Towards the logical prediction of control targets for biological networks

Loïc Paulevé

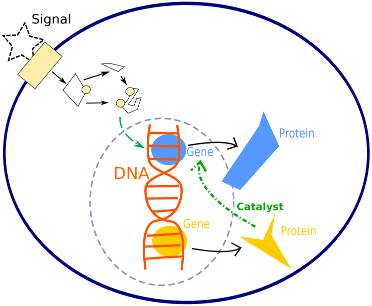
LRI, CNRS / Université Paris-Sud, Orsay, France – BioInfo team

loic.pauleve@lri.fr

http://loicpauleve.name

Open University Digicosme - 2016, May 18

Biological networks



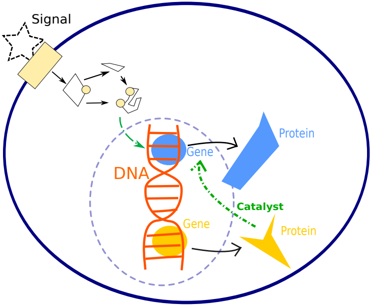
Prediction

- Cell response w.r.t. signal+environment
- Long-term behaviours (differentiation)

Control

- Mutations/Perturbations for modifying cell behaviour
- Trans/De-differentiation

Biological networks



Prediction

- Cell response w.r.t. signal+environment
- Long-term behaviours (differentiation)

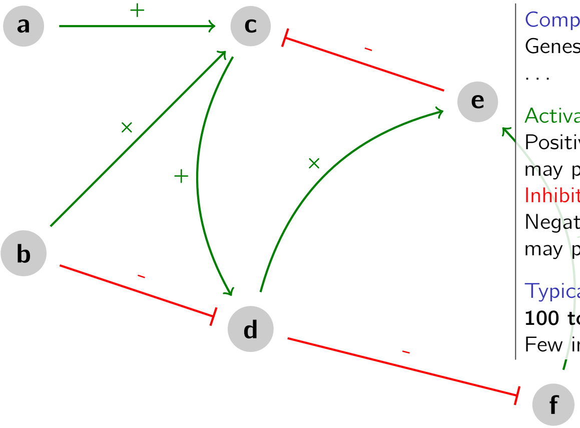
Control

- Mutations/Perturbations for modifying cell behaviour
- Trans/De-differentiation

$\Rightarrow$ { **Computational models of dynamics**
−Formal verification
−Automatic reasoning

Interaction networks

E.g., Signalling Networks, Gene Regulatory Networks



Components
Genes, proteins, complexes, ...

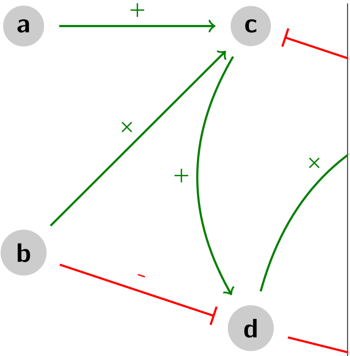
Activations
Positive influence (*a* increase may promote *c* increase).

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

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

Logical models of qualitative dynamics

E.g., Boolean networks, Automata networks



Local state of node

Discrete variable (Boolean, multi-valued).
No population ⇒ qualitative level

Rules for updating the state w.r.t. to the state of its regulators:

Function based
Boolean/Thomas networks

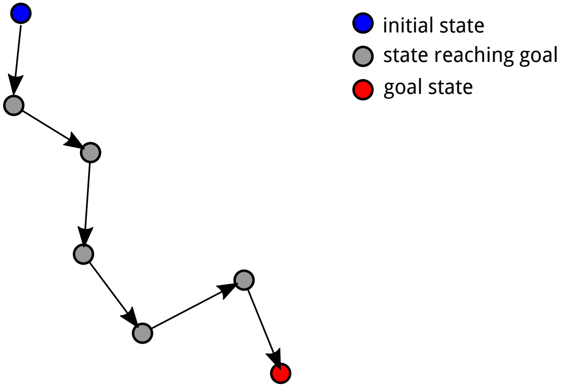
Transition based
Petri nets/**Automata networks**

Dynamics

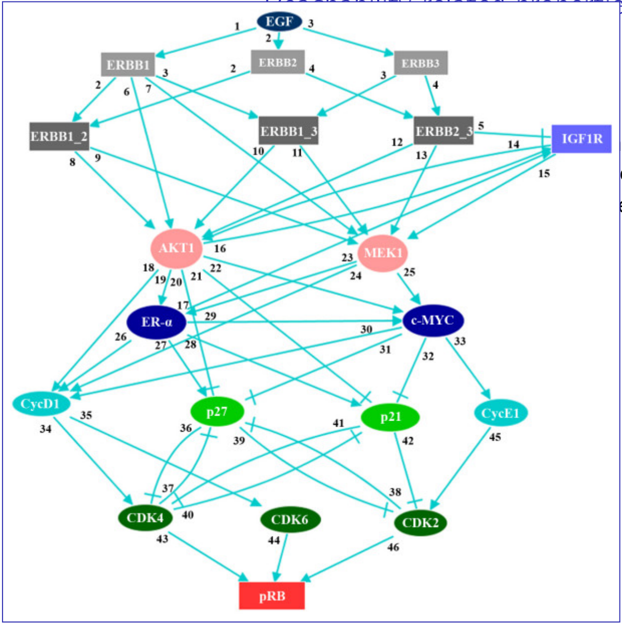
Transitions between global states
Can be non-deterministic
(e.g., different ordering of node changes, but not only).

Reachability-related properties

Global state graph



Reachability-related properties



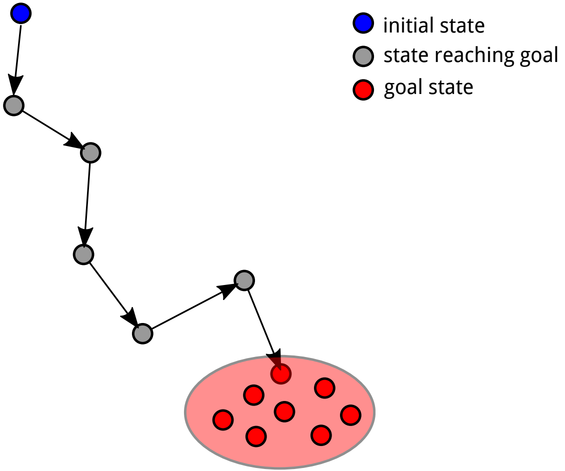
te

ching goal

e

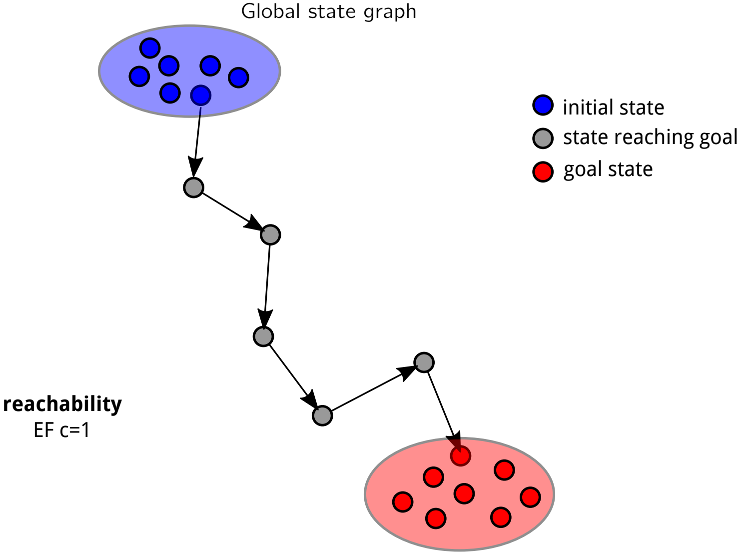
Reachability-related properties

Global state graph



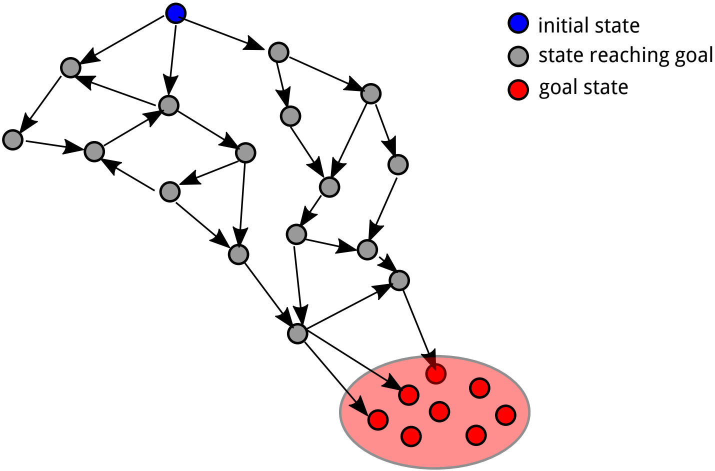
reachability
EF c=1

Reachability-related properties



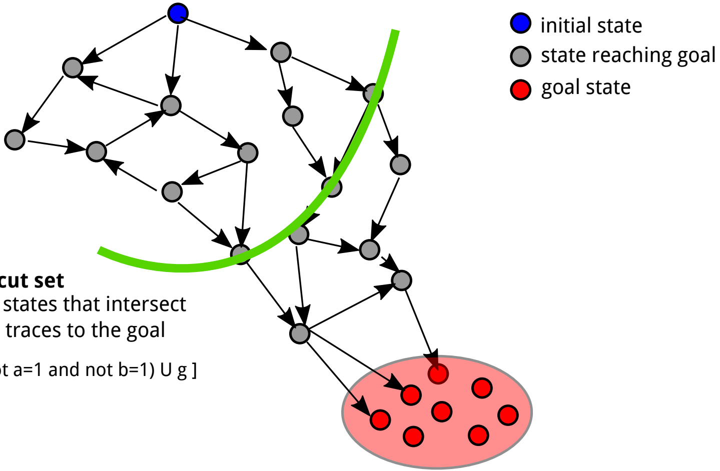
Reachability-related properties

Global state graph



Reachability-related properties

Global state graph

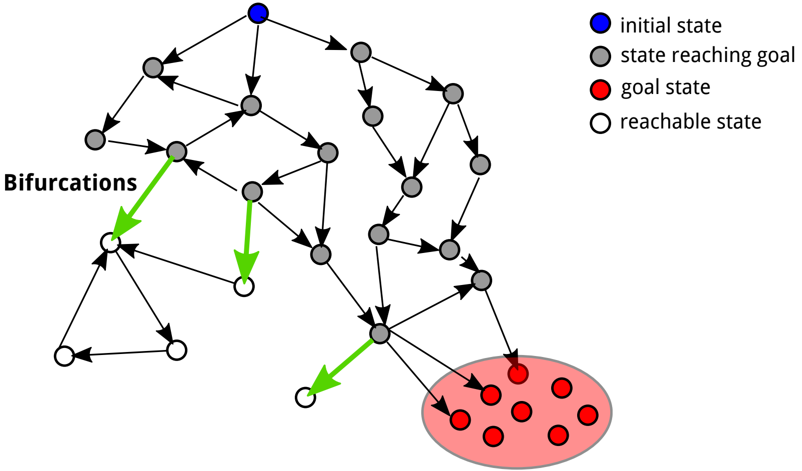


cut set
set of local states that intersect with all the traces to the goal

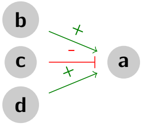
not E [(not a=1 and not b=1) U g]

Reachability-related properties

Global state graph



Challenges for scalability



Modelling issues

- Partially-specified interactions.
- Boolean networks need to be fully specified (deterministic Boolean function f_a).
- Intractable enumeration of all models.

Analysis issues

- Combinatorial explosion of behaviours (e.g. 2^{100} - 10^{30} to $2^{10\,000}$ - 10^{3000} states).
- Reachability is PSPACE-complete
- Large range of control candidate to consider
- Large range of initial conditions to consider

Outline

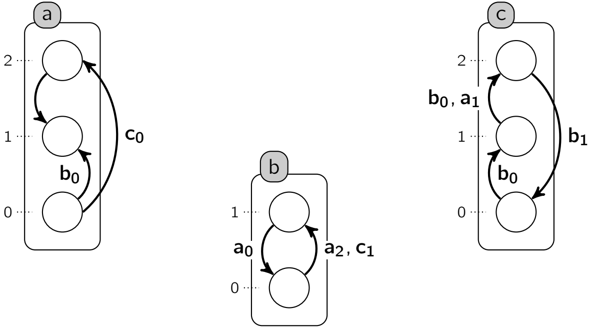
Static analysis for Automata Networks dynamics

- Dedicated to transient reachability analysis
- Highly scalable
- Correct results (over-/under-approximations). . .
- but incomplete.

Menu:

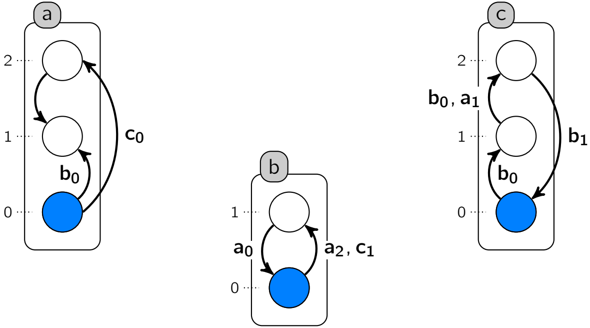
1. Local Causality Analysis
2. Application to reachability-related properties
3. Current/future work

Automata Networks



Asynchronous semantics (one transition at a time):

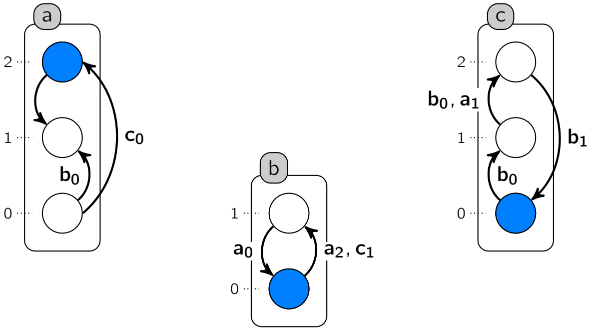
Automata Networks



Asynchronous semantics (one transition at a time):

$\langle a_0, b_0, c_0 \rangle$

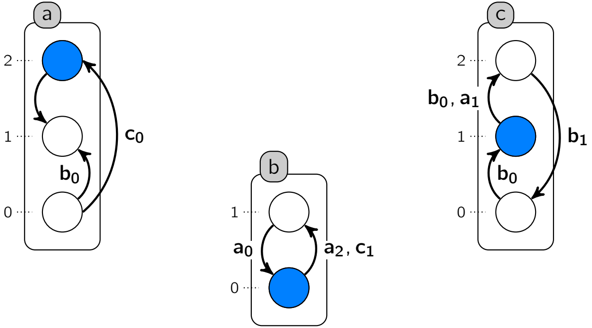
Automata Networks



Asynchronous semantics (one transition at a time):

$$\langle a_2, b_0, c_0 \rangle$$
$$\nearrow$$
$$\langle a_0, b_0, c_0 \rangle$$
$$\searrow$$
$$\langle a_1, b_0, c_0 \rangle$$

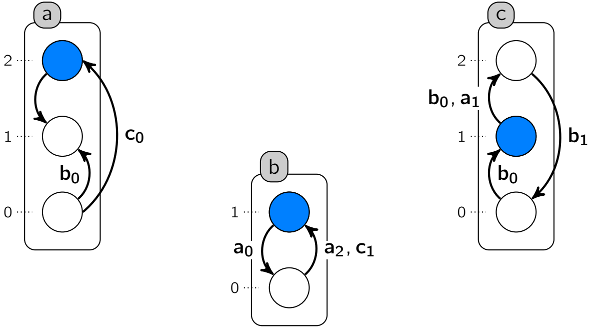
Automata Networks



Asynchronous semantics (one transition at a time):

$$\langle a_2, b_0, c_0 \rangle \longrightarrow \langle a_2, b_0, c_1 \rangle$$
$$\nearrow$$
$$\langle a_0, b_0, c_0 \rangle$$
$$\searrow$$
$$\langle a_1, b_0, c_0 \rangle$$

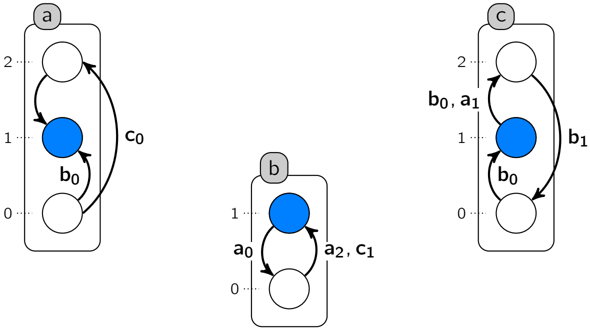
Automata Networks



Asynchronous semantics (one transition at a time):

$$\begin{array}{l} \qquad \langle a_2, b_0, c_0 \rangle \longrightarrow \langle a_2, b_0, c_1 \rangle \longrightarrow \langle a_2, b_1, c_1 \rangle \\ \quad \nearrow \\ \langle a_0, b_0, c_0 \rangle \\ \quad \searrow \\ \qquad \langle a_1, b_0, c_0 \rangle \end{array}$$

Automata Networks



Asynchronous semantics (one transition at a time):

$$
\begin{array}{l}
\qquad \langle a_2, b_0, c_0 \rangle \longrightarrow \langle a_2, b_0, c_1 \rangle \longrightarrow \langle a_2, b_1, c_1 \rangle \longrightarrow \langle a_1, b_1, c_1 \rangle \\
\quad \nearrow \\
\langle a_0, b_0, c_0 \rangle \\
\quad \searrow \\
\qquad \langle a_1, b_0, c_0 \rangle \longrightarrow \cdots
\end{array}
$$

Automata Network modelling of Biological Networks

Transition-centered specification

- .. in opposition to function-centered of Boolean/Thomas networks
- explicit context/causality of state changes
- closely related to (safe) Petri nets
- step semantics (purely async, purely sync, mixed)

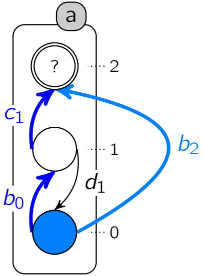
Modelling

- any Boolean/Thomas networks can be encoded;
- in case of logical rules uncertainty: model the union of Boolean/Thomas networks (over-approximation of behaviours)
- encoding of SBGN Process Description models [Rougny et al. BMC Systems Biology, in press] (includes reaction networks, e.g., Biocham models).

Tools

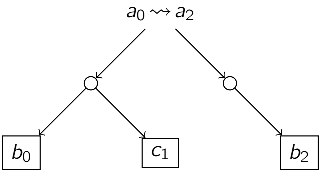
- models can be converted from SBML-qual/GINsim using `logicalmodel` (`https://github.com/colomoto/logicalmodel`)
- analysis using `Pint` (`http://loicpauleve.name/pint`)

Local Causality

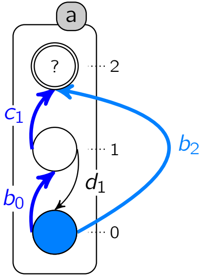


$$local\text{-}paths(a_0 \rightsquigarrow a_2) = \{a_0 \xrightarrow{b_0} a_1 \xrightarrow{c_1} a_2,\ a_0 \xrightarrow{b_2} a_2\}$$

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

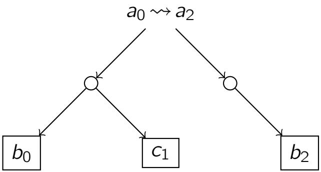


Local Causality



$$local\text{-}paths(a_0 \rightsquigarrow a_2) = \{a_0 \xrightarrow{b_0} a_1 \xrightarrow{c_1} a_2,\ a_0 \xrightarrow{b_2} a_2\}$$

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

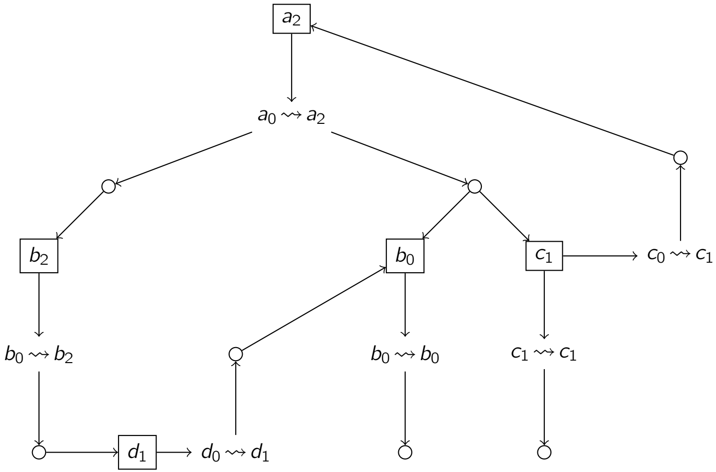


For any trace π starting at some global state s with $a_0 \in s$ and reaching a_2:

- 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_0 and c_1, or b_2 are reached before a_2 in π.

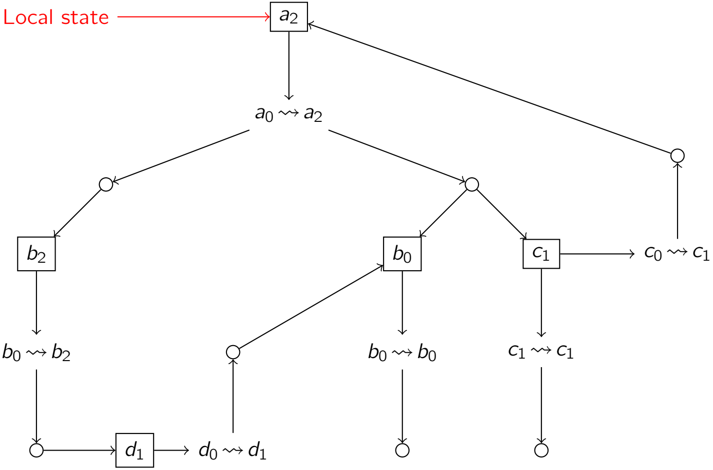
Local Causality Graph

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



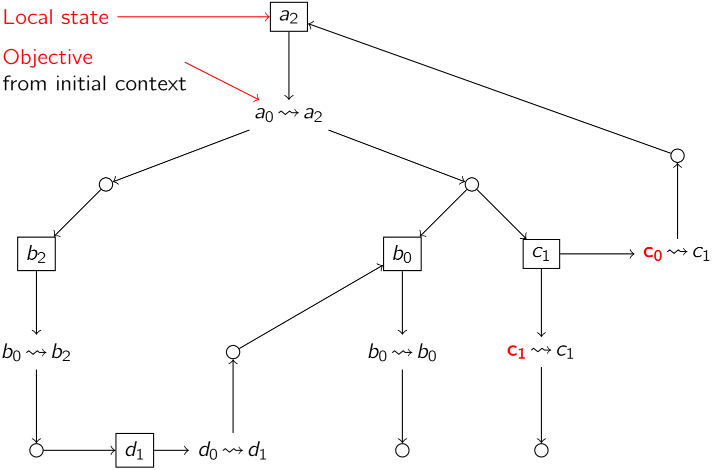
Local Causality Graph

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



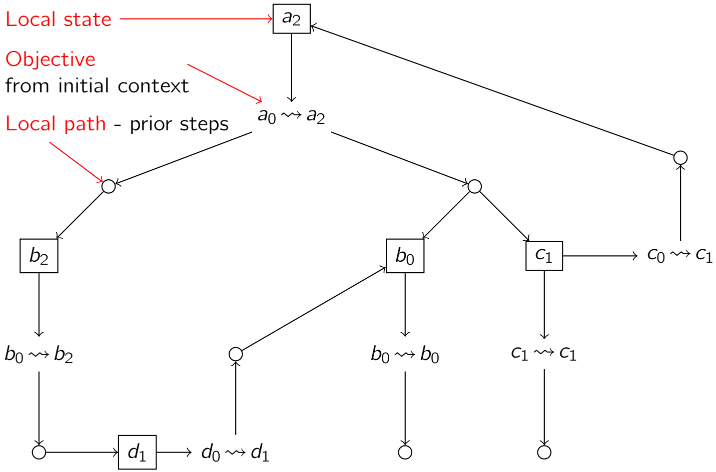
Local Causality Graph

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



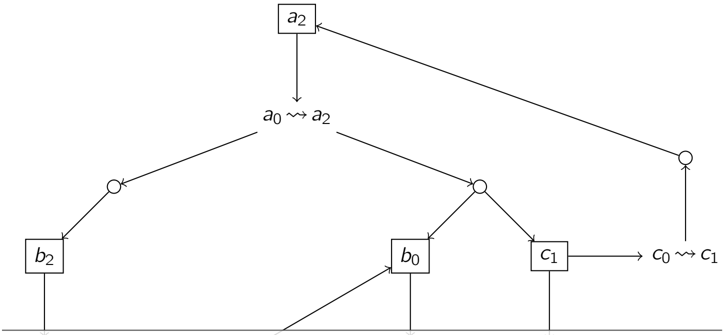
Local Causality Graph

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



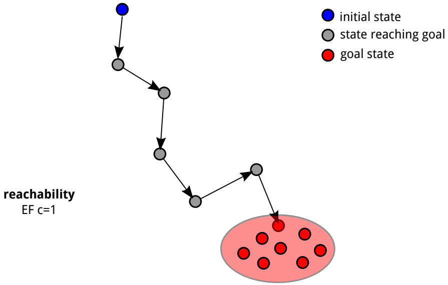
Local Causality Graph

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



Complexity
Nb of objectives: poly(automata size) x nb automata
Nb of local paths: exp(automata size), poly(local transitions)
Usually, automata size is very small (2 for Boolean networks)
$\Rightarrow$ highly tractable for large networks of small automata

Application to reachability



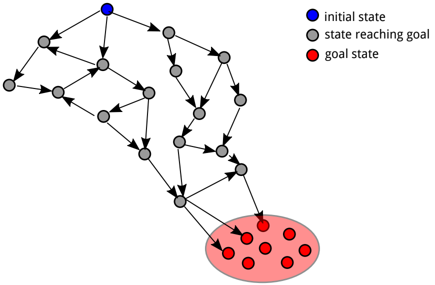
Local causality analysis

- Necessary (OA) or sufficient (UA) conditions

$$\mathrm{UA}(\mathbf{s_0} \to^* \mathbf{s}) \Rightarrow s_0 \to^* s \Rightarrow \mathrm{OA}(\mathbf{s_0} \to^* \mathbf{s})$$

- Model reduction which preserves all minimal traces
- OA: linear with LCG; UA: NP; reduction: linear

Application to reachability



Local causality analysis

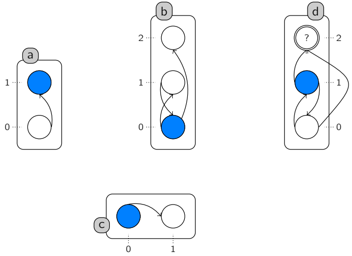
- Necessary (OA) or sufficient (UA) conditions

$$\mathrm{UA}(\mathbf{s_0} \rightarrow^* \mathbf{s}) \Rightarrow s_0 \rightarrow^* s \Rightarrow \mathrm{OA}(\mathbf{s_0} \rightarrow^* \mathbf{s})$$

- Model reduction which preserves all minimal traces
- OA: linear with LCG; UA: NP; reduction: linear

Necessary conditions for reachability

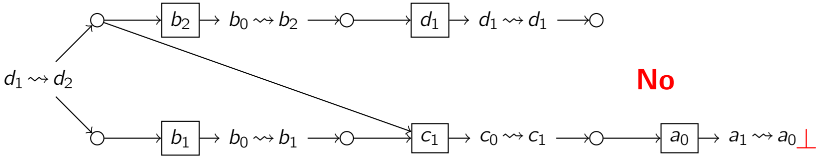
Example



Necessary condition $\text{OA}(s_0 \to^* d_2)$

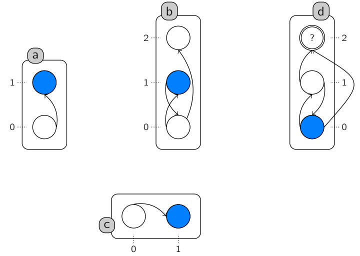
There exists a traversal of the LCG s.t.:

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



Necessary conditions for reachability

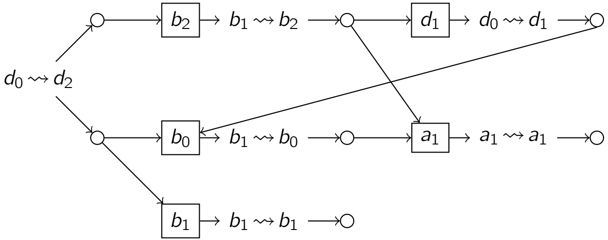
Example



Necessary condition $OA(s_0 \to^* d_2)$

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

- objective $\to$ follow at least one solution;
- local state $\to$ follow all objectives;
- local path $\to$ follow all local states;
- no cycle.



Inconc

UA($s_0 \rightarrow^* s$); goal-oriented reduction

Sufficient condition for reachability UA($s_0 \rightarrow^* s$)

- Based on the Local Causality Graph
- Considers a broader set of objectives
- Simple version linear with LCG; more efficient is NP
  (basic idea: choose a single local path for each objective)

[Paulevé et al, MSCS, 2012; Folschette et al, TCS, 2015]

Goal-oriented model reduction

- Based on the Local Causality Graph;
- Considers a broader set of objectives (than OA and UA)
- Linear with LCG
- Preserves all the minimal traces to the goal, whatever step semantics

[submitted]

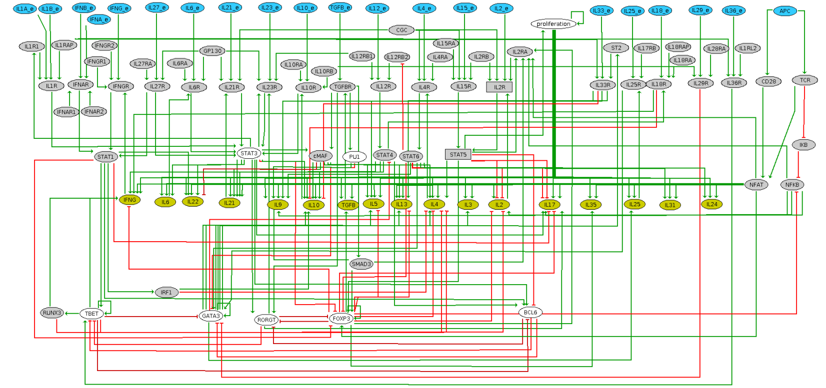
Experiments

For each model

- select an initial state;
- select a goal (activation of a node).

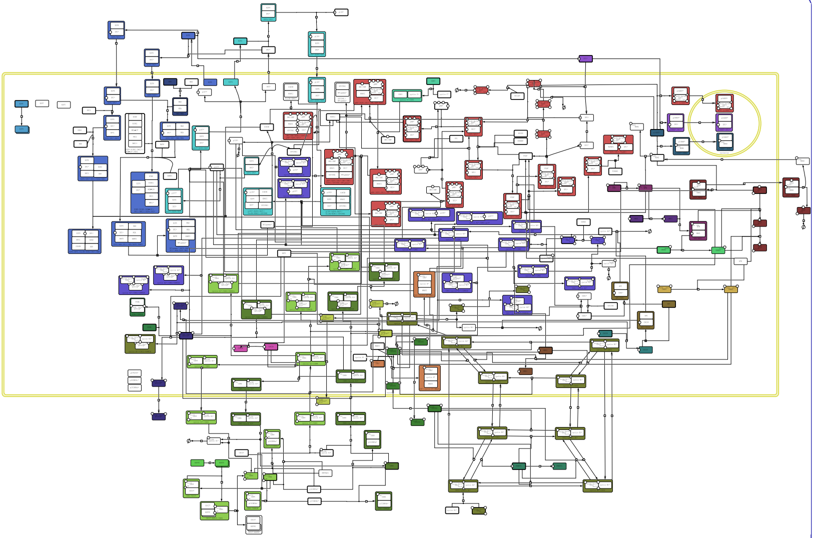
				Verification of goal reachability		
Model	$\lvert T \rvert$	# states	$\lvert$unf$\rvert$	NuSMV (EF g)	its-reach	pint
TCell-d (101) profile 1	384	$\approx 2.7 \cdot 10^8$	257	3s 40Mb	0.5s 24Mb	**0.05s**
TCell-d (101) profile 2	384	KO	KO	KO	0.5s 23Mb	**1.5s**
RBE2F (370)	742	KO	KO	KO	KO	**0.3s**
MAPK-Schoeberl (309)	1251	KO	KO	KO	KO	**90s**

Experiments

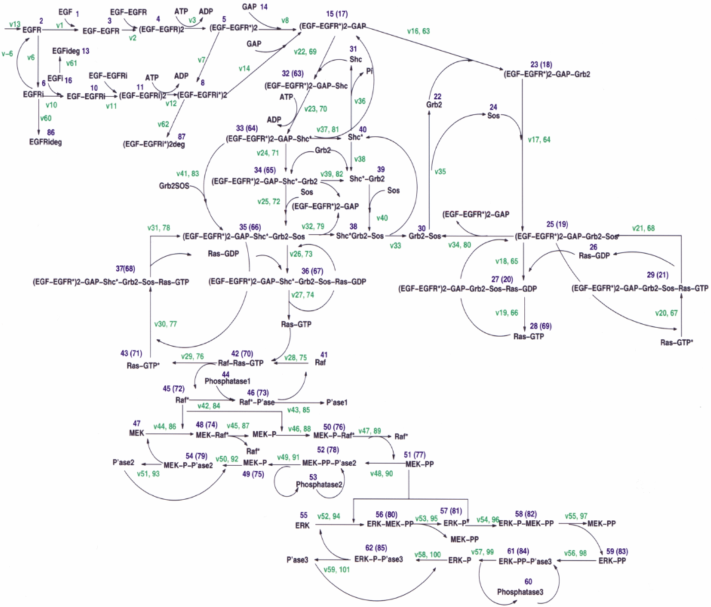


[Abou-Jaoudé et al, Frontiers in Bioengineering and Biotechnology, 2015]

Experiments



[Calzone et al, Mol Syst Biol, 2008]



[Schoeberl et al, Nature Biotechnology, 2002]

Experiments

For each model

- select an initial state;
- select a goal (activation of a node).

				Verification of goal reachability		
Model	$\|T\|$	# states	|unf|	`NuSMV` (EF g)	`its-reach`	`pint`
TCell-d (101) profile 1	384	$\approx 2.7 \cdot 10^8$	257	3s 40Mb	0.5s 24Mb	**0.05s**
TCell-d (101) profile 2	384	KO	KO	KO	0.5s 23Mb	**1.5s**
RBE2F (370)	742	KO	KO	KO	KO	**0.3s**
MAPK-Schoeberl (309)	1251	KO	KO	KO	KO	**90s**

Experiments

For each model

- select an initial state;
- select a goal (activation of a node).

				Verification of goal reachability		
Model	$\lvert T \rvert$	# states	$\lvert$unf$\rvert$	`NuSMV` (EF g)	`its-reach`	`pint`
TCell-d (101)	384	$\approx 2.7 \cdot 10^8$	257	3s 40Mb	0.5s 24Mb	**0.05s**
profile 1	**0**	**1**	**1**			
TCell-d (101)	384	KO	KO	KO	0.5s 23Mb	**1.5s**
profile 2	**161**	**75,947,684**	**KO**	**474s 260Mb**	**0.3s 19Mb**	
RBE2F (370)	742	KO	KO	KO	KO	**0.3s**
	56	**2,350,494**	**28,856**	**5s 377Mb**	**5s 170Mb**	
MAPK-Schoeberl	1251	KO	KO	KO	KO	**90s**
(309)	**429**	**KO**	**KO**	**KO**	**KO**	

In all cases, reduction step took less than 0.1s

Experiments

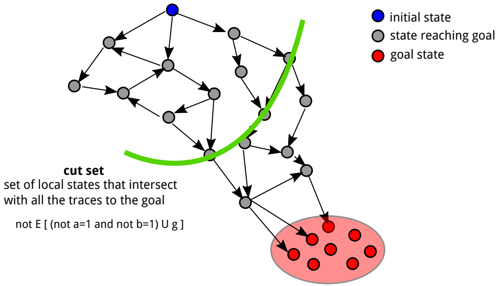
Goal-oriented reduction

Goal-oriented LTL/CTL model checking

- requires all the minimal traces
- cut set verification: `not E [` $(a \neq i \wedge b \neq j)$ `U` g `]`

	Wnt (32)	TCell-r (40)	EGF-r (104)	TCell-d (101)	RBE2F (370)
`NuSMV`	44s 55Mb **9.1s 27Mb**	KO **2.4s 34Mb**	KO **13s 33Mb**	KO **600s 360Mb**	KO **6s 29Mb**
`its-ctl`	105s 2.1Gb **16s 720Mb**	492s 10Gb **11s 319Mb**	KO **21s 875Mb**	KO **KO**	KO **179s 1.8Gb**

Application to cut sets



Cut set for g from s_0

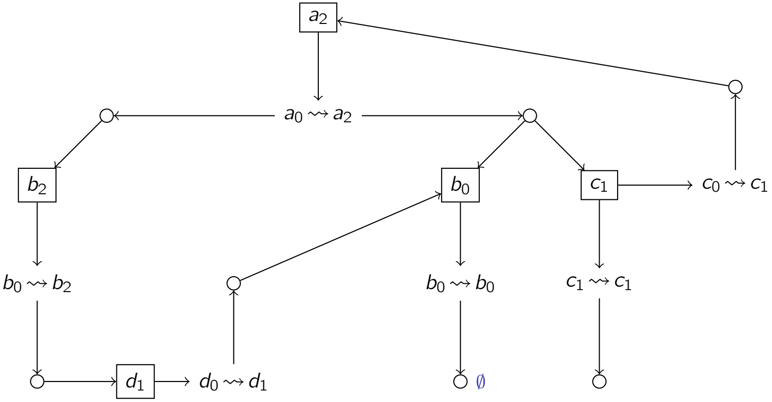
- Set of local states C such that g is not reachable if all transitions involving a local state in C are removed.

From Local Causality Graph

- Direct computation of cut sets (no enumeration of candidates)
- Under-approximation: some may be missed.

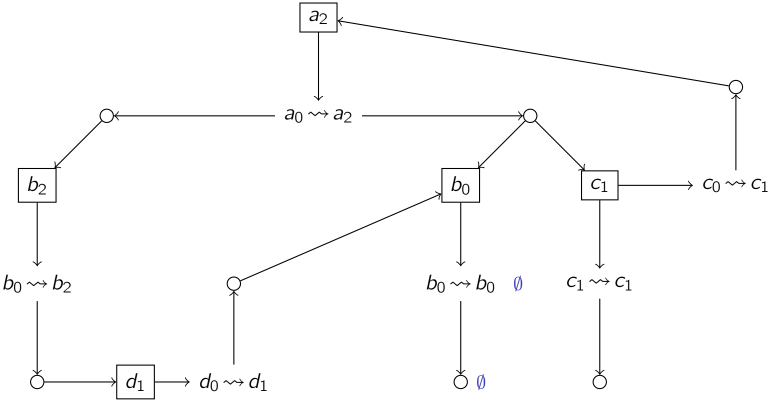
Cut Sets Under-approximation

Example



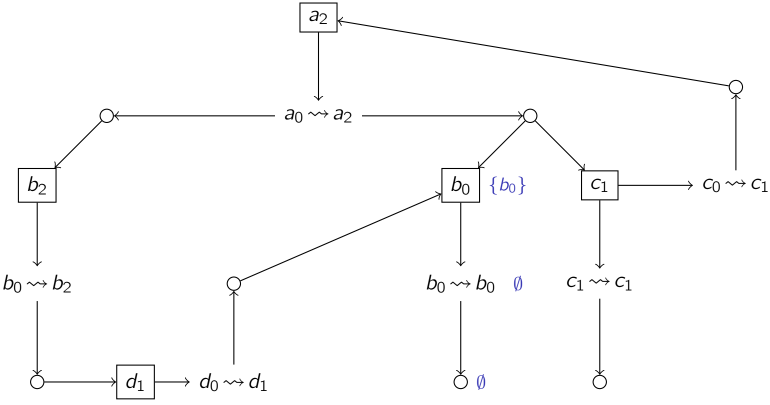
Cut Sets Under-approximation

Example



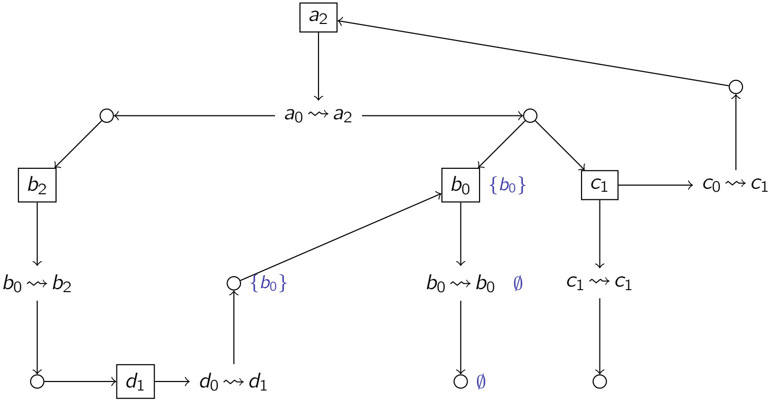
Cut Sets Under-approximation

Example



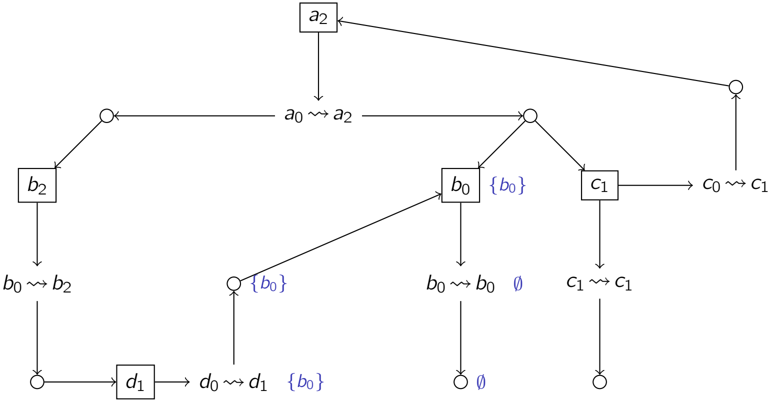
Cut Sets Under-approximation

Example



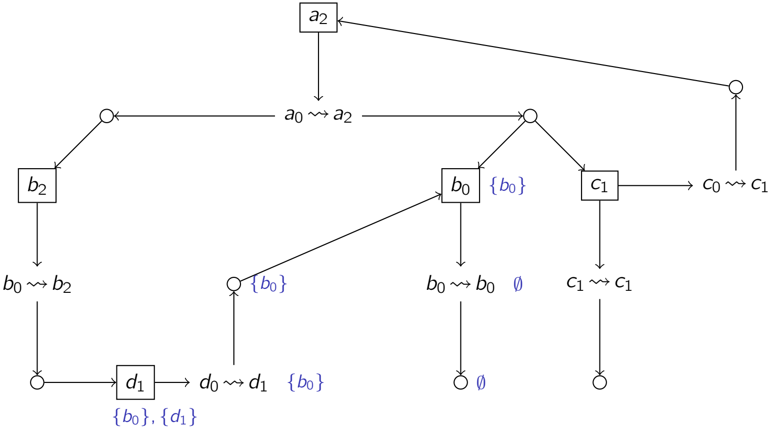
Cut Sets Under-approximation

Example



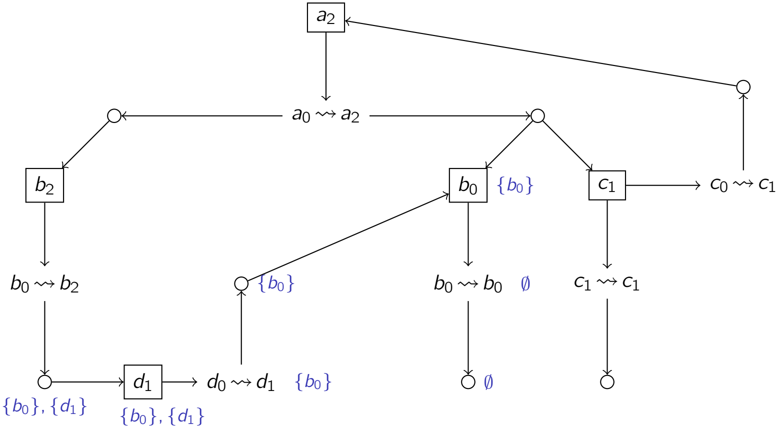
Cut Sets Under-approximation

Example



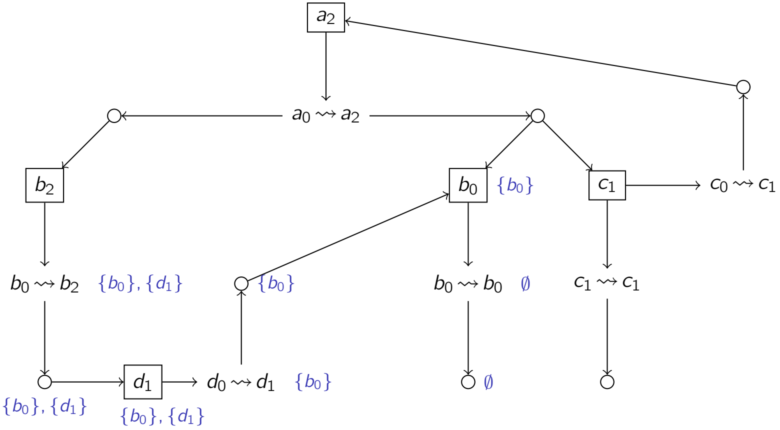
Cut Sets Under-approximation

Example



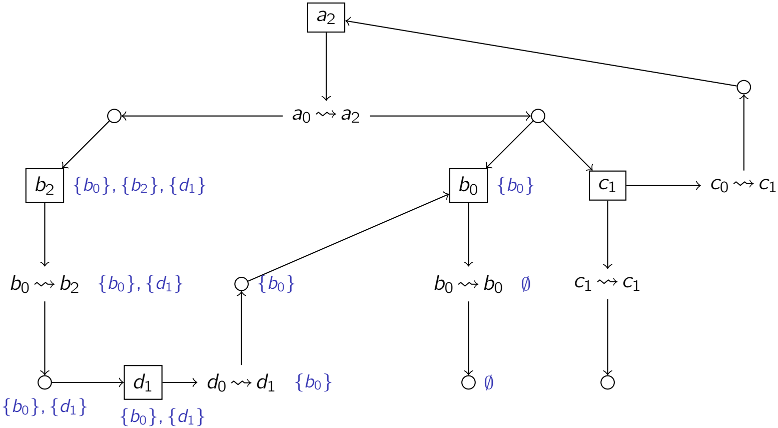
Cut Sets Under-approximation

Example



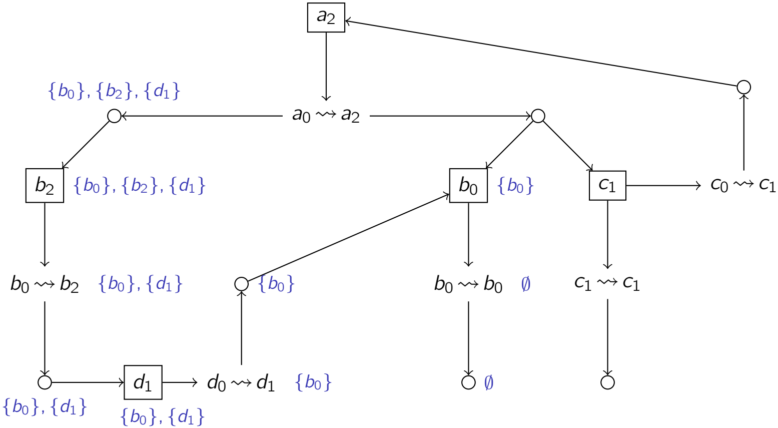
Cut Sets Under-approximation

Example



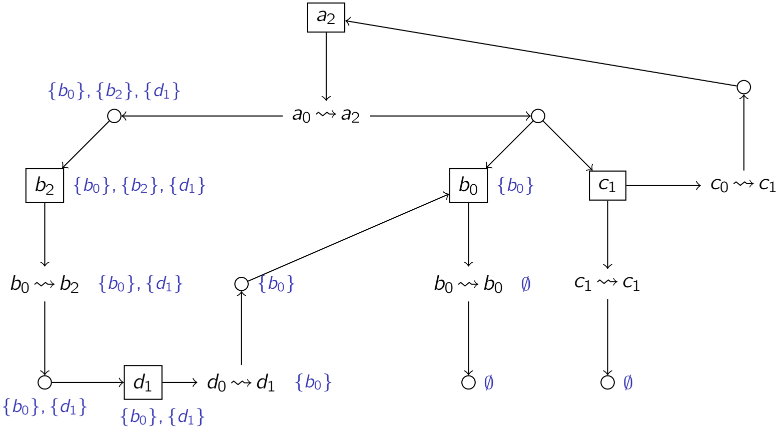
Cut Sets Under-approximation

Example



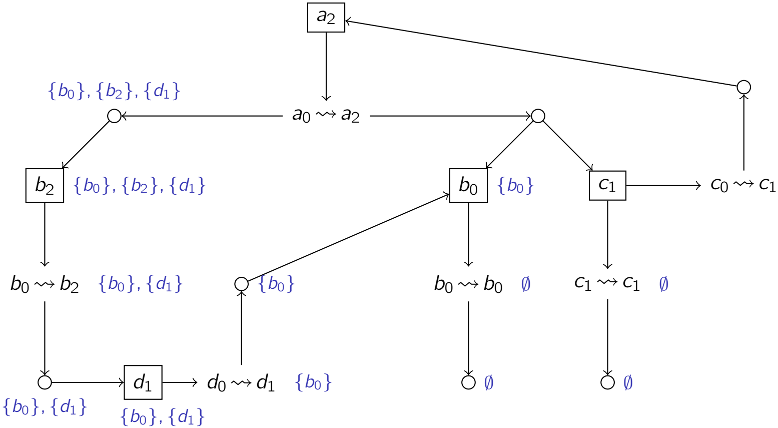
Cut Sets Under-approximation

Example



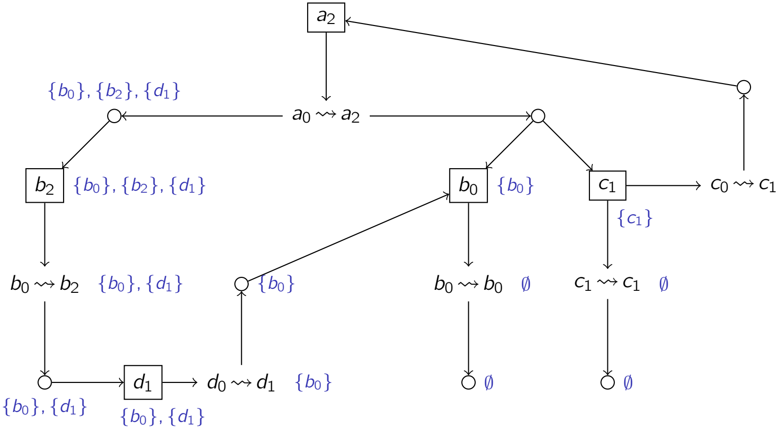
Cut Sets Under-approximation

Example



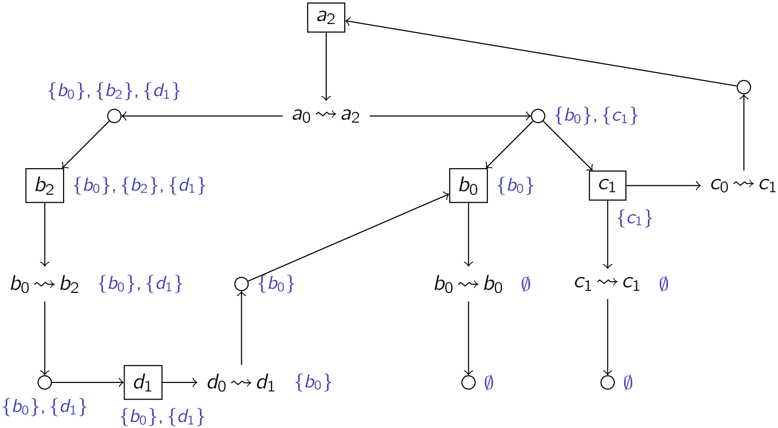
Cut Sets Under-approximation

Example



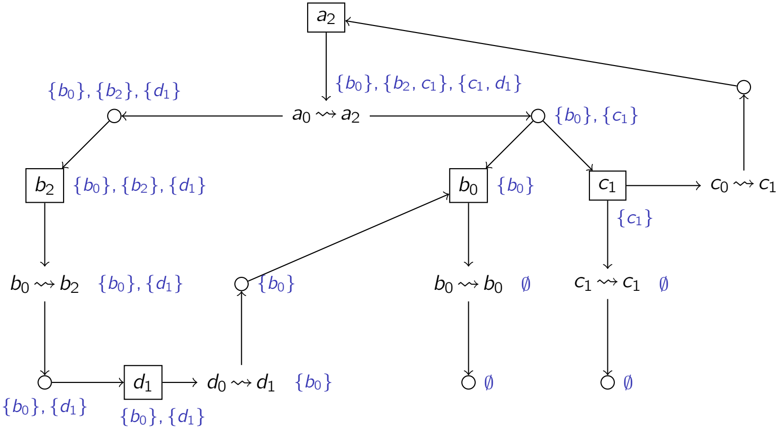
Cut Sets Under-approximation

Example



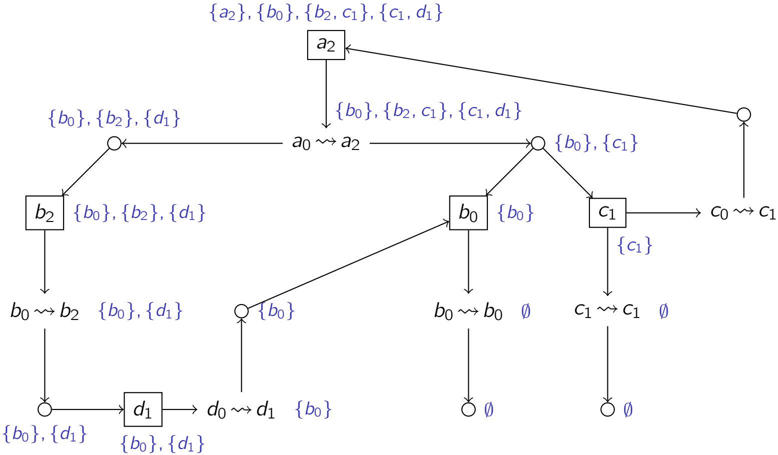
Cut Sets Under-approximation

Example



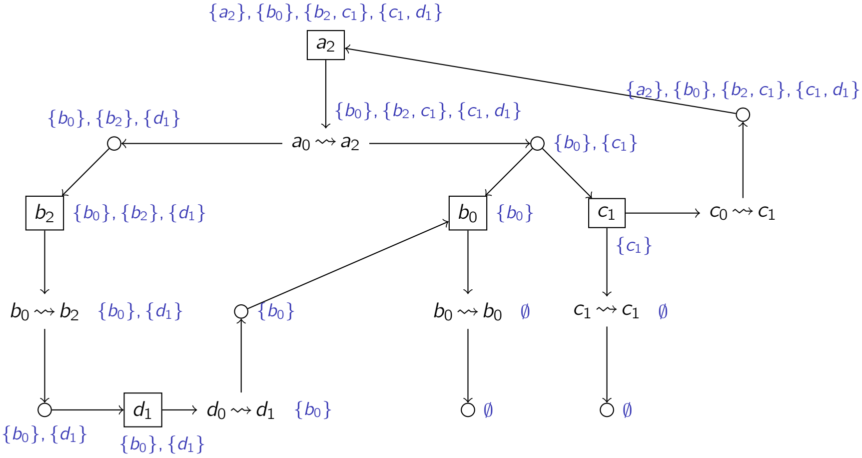
Cut Sets Under-approximation

Example



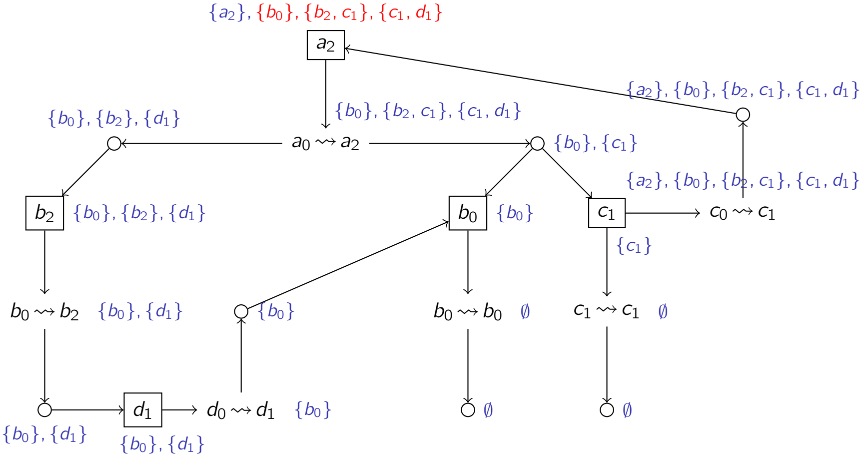
Cut Sets Under-approximation

Example



Cut Sets Under-approximation

Example



Experiments

OCaml implementation (`Pint`)

```
pint-reach --cutsets N -i model.an g=1
```

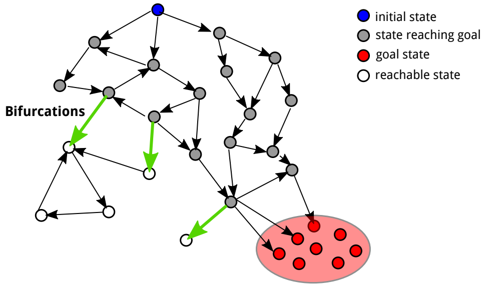
N-cut sets: cut sets of cardinality at most N.

	TCell-d (101)	RBE2F (370)	MAPK-Schoeberl (309)	PID (21,000)
4-cut sets	0.03s (27)	0.06s (57)	0.1s (34)	39s (37)
6-cut sets	0.03s (27)	0.76s (334)	0.5s (43)	2.6h (1257)

[Paulevé et al at CAV 2013]

To be benchmarked: SAT implementation.

Application to bifurcations



Bifurcation from s_0 to g

- local transition (e.g. $t_b = c_0 \xrightarrow{a_0, b_1} c_1$)
- $s_0 \to^* s_b \to^* g$; and $s_b \cdot t_b \not\to^* g$.
- relaxed: $\mathrm{UA}(s_0 \to^* s_b)$, $\mathrm{UA}(s_b \to^* g)$, $\neg\,\mathrm{OA}(s_b \cdot t_b \to^* g)$
  $\Rightarrow$ **under-approximation** of bifurcations

Application to bifurcations

Implementation

Given s_0 and g (goal), find s_b and t_b such that:

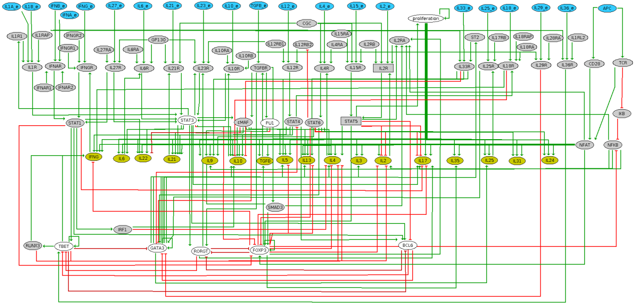
$$\mathrm{UA}(s_0 \rightarrow^* s_b) \wedge \mathrm{UA}(s_b \rightarrow^* g) \wedge \neg\, \mathrm{OA}(s_b \cdot t_b \rightarrow^* g)$$

- OA and UA can be implemented in SAT;
- when tractable, $\mathrm{UA}(s_0 \rightarrow^* s_b)$ can be replaced with an exact checking (e.g., prefix of unfolding).
- we used ASP solver (convenient input language).

[submitted]

Application to bifurcations

Experiments



[Abou-Jaoudé et al, Frontiers in Bioengineering and Biotechnology, 2015]

101 automata, 381 local transitions

s_0	Goal	Nb states	$s_0 \to^* g$			$UA(s_0 \to^* g)$	
			$\lvert\text{unf-prefix}(s_0)\rvert$	$\lvert t_b\rvert$	Time	$\lvert t_b\rvert$	Time
th17	RORGT_1	$\approx 4 \cdot 10^9$	2860	9	23.9s	8	29.04s
	BCL6_1			5	26.2s	4	26.64s
HTG	BCL6_1	KO	KO	-	-	6	61.9s
	GATA3_1			-	-	7	34.16s

Conclusion

Static analysis for transient reachability

- Scalable to large networks of small automata.
- Applications to reachability, cut sets, bifurcations.
- Model reduction which preserves all traces to a given goal.

Step semantics

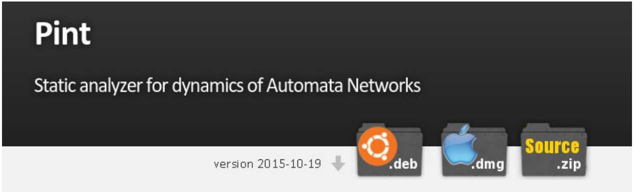
- Reachability over-approximation, cut sets, and model reduction work for most step semantics (async, sync, mixed).
- Reachability under-approximation works only if async transitions are possible.

Comments

- Gives correct, but incomplete results.
- Exploits the low scope of transitions in logical networks: each local transitions depend on a few automata (same apply for the goal).

Software: Pint

http://loicpauleve.name/pint



- Input: automata networks
  - convert SBML-qual/GINsim with LogicalModels[1]
  - scripts for CellNetAnalyser, Biocham, etc.
- Command line tools:
  - Static analysis for reachability, cut sets, fixed points
  - Model reduction w.r.t. reachability property
  - Inference of Interaction graph/Thomas parameters
  - Interface with model-checkers (NuSMV, ITS, mole).
- OCaml library (possible C/C++ bindings)

[1] https://github.com/colomoto/logicalmodel

Current/future work

Reducing approximations

- Coupling with Petri net unfoldings
- LCG and unfoldings exploit concurrency
- Adapt algorithms to unfoldings: higher complexity, but complete results.

Towards general cell reprogramming

- Predicts mutations/perturbations to trigger an attractor change
- Reachability properties are central
- Large number of candidates, hopefully restrictable by
  - topology
  - unfolding
  - LCG

Towards the logical prediction of control targets for biological networks

Loïc Paulevé

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

Open University Digicosme - 2016, May 18