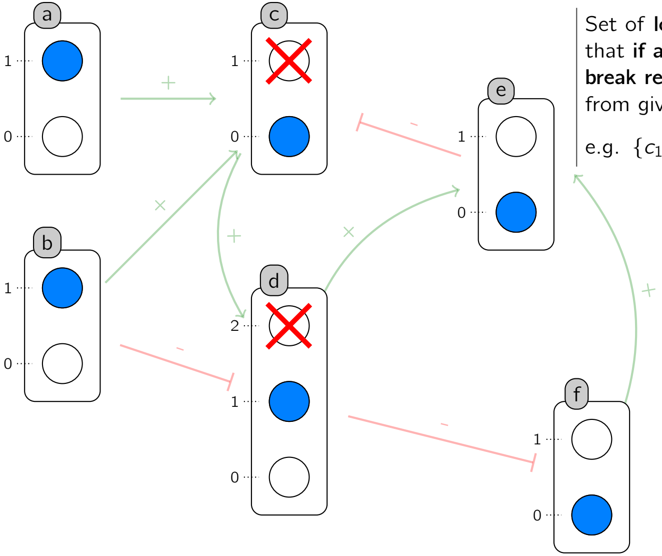
No population
⇒ qualitative levels

Automata networks
- Sync / async
- Boolean/multi-valued
- . . .

Specify partial or complete cooperations.

How modify the system to prevent e_1 reachability?

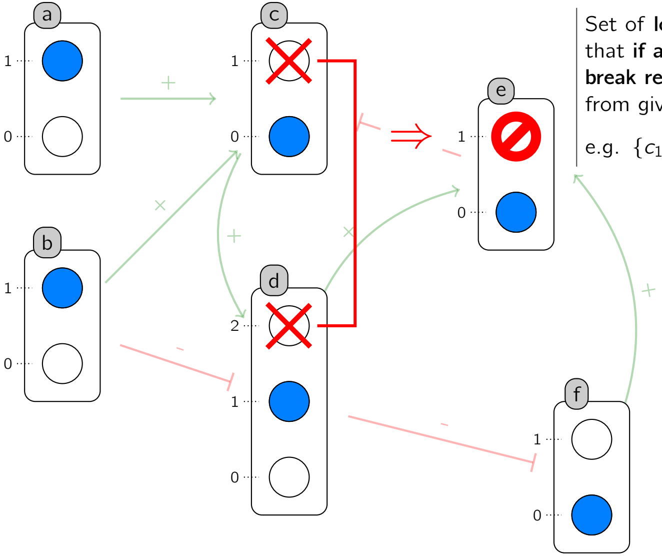
Cut Sets for Reachability



Set of **local states** that **if all disabled break reachability** from given initial states

e.g. $\{c_1, d_2\}$

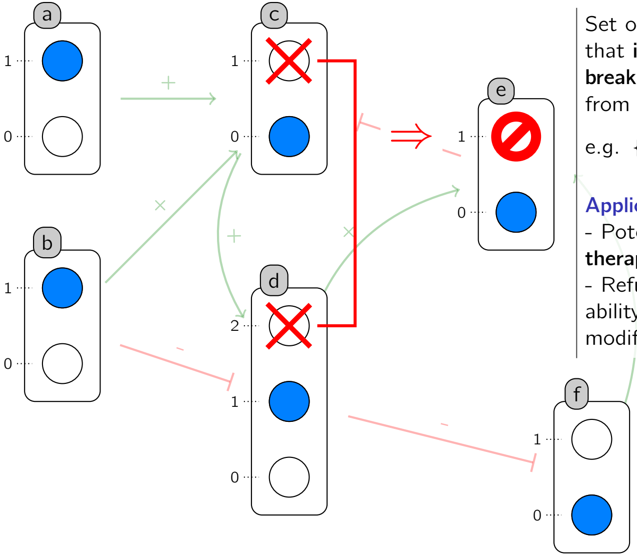
Cut Sets for Reachability



Set of **local states** that **if all disabled break reachability** from given initial states

e.g. $\{c_1, d_2\}$

Cut Sets for Reachability



Set of **local states** that **if all disabled break reachability** from given initial states

e.g. $\{c_1, d_2\}$

Applications

- Potential **therapeutic targets**
- Refute model if reachability still occurs in the modified (real) system

Challenges

Naive algorithm – $\mathcal{M}$: automata network; ς: set of initial states
$\mathrm{CutSets} \leftarrow \emptyset$
For $\omega \in \wp$(**Local States**) ordered by cardinality:
  if $(\nexists \omega' \in \mathrm{CutSets} : \omega' \subset \omega)$ and $\forall s \in \varsigma, (\mathcal{M} \ominus \omega, s) \not\models \mathrm{EF}\ z_i$:
    $\mathrm{CutSets} \leftarrow \mathrm{CutSets} \cup \{\omega\}$.

Challenges

Naive algorithm – $\mathcal{M}$: automata network; ς: set of initial states
$\mathrm{CutSets} \leftarrow \emptyset$
For $\omega \in \wp$(**Local States**) ordered by cardinality:
if $(\nexists \omega' \in \mathrm{CutSets} : \omega' \subset \omega)$ and $\forall s \in \varsigma, (\mathcal{M} \ominus \omega, s) \nvDash \mathrm{EF}\ z_i$:
$\mathrm{CutSets} \leftarrow \mathrm{CutSets} \cup \{\omega\}$.

Goal: **scalable** with biological networks complexity

- numerous automata, all different;
- few local states per automaton.

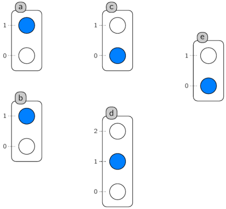
Contribution: **under-approximation of cut sets**

- some will be missed, some will be too thick (non-minimal, for the model);
- handle **networks with more than 9,000 nodes**.

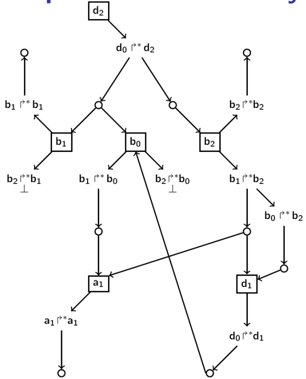
No candidate enumeration, no model-checking.

Overview

Automata network



Graph of Local Causality



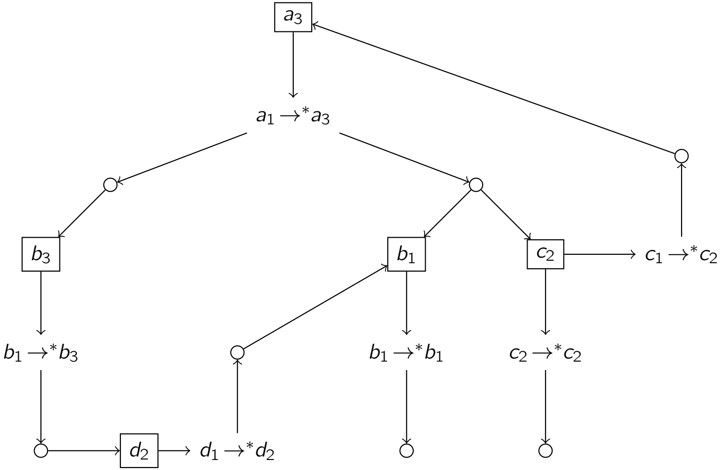
Under-approximation of Cut Sets

Prior work: over-/under-approximation of reachability in large-scale biological networks.

[Paulevé et al. in Math. Struct. in Comp. Sci. 2012]

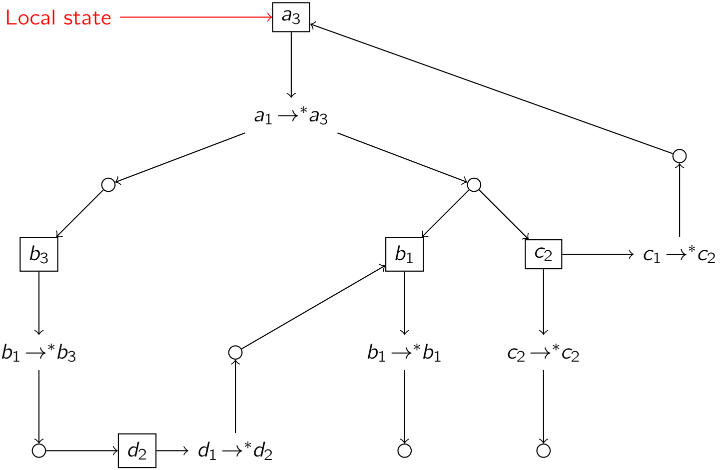
Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1, 2\}; d \mapsto \{2\}\}$.



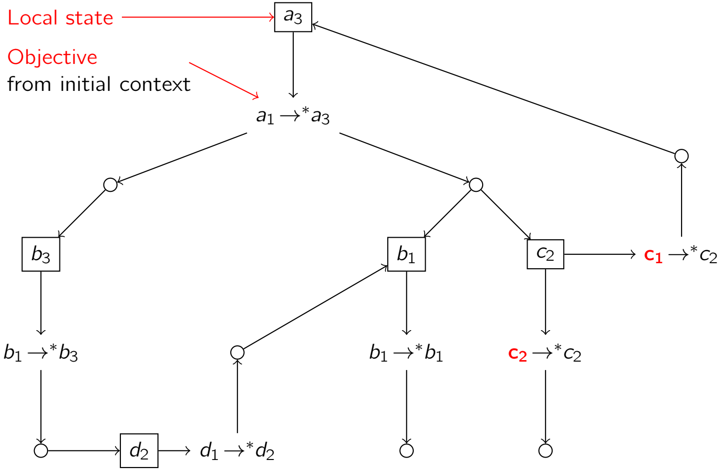
Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1, 2\}; d \mapsto \{2\}\}$.



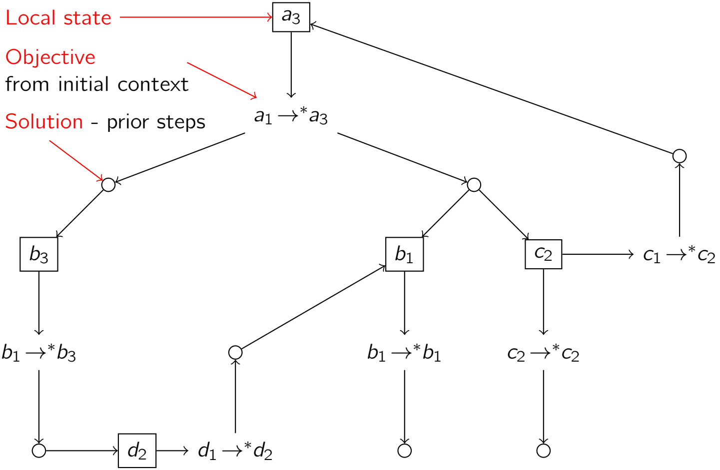
Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{\mathbf{1}, \mathbf{2}\}; d \mapsto \{2\}\}$.



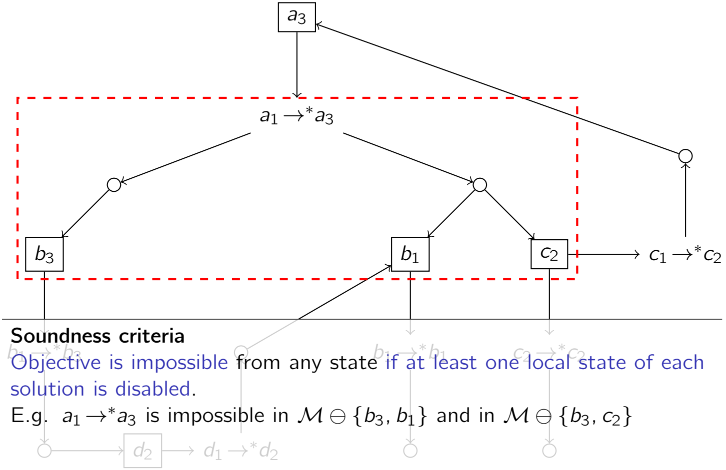
Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1,2\}; d \mapsto \{2\}\}$.



Graph of Local Causality

- Causality of a_3.
- Initial context $\varsigma = \{a \mapsto \{1\}; b \mapsto \{1\}; c \mapsto \{1,2\}; d \mapsto \{2\}\}$.

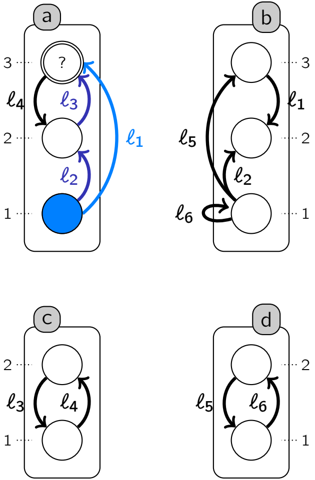


Soundness criteria

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

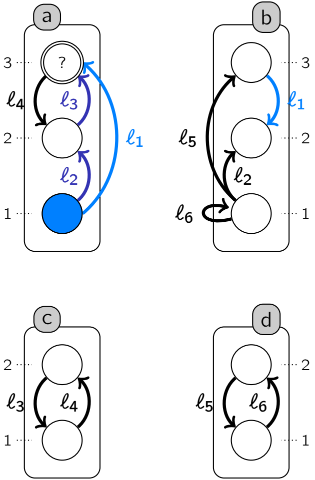
E.g. $a_1 \rightarrow^* a_3$ is impossible in $\mathcal{M} \ominus \{b_3, b_1\}$ and in $\mathcal{M} \ominus \{b_3, c_2\}$

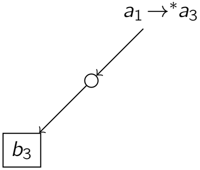
Computing GLC for Automata Networks



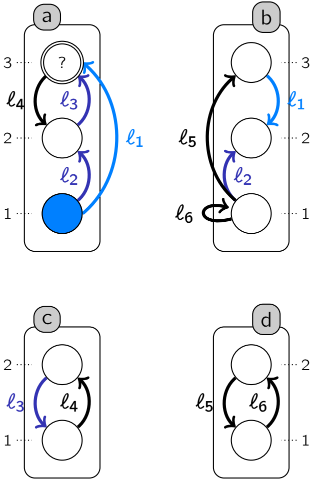
$a_1 \rightarrow^* a_3$

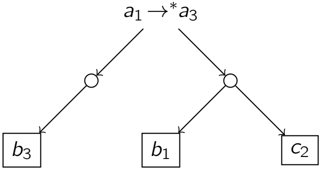
Computing GLC for Automata Networks





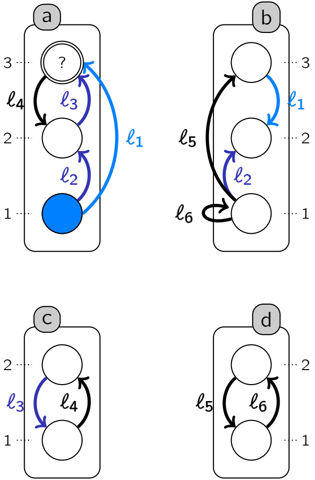
Computing GLC for Automata Networks

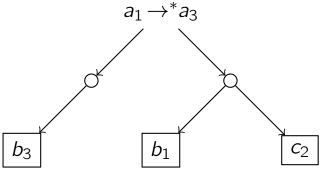




(ignore order, count, synchronism)

Computing GLC for Automata Networks





(ignore order, count, synchronism)

Complexity (construction + size of GLC)

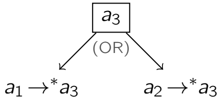
- polynomial in the total number of local states;
- exponential in the number of local states within one automaton

⇒ efficient with a small number of local states per automaton, whereas a very large number of automata can be handled.

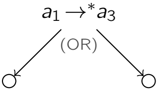
Cut Sets Under-Approximation

Associate to each node sets of local states intersecting *any* trace from given context.

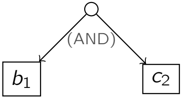
$$\mathbb{V} : \text{nodes} \mapsto \wp(\wp^{\leq N}(\mathcal{O}bs)), \mathcal{O}bs \subset \textbf{LS}$$



$$\mathbb{V}(a_3) = \mathbb{V}(a_1 \to^* a_3) \tilde{\times} \mathbb{V}(a_2 \to^* a_3) \cup \{\{a_3\}\}$$



$$\mathbb{V}(a_1 \to^* a_3) = \mathbb{V}(sol^1) \tilde{\times} (sol^2)$$



$$\mathbb{V}(sol^1) = \mathbb{V}(b_1) \cup \mathbb{V}(c_2)$$

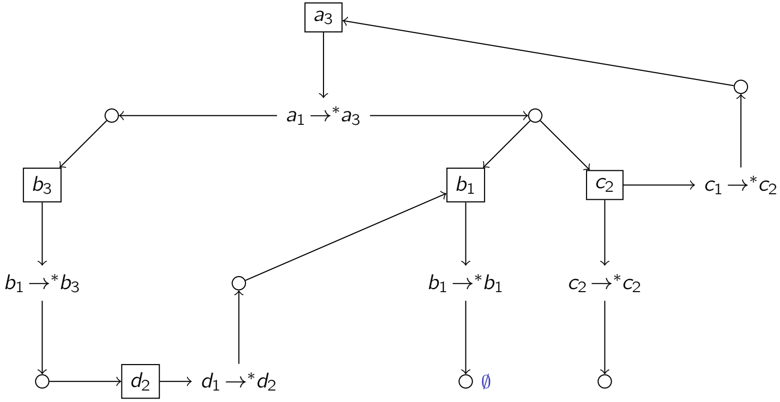
$\{e^1, \dots, e^n\} \tilde{\times} \{f^1, \dots, f^m\} \stackrel{\Delta}{=} \{e^i \cup f^j \mid i \in [1;n] \wedge j \in [1;m]\}$; $e^i, f^j \in \wp^{\leq N}(\mathcal{O}bs)$

Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

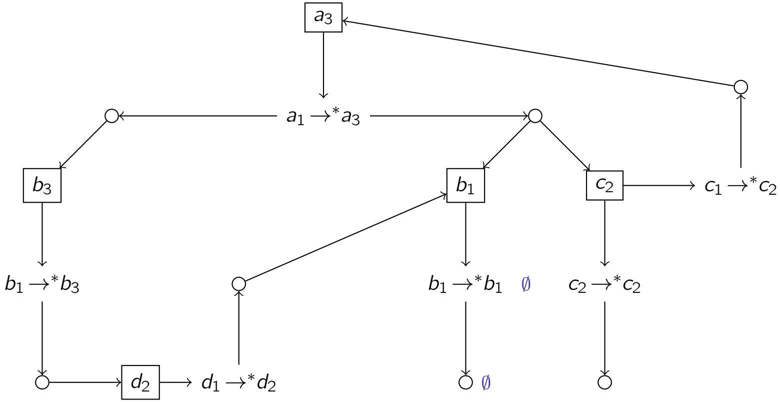


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

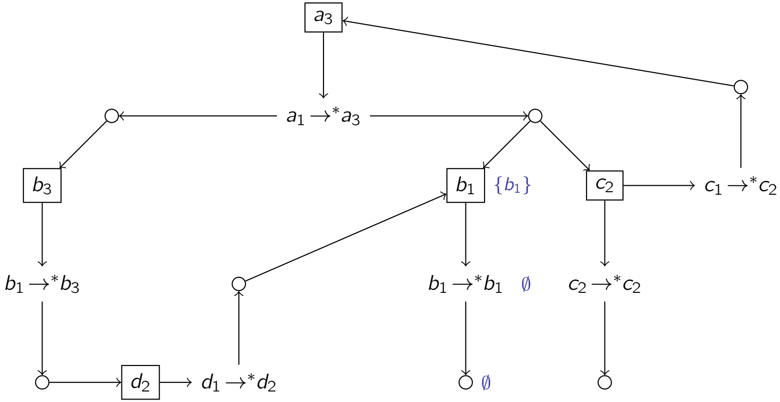


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

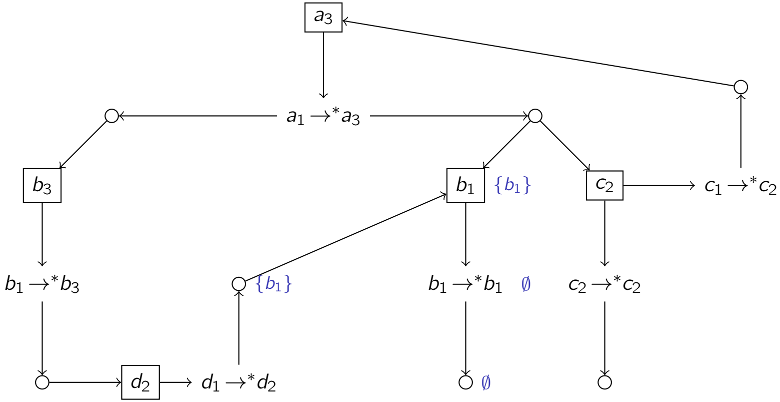


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

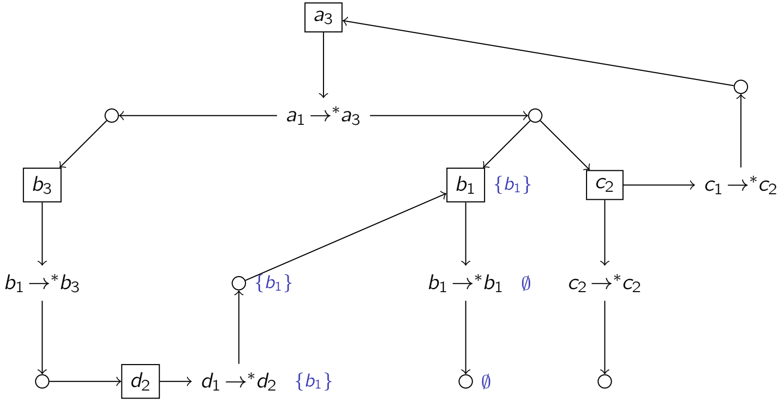


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

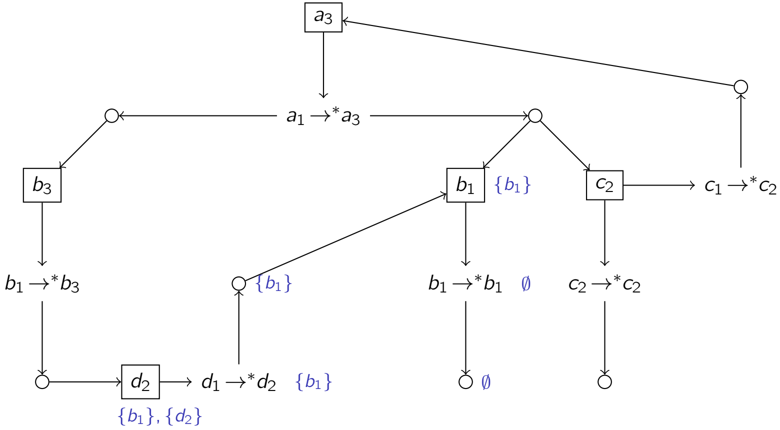


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

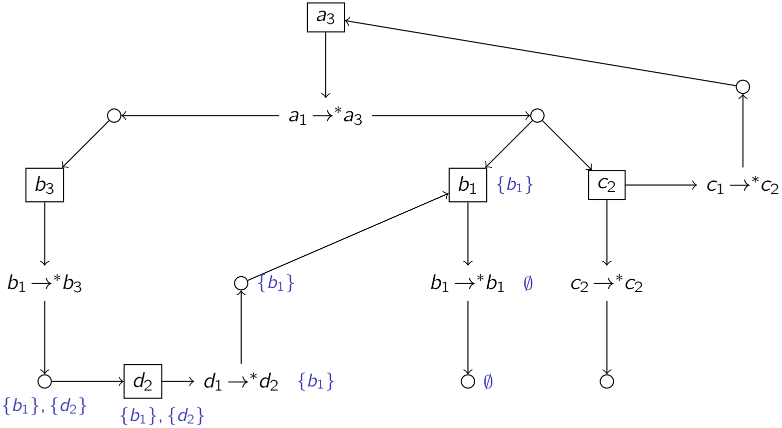


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

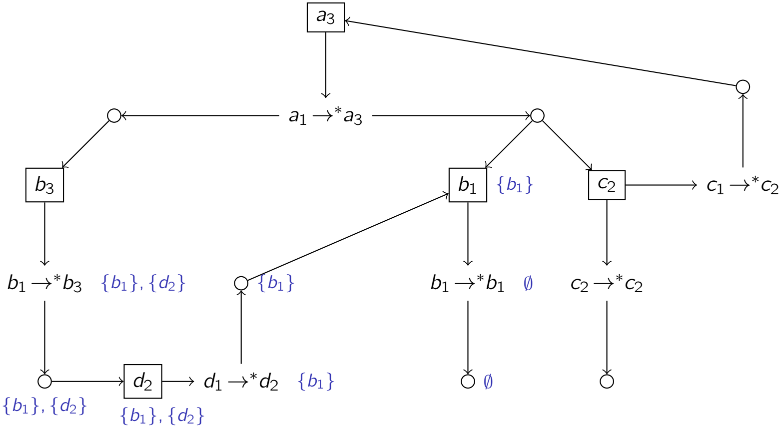


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

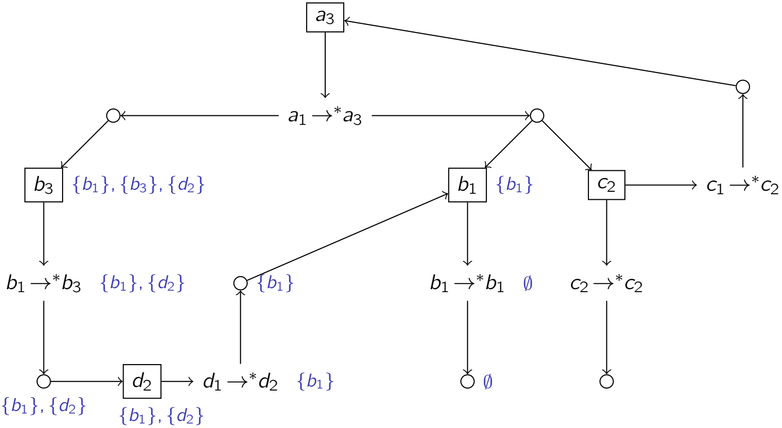


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

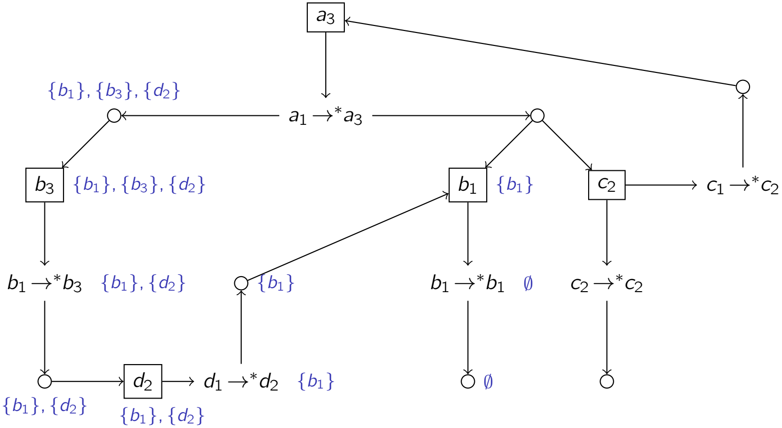


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

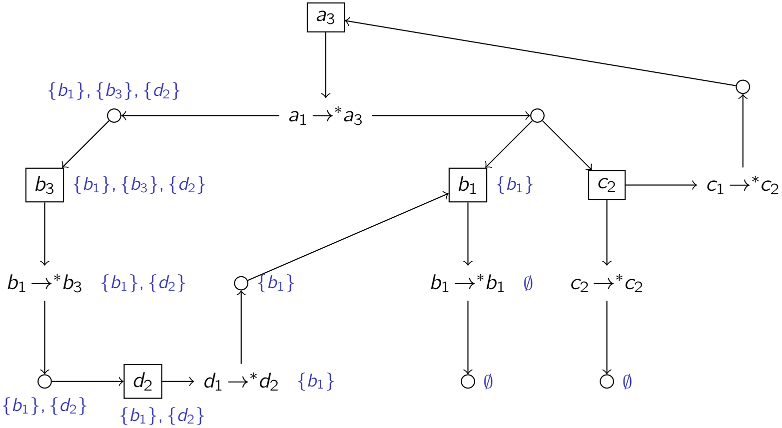


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

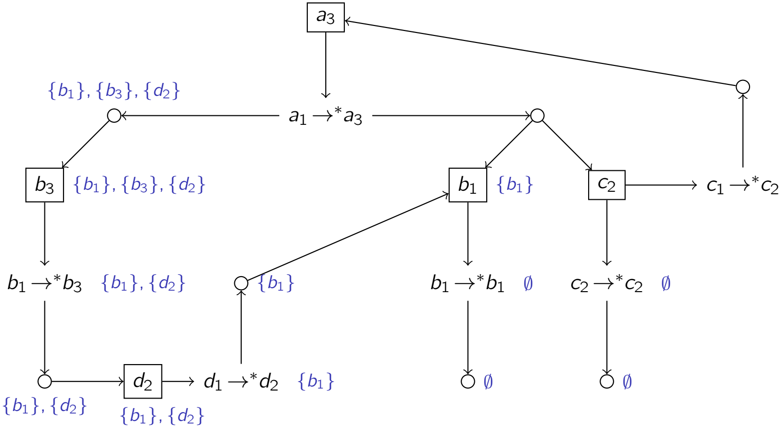


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

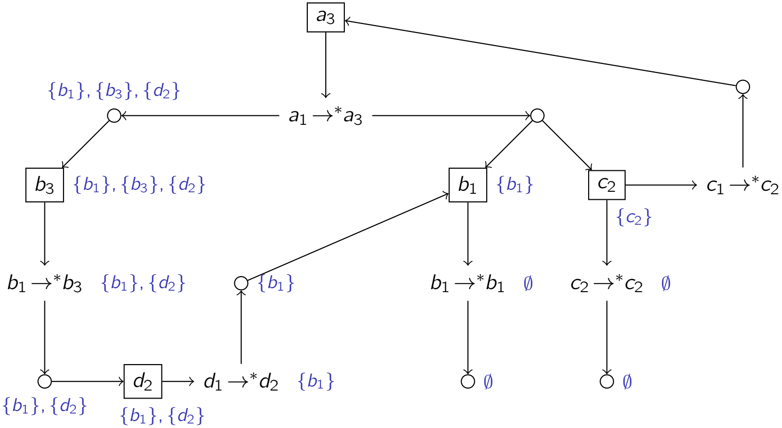


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

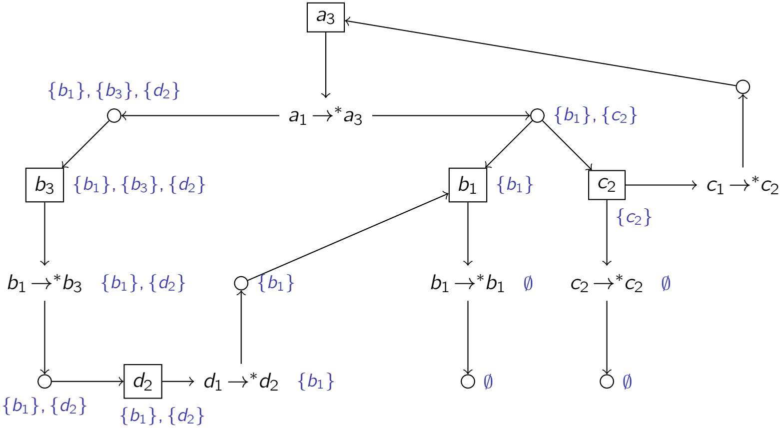


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

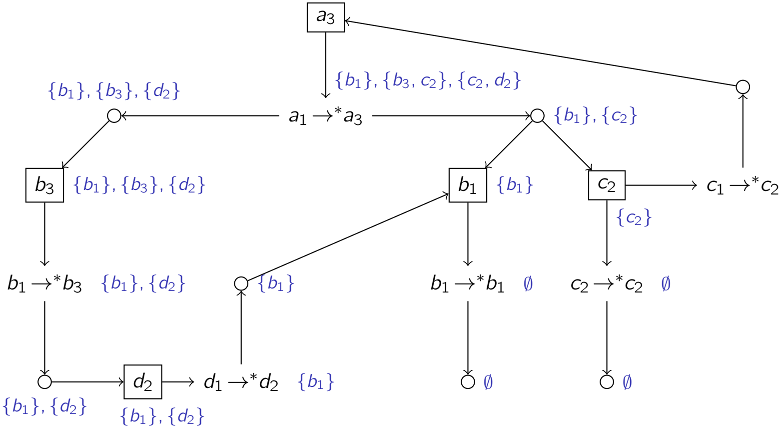


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

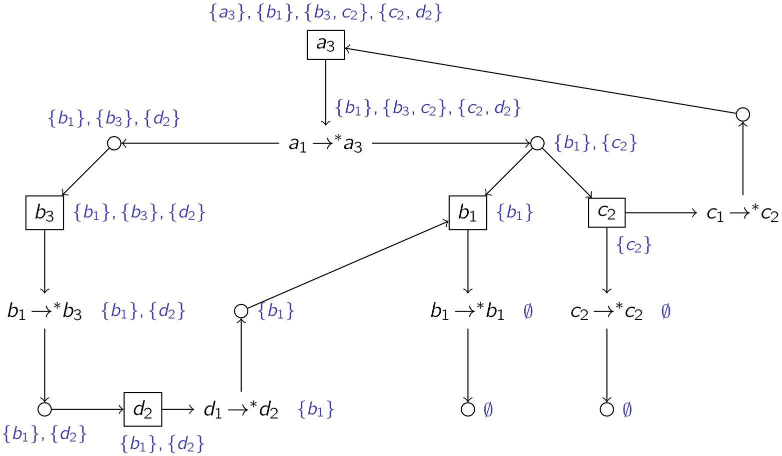


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

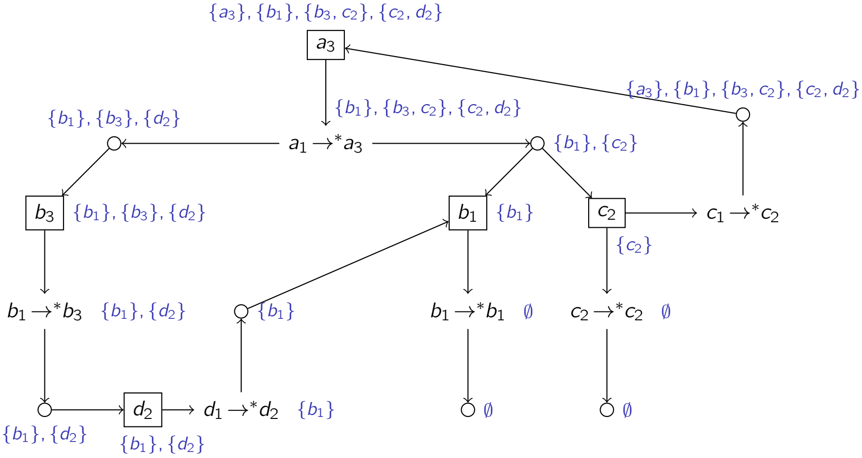


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.

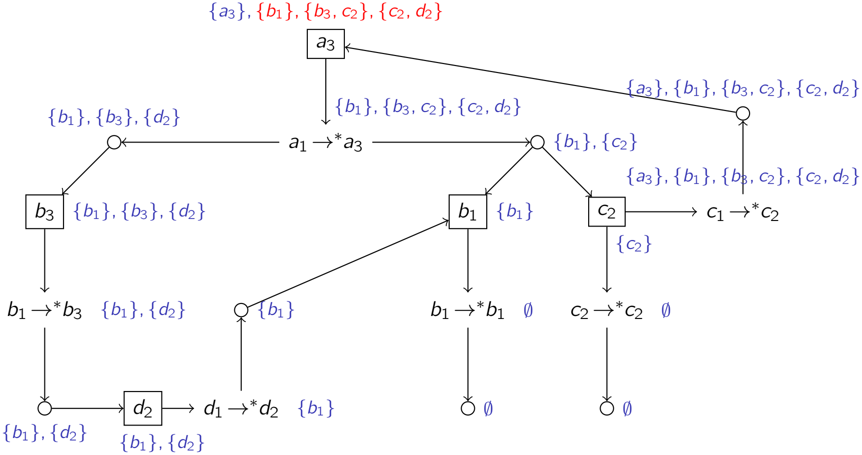


Cut Sets Under-approximation

Example

Sketch

- Follow the topological order of GLC.
- SCCs: arbitrary/random order for updating nodes having child modified.
- Always converges.



Formal analysis of the whole PID

Pathway Interaction Database `http://pid.nci.nih.gov`

- Inductions, inhibitions, transcriptional regulation, complex formations, . . .
- +9,000 interacting components.

Graph of Local Causality for (independent) reachability of active SNAIL, p15, p21

- From Process Hitting model (sub-class of Asynchronous ANs)
  +21,000 concurrent automata (biological and logical); largest: 16 local states.
- ≈20,000 nodes involving ≈1,600 biological components.

Extracted Cut Sets

N	Visited nodes	Exec. time	$SNAIL_1$	$p15INK4b_1$	$p21CIP1_1$
1	29,022	0.9s	1	1	1
2	36,602	1.6s	+6	+6	+0
3	44,174	5.4s	+0	+92	+0
4	54,322	39s	+30	+60	+0
5	68,214	8.3m	+90	+80	+0
6	90,902	2.6h	+930	+208	+0

Implemented in PINT `http://process.hitting.free.fr` (OCaml);
Dedicated data structures to efficiently compute cross products between million of sets.

Discussion

Summary

- Cut sets for transient reachability from a set of initial states
  $\Rightarrow$ sets of local states necessary for reachability.
- Tractable on very large-scale biological networks.

Quality of under-approximation

- Graph of Local Causality abstracts a lot of details around synchronisations.
- The less sync the AN, the more accurate the cut sets.
- Suited for qualitative biological networks.

Future work

- Take into account the time scales of interactions.
- Cut sets that do not break other dynamical properties.
- Cut sets for other dynamical properties.

Thank you for your attention.