Abstraction and Verification of Large-scale Biological Networks

PPS - March 7th, 2013

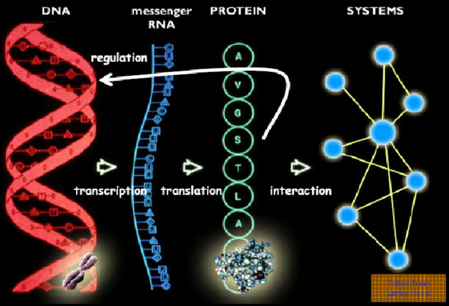
Loïc Paulevé

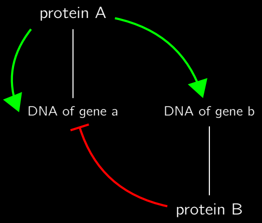
ETH Zürich (BISON group, Heinz Koeppl)

http://loicpauleve.name

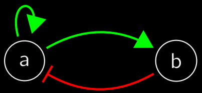
Biological Regulatory Networks (BRNs)

The Interaction Graph





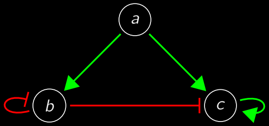
Interaction graph

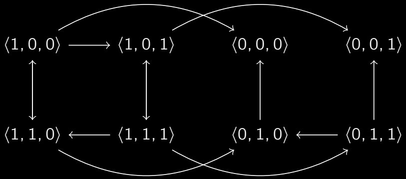


Boolean/Discrete Networks

- Each component has a finite set of qualitative levels ($\{0, 1, 2\}$).
- Functions associate the next level given the state of the regulators.

Boolean network example





$$f^a(a, b, c) = 0$$

$$f^b(a, b, c) = \begin{cases} 1 & \text{if } a = 1 \text{ and } b = 0 \\ 0 & \text{otherwise} \end{cases}$$

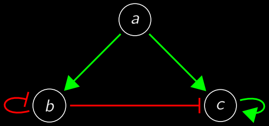
$$f^c(a, b, c) = \begin{cases} 1 & \text{if } b = 0 \text{ and } (a = 1 \text{ or } c = 1) \\ 0 & \text{otherwise} \end{cases};$$

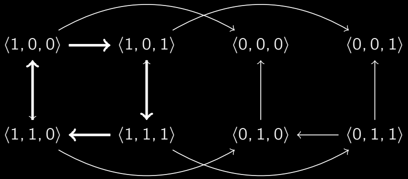
[René Thomas in Journal of Theoretical Biology, 1973]
[Richard, Comet, Bernot in Modern Formal Methods and App., 2006]

Boolean/Discrete Networks

- Each component has a finite set of qualitative levels ($\{0, 1, 2\}$).
- Functions associate the next level given the state of the regulators.

Boolean network example





$$f^a(a, b, c) = 0$$

$$f^b(a, b, c) = \begin{cases} 1 & \text{if } a = 1 \text{ and } b = 0 \\ 0 & \text{otherwise} \end{cases}$$

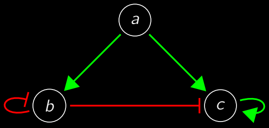
$$f^c(a, b, c) = \begin{cases} 1 & \text{if } b = 0 \text{ and } (a = 1 \text{ or } c = 1) \\ 0 & \text{otherwise} \end{cases};$$

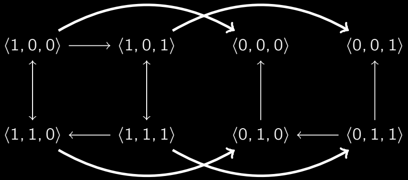
[René Thomas in Journal of Theoretical Biology, 1973]
[Richard, Comet, Bernot in Modern Formal Methods and App., 2006]

Boolean/Discrete Networks

- Each component has a finite set of qualitative levels ($\{0, 1, 2\}$).
- Functions associate the next level given the state of the regulators.

Boolean network example





$$f^a(a, b, c) = 0$$

$$f^b(a, b, c) = \begin{cases} 1 & \text{if } a = 1 \text{ and } b = 0 \\ 0 & \text{otherwise} \end{cases}$$

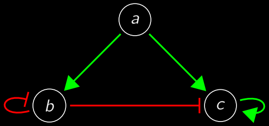
$$f^c(a, b, c) = \begin{cases} 1 & \text{if } b = 0 \text{ and } (a = 1 \text{ or } c = 1) \\ 0 & \text{otherwise} \end{cases};$$

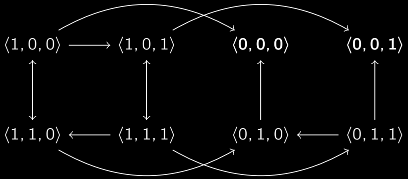
[René Thomas in Journal of Theoretical Biology, 1973]
[Richard, Comet, Bernot in Modern Formal Methods and App., 2006]

Boolean/Discrete Networks

- Each component has a finite set of qualitative levels ($\{0, 1, 2\}$).
- Functions associate the next level given the state of the regulators.

Boolean network example





$$f^a(a, b, c) = 0$$

$$f^b(a, b, c) = \begin{cases} 1 & \text{if } a = 1 \text{ and } b = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$f^c(a, b, c) = \begin{cases} 1 & \text{if } b = 0 \text{ and } (a = 1 \text{ or } c = 1) \\ 0 & \text{otherwise} \end{cases};$$

[René Thomas in Journal of Theoretical Biology, 1973]
[Richard, Comet, Bernot in Modern Formal Methods and App., 2006]

Motivation and Challenges

Prove dynamical properties Validate/Refute a model

- Fixed points (steady states) analysis;
- Reachability properties;
- Attractors characterisation.

Control dynamical properties Therapeutic targets

- Necessary or sufficient conditions.
- Key components/influences/parameters.

Motivation and Challenges

Prove dynamical properties   Validate/Refute a model

- Fixed points (steady states) analysis;
- Reachability properties;
- Attractors characterisation.

Control dynamical properties   Therapeutic targets

- Necessary or sufficient conditions.
- Key components/influences/parameters.

Large-scale models

- Lack of details (knowledge) for some interactions
  $\longrightarrow$ avoid model/parameters enumeration.
- Numerous environment inputs: uncertainity for the initial conditions
  $\longrightarrow$ handle multiple initial states at once.
- Work around the state-space combinatoric explosion
  $\longrightarrow$ abstraction techniques.

Approach, Results

Methods

- New formalism: Process Hitting (class of Asynchronous Automata Networks).
- Dedicated abstract interpretation of dynamics.
- Static causality analysis.

Fixed Point Enumeration

- Reduction to the n-cliques of a n-partite graph.

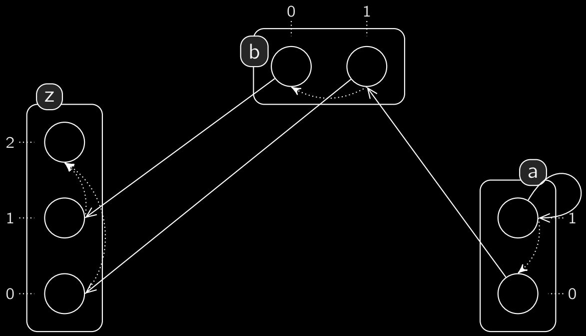
Successive reachability properties $\mathrm{EF}\ a_i \wedge (\mathrm{EF}\ b_j \wedge \dots)$

- Reduced complexity but may be inconclusive (Yes/No/Inconc): poly(#automata), exp(#local states within one automata).
- Necessary/sufficient patterns in a Graph of Local Causality.
- Identification of cut sets for reachability (towards control).

Outline

The Process Hitting Framework

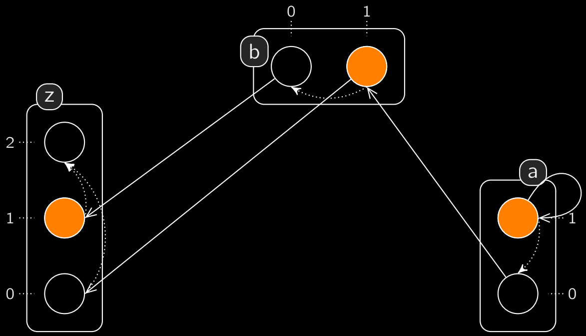
[Paulevé, Magnin, Roux in TCSB 2011]



- Automata: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1\rangle$, $\langle a_0, b_1, z_1\rangle$, $\langle a_0, b_0, z_1\rangle$, . . . ;
- Restriction of Asynchronous Automata Networks.

The Process Hitting Framework

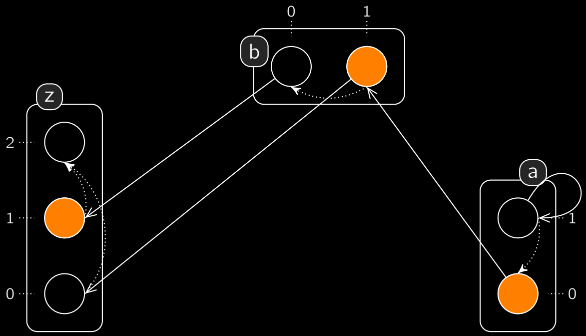
[Paulevé, Magnin, Roux in TCSB 2011]



- Automata: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1 \rangle$, $\langle a_0, b_1, z_1 \rangle$, $\langle a_0, b_0, z_1 \rangle$, ...;
- Restriction of Asynchronous Automata Networks.

The Process Hitting Framework

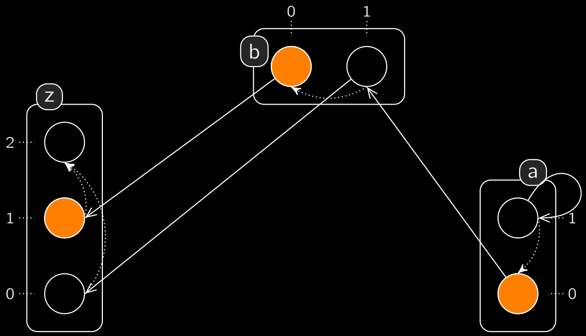
[Paulevé, Magnin, Roux in TCSB 2011]



- Automata: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1\rangle$, $\langle a_0, b_1, z_1\rangle$, $\langle a_0, b_0, z_1\rangle$, . . . ;
- Restriction of Asynchronous Automata Networks.

The Process Hitting Framework

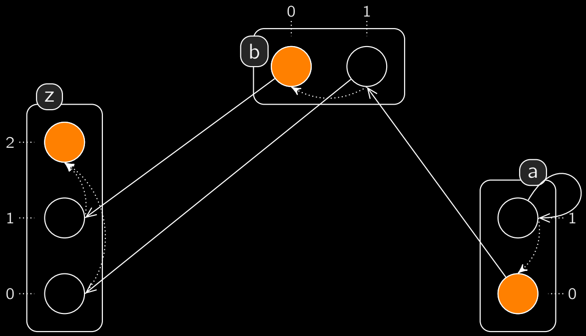
[Paulevé, Magnin, Roux in TCSB 2011]



- Automata: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, ...;
- States: $\langle a_1, b_1, z_1\rangle$, $\langle a_0, b_1, z_1\rangle$, $\langle a_0, b_0, z_1\rangle$, ...;
- Restriction of Asynchronous Automata Networks.

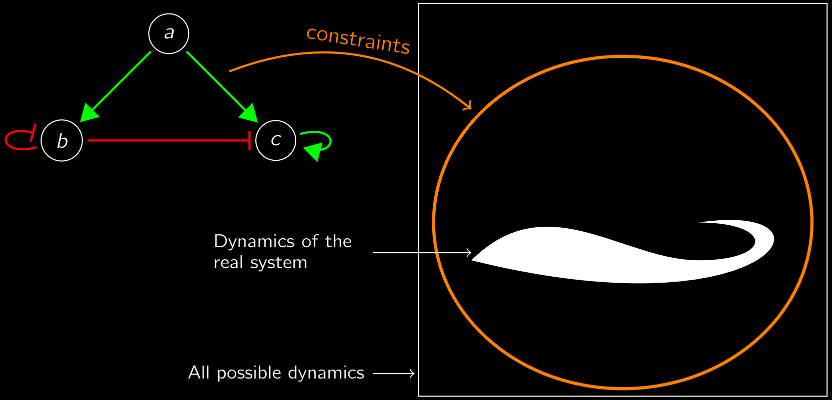
The Process Hitting Framework

[Paulevé, Magnin, Roux in TCSB 2011]



- Automata: a,b,z; Processes: $a_0, a_1, b_0, b_1, z_0, z_1, z_2$;
- Actions: a_0 *hits* b_1 to make it *bounce* to b_0, . . . ;
- States: $\langle a_1, b_1, z_1\rangle$, $\langle a_0, b_1, z_1\rangle$, $\langle a_0, b_0, z_1\rangle$, . . . ;
- Restriction of Asynchronous Automata Networks.

Generalised Dynamics of Regulatory Networks



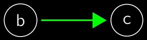
Dynamics over-approximation

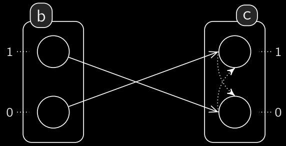
- A component can not increase if none effective activator is present.
- A component can not decrease if none effective inhibitor is present.

Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

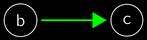


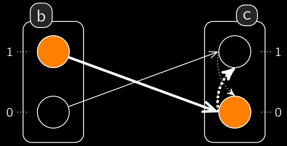


Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

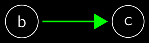


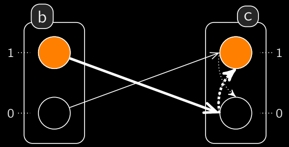


Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

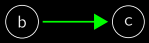


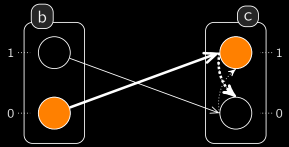


Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

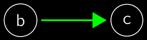


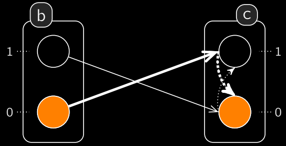


Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

Boolean case:

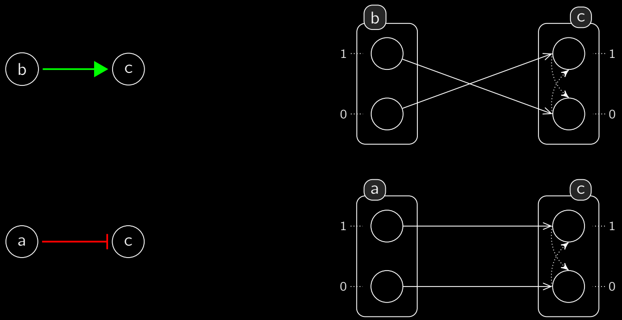




Generalized Dynamics of BRNs

- Idea: the most permissive dynamics [Paulevé, Magnin, Roux in TCSB 2011].
- Without knowledge of functions between components.

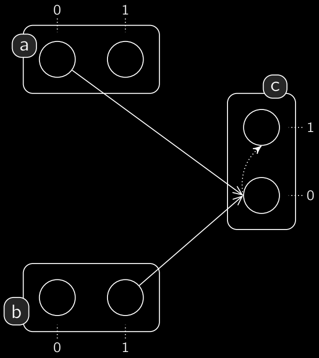
Boolean case:



Note: this construction can be easily extended to multi-valued components.

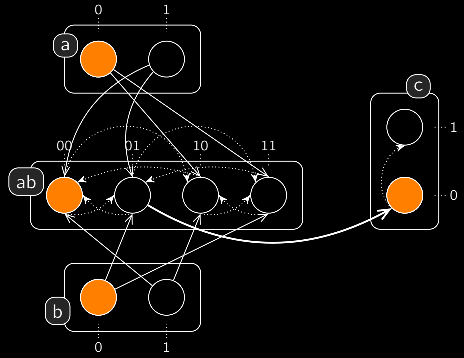
Refining with Cooperation

- Idea: $c_0 \rightarrow c_1$ when a_0 and b_1 are present.
- Introduction of a cooperative automata reflecting the state of a and b.



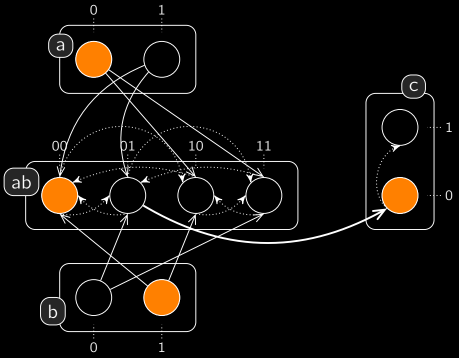
Refining with Cooperation

- Idea: $c_0 \rightarrow c_1$ when a_0 and b_1 are present.
- Introduction of a cooperative automata reflecting the state of a and b.



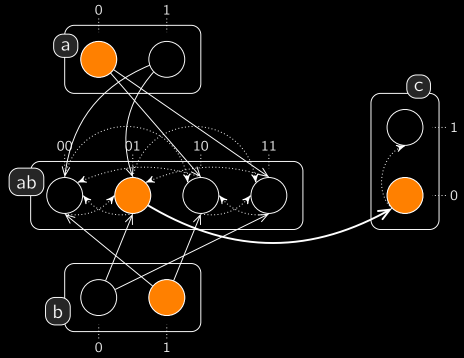
Refining with Cooperation

- Idea: $c_0 \rightarrow c_1$ when a_0 <u>and</u> b_1 are present.
- Introduction of a cooperative automata reflecting the state of a and b.



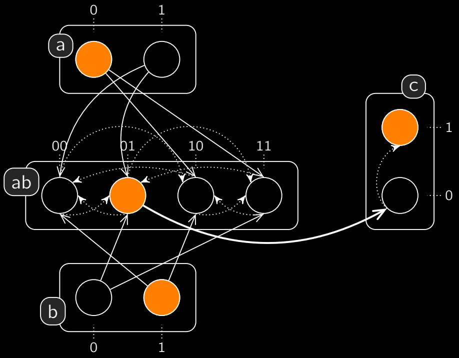
Refining with Cooperation

- Idea: $c_0 \rightarrow c_1$ when a_0 and b_1 are present.
- Introduction of a cooperative automata reflecting the state of a and b.



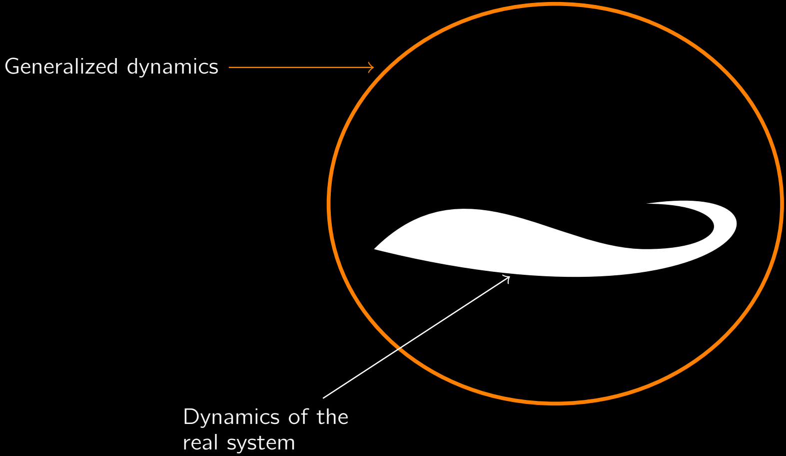
Refining with Cooperation

- Idea: $c_0 \rightarrow c_1$ when a_0 and b_1 are present.
- Introduction of a cooperative automata reflecting the state of a and b.

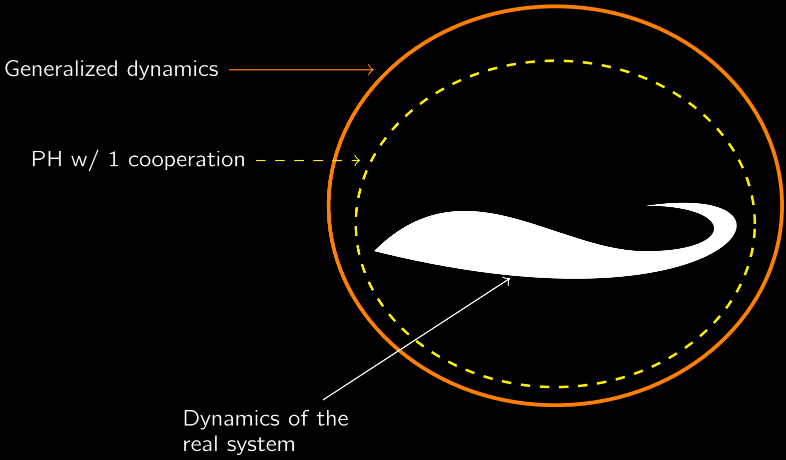


$\Rightarrow$ introduce a temporal shift; similar to complexes.

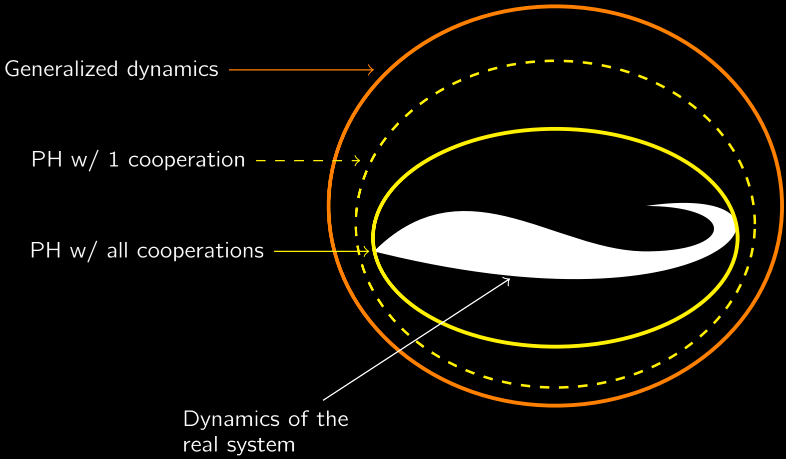
Abstraction Relationships



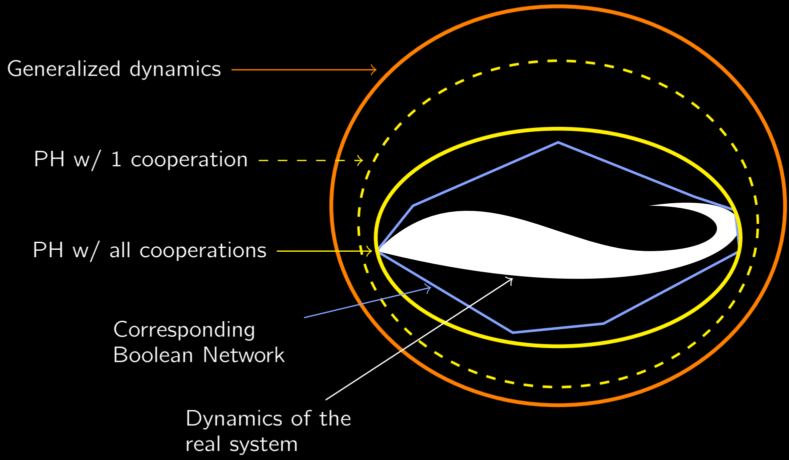
Abstraction Relationships



Abstraction Relationships



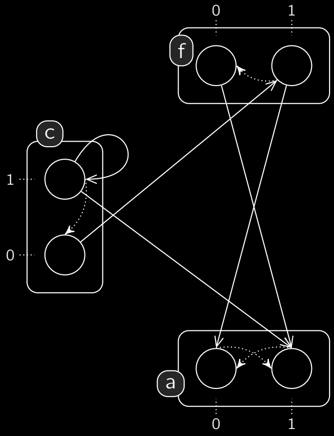
Abstraction Relationships

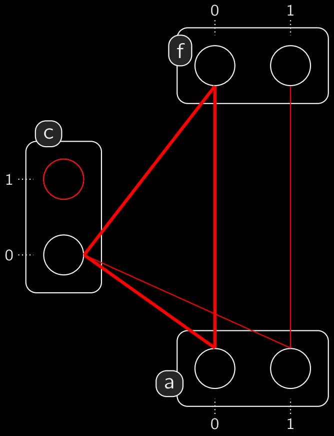


Outline

Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]



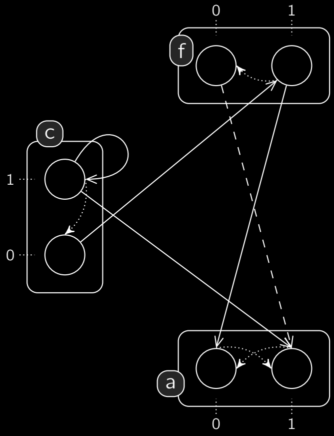


n-cliques are fixed points

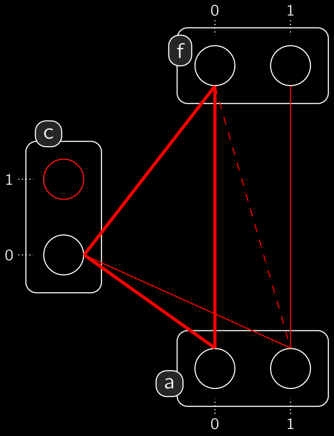
Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]

Process Hitting



Hitless graph

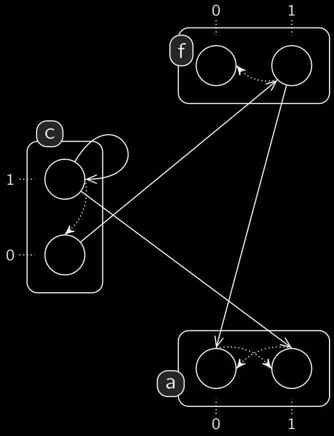


n-cliques <u>are</u> fixed points

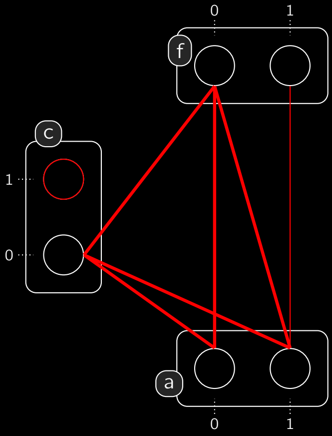
Fixed Points

[Paulevé, Magnin, Roux in TCSB 2011]

Process Hitting



Hitless graph

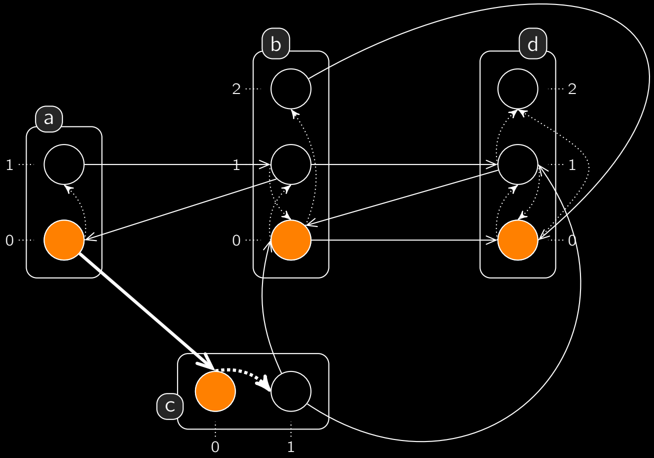


n-cliques are fixed points

Outline

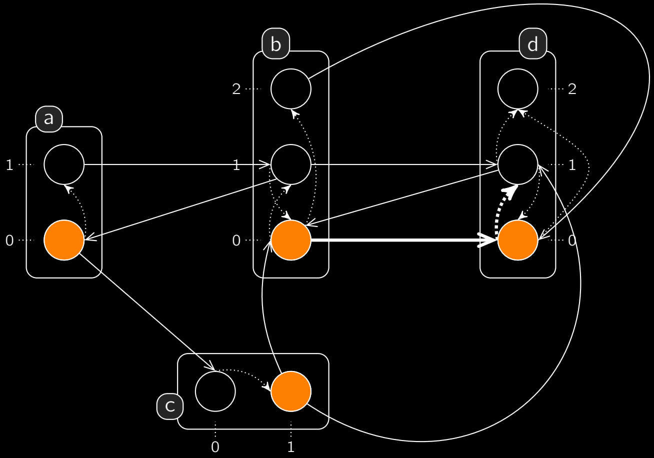
Loïc Paulevé

Scenarios



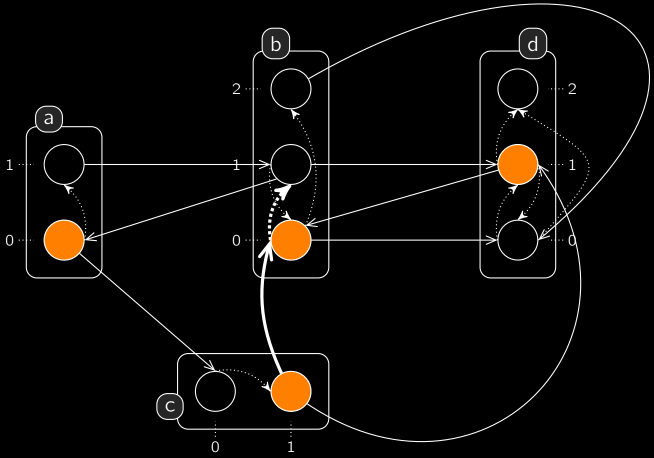
$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$

Scenarios



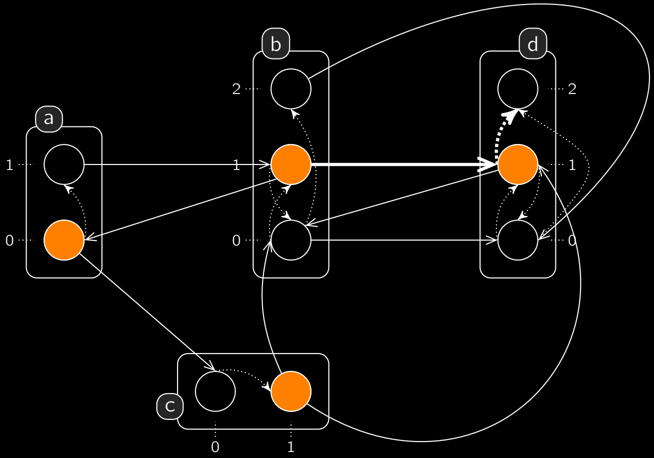
$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$

Scenarios



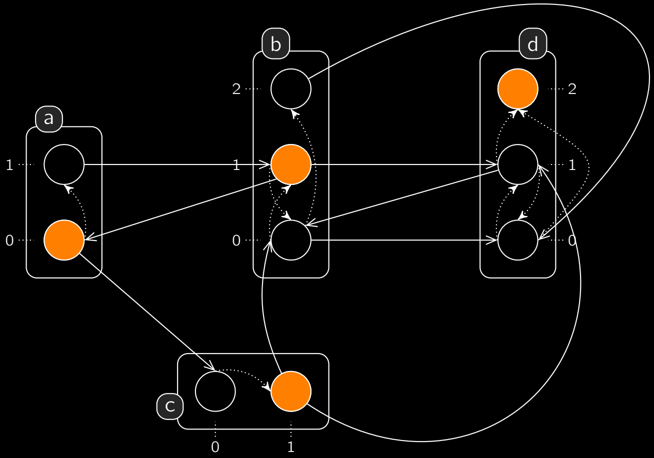
$$a_0 \rightarrow c_0 \upharpoonright c_1 :: b_0 \rightarrow d_0 \upharpoonright d_1 :: c_1 \rightarrow b_0 \upharpoonright b_1 :: b_1 \rightarrow d_1 \upharpoonright d_2$$

Scenarios



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

Scenarios



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

Two Complementary Abstractions of Scenarios

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

Abstraction by Objective Sequences

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

Two Complementary Abstractions of Scenarios

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

Abstraction by Objective Sequences

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

Two Complementary Abstractions of Scenarios

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

Abstraction by Objective Sequences

- $c_0 \Rsh^* c_1 :: d_0 \Rsh^* d_1 :: b_0 \Rsh^* b_1 :: d_1 \Rsh^* d_2$;
- $b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$
- $d_0 \Rsh^* d_2$, ...

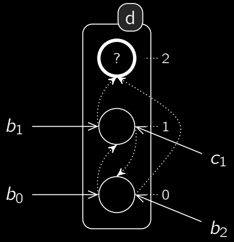
Two Complementary Abstractions of Scenarios

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

Abstraction by Objective Sequences

- $c_0 \Rsh^* c_1 :: d_0 \Rsh^* d_1 :: b_0 \Rsh^* b_1 :: d_1 \Rsh^* d_2$;
- $b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$
- $d_0 \Rsh^* d_2$, ...

Abstraction by Bounce Sequences



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

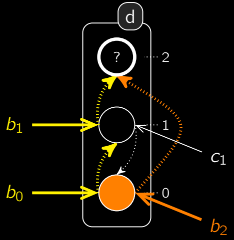
Two Complementary Abstractions of Scenarios

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

Abstraction by Objective Sequences

- $c_0 \Rsh^* c_1 :: d_0 \Rsh^* d_1 :: b_0 \Rsh^* b_1 :: d_1 \Rsh^* d_2$;
- $b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$
- $d_0 \Rsh^* d_2$, ...

Abstraction by Bounce Sequences



E.g.: $b_0 \rightarrow d_0 \Rsh d_1 :: b_1 \rightarrow d_1 \Rsh d_2$ ($d_0 \Rsh^* d_2$)
$\Rightarrow$ can be computed off-line:

- $\mathbf{BS}(d_0 \Rsh^* d_2) = \{b_0 \rightarrow d_0 \Rsh d_1 :: b_1 \rightarrow d_1 \Rsh d_2,$
  $b_2 \rightarrow d_0 \Rsh d_2\}$;
- $\mathbf{BS}^\wedge(d_0 \Rsh^* d_2) = \{\{b_0, b_1\}, \{b_2\}\}$.

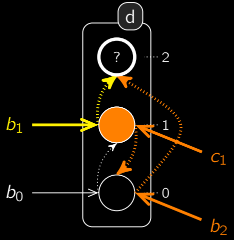
Two Complementary Abstractions of Scenarios

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

Abstraction by Objective Sequences

- $c_0 \Rsh^* c_1 :: d_0 \Rsh^* d_1 :: b_0 \Rsh^* b_1 :: d_1 \Rsh^* d_2$;
- $b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$
- $d_0 \Rsh^* d_2$, ...

Abstraction by Bounce Sequences



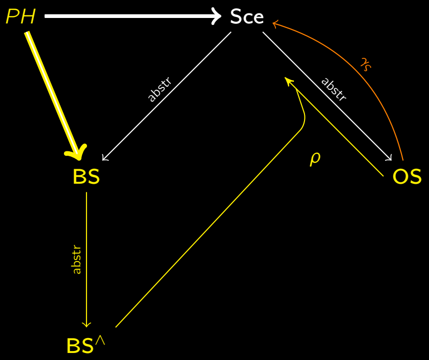
E.g.: $b_0 \rightarrow d_0 \Rsh d_1 :: b_1 \rightarrow d_1 \Rsh d_2$ $(d_0 \Rsh^* d_2)$
$\Rightarrow$ can be computed off-line:

- $\mathbf{BS}(d_0 \Rsh^* d_2) = \{b_0 \rightarrow d_0 \Rsh d_1 :: b_1 \rightarrow d_1 \Rsh d_2,$
  $b_2 \rightarrow d_0 \Rsh d_2\}$;
- $\mathbf{BS}^\wedge(d_0 \Rsh^* d_2) = \{\{b_0, b_1\}, \{b_2\}\}$.
- $\mathbf{BS}(d_1 \Rsh^* d_2) = \{b_1 \rightarrow d_1 \Rsh d_2,$
  $c_1 \rightarrow d_1 \Rsh d_0 :: b_2 \rightarrow d_0 \Rsh d_2\}$;
- $\mathbf{BS}^\wedge(d_1 \Rsh^* d_2) = \{\{b_1\}, \{b_2, c_1\}\}$.

Abstract Interpretation of Scenarios

Inputs

- Context: For each automata, subset of initial processes.
  E.g. $\varsigma = \langle a_0, \{b_0, b_2\}, c_0, d_0 \rangle$.
- Objective Sequence (**OS**): reachability property.
  E.g. $\omega = b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$ (or $\mathtt{EF}\ (b_1 \wedge \mathtt{EF}\ d_2)$).



Overall approach

- 2 complementary abstractions;
- Concretization: $\gamma_\varsigma : \mathbf{OS} \mapsto \wp(\mathbf{Sce})$;
- Refinements: $\rho : \mathbf{OS} \mapsto \wp(\mathbf{OS})$;
- $\gamma_\varsigma(\omega) = \gamma_\varsigma(\rho(\omega))$.

Objective Sequence Refinements

$$\gamma_\varsigma(\omega) = \{\delta \in \textbf{Sce} \mid \omega \text{ abstracts } \delta \wedge \mathrm{support}(\delta) \subseteq \varsigma\}.$$

Objective Refinement by $\textbf{BS}^\wedge$: $\rho^\wedge$

$\textbf{Obj} \times \wp(\textbf{BS}^\wedge)$	$\wp(\textbf{OS})$
$d_0 \Rsh^* d_2$, $\{\{b_0, b_1\}, \{b_2\}\}$	$\star \Rsh^* b_0 :: b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_1 :: b_1 \Rsh^* b_0 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_2 :: d_0 \Rsh^* d_2$
$\gamma_\varsigma(d_0 \Rsh^* d_2)$	$= \gamma_\varsigma(\rho^\wedge(d_0 \Rsh^* d_2, \textbf{BS}^\wedge(d_0 \Rsh^* d_2)))$

Objective Sequence Refinements

$$\gamma_\varsigma(\omega) = \{\delta \in \textbf{Sce} \mid \omega \text{ abstracts } \delta \wedge \mathrm{support}(\delta) \subseteq \varsigma\}.$$

Objective Refinement by $\mathbf{BS}^\wedge$: $\rho^\wedge$

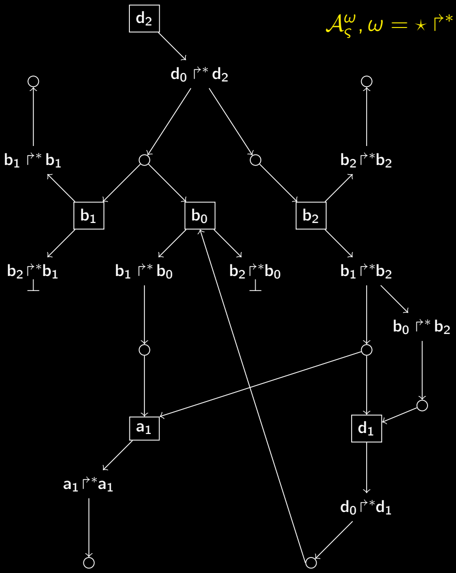
$\mathbf{Obj} \times \wp(\mathbf{BS}^\wedge)$	$\wp(\mathbf{OS})$
$d_0 \Rsh^* d_2$, $\{\{b_0, b_1\}, \{b_2\}\}$	$\star \Rsh^* b_0 :: b_0 \Rsh^* b_1 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_1 :: b_1 \Rsh^* b_0 :: d_0 \Rsh^* d_2$, $\star \Rsh^* b_2 :: d_0 \Rsh^* d_2$
$\gamma_\varsigma(d_0 \Rsh^* d_2)$	$= \gamma_\varsigma(\rho^\wedge(d_0 \Rsh^* d_2, \mathbf{BS}^\wedge(d_0 \Rsh^* d_2)))$

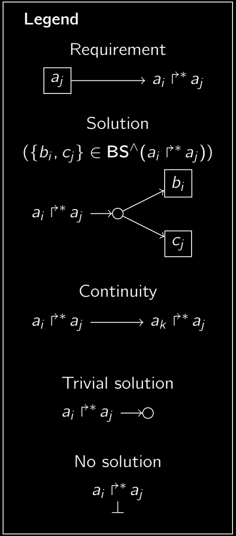
Generalization to **OS** refinements: $\widetilde{\rho}$

$\mathbf{OS} \times \wp(\mathbf{BS}^\wedge)$	$\wp(\mathbf{OS})$
$\omega, \mathbf{BS}^\wedge$	$\mathrm{interleave}\begin{pmatrix}\omega' \\ \omega_{1..n-1}\end{pmatrix} :: \omega_{n..\lvert\omega\rvert}$ where $n \in \mathbb{I}^\omega$ and $\omega' :: \omega_n \in \rho^\wedge(\omega_n, \mathbf{BS}^\wedge(\omega_n))$
$\gamma_\varsigma(\omega)$	$= \gamma_\varsigma(\widetilde{\rho}(\omega, \mathbf{BS}^\wedge))$

Graph of Local Causality

$$\mathcal{A}^{\omega}_{\varsigma}, \omega = \star \Rsh^* d_2, \varsigma = \langle a_1, \{b_1, b_2\}, c_1, d_0 \rangle$$





Approximations of Successive Reachability

Over-approximations

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

No / Inconc

$\Uparrow$

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

$\Uparrow$

Under-approximations

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

[Paulevé, Magnin, Roux in Mathematical Structures in Computer Science, 2012]

Approximations of Successive Reachability

Over-approximations

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

No / Inconc

$\Uparrow$

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

$\Uparrow$

Under-approximations

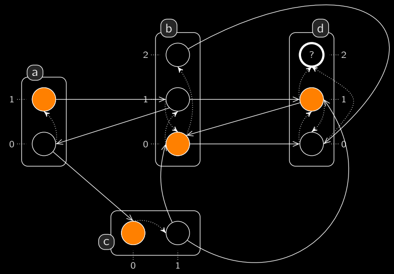
- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

[Paulevé, Magnin, Roux in Mathematical Structures in Computer Science, 2012]

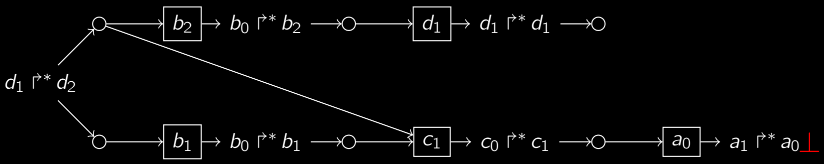
Un-ordered Over-approximation

Example



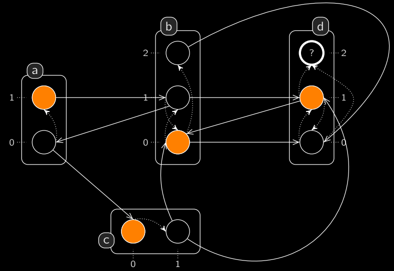
Necessary condition for reaching d_2:
There exists a traversal of $\mathcal{A}_\varsigma^\omega$ such that:

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



Un-ordered Over-approximation

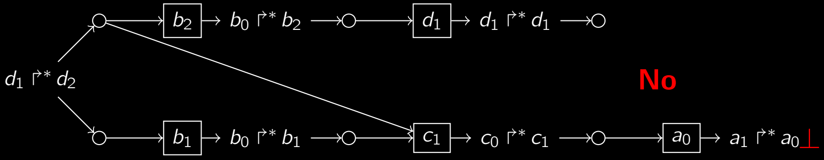
Example



Necessary condition for reaching d_2:

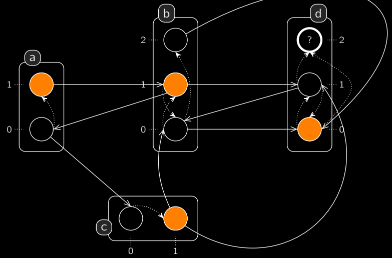
There exists a traversal of $\mathcal{A}^{\omega}_{\varsigma}$ such that:

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



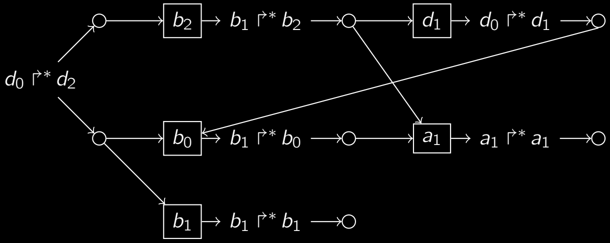
Un-ordered Over-approximation

Example



Necessary condition for reaching d_2:
There exists a traversal of $\mathcal{A}^\omega_\varsigma$ such that:

- 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 occurences order constraints.

No / Inconc

$\Uparrow$

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

$\Uparrow$

Under-approximations

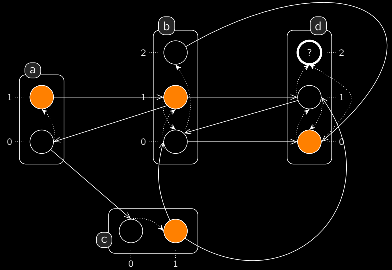
- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

[Paulevé, Magnin, Roux in Mathematical Structures in Computer Science, 2012]

Un-ordered Under-approximation

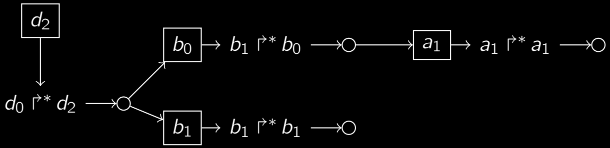
Example



Sufficient condition for reaching d_2:

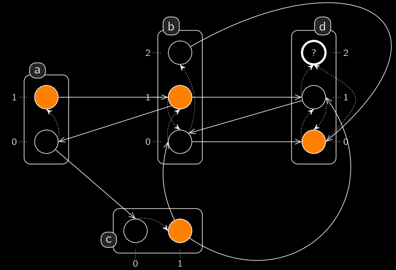
- $\lceil \mathcal{B}^{\omega}_{\varsigma} \rceil$ has no cycle;
- each objective has at least one solution.

$\lceil \mathcal{B}^{\omega}_{\varsigma} \rceil$: saturated $\mathcal{A}^{\omega}_{\varsigma}$.



Un-ordered Under-approximation

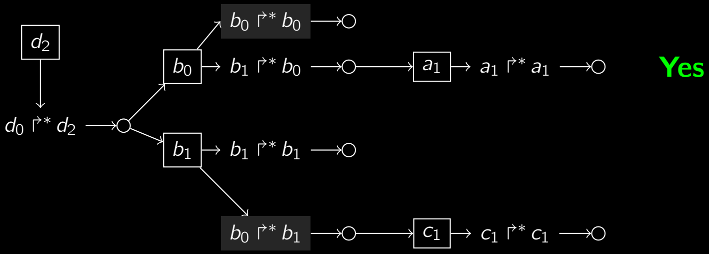
Example



Sufficient condition for reaching d_2:

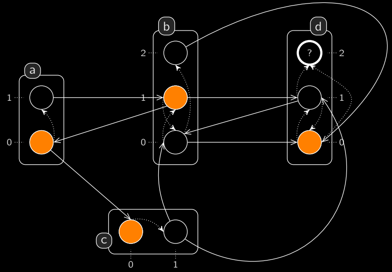
- $\lceil \mathcal{B}^{\omega}_{\varsigma} \rceil$ has no cycle;
- each objective has at least one solution.

$\lceil \mathcal{B}^{\omega}_{\varsigma} \rceil$: saturated $\mathcal{A}^{\omega}_{\varsigma}$.



Un-ordered Under-approximation

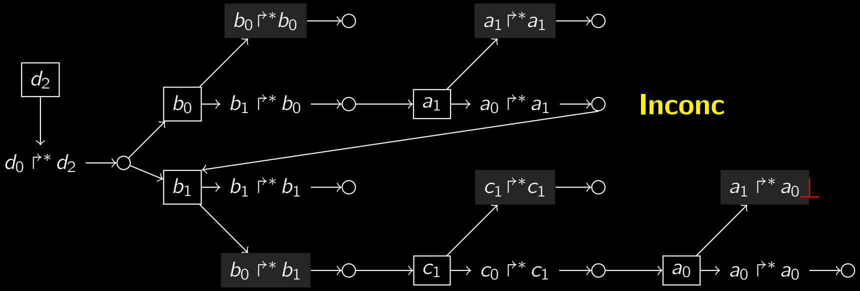
Example



Sufficient condition for reaching d_2:

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

$\lceil \mathcal{B}^\omega_\varsigma \rceil$: saturated $\mathcal{A}^\omega_\varsigma$.



Static Analysis of Successive Reachability

Over-approximations

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

No / Inconc

$\Uparrow$

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

$\Uparrow$

Under-approximations

- Un-ordered approximation.
- Ordered approximation.

Yes / Inconc

[Paulevé, Magnin, Roux in Mathematical Structures in Computer Science, 2012]

Complexity

Graph of Local Causality $\mathcal{A}^\omega_\varsigma$, $\lceil \mathcal{B}^\omega_\varsigma \rceil$

- **BS**$^\wedge$: exp(#processes within one automata).
- $\mathcal{A}^\omega_\varsigma$ (and $\lceil \mathcal{B}^\omega_\varsigma \rceil$): poly(#processes) $\times$ exp(#processes within one automata).

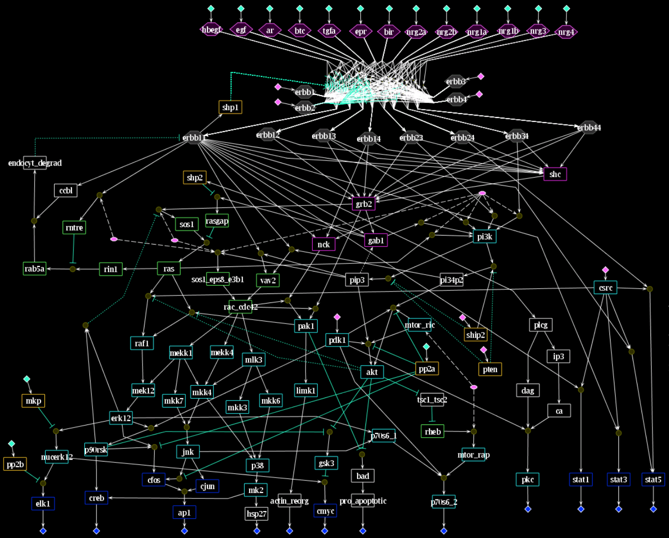
Analyses

- Over-approximations: polynomial in the size of $\mathcal{A}^\omega_\varsigma$.
- Different strategies of under-approximation:
  - global: polynomial in the size of $\lceil \mathcal{B}^\omega_\varsigma \rceil$;
  - per solution: $\times$ exponential in the size of **BS**$^\wedge$.

$\Longrightarrow$ efficient with a small number of processes per automata, while a very large number of automata can be handled.

EGFR/ErbB Signalling Network

(104 components)



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

Process Hitting

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

Execution times

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

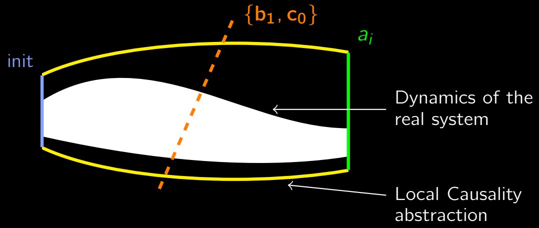
Model	autom.	procs	actions	states	Biocham[1]	libDDD[2]	PINT[3]
egfr20	35	196	670	2^{64}	[3s-KO]	[1s-150s]	0.007s
tcrsig40	54	156	301	2^{73}	[1s-KO]	[0.6s-KO]	0.004s
tcrsig94	133	448	1124	2^{194}	KO	KO	0.030s
egfr104	193	748	2356	2^{320}	KO	KO	0.050s

[1] http://contraintes.inria.fr/biocham (using NuSMV2)
[2] http://move.lip6.fr/software/DDD
[3] http://process.hitting.free.fr

Outline

Cut Sets of Processes for Reachability

Goal sets of processes whose action is necessary for a given process reachability.



Results

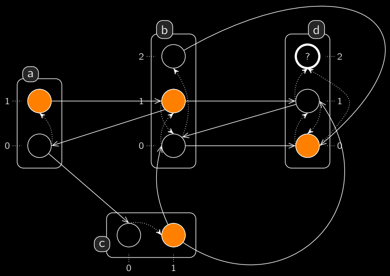
- Efficient under-approximation using the Graph of Local Causality: no candidate enumeration, no model-checking.
- Applicable to any automata network.

Application

- Formal identification of therapeutic targets.
- Models of very large biological networks: PID (+9000 components): computation of 1- to 5-sets between 1s and 8min.

[Paulevé, Andrieux, Koeppl at CAV 2013]

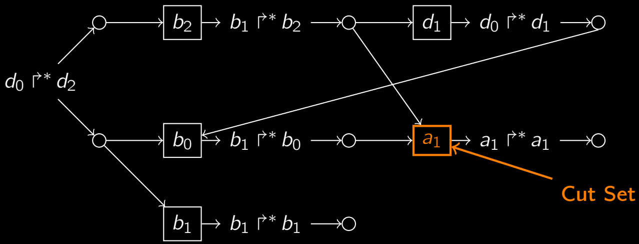
Extraction of Cut Sets



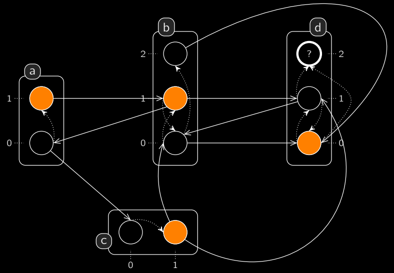
Necessary condition for reaching d_2:

There exists a traversal of $\mathcal{A}^{\omega}_{\varsigma}$ such that:

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



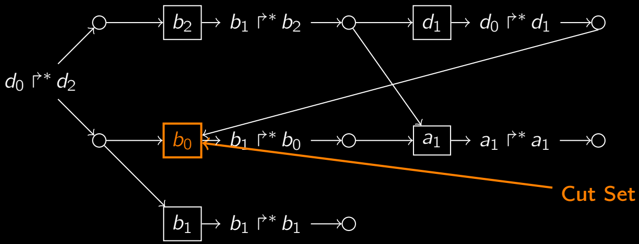
Extraction of Cut Sets



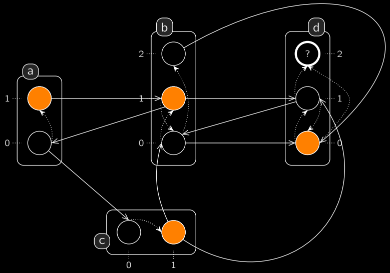
Necessary condition for reaching d_2:

There exists a traversal of $\mathcal{A}^{\omega}_{\varsigma}$ such that:

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



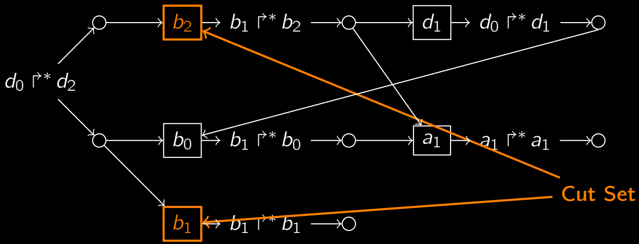
Extraction of Cut Sets



Necessary condition for reaching d_2:

There exists a traversal of $\mathcal{A}_\varsigma^\omega$ such that:

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



Outline

Conclusion

The Process Hitting framework

- Qualitative asynchronous modelling.
- Different levels of dynamics abstractions (partial knowledge on cooperations).
- Automatic encoding of Boolean Networks (over-approximation).

Abstract causality analysis

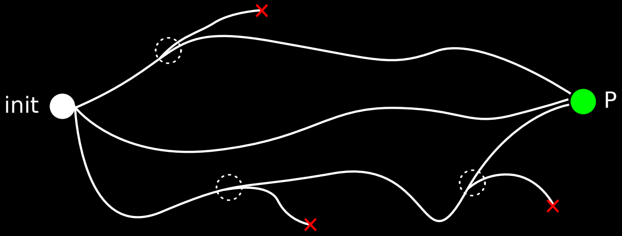
- Local causality reasonning.
- Over- and under-approximation of reachability properties.
- Extract necessary sets of processes (potential therapeutic targets).
- Tractable on very large networks.

Implementation: PINT software - `http://process.hitting.free.fr`

Future work

Process Hitting with Priorities

- Static split of actions into priority classes.
- An action can be played only if none action with higher priority can be played.
- $\Longrightarrow$ different time-scales;
- $\Longrightarrow$ enhanced expressivity (with 3 classes: Petri Nets).



Link with static analyse of Boolean networks

- Relate Graph of Local Causality with Interaction Graph/Boolean functions constraints.

Thank you for your attention.