Around Reachability in Automata Networks

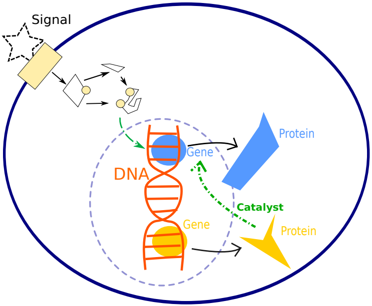
Loïc Paulevé

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

Journées annuelles GT BIOSS - 1-2 Juillet 2016

CNRS MASTODONS

Reachability in models of biological networks



Validation

- Ability to reproduce data

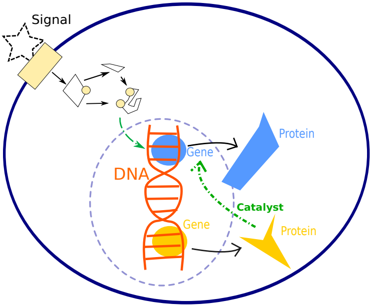
Prediction

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

Control

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

Reachability in models of biological networks



Validation

- Ability to reproduce data

Prediction

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

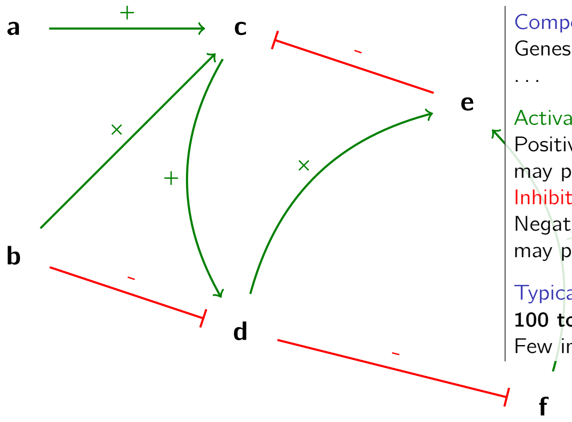
Control

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

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

Influence networks

E.g., Signalling Networks, Gene Regulatory Networks



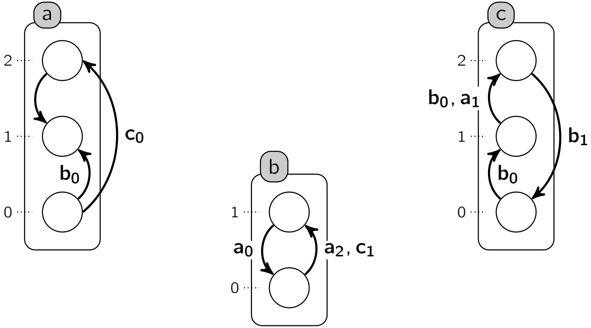
Components
Genes, proteins, complexes, ...

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

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

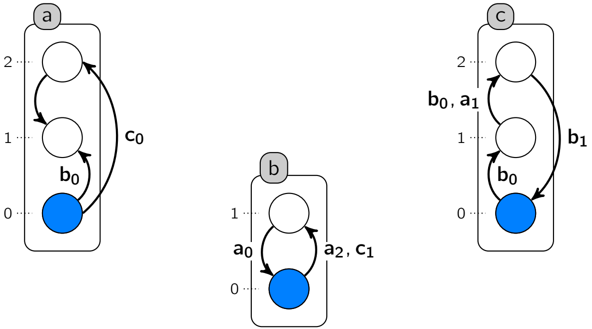
Typical settings
100 to +10,000 components
Few information on kinetics

Automata Networks



Asynchronous semantics (one transition at a time):

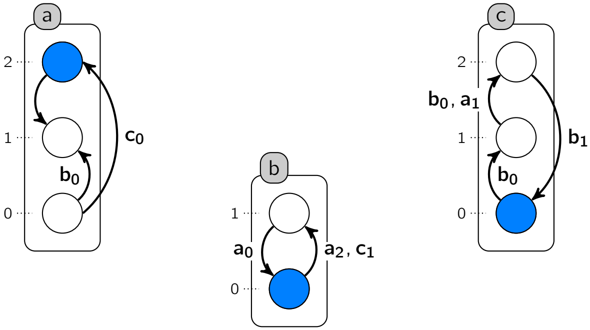
Automata Networks



Asynchronous semantics (one transition at a time):

$\langle a_0, b_0, c_0 \rangle$

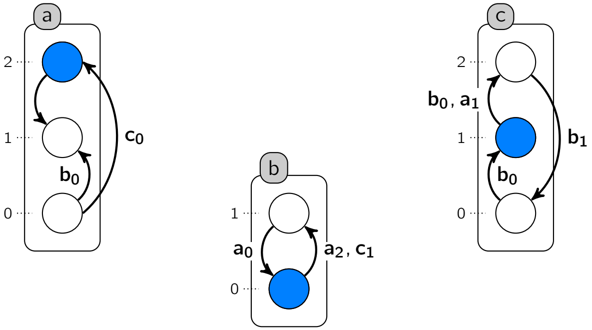
Automata Networks



Asynchronous semantics (one transition at a time):

$$\begin{array}{ll} & \langle a_2, b_0, c_0\rangle \\ & \nearrow \\ \langle a_0, b_0, c_0\rangle & \\ & \searrow \\ & \langle a_1, b_0, c_0\rangle \end{array}$$

Automata Networks



Asynchronous semantics (one transition at a time):

$$\langle a_2, b_0, c_0\rangle \longrightarrow \langle a_2, b_0, c_1\rangle$$

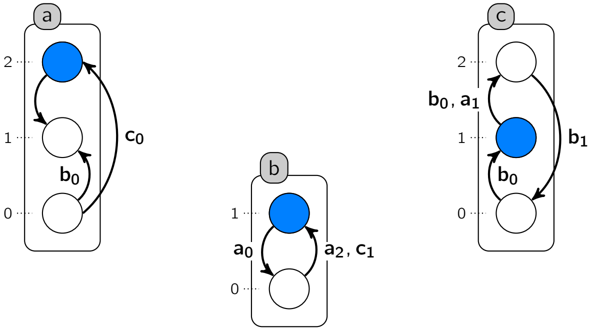
$$\nearrow$$

$$\langle a_0, b_0, c_0\rangle$$

$$\searrow$$

$$\langle a_1, b_0, c_0\rangle$$

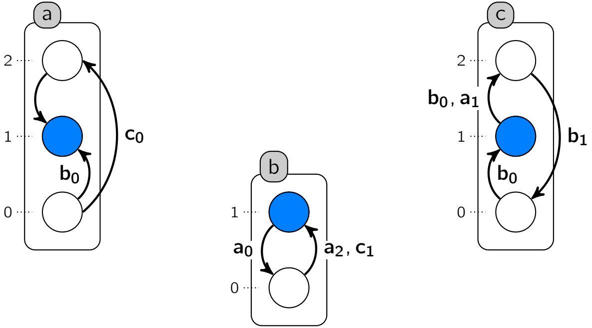
Automata Networks



Asynchronous semantics (one transition at a time):

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

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`)

Motivations

Bad news: reachability is PSPACE-complete.

Motivation

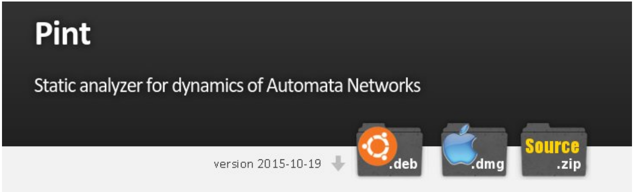
- Identify control targets without exact verification
- Model reduction preserving reachability

Approach

- Necessary/sufficient conditions with lower complexity
- Relax completeness, ensure correctness
  Under-approximation: no false positive, but false negatives
- Relax correctness, ensure completeness
  Over-approximation: no false negative, but false positives

Software: Pint

http://loicpauleve.name/pint



- Input: automata networks
  - convert SBML-qual/GINsim with `LogicalModels`
  - 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)

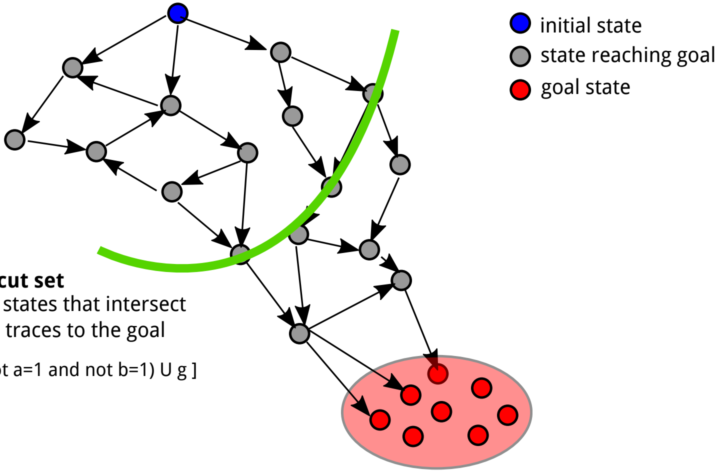
```
model.an:
a [0, 1]
b [0, 1, 2]
c [0, 1]

a 0 -> 1 when b=0 and c=1
a 1 -> 0 when b=1
a 1 -> 0 when b=2
a 1 -> 0 when c=0

b 0 -> 1 when a=1
b 1 -> 2 when a=1
```

Cut sets for (transient) reachability

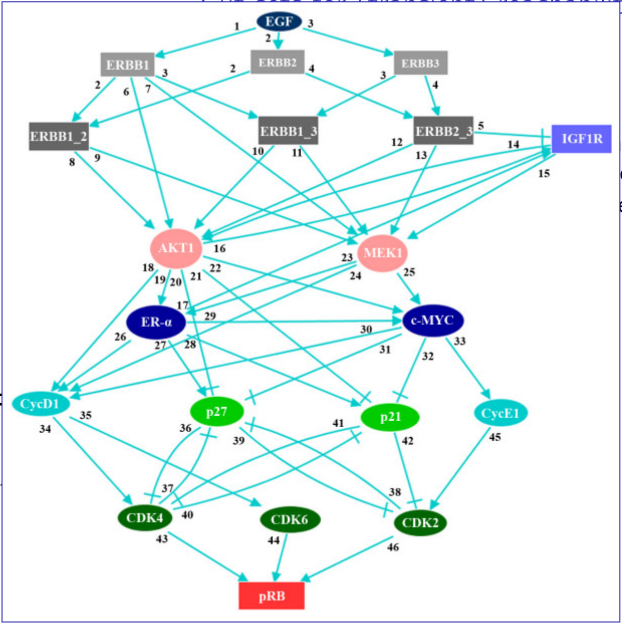
Global state graph



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]

Cut sets for (transient) reachability



te

ching goal

e

c

set of local

with all the

not E [(no

Cut sets for (transient) reachability

Experiments

Under-approximation of N-cut sets (cardinality at most N)

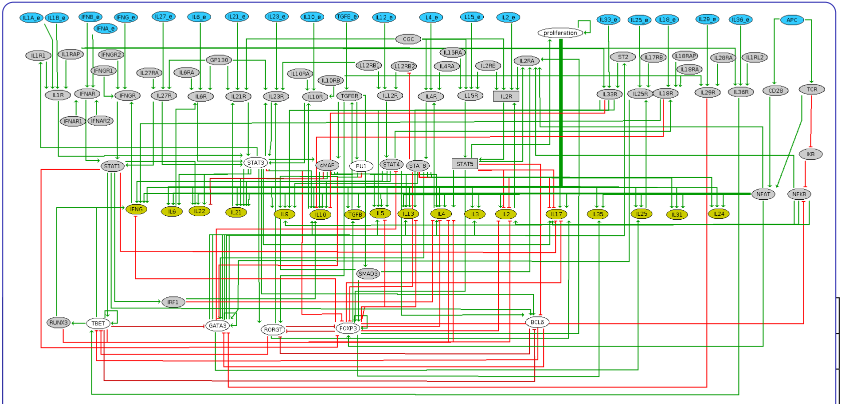
```
$ pint-reach --cutsets 4 --no-init-cutsets -i TCell-d.an BCL6=1
"GP130"=1
"STAT3"=1
"CD28"=1,"IL6R"=1
...
"IL6RA"=1,"TCR"=1
```

	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]

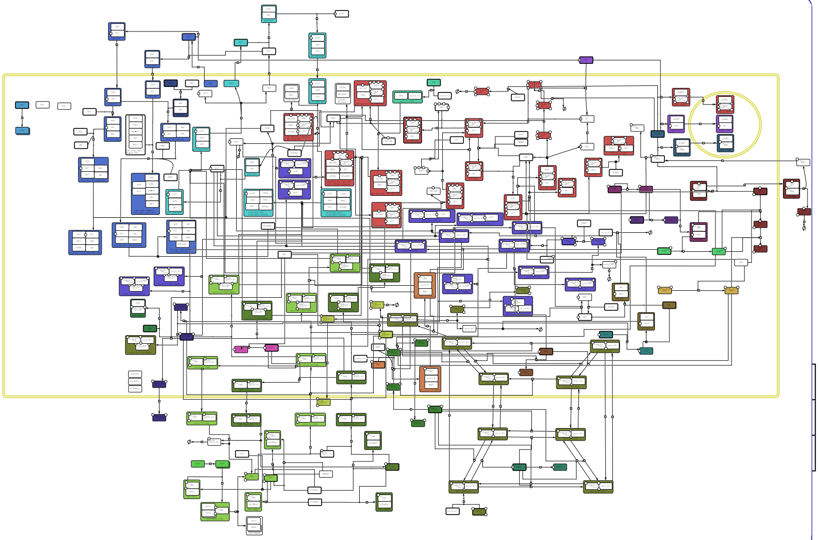
Cut sets for (transient) reachability

Experiments

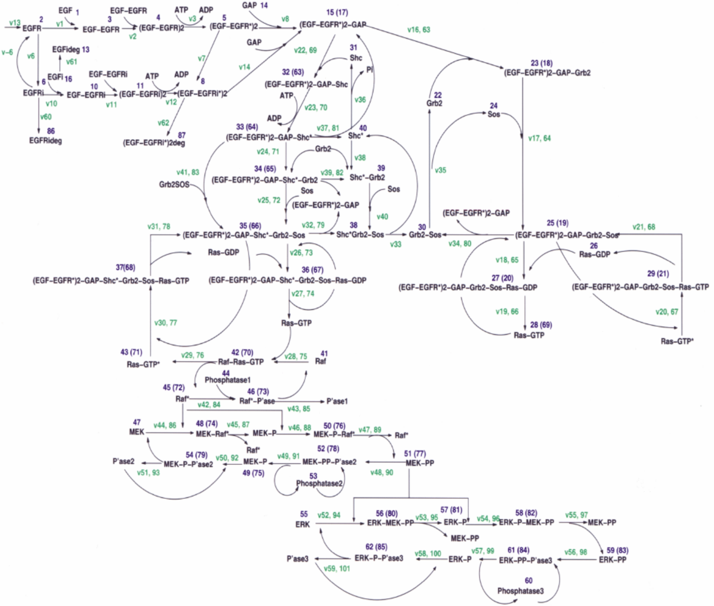


[Abou-Jaoudé et al, Frontiers in Bioengineering and Biotechnology, 2015]

Cut sets for (transient) reachability



[Calzone et al, Mol Syst Biol, 2008]



[Schoeberl et al, Nature Biotechnology, 2002]

Cut sets for (transient) reachability

Experiments

Under-approximation of N-cut sets (cardinality at most N)

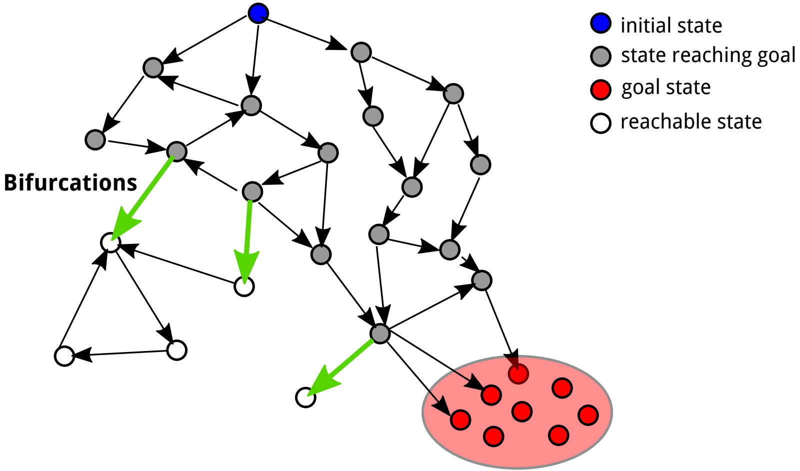
```
$ pint-reach --cutsets 4 --no-init-cutsets -i TCell-d.an BCL6=1
"GP130"=1
"STAT3"=1
"CD28"=1,"IL6R"=1
...
"IL6RA"=1,"TCR"=1
```

	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)

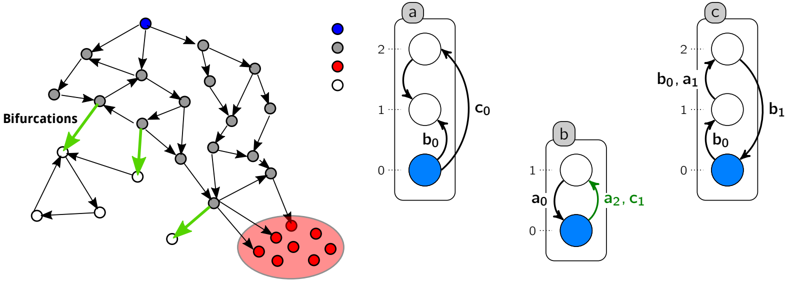
[Paulevé et al at CAV 2013]

Bifurcations for reachability

Global state graph



Bifurcations for reachability

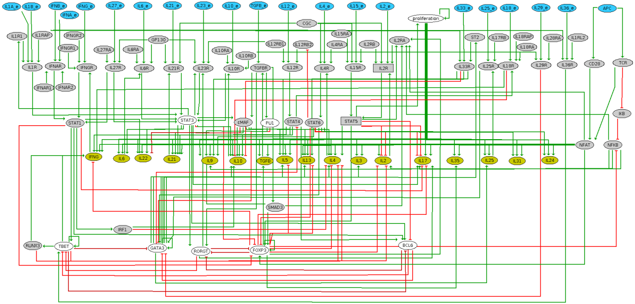


- Identify when and how a system loses a capability.
- Not for control as is: bifurcation transition may be necessary for goal.
- ASP implementation combining necessary and sufficient conditions on reachability:
  Under-approximation of bifurcations; complexity: NP

Joint work with L. F. Fitime, C. Guziolowski, O. Roux [WCB'16]

Bifurcations for reachability

Experiments

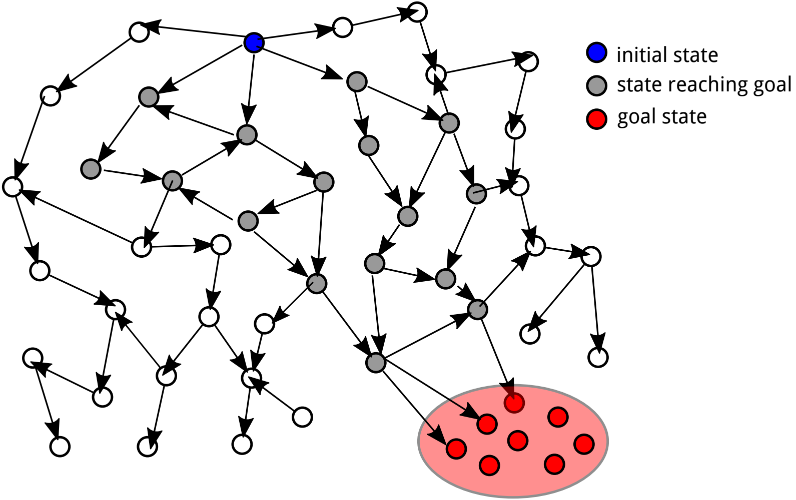


[Abou-Jaoudé et al, Frontiers in Bioengineering and Biotechnology, 2015]

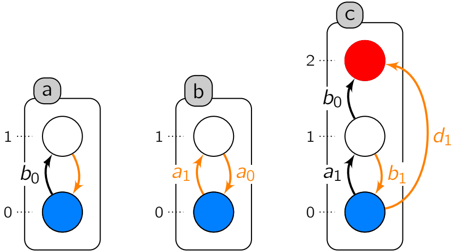
101 automata, 381 local transitions

s_0	Goal	Nb states	$\lvert t_b \rvert$	Time
th17	RORGT_1	$\approx 4 \cdot 10^9$	8	$29.04s$
	$\mathrm{BCL6}_1$		4	$26.64s$
HTG	$\mathrm{BCL6}_1$	KO	6	$61.9s$
	$\mathrm{GATA3}_1$		7	$34.16s$

Goal-oriented reduction



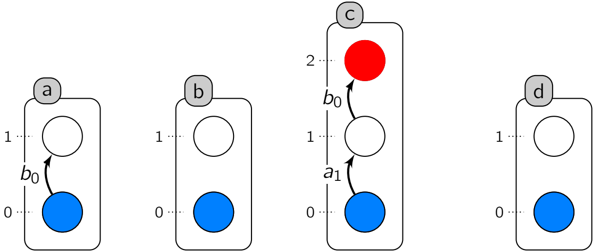
Goal-oriented reduction



- Preserves all minimal traces to goal
- Low complexity: poly. with local transition, exp. with nb states per automaton (typically 2 to 4)

[Paulevé at CMSB 2016]

Goal-oriented reduction



- Preserves all minimal traces to goal
- Low complexity: poly. with local transition, exp. with nb states per automaton (typically 2 to 4)

[Paulevé at CMSB 2016]

Goal-oriented reduction

Experiments

For each model: select an initial state; select a goal (activation of a node).

```
$ pint-export -i model.an --reduce-for-goal g=1 -o reduced.an
$ pint-nusmv -i reduced.an g=1
```

			Verification of goal reachability			
Model	$\|T\|$	# states	NuSMV (EF g)		its-reach	
VPC (88)	332	KO	KO		1s	50Mb
	219	$\mathbf{1.8 \cdot 10^9}$	**236s**	**156Mb**	**0.8s**	**21Mb**
TCell-d (101)	384	$\approx 2.7 \cdot 10^8$	3s	40Mb	0.5s	24Mb
profile 1	**0**	**1**				
TCell-d (101)	384	KO	KO		0.5s	23Mb
profile 2	**161**	**75,947,684**	**474s**	**260Mb**	**0.3s**	**19Mb**
EGF-r (104)	378	$\approx 2.7 \cdot 10^{16}$	KO		1.36s	60Mb
	69	**62,914,560**	**11s**	**33Mb**	**0.3s**	**17Mb**
RBE2F (370)	742	KO	KO		KO	
	56	**2,350,494**	**5s**	**377Mb**	**5s**	**170Mb**

In all cases, reduction step took less than 0.1s

Goal-oriented reduction

Experiments

Verification of cut sets

- requires all the minimal traces
- cut set verification: `not E [ (`$a \neq 1 \wedge b \neq 1$`) U` $g = 1$ `]`

```
$ pint-export -i model.an --reduce-for-goal g=1 -o reduced.an
$ pint-nusmv -i reduced.an --is-cutset a=1,b=1 g=1
```

	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**

Conclusion

Summary

- Automata networks (generalize logical networks); asynchronous semantics
- Avoid concrete reachability verification.
- Use to a variety of bio-inspired problems: reachability, cut sets, bifurcations
- Tractable on very large networks with guaranteed results.

Methodology

Intuition: exploit the low scope of transitions in logical networks.

- Static analysis by abstract interpretation
- Intermediate representation (Local Casuality Graph) to reason on necessary/sufficient conditions
- Implementation mixes algorithms on graphs and SAT (ASP).

What's next

- Over-approximation of cut sets, bifurcations.
- Move towards completeness: embed in exact approaches (notably unfoldings, model-checking).
- Use for predicting targets for cellular reprogramming.

Around Reachability in Automata Networks

Loïc Paulevé

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

Journées annuelles GT BIOSS - 1-2 Juillet 2016

CNRS MASTODONS