

Abstractions for Capturing Dynamics of Very Large Biological Networks

JFLI-LRI-NII - 10th October 2013

Loïc Paulevé

CNRS / LRI

`loic.pauleve@lri.fr`

`http://loicpauleve.name`

Formal verification of dynamical properties

- **Validate/refute** a model.
- **Exhaustive** analysis.
- How to handle large-scale networks?

Process Hitting

- **Qualitative modelling** of biological networks **dynamics**.
- **Subclass of automata networks**/Petri nets.
- Supports **partial knowledge** on cooperations between components.

Abstract Interpretation of Dynamics

- **Compact approximate representation** of possible dynamics.
- **Over-approximation** and **under-approximation** of dynamics.
- Application: **reachability** verification, **cut sets** computation.

Outline

- 1 Fixed points
- 2 Reachability
- 3 Cut sets for reachability

Outline

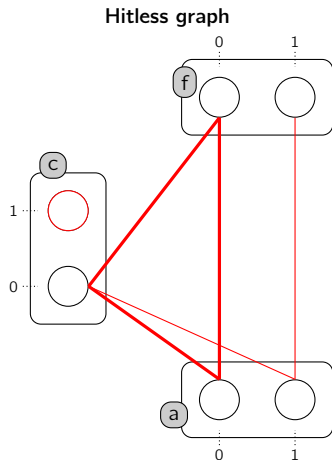
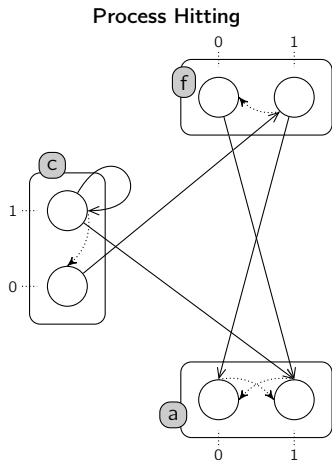
① Fixed points

② Reachability

③ Cut sets for reachability

Fixed Points

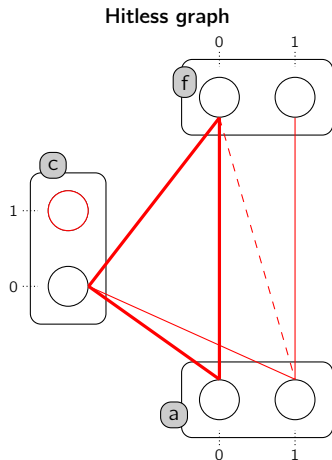
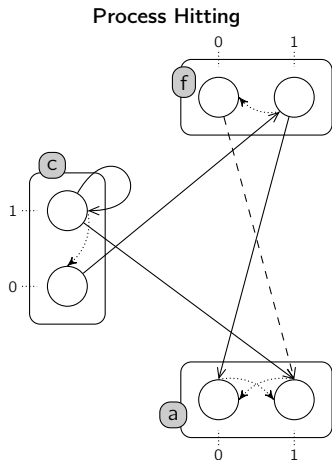
[Paulevé, Magnin, Roux in TCSB 2011]



n -cliques are fixed points

Fixed Points

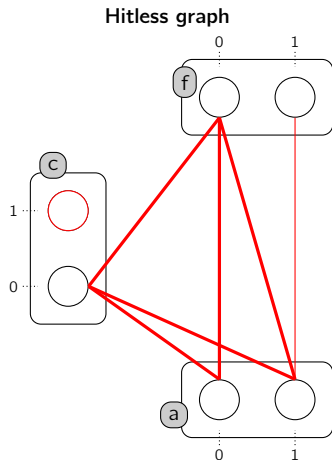
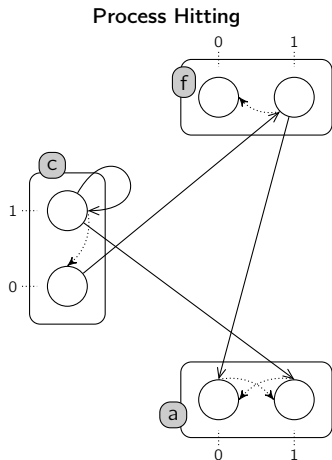
[Paulevé, Magnin, Roux in TCSB 2011]



n -cliques are fixed points

Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]



n -cliques are fixed points

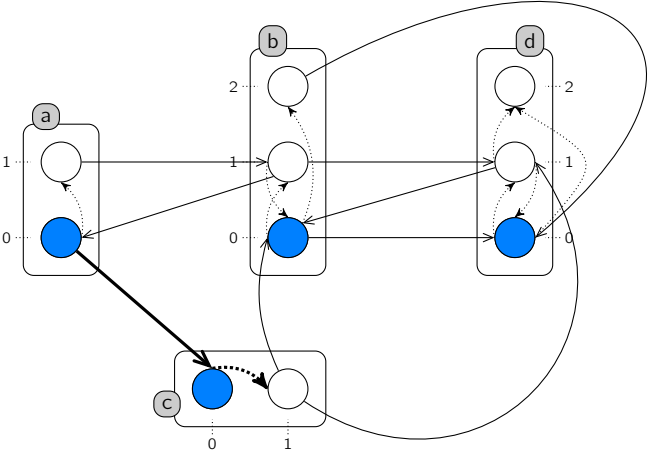
Outline

① Fixed points

② Reachability

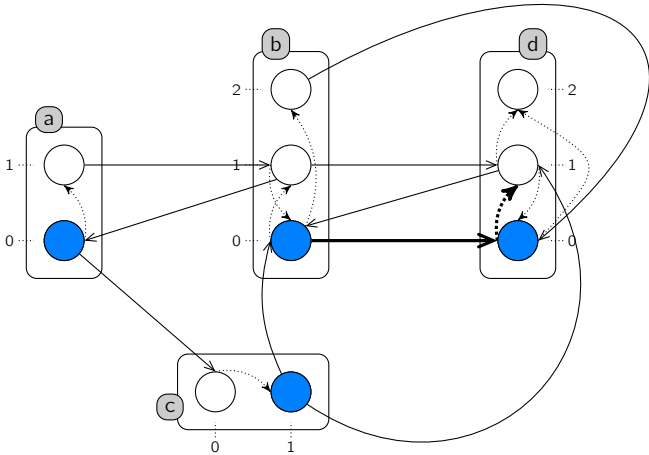
③ Cut sets for reachability

Scenarios



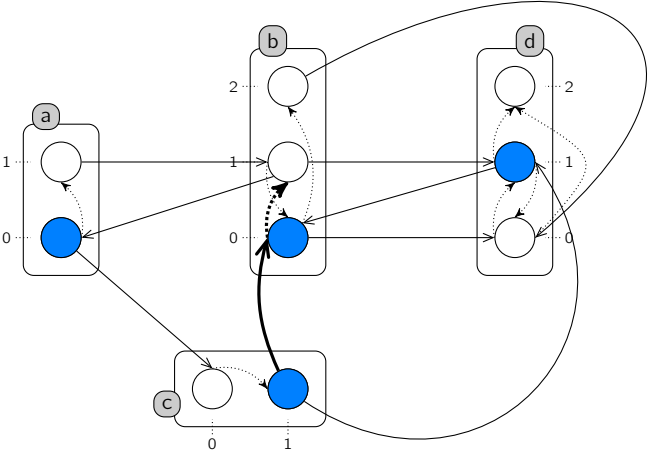
$$a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$$

Scenarios



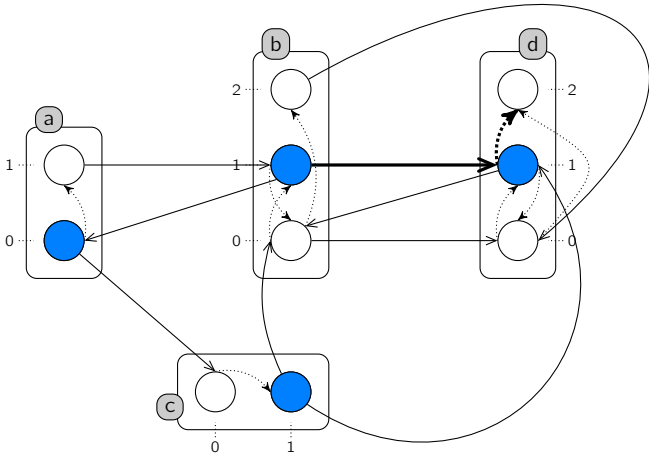
$$a_0 \rightarrow c_0 \uparrow c_1 :: \mathbf{b_0 \rightarrow d_0 \uparrow d_1} :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$$

Scenarios



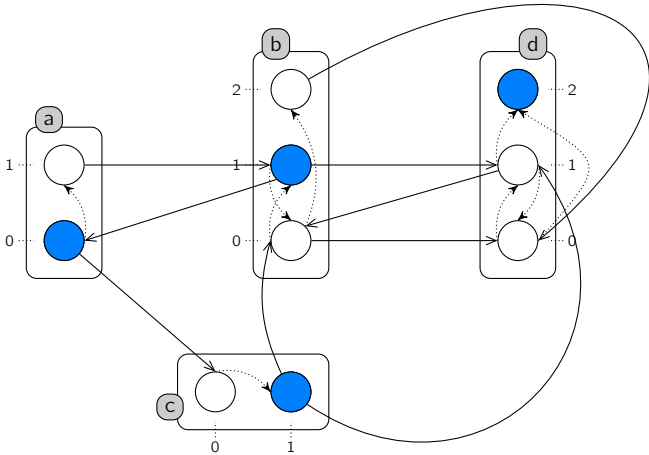
$$a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: \mathbf{c_1 \rightarrow b_0 \uparrow b_1} :: b_1 \rightarrow d_1 \uparrow d_2$$

Scenarios



$$a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: \mathbf{b_1 \rightarrow d_1 \uparrow d_2}$$

Scenarios



$$a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2$$

Static Analysis of Successive Reachability Properties

[Paulevé, Magnin, Roux in MSCS 2012]

Successive Reachability ω

- Given a Process Hitting \mathcal{PH} with an initial state,
- is it possible to reach the process a_i ? ...
- then the process b_j ? ... etc.

Static Analysis of Successive Reachability Properties

[Paulevé, Magnin, Roux in MSCS 2012]

Successive Reachability ω

- Given a Process Hitting \mathcal{PH} with an initial state,
- is it possible to reach the process a_i ? ...
- then the process b_j ? ... etc.

Difficulties: combinatorial explosion of dynamics to explore.

Static Analysis of Successive Reachability Properties

[Paulevé, Magnin, Roux in MSCS 2012]

Successive Reachability ω

- Given a Process Hitting \mathcal{PH} with an initial state,
- is it possible to reach the process a_i ? ...
- then the process b_j ? ... etc.

Difficulties: combinatorial explosion of dynamics to explore.

Chosen approach

Over-approximations

\mathcal{PH} does not satisfy $\mathcal{P} \implies \omega$ is impossible.

Under-approximations

\mathcal{PH} satisfies $\mathcal{Q} \implies \omega$ is possible.

Requirement: checking \mathcal{P} (\mathcal{Q}) is fast.

Two Complementary Abstractions of Scenarios

$$a_0 \rightarrow c_0 \uparrow^* c_1 :: b_0 \rightarrow d_0 \uparrow^* d_1 :: c_1 \rightarrow b_0 \uparrow^* b_1 :: b_1 \rightarrow d_1 \uparrow^* d_2$$

Abstraction by Objective Sequences

- $c_0 \uparrow^* c_1 :: d_0 \uparrow^* d_1 :: b_0 \uparrow^* b_1 :: d_1 \uparrow^* d_2$;

Two Complementary Abstractions of Scenarios

$$a_0 \rightarrow c_0 \uparrow^* c_1 :: b_0 \rightarrow d_0 \uparrow^* d_1 :: c_1 \rightarrow b_0 \uparrow^* b_1 :: b_1 \rightarrow d_1 \uparrow^* d_2$$

Abstraction by Objective Sequences

- $c_0 \uparrow^* c_1 :: d_0 \uparrow^* d_1 :: b_0 \uparrow^* b_1 :: d_1 \uparrow^* d_2$;
- $b_0 \uparrow^* b_1 :: d_0 \uparrow^* d_2$

Two Complementary Abstractions of Scenarios

$$a_0 \rightarrow c_0 \uparrow^* c_1 :: b_0 \rightarrow \mathbf{d}_0 \uparrow^* d_1 :: c_1 \rightarrow b_0 \uparrow^* b_1 :: b_1 \rightarrow d_1 \uparrow^* \mathbf{d}_2$$

Abstraction by Objective Sequences

- $c_0 \uparrow^* c_1 :: d_0 \uparrow^* d_1 :: b_0 \uparrow^* b_1 :: d_1 \uparrow^* d_2$;
- $b_0 \uparrow^* b_1 :: d_0 \uparrow^* d_2$
- $d_0 \uparrow^* d_2, \dots$

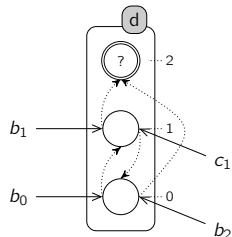
Two Complementary Abstractions of Scenarios

$$a_0 \rightarrow c_0 \uparrow^* c_1 :: \mathbf{b_0 \rightarrow d_0 \uparrow^* d_1} :: c_1 \rightarrow b_0 \uparrow^* b_1 :: \mathbf{b_1 \rightarrow d_1 \uparrow^* d_2}$$

Abstraction by Objective Sequences

- $c_0 \uparrow^* c_1 :: d_0 \uparrow^* d_1 :: b_0 \uparrow^* b_1 :: d_1 \uparrow^* d_2$;
- $b_0 \uparrow^* b_1 :: d_0 \uparrow^* d_2$
- $d_0 \uparrow^* d_2, \dots$

Abstraction by Bounce Sequences



E.g.: $b_0 \rightarrow d_0 \uparrow^* d_1 :: b_1 \rightarrow d_1 \uparrow^* d_2$ ($d_0 \uparrow^* d_2$)

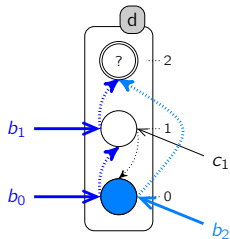
Two Complementary Abstractions of Scenarios

$$a_0 \rightarrow c_0 \uparrow^* c_1 :: b_0 \rightarrow d_0 \uparrow^* d_1 :: c_1 \rightarrow b_0 \uparrow^* b_1 :: b_1 \rightarrow d_1 \uparrow^* d_2$$

Abstraction by Objective Sequences

- $c_0 \uparrow^* c_1 :: d_0 \uparrow^* d_1 :: b_0 \uparrow^* b_1 :: d_1 \uparrow^* d_2$;
- $b_0 \uparrow^* b_1 :: d_0 \uparrow^* d_2$
- $d_0 \uparrow^* d_2, \dots$

Abstraction by Bounce Sequences



E.g.: $b_0 \rightarrow d_0 \uparrow^* d_1 :: b_1 \rightarrow d_1 \uparrow^* d_2$ ($d_0 \uparrow^* d_2$)

\Rightarrow can be computed off-line:

- $\text{sol}(d_0 \uparrow^* d_2) = \{b_0 \rightarrow d_0 \uparrow^* d_1 :: b_1 \rightarrow d_1 \uparrow^* d_2, b_2 \rightarrow d_0 \uparrow^* d_2\}$;
- $\text{sol}^\wedge(d_0 \uparrow^* d_2) = \{\{b_0, b_1\}, \{b_2\}\}$.

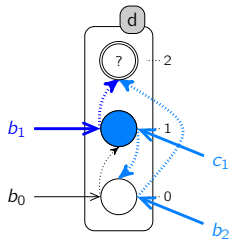
Two Complementary Abstractions of Scenarios

$$a_0 \rightarrow c_0 \uparrow^* c_1 :: b_0 \rightarrow d_0 \uparrow^* d_1 :: c_1 \rightarrow b_0 \uparrow^* b_1 :: b_1 \rightarrow d_1 \uparrow^* d_2$$

Abstraction by Objective Sequences

- $c_0 \uparrow^* c_1 :: d_0 \uparrow^* d_1 :: b_0 \uparrow^* b_1 :: d_1 \uparrow^* d_2$;
- $b_0 \uparrow^* b_1 :: d_0 \uparrow^* d_2$
- $d_0 \uparrow^* d_2, \dots$

Abstraction by Bounce Sequences



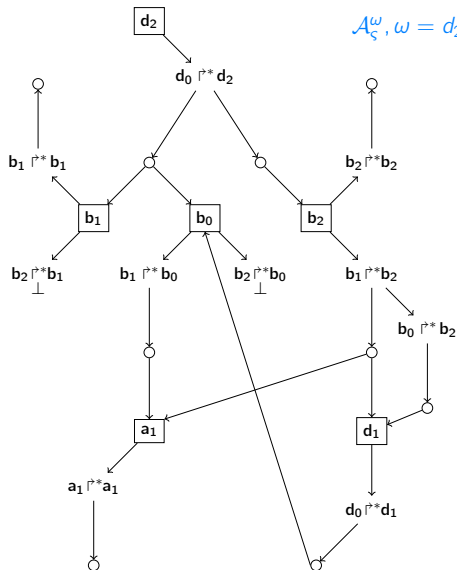
E.g.: $b_0 \rightarrow d_0 \uparrow^* d_1 :: b_1 \rightarrow d_1 \uparrow^* d_2$ ($d_0 \uparrow^* d_2$)

\Rightarrow can be computed off-line:

- $\text{sol}(d_0 \uparrow^* d_2) = \{b_0 \rightarrow d_0 \uparrow^* d_1 :: b_1 \rightarrow d_1 \uparrow^* d_2, b_2 \rightarrow d_0 \uparrow^* d_2\}$;
- $\text{sol}^\wedge(d_0 \uparrow^* d_2) = \{\{b_0, b_1\}, \{b_2\}\}$.
- $\text{sol}(d_1 \uparrow^* d_2) = \{b_1 \rightarrow d_1 \uparrow^* d_2, c_1 \rightarrow d_1 \uparrow^* d_0 :: b_2 \rightarrow d_0 \uparrow^* d_2\}$;
- $\text{sol}^\wedge(d_1 \uparrow^* d_2) = \{\{b_1\}, \{b_2, c_1\}\}$.

Graph of Local Causality

$$\mathcal{A}_s^\omega, \omega = d_2, s = \langle a_1, \{b_1, b_2\}, c_1, d_0 \rangle$$

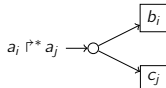

Legend

Requirement



Solution

$$(\{b_i, c_j\} \in \text{sol}^\wedge(a_i \uparrow^* a_j))$$



Continuity

$$a_i \uparrow^* a_j \longrightarrow a_k \uparrow^* a_j$$

Trivial solution

$$a_i \uparrow^* a_j \longrightarrow \bigcirc$$

No solution

$$a_i \uparrow^* a_j \longrightarrow \perp$$

Abstract Interpretation

- ω : successive reachability property (e.g. $a_i :: b_j :: \dots :: z_l$).
- ς : set of initial states.

$$\gamma_\varsigma(\omega) = \{\delta \in \mathbf{Sce} \mid \delta \text{ concretizes } \omega \wedge \delta \text{ starts in } \varsigma\}$$

Objective refinement using sol^\wedge : ρ^\wedge

$\text{Obj} \times \wp(\text{sol}^\wedge)$	$\wp(\text{OS})$
$d_0 \overset{*}{\mapsto} d_2$,	$\star \overset{*}{\mapsto} b_0 :: b_0 \overset{*}{\mapsto} b_1 :: d_0 \overset{*}{\mapsto} d_2,$
$\{\{b_0, b_1\}, \{b_2\}\}$	$\star \overset{*}{\mapsto} b_1 :: b_1 \overset{*}{\mapsto} b_0 :: d_0 \overset{*}{\mapsto} d_2,$ $\star \overset{*}{\mapsto} b_2 :: d_0 \overset{*}{\mapsto} d_2$
$\gamma_\varsigma(d_0 \overset{*}{\mapsto} d_2)$	$= \gamma_\varsigma(\rho^\wedge(d_0 \overset{*}{\mapsto} d_2, \text{sol}^\wedge(d_0 \overset{*}{\mapsto} d_2)))$

Abstract Interpretation

- ω : successive reachability property (e.g. $a_i :: b_j :: \dots :: z_l$).
- ς : set of initial states.

$$\gamma_{\varsigma}(\omega) = \{\delta \in \mathbf{Sce} \mid \delta \text{ concretizes } \omega \wedge \delta \text{ starts in } \varsigma\}$$

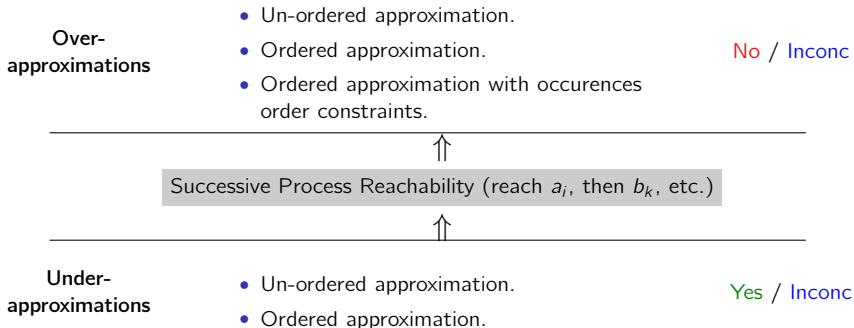
Objective refinement using sol^\wedge : ρ^\wedge

$\text{Obj} \times \rho(\text{sol}^\wedge)$	$\rho(\text{OS})$
$d_0 \overset{*}{\mapsto} d_2$, $\{\{b_0, b_1\}, \{b_2\}\}$	$\star \overset{*}{\mapsto} b_0 :: b_0 \overset{*}{\mapsto} b_1 :: d_0 \overset{*}{\mapsto} d_2,$ $\star \overset{*}{\mapsto} b_1 :: b_1 \overset{*}{\mapsto} b_0 :: d_0 \overset{*}{\mapsto} d_2,$ $\star \overset{*}{\mapsto} b_2 :: d_0 \overset{*}{\mapsto} d_2$
$\gamma_{\varsigma}(d_0 \overset{*}{\mapsto} d_2)$	$= \gamma_{\varsigma}(\rho^\wedge(d_0 \overset{*}{\mapsto} d_2, \text{sol}^\wedge(d_0 \overset{*}{\mapsto} d_2)))$

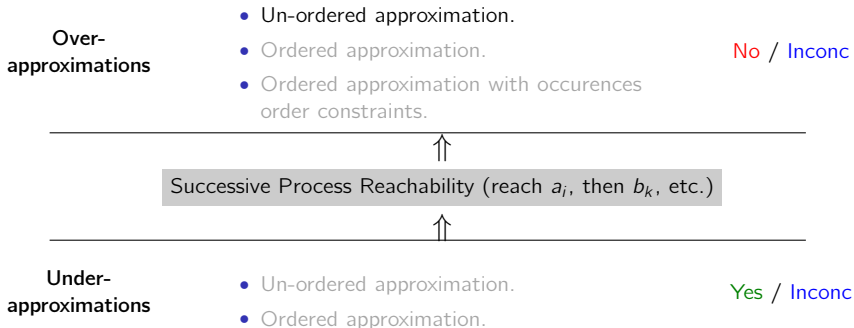
Generalization to OS refinements: $\tilde{\rho}$

$\text{OS} \times \rho(\text{sol}^\wedge)$	$\rho(\text{OS})$
$\omega, \text{sol}^\wedge$	interleave $\begin{pmatrix} \omega' \\ \omega_{1..n-1} \end{pmatrix} :: \omega_{n.. \omega }$ where $n \in \mathbb{I}^\omega$ and $\omega' :: \omega_n \in \rho^\wedge(\omega_n, \text{sol}^\wedge(\omega_n))$
$\gamma_{\varsigma}(\omega)$	$= \gamma_{\varsigma}(\tilde{\rho}(\omega, \text{sol}^\wedge))$

Approximations of Successive Reachability

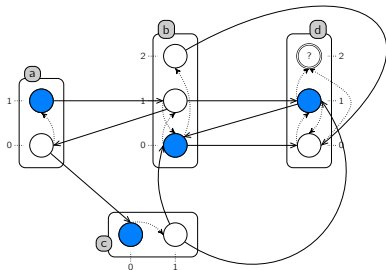


Approximations of Successive Reachability



Over-approximation of Reachability

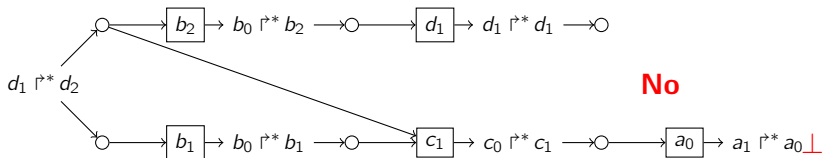
Example



Necessary condition for reaching d_2 :

There exists a traversal of the GLC s.t.:

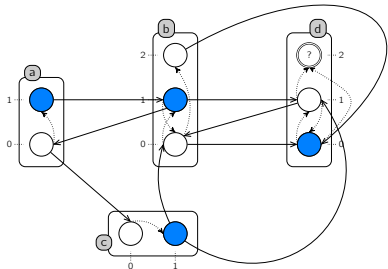
- objective \rightarrow follow at least one solution;
- process \rightarrow follow all objectives;
- no cycle.



No

Over-approximation of Reachability

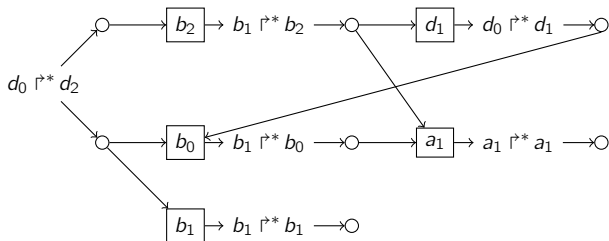
Example



Necessary condition for reaching d_2 :

There exists a traversal of the GLC s.t.:

- objective \rightarrow follow at least one solution;
- process \rightarrow follow all objectives;
- no cycle.



Inconc

Approximations of Successive Reachability

**Over-
approximations**

- Un-ordered approximation.
- Ordered approximation.
- Ordered approximation with occurrences order constraints.

No / Inconc



Successive Process Reachability (reach a_i , then b_k , etc.)



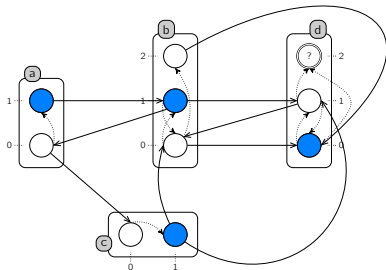
**Under-
approximations**

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

Un-ordered Under-approximation

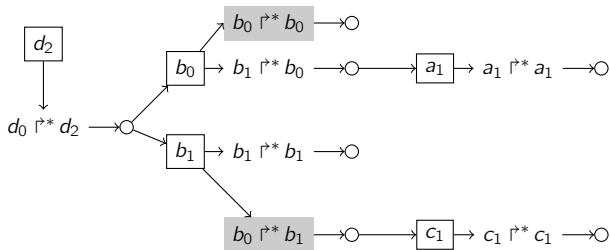
Example



Sufficient condition for reaching d_2 :

- $\lceil \mathcal{B}_\zeta^\omega \rceil$ has no cycle;
- each objective has at least one solution.

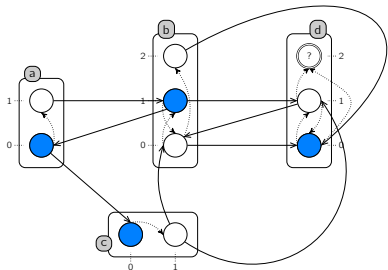
$\lceil \mathcal{B}_\zeta^\omega \rceil$: saturated \mathcal{A}_ζ^ω .



Yes

Un-ordered Under-approximation

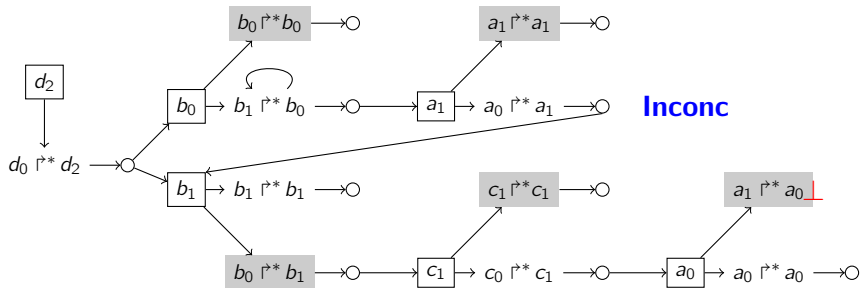
Example



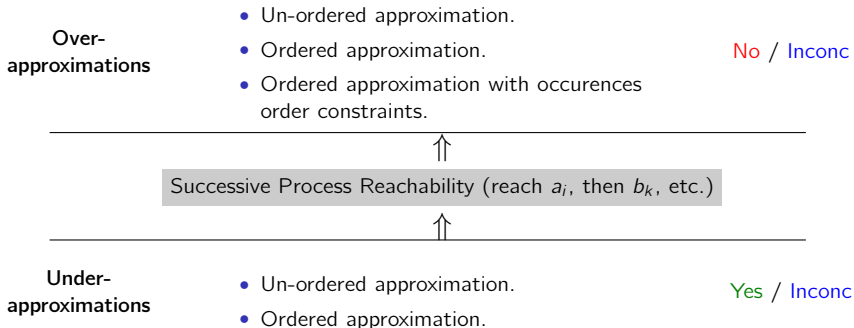
Sufficient condition for reaching d_2 :

- $\lceil \mathcal{B}_\zeta^\omega \rceil$ has no cycle;
- each objective has at least one solution.

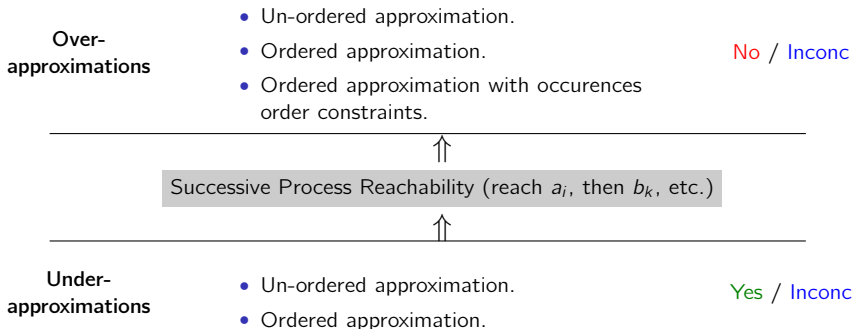
$\lceil \mathcal{B}_\zeta^\omega \rceil$: saturated \mathcal{A}_ζ^ω .



Static Analysis of Successive Reachability



Static Analysis of Successive Reachability

**Still inconclusive?**

- Require new analyses of the abstract structure
- \Rightarrow drive refinements of ω .

Abstract Structures \mathcal{A}_ζ^ω , $[\mathcal{B}_\zeta^\omega]$

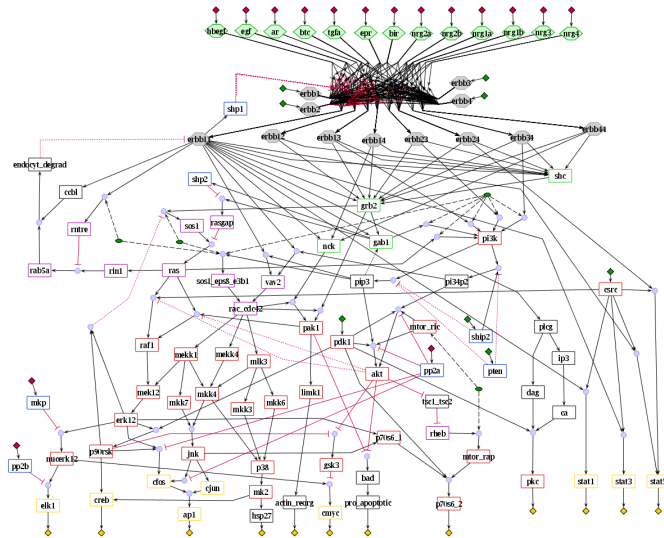
- sol^\wedge computation: **exponential** in the number of **processes within a single sort**.
- Size of sol^\wedge : combinaisons of $|\mathbf{LS}_a|$ processes $\binom{|\mathbf{LS}|}{|\mathbf{LS}_a|}$.
- Size of \mathcal{A}_ζ^ω (and $[\mathcal{B}_\zeta^\omega]$): **polynomial in processes number** \times size of sol^\wedge .

Analyses

- **Over-approximations**: **polynomial** in the size of \mathcal{A}_ζ^ω .
- Different strategies of **under-approximation**:
 - global: **polynomial** in the size of $[\mathcal{B}_\zeta^\omega]$;
 - per solution: \times exponential in the size of sol^\wedge .

\implies efficient with a **small number of processes per sort**, while a **very large number of sorts** can be handled.

EGFR/ErbB Signalling Network (104 components)



[Samaga, *et al.* in PLoS Comput Biol, 2009]

Process Hitting
193 sorts,
748 processes,
2356 actions:
 $\approx 2 \cdot 10^{96}$ states.

Execution times

- Real biological models.
- Wide-range of biological/arbitrary reachability analysis.
- Always conclusive.

Model	sorts	procs	actions	states	Biocham ¹	libDDD ²	PINT ³
egfr20	35	196	670	2^{64}	[3s-KO]	[1s-150s]	0.007s
tcrsig40	54	156	301	2^{73}	[1s-KO]	[0.6s-KO]	0.004s
tcrsig94	133	448	1124	2^{194}	KO	KO	0.030s
egfr104	193	748	2356	2^{320}	KO	KO	0.050s

¹ <http://contraintes.inria.fr/biocham> (using NuSMV2)

² <http://move.lip6.fr/software/DDD>

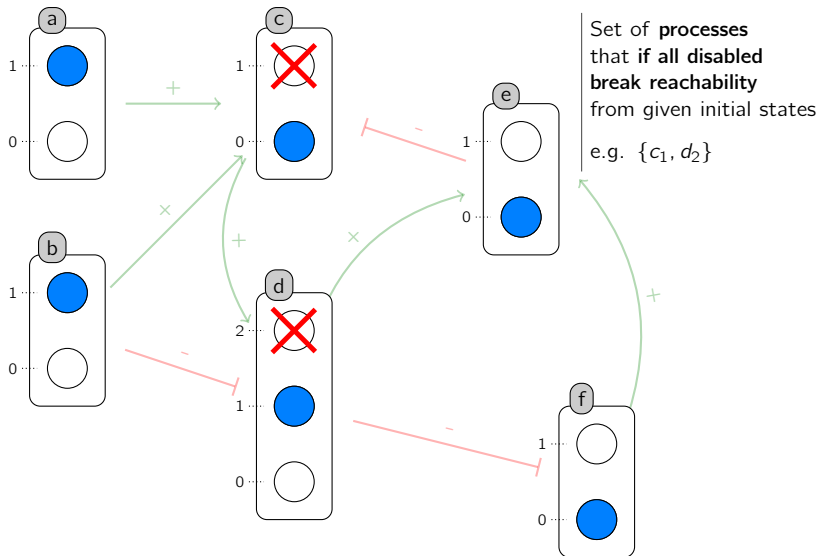
³ <http://loicpauleve.name/pint>

Outline

- ① Fixed points
- ② Reachability
- ③ Cut sets for reachability

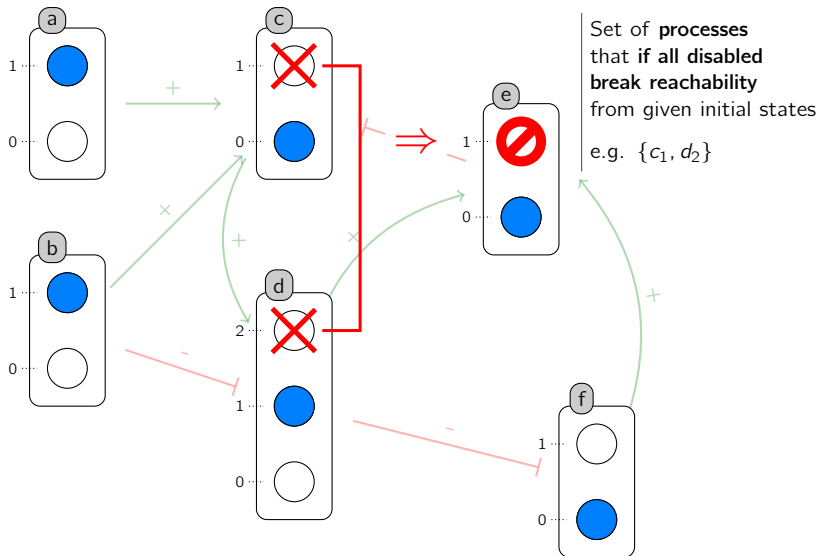
Cut Sets for Reachability

[Paulevé et al. at CAV'13]



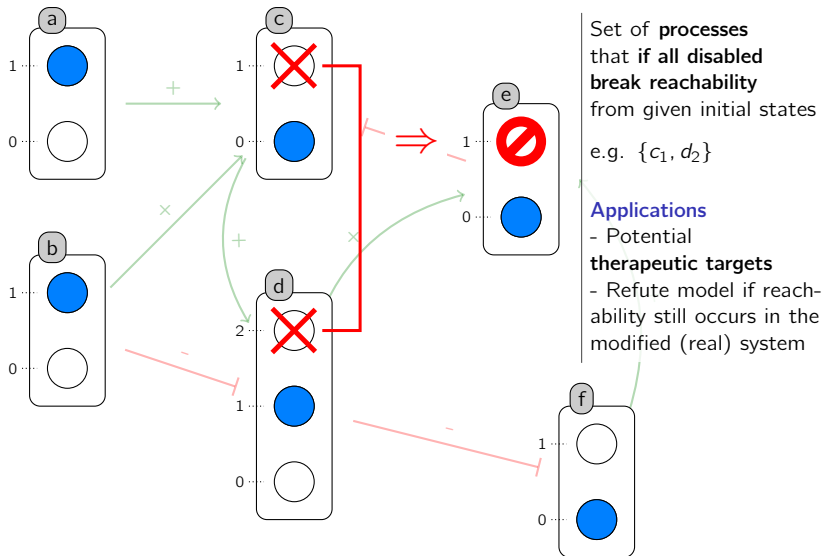
Cut Sets for Reachability

[Paulevé et al. at CAV'13]



Cut Sets for Reachability

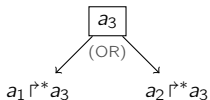
[Paulevé et al. at CAV'13]



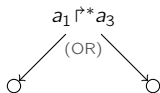
Cut Sets Under-Approximation

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

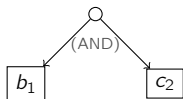
$$\mathbb{V} : \text{nodes} \mapsto \wp(\wp^{\leq N}(\mathcal{Obs})), \mathcal{Obs} \subset \text{LS}$$



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



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



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

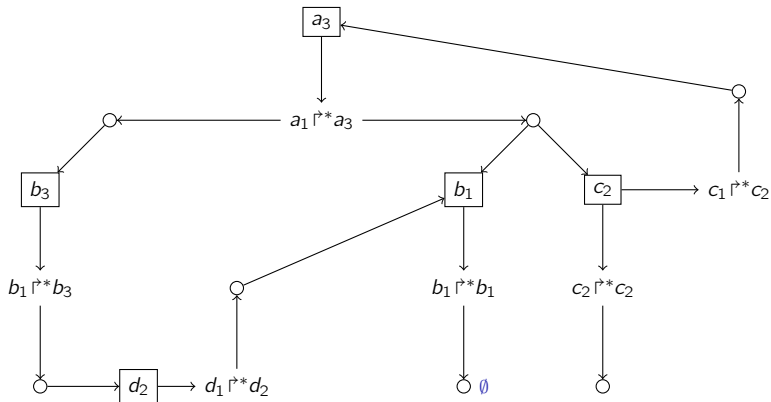
$$\{e^1, \dots, e^n\} \tilde{\times} \{f^1, \dots, f^m\} \triangleq \{e^i \cup f^j \mid i \in [1;n] \wedge j \in [1;m]\}; e^i, f^j \in \wp^{\leq N}(\mathcal{Obs})$$

Cut Sets Under-approximation

Example

Sketch

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

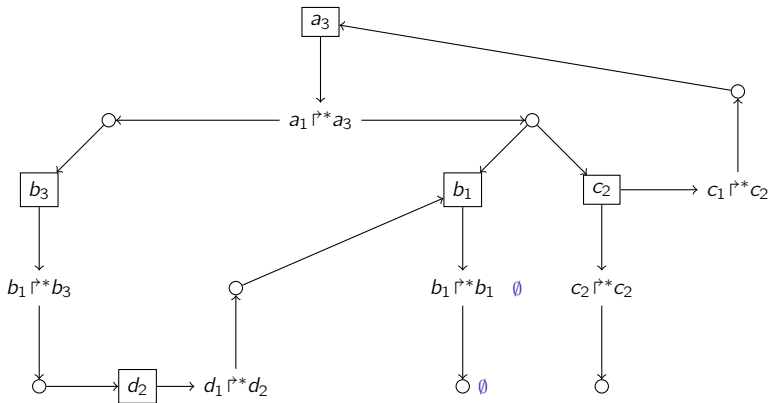


Cut Sets Under-approximation

Example

Sketch

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

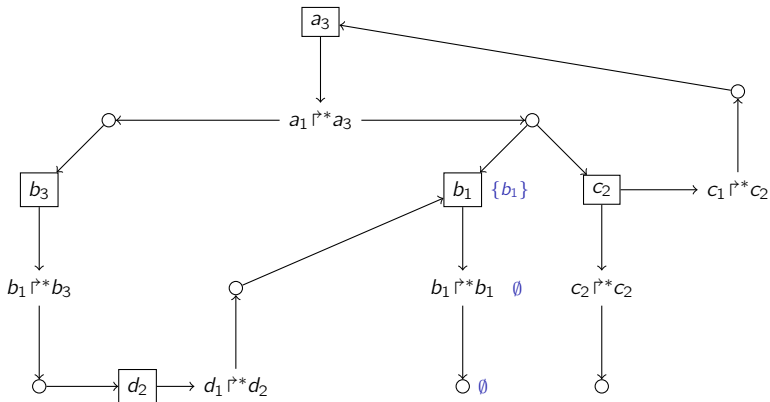


Cut Sets Under-approximation

Example

Sketch

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

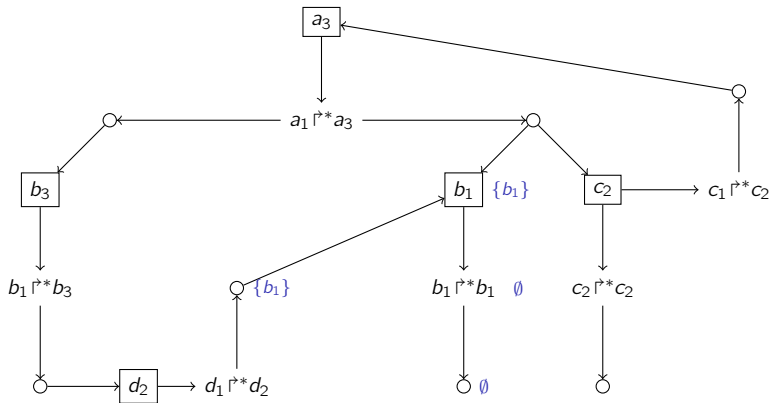


Cut Sets Under-approximation

Example

Sketch

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

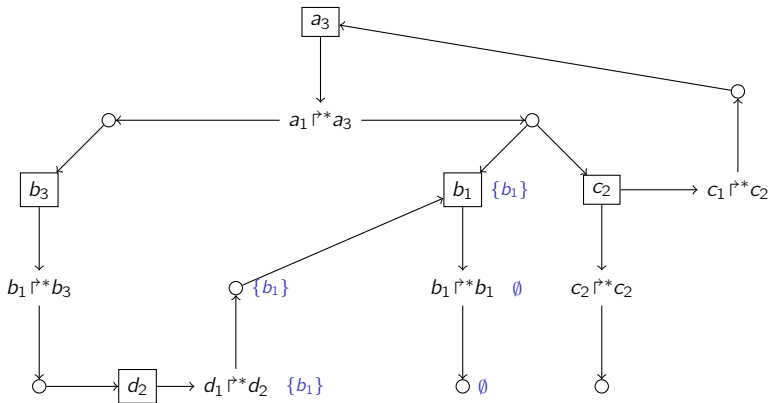


Cut Sets Under-approximation

Example

Sketch

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

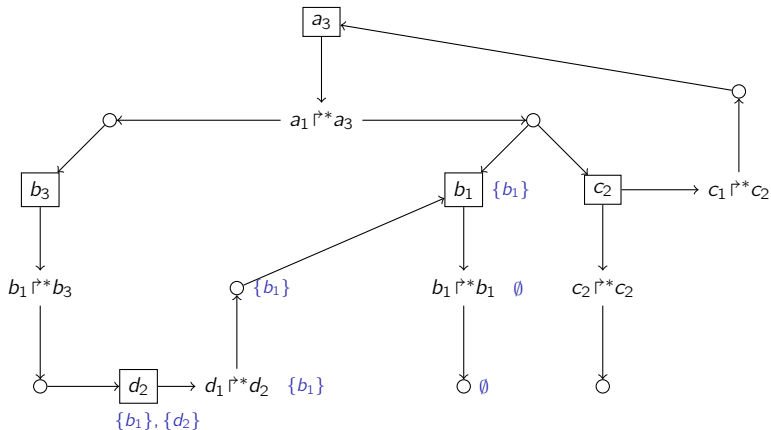


Cut Sets Under-approximation

Example

Sketch

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

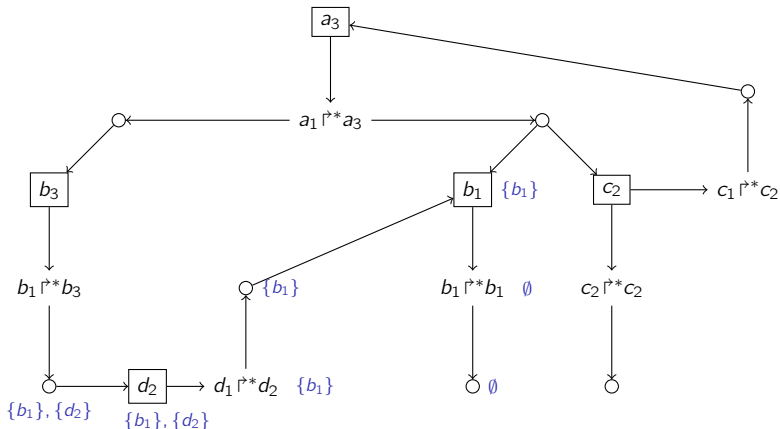


Cut Sets Under-approximation

Example

Sketch

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

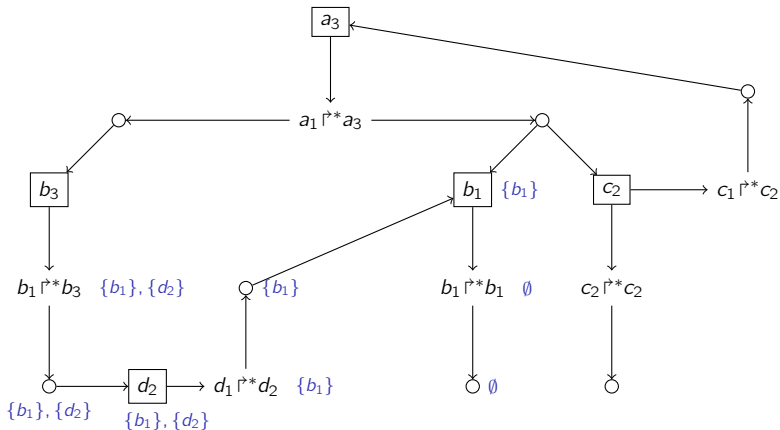


Cut Sets Under-approximation

Example

Sketch

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

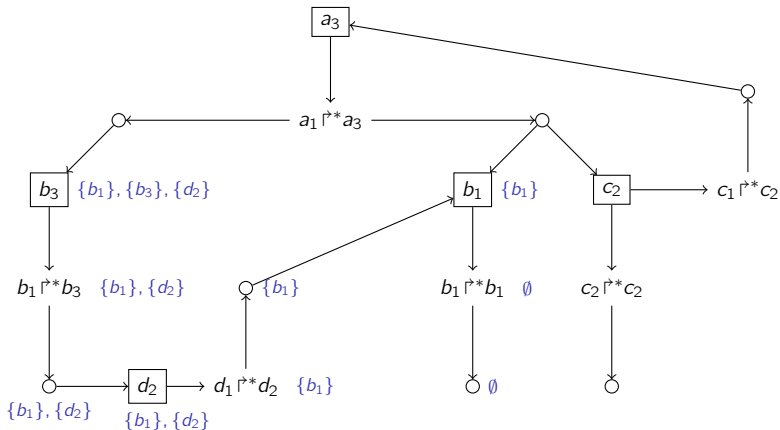


Cut Sets Under-approximation

Example

Sketch

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

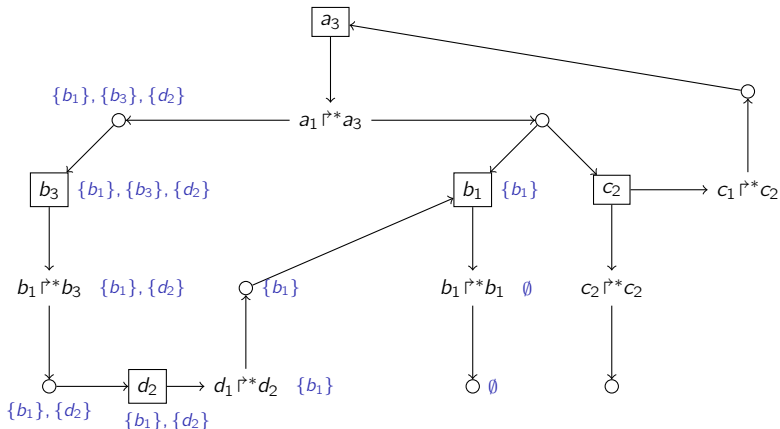


Cut Sets Under-approximation

Example

Sketch

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

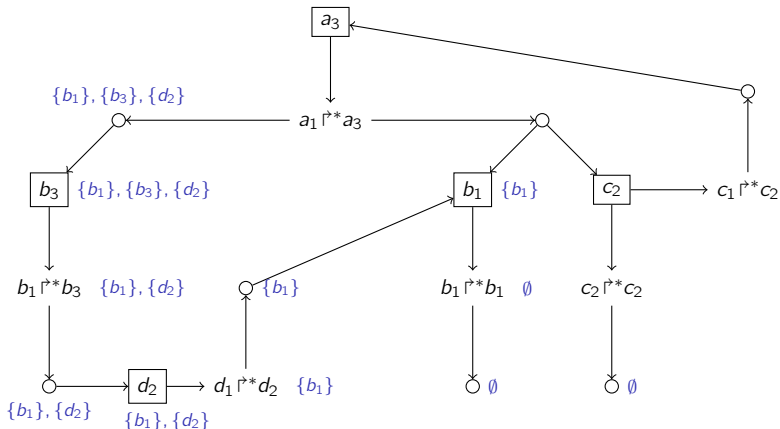


Cut Sets Under-approximation

Example

Sketch

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

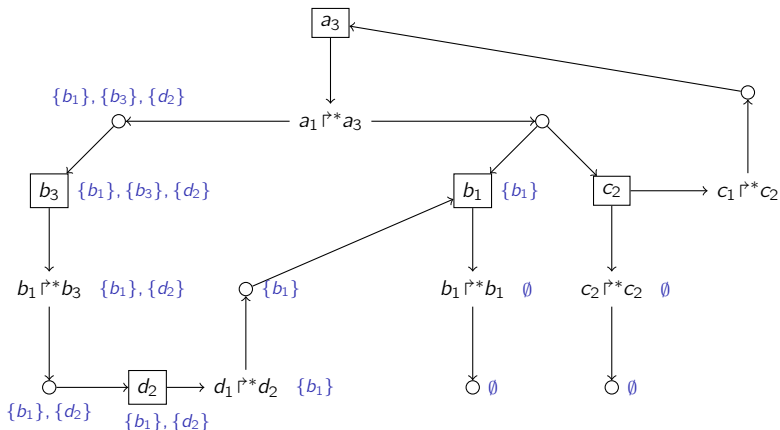


Cut Sets Under-approximation

Example

Sketch

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

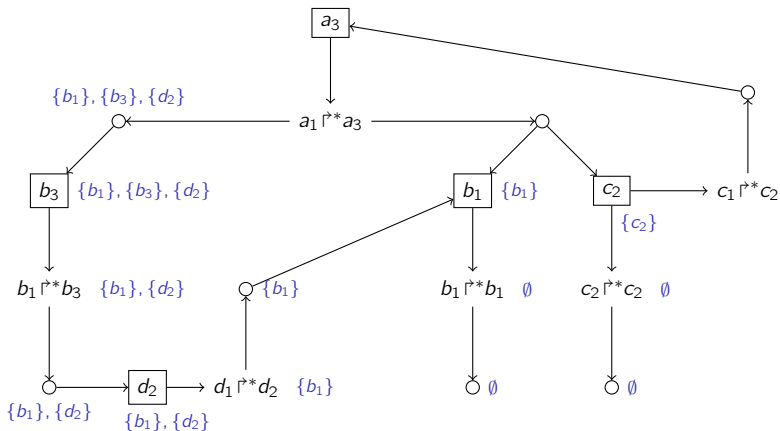


Cut Sets Under-approximation

Example

Sketch

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

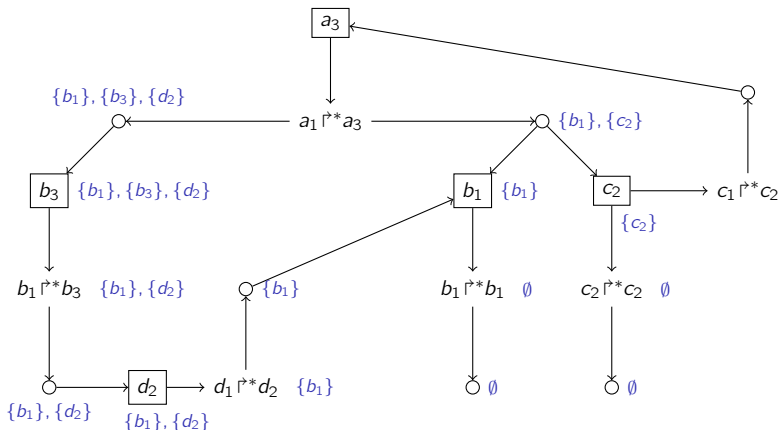


Cut Sets Under-approximation

Example

Sketch

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

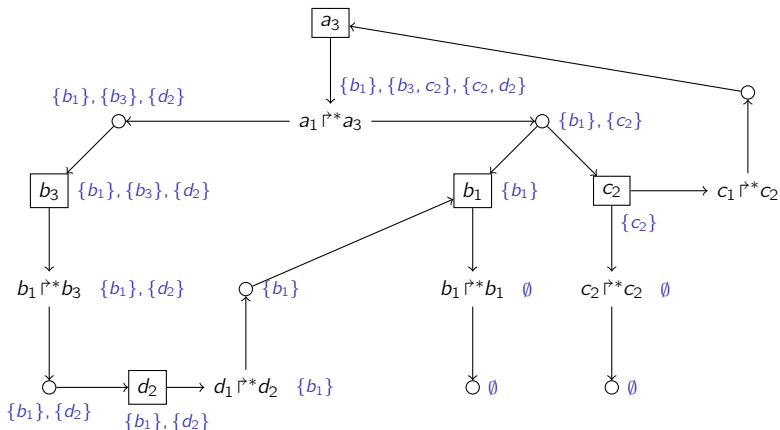


Cut Sets Under-approximation

Example

Sketch

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

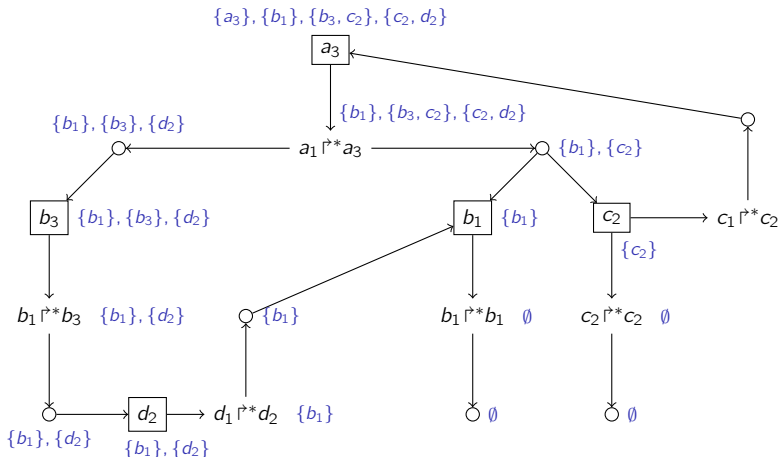


Cut Sets Under-approximation

Example

Sketch

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

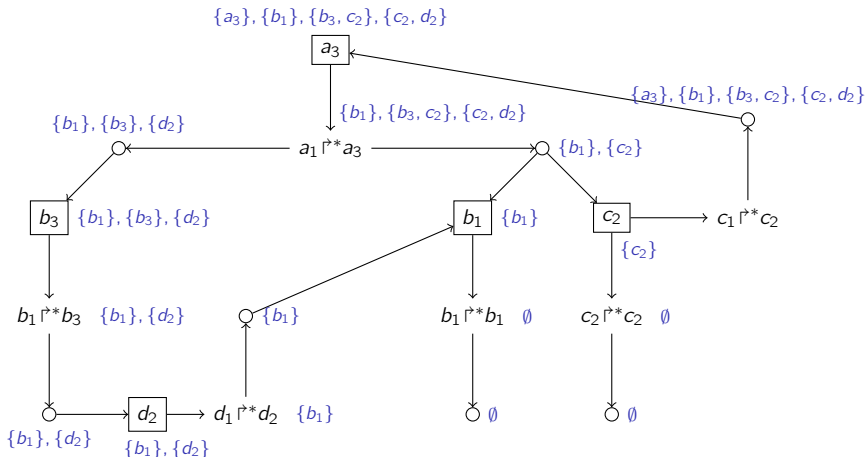


Cut Sets Under-approximation

Example

Sketch

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

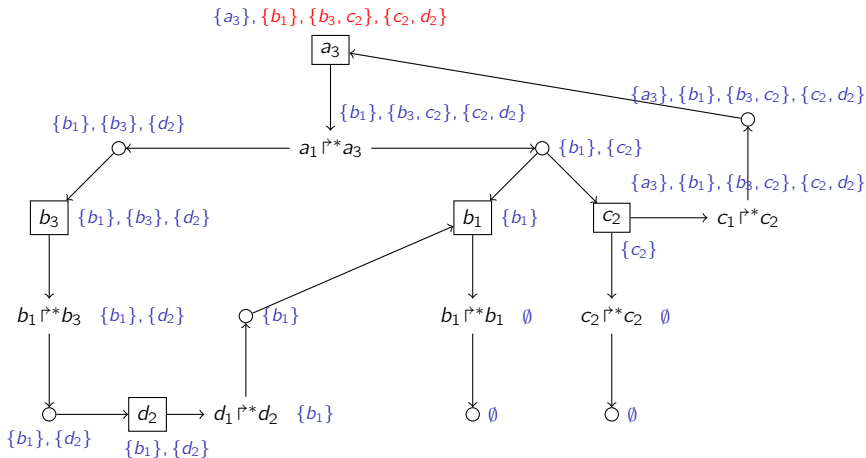


Cut Sets Under-approximation

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.
- $\approx 20,000$ nodes involving $\approx 1,600$ 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://loicpauleve.name/pint> (OCaml);

Dedicated data structures to efficiently compute cross products between million of sets.

Abstract interpretation of dynamics

- Do not build (even symbolically) the transition graph.
- Compact static structure for representing the possible transitions.
⇒ Graph of Local Causality (GLC)

Static analysis using the GLC

- Over-approximation: successive reachability properties + cut sets
Can be defined for any automata network/Petri net.
- Under-approximation: successive reachability properties
Specific to Process Hitting

Extensions

- Process Hitting with Priorities (Maxime Folschette)
⇒ new under-approximation conditions.
- Other dynamical properties: complex attractors, etc.

Thank you for your attention.