Teatime capsid:

Automatic transformation from reaction models to Boolean models using answer-sets constrained by a topology and an abstracted dynamic.

Athénaïs Vaginay

2020 january 23rd (enfin \o/ :D)

Teatime capsid:

Automatic transformation from **reaction models** to **Boolean models** using **answer-sets** constrained by a topology and an abstracted dynamic.

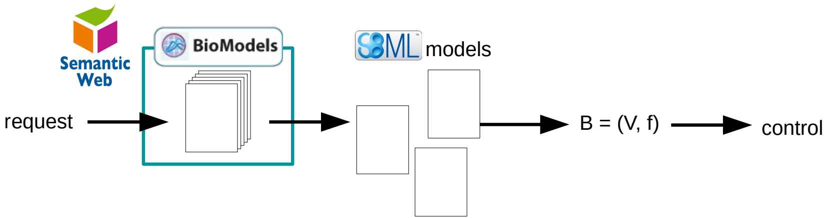
Athénaïs Vaginay

2020 january 23rd (enfin \o/ :D)

what did we talked about one year ago ?

(time flies...)

Project Thesis Overview (which i like sooooo much ♡)



as automatized as possible... ;-)

Modèle : définition du Dictionnaire de l'Académie française

Modèle, n.m., XVIe siècle, *modelle*. Emprunté de l'italien *modello*, de même sens.
4. Sciences. Représentation physique, graphique ou, plus généralement, mathématique qui formalise les relations unissant les différents éléments d'un système, d'un processus, d'une structure, en vue de faciliter la compréhension de certains mécanismes ou de permettre la validation d'une hypothèse. [. . .]

(To find your way around a city, you do not need to have a map at scale 1:1. . .)

https://www.dictionnaire-academie.fr/article/A9M2435

Biological levels

Many levels in biology → multi scale biological models.

vertical hierarchy

biosphere
ecosystem
community
population
organism
organ
tissue
→ cell ←
macromolecule
molecule
atom

horizontal modularity
(signaling, gene regulation, metabolism)

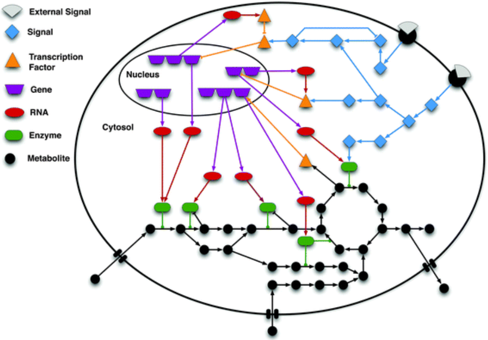


Figure: Gonçalves et al. "Bridging the Layers: Towards Integration of Signal Transduction, Regulation and Metabolism into Mathematical Models." Molecular BioSystems 9, no. 7 (2013): 1576. https://doi.org/10.1039/c3mb25489e

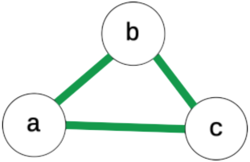
modelling framework

a LOT of formalisms

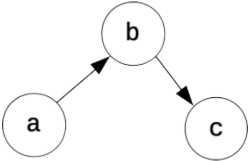
- statistical model
- mechanistic models

formalisms — statistical models

based on correlations, undirectional.

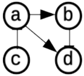


for an underlying mechanism that looks like:

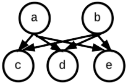


formalisms — mechanistic models

Boolean network



Bayesian network

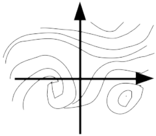


Process algebras

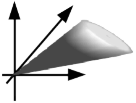
((b(x,de)[E]) || (B(y, dI)[I]))

bh(x, dE) bh(y, dI) (E || I)

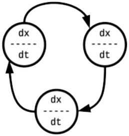
Differential equation



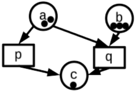
Constraint based model



Hybrid systems



Petri Nets



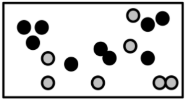
Interacting state machine

Compartment based

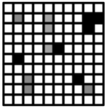
Rule based

...

Agent-based model



Cellular automata



Comparison of 2 formalisms:

differentials equation + events

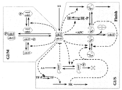


FIG. 2. The wiring diagram of the fission-yeast cell-cycle engine. In the middle of the diagram is Cdc2/Cdc13 (MPF), which is regulated by proteolysis of the Cdc13 component, phosphorylation of Cdc2 subunit, and stoichiometric inhibition of the complex. These processes are arranged according to the cell cycle transitions in which they are involved.

TABLE I. The differential and algebraic equations describing the wiring diagram in Fig. 2.

$$\frac{d[\mathrm{Cdc13}_T]}{dt} = k_1 M - (k_2' + k_2''[\mathrm{Ste9}] + k_2'''[\mathrm{Slp1}])[\mathrm{Cdc13}_T], \quad (1)$$

$$\frac{d[\mathrm{preMPF}]}{dt} = k_{wee}([\mathrm{Cdc13}_T] - [\mathrm{preMPF}]) - k_{25}[\mathrm{preMPF}] - (k_2' + k_2''[\mathrm{Ste9}] + k_2'''[\mathrm{Slp1}])[\mathrm{preMPF}], \quad (2)$$

$$\frac{d[\mathrm{Ste9}]}{dt} = (k_3' + k_3''[\mathrm{Slp1}])\frac{1-[\mathrm{Ste9}]}{J_3 + 1 - [\mathrm{Ste9}]} - (k_4'[\mathrm{SK}] + k_4[\mathrm{MPF}])\frac{[\mathrm{Ste9}]}{J_4 + [\mathrm{Ste9}]}, \quad (3)$$

$$\frac{d[\mathrm{Slp1}_T]}{dt} = k_5' + k_5''\frac{[\mathrm{MPF}]^4}{J_5^4 + [\mathrm{MPF}]^4} - k_6[\mathrm{Slp1}_T], \quad (4)$$

$$\frac{d[\mathrm{Slp1}]}{dt} = k_7[\mathrm{IEP}]\frac{[\mathrm{Slp1}_T]-[\mathrm{Slp1}]}{J_7 + [\mathrm{Slp1}_T] - [\mathrm{Slp1}]} - k_8\frac{[\mathrm{Slp1}]}{J_8 + [\mathrm{Slp1}]} - k_6[\mathrm{Slp1}], \quad (5)$$

$$\frac{d[\mathrm{IEP}]}{dt} = k_9[\mathrm{MPF}]\frac{1-[\mathrm{IEP}]}{J_9 + 1 - [\mathrm{IEP}]} - k_{10}\frac{[\mathrm{IEP}]}{J_{10} + [\mathrm{IEP}]}, \quad (6)$$

$$\frac{d[\mathrm{Rum1}_T]}{dt} = k_{11} - (k_{12} + k_{12}'[\mathrm{SK}] + k_{12}''[\mathrm{MPF}])[\mathrm{Rum1}_T], \quad (7)$$

$$\frac{d[\mathrm{SK}]}{dt} = k_{13}[\mathrm{TF}] - k_{14}[\mathrm{SK}], \quad (8)$$

$$\frac{dM}{dt} = \mu M, \quad (9)$$

$$[\mathrm{Trimer}] = \frac{2[\mathrm{Cdc13}_T][\mathrm{Rum1}_T]}{\Sigma + \sqrt{\Sigma^2 - 4[\mathrm{Cdc13}_T][\mathrm{Rum1}_T]}}, \quad (10)$$

$$[\mathrm{MPF}] = \frac{([\mathrm{Cdc13}_T] - [\mathrm{preMPF}])([\mathrm{Cdc13}_T] - [\mathrm{Trimer}])}{[\mathrm{Cdc13}_T]}, \quad (11)$$

$$[\mathrm{TF}] = G(k_{15}M, k_{16}' + k_{16}''[\mathrm{MPF}], J_{15}, J_{16}), \quad (12)$$

where

$k_{wee} = k_{wee}' + (k_{wee}'' - k_{wee}')G(V_{awee}, V_{iwee}[\mathrm{MPF}], J_{awee}, J_{iwee})$,

$k_{25} = k_{25}' + (k_{25}'' - k_{25}')G(V_{a25}[\mathrm{MPF}], V_{i25}, J_{a25}, J_{i25})$,

$\Sigma = [\mathrm{Cdc13}_T] + [\mathrm{Rum1}_T] + K_{diss}$,

$$G(a,b,c,d) = \frac{2ad}{b-a+bc+ad+\sqrt{(b-a+bc+ad)^2 - 4ad(b-a)}}.$$

TABLE II. Parameter values for wild-type cells. All constants have units min^{-1}, except the Js, which are dimensionless Michaelis constants, and K_{diss}, which is a dimensionless equilibrium constant for trimer dissociation.

Cdc13 synthesis and degradation:
$k_1 = 0.03$, $k_2' = 0.03$, $k_2'' = 1$, $k_2''' = 0.1$.
Ste9 activation and inactivation:
$k_3' = 1$, $k_3'' = 10$, $J_3 = 0.01$, $k_4' = 2$, $k_4 = 35$, $J_4 = 0.01$.
Slp1 synthesis, degradation, activation and inactivation:
$k_5' = 0.005$, $k_5'' = 0.3$, $k_6 = 0.1$, $J_5 = 0.3$
$k_7 = 1$, $k_8 = 0.25$, $J_7 = 0.001$, $J_8 = 0.001$.
IE activation and inactivation:
$k_9 = 0.1$, $k_{10} = 0.04$, $J_9 = 0.01$, $J_{10} = 0.01$.
Rum1 synthesis, degradation and inhibition:
$k_{11} = 0.1$, $k_{12} = 0.01$, $k_{12}' = 1$, $k_{12}'' = 3$, $K_{diss} = 0.001$.
SK synthesis and degradation:
$k_{13} = 0.1$, $k_{14} = 0.1$.
TF activation and inactivation:
$k_{15} = 1.5$, $k_{16}' = 1$, $k_{16}'' = 2$, $J_{15} = 0.01$, $J_{16} = 0.01$.
Wee1 activation and inactivation:
$V_{awee} = 0.25$, $V_{iwee} = 1$, $J_{awee} = 0.01$, $J_{iwee} = 0.01$.
Cdc25 activation and inactivation:
$V_{a25} = 1$, $V_{i25} = 0.25$, $J_{a25} = 0.01$, $J_{i25} = 0.01$.
Rate of tyr-phosphorylation and dephosphorylation:
$k_{wee}' = 0.15$, $k_{wee}'' = 1.3$, $k_{25}' = 0.05$, $k_{25}'' = 5$.
Growth rate:
$\mu = 0.005$.

from *"Mathematical model of the cell division cycle of fission yeast". Chaos: An Interdisciplinary Journal of Nonlinear Science. Novak et al. 2001*

Boolean network models

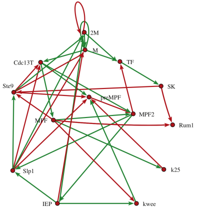


Fig. 2. Interaction network of the Boolean model. Green links correspond to $T_{ik} = +1$ and red links to $T_{ik} = -1$.

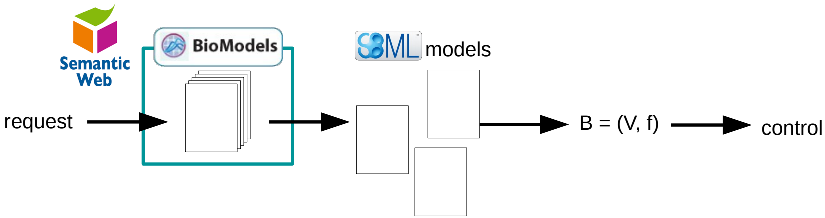
The transition from differential equations to Boolean networks: A case study in simplifying a regulatory network model. Davidich 2008.

community standards

standards needed to allow model sharing (and tools to communicate)

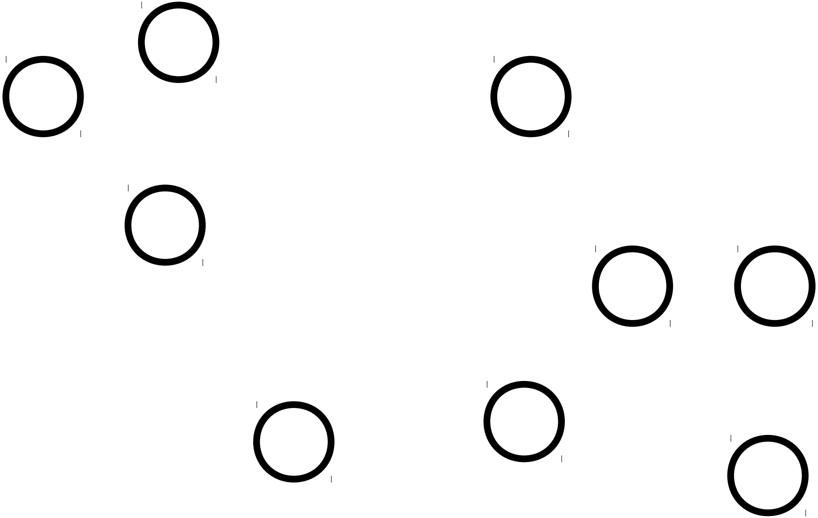
SBML, based on reaction model, is agnostic to the formalism.

Project Thesis Overview (sometimes I wonder if I'll have time ¯_(ツ)_/¯)

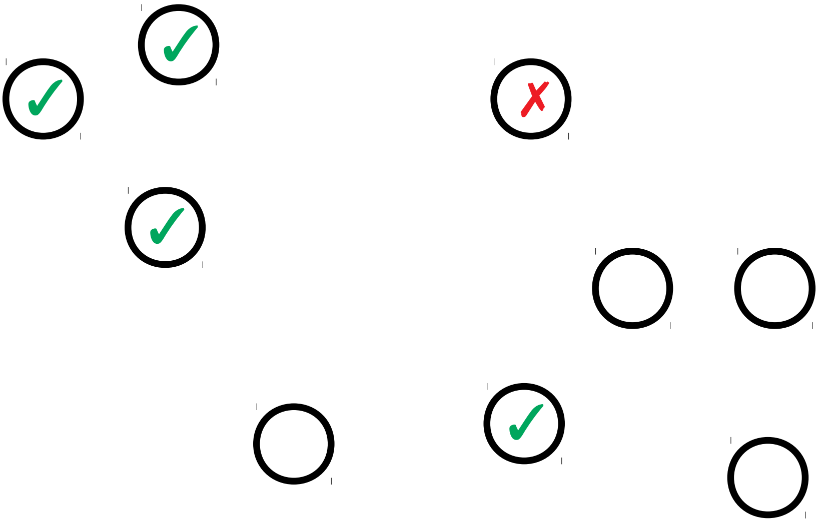


as automatized as possible... ;-)

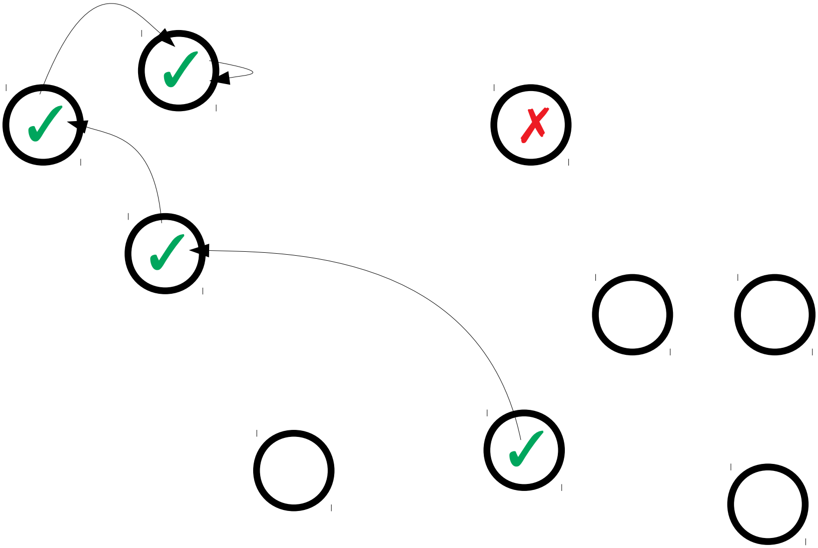
About control — the possible states of a system



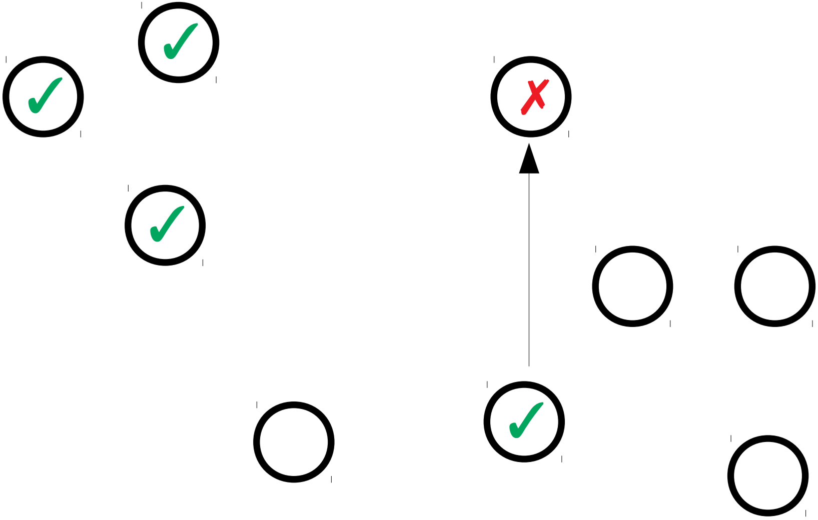
About control — some are “normal”, some are ‘pathological”



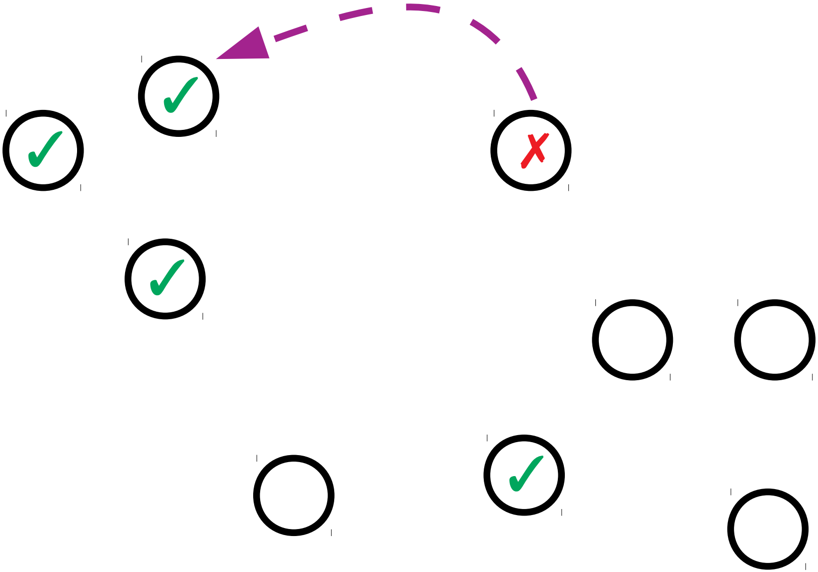
About control — behaviour of the **normal** system



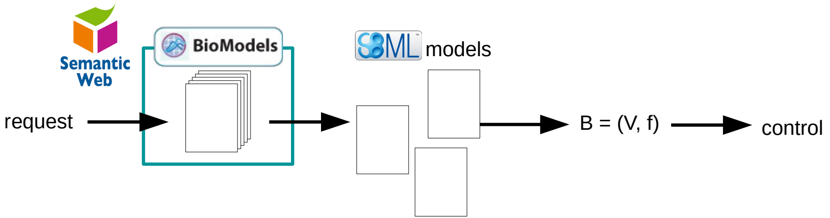
About control — behavior of the **broken** system



About control — behavior of the **repaired** system

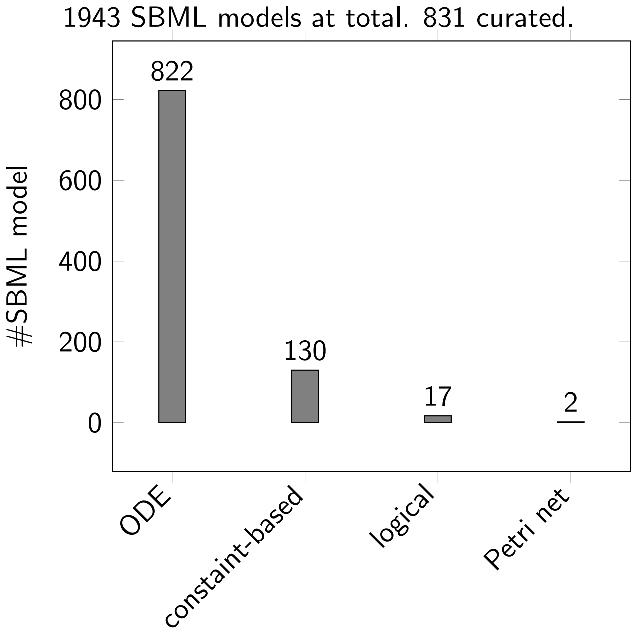


Project Thesis Overview

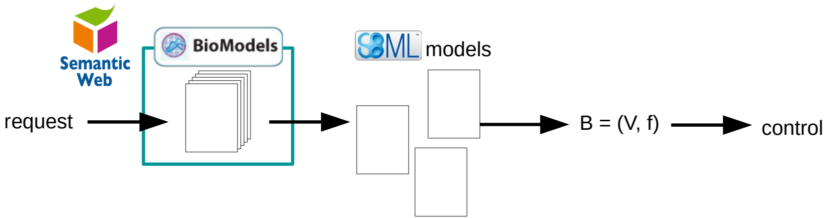


as automatized as possible... ;-)

BioModels content (updated 2019 dec 12)

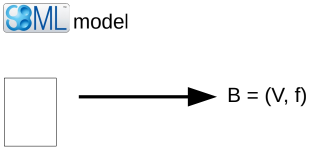


Project Thesis Overview



as automatized as possible... ;-)

Dish of the day ("Et bon appétit, bien sûr")



as automatized as possible... ;-)

Steps overwiew

- what is a boolean network. Concretely...? (easy-ish)
- retrieve the list of components (super archi easy)
- retrieve the influences between component (easy : use the Theorems of Fages 2008)
- retrieve the binarized behavior of the system (easy : just solve deterministically the equa diff then apply a binarization threshold)
- overapproximate the dynamic using the reachability (mediumly hard...)
- encode this in ASP (OMG, haaaaaaaaaard)

Boolean network formalism = network of Boolean automata

$$\mathbb{B} = \{0, 1\}$$

$$B = f : \mathbb{B}^n \to \mathbb{B}^n$$

$$B = (V, f_i : \mathbb{B}^n \to \mathbb{B} \; \forall i \in V)$$

boolean network formalism = network of boolean automata

- automata
- automata configuration
- system configuration
- influence
- fonctions locales de transitions
- updating scheme
- transition graph
- reachability

Automates

Warning

any resemblance to elements of fiction already existing **can not** be only fortuitous. . .

3 automata: $P, N, J \in V$

- Phillis (P) girlfriend of Joshua
- Nikki (N) is the mother of Joshua and does not like Phillis
- Joshua (J) is the son of Nikki and boyfriend of Phillis

Boolean automata $\rightarrow$ theeir state is Boolean

P, N and J can be either sad (0) or happy (1).
Automata state: boolean $\forall i \in V, x_i \in \mathbb{B} = \{0, 1\}$

Boolean automata → theeir state is Boolean

P, N and J can be either sad (0) or happy (1).
Automata state: boolean $\forall i \in V, x_i \in \mathbb{B} = \{0, 1\}$

system configuration : $x \in \mathbb{B}^n$

Here $n = 3 \rightarrow 2^3 = 8$ different configurations.

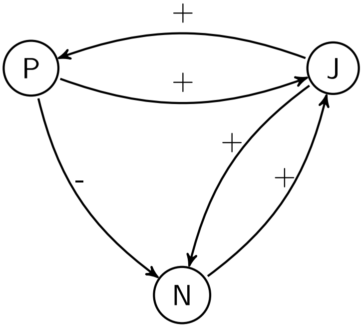
P	N	J	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

$\Delta(x, y) := i \in 1, ..., n | x_i \neq y_i$: set of component which differ between two configurations x and y.

des relations tout à fait classiques...

- P loves J. $IN(P) = \{J\}$, $k_P = 1$
- N loves her son J but not P $IN(N) = \{J, P\}$, $k_N = 2$
- J loves P and his dear mom N (differently, but still...) $IN(J) = \{P, N\}$, $k_J = 2$

influence graph:

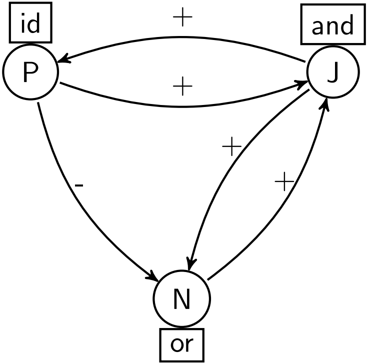


$IN(x_i) = \{x_{i_1}, \dots x_{i_k}\}$: the set of k parents nodes of x_i.
k: the "indegree" of x_i. K: the maximum indegree of a BN.

des dynamiques tout à fait classiques...

- P happy if J happy: $f_P = x_J$
- N happy if J happy or if P not happy: $f_N = x_J \vee \neg x_P$
- J happy when both N and P happy: $f_J = x_P \wedge x_N$

boolean network:



fonction locale de transition : $f_{x_i} : \mathbb{B}^{k_i} \to \mathbb{B}$

the truth (table) of the crisis

$$f_P = x_J$$
$$f_N = x_J \vee \neg x_P$$
$$f_J = x_P \wedge x_K$$

x_P	x_N	x_J	$f_P(x)$	$f_N(x)$	$f_J(x)$
0	0	0	0	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

transition graph

The transition graph describes the behavior of the system.

$$\mathscr{G} = \left(V_{\mathscr{G}} = \mathbb{B}^n, E_{\mathscr{G}} \subseteq \mathbb{B}^n \times \mathbb{B}^n \right)$$

It depends on the updating scheme.

updating scheme = organisation of the updates in time

Any order or any number of automata can be updated at each time step $\rightarrow$ infinite number of possibility.

Some classical:

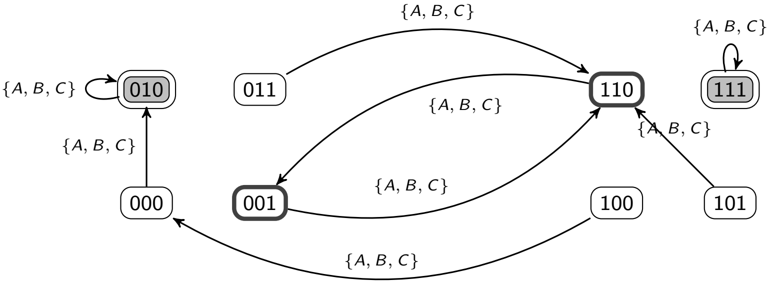
- (deterministic) synchonous (or parallele)
- (pure) asynchronous
- generalized asynchrous (or elementary)
- deterministic sequential block
- ...

One weird that we will use:

- most permissive (in $\mathbb{M} = \{0, *, 1\}$)

synchonous $\mathscr{G}^\pi$: $\{A, B, C\}$

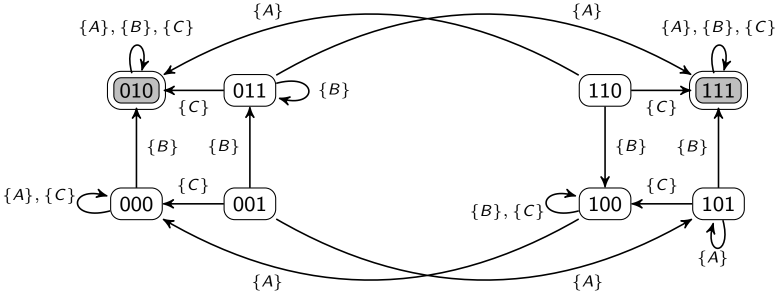
all components get updated at each time step : $x \xrightarrow{\pi} y$ iff $f(x) = y$



$$f_P = x_J;\, f_N = x_J \vee \neg x_P;\, f_J = x_P \wedge x_K$$

true asynchonous $\mathscr{G}^{\alpha}$: $\{\{A\},\{B\},\{C\}\}$

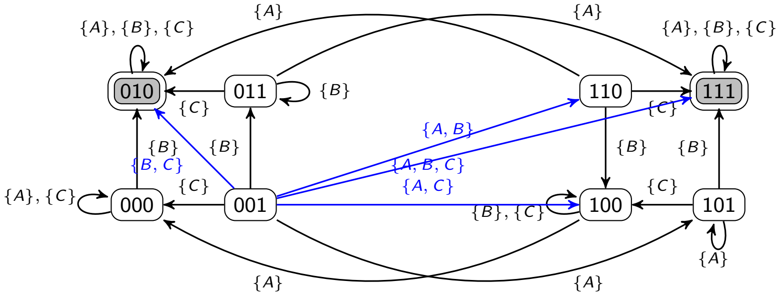
only one component (chosen non deteministicaly) is updated at each time step: $x \xrightarrow{\alpha} y$ iff $\exists i \in V, \Delta(x,y) = i \wedge y_i = f_i(x)$



$f_P = x_J$ $\qquad$ $f_N = x_J \vee \neg x_P$ $\qquad$ $f_J = x_P \wedge x_K$

generalized asynchrous $\mathscr{G}^{\epsilon}_{010}$:$\{\{A\},\{B\},\{C\},\{A,B\},\{A,C\},\{B,C\},\{A,B,C\}\}$

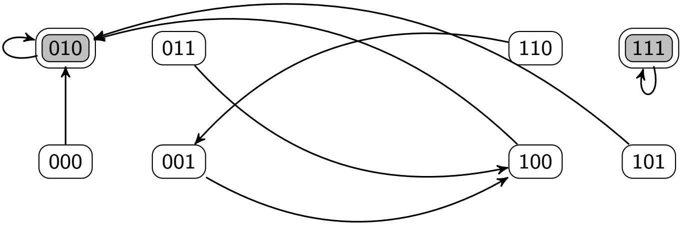
any number of component get updated non deterministically in one time step: $x \xrightarrow{\epsilon} y$ iff $\forall i \in \Delta(x,y), y_i = f_i(x)$



$f_P = x_J$ $\qquad$ $f_N = x_J \vee \neg x_P$ $\qquad$ $f_J = x_P \wedge x_K$

sequential blocs $\mathscr{G}^{\beta_1} = (\{A, B\}, \{C\})$

updates following a given block order



$f_P = x_J$ $\quad f_N = x_J \vee \neg x_P$ $\quad f_J = x_P \wedge x_K$

Trajectory and attractors (cycles and fixed points)

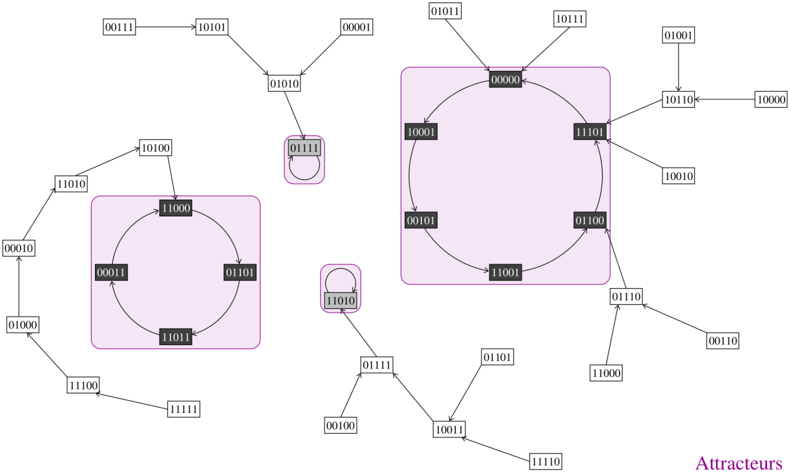
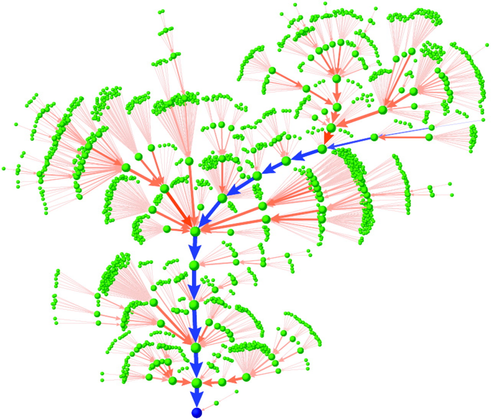


Image from Sylvain Sené "Réseaux d'automates et systèmes biologiques une approche par le calcul naturel", Biorégul 2019

Trajectory and attractors — biology mapping example

Any boolean model has at least 1 attractor.
Attractors are mapped to (biological) propreties (they represent physiological functions, cellular types, . . .).



Dynamical trajectories of the 1,764 protein states (green nodes) flowing to the G1 fixed point (blue node). Arrows between states indicate the direction of dynamic flow from one state to another. The cell-cycle sequence is colored blue. The size of a node and the thickness of an arrow are proportional to the logarithm of the traffic flow passing through them.

From "The yeast cell-cycle network is robustly designed." Li et al. 2004. (Fig. 2)

reachability

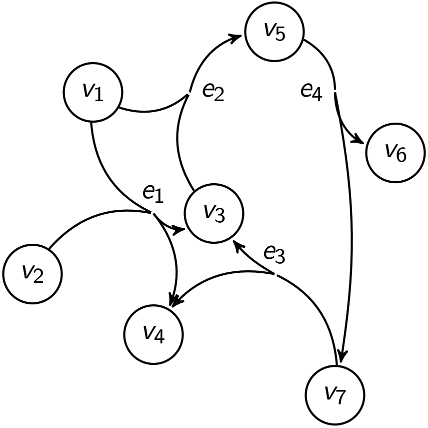
The fact that their exist a sequence with which a configuration is accessible from another configuration.

Steps overwiew

- what is a boolean network. Concretely...? (easy-ish)
- retrieve the list of components (super archi easy)
- retrieve the influences between component (easy : use the Theorems of Fages 2008)
- retrieve the binarized behavior of the system (easy : just solve deterministically the equa diff then apply a binarization threshold)
- overapproximate the dynamic using the reachability (mediumly hard...)
- encode this in ASP (OMG, haaaaaaaaaard)

core : Reactions model

$$M = \{e_i \texttt{ for } R_i\texttt{=[} M_i \texttt{]=>}P_i\}_{i=1\ldots n}$$



hypergraph = nodes + hyperedges

Differential semantic for reaction models

$$M = \{e_i \texttt{ for } R_i\texttt{=[} M_i \texttt{]=>} P_i\}_{i=1\ldots n}$$

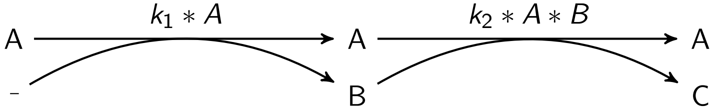
$$dx/dt = \sum_{i=1}^{n} (P_i(x) - R_i(x)) * e_i$$

an example :

$k_1 * A$ `for _=[ A ]=>B`

$k_2 * A * B$ `for B=[ A ]=>C`

$$dA/dt = 0$$
$$dB/dt = k1 * A - k2 * A * B$$
$$dC/dt = k2 * A * B$$



Steps overwiew

- what is a boolean network. Concretely...? (easy-ish)
- retrieve the list of components (super archi easy)
- retrieve the influences between component (easy : use the Theorems of Fages 2008)
- retrieve the binarized behavior of the system (easy : just solve deterministically the equa diff then apply a binarization threshold)
- overapproximate the dynamic using the reachability (mediumly hard...)
- encode this in ASP (OMG, haaaaaaaaaard)

retrieve the influence graph from the SBML file

influence graph: a directed graph with labeled edges
($+$: positive influence; $-$: negative influence).

Theorem from Fages et al. 2008 prove the equivalence between Reaction Models and Influence Graph. Either:

- Parse the rules, and determine the influence from the stoichiometric coefficients given in the rules.
- Use the jacobian matrix J of the ODE model formed of the partial derivatives $J_{i,j} = \frac{\partial i}{\partial j}$.

Steps overwiew

- what is a boolean network. Concretely...? (easy-ish)
- retrieve the list of components (super archi easy)
- retrieve the influences between component (easy : use the Theorems of Fages 2008)
- retrieve the binarized behavior of the system (easy : just solve deterministically the equa diff then apply a binarization threshold)
- overapproximate the dynamic using the reachability (mediumly hard...)
- encode this in ASP (OMG, haaaaaaaaaard)

Solve the equa diff deterministically

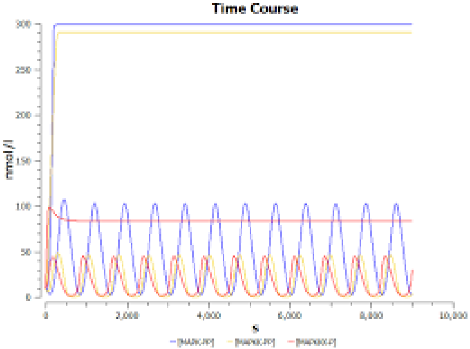


Figure: from “Building and Simulating Models using COPASI”, © 2016, Nicolas LeNovère, Viji Chelliah, Bhupinder Virk. creative commons Attribution - Share Alike 4.0 licence.

Scale between 0 and 1, then apply a threasold

ρ_t^x : continuous observation $\in [0;1]$ of a component x at time t,
β_t^x : its Boolean value ; determined with the following :

$$\beta_t^x \triangleq \begin{cases} 1 & \text{when } \rho_t^x <= 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Other methods exist, but this one is the most simple one.

Steps overwiew

- what is a boolean network. Concretely...? (easy-ish)
- retrieve the list of components (super archi easy)
- retrieve the influences between component (easy : use the Theorems of Fages 2008)
- retrieve the binarized behavior of the system (easy : just solve deterministically the equa diff then apply a binarization threshold)
- overapproximate the dynamic using the reachability (mediumly hard...)
- encode this in ASP (OMG, haaaaaaaaaard)

meta-configuration in most permissive semantics

Reachability : the fact that their exist a sequence with which a configuration is accessible from another configuration.

Most permissive semantics consists in **approximating** the dynamic using the **reachability** between states and not directly the state transition.

A meta-configuration concatenates together all the configurations that are reachable from a given configuration.

One Semantic to approximate them all,
One Semantic to find them,
One Semantic to bring them all
and in a vector bind them.

meta-configurations : example 1

Concrete ($\rightarrow$)

$$
\begin{tikzcd}
\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \arrow[r] & \begin{pmatrix}1\\0\\1\\0\end{pmatrix} \arrow[d] \\
\begin{pmatrix}1\\0\\0\\1\end{pmatrix} \arrow[u] & \begin{pmatrix}1\\0\\1\\1\end{pmatrix} \arrow[l]
\end{tikzcd}
$$

Meta-state ($\rightrightarrows$)

$$
\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \rightrightarrows \begin{pmatrix}1\\0\\\{0,1\}\\0\end{pmatrix} \rightrightarrows \begin{pmatrix}1\\0\\\{0,1\}\\\{0,1\}\end{pmatrix}
$$

cycle of the BN transition graph is over-approximated by a fixed-point in the meta-state semantics

meta-configurations : example 2

Concrete ($\rightarrow$)

$$\begin{tikzcd}
\begin{pmatrix}0\\1\\1\\0\end{pmatrix} \arrow[r] \arrow[d] \arrow[dr] & \begin{pmatrix}0\\1\\1\\1\end{pmatrix} \arrow[d] \\
\begin{pmatrix}0\\1\\0\\0\end{pmatrix} \arrow[r] & \begin{pmatrix}0\\1\\0\\1\end{pmatrix}
\end{tikzcd}$$

Meta-state ($\rightrightarrows$)

$$\begin{tikzcd}
\begin{pmatrix}0\\1\\1\\0\end{pmatrix} \arrow[r, shift left] \arrow[r, shift right] \arrow[d, shift left] \arrow[d, shift right] & \begin{pmatrix}0\\1\\1\\\{0,1\}\end{pmatrix} \arrow[d, shift left] \arrow[d, shift right] \\
\begin{pmatrix}0\\1\\\{0,1\}\\0\end{pmatrix} \arrow[r, shift left] \arrow[r, shift right] & \begin{pmatrix}0\\1\\\{0,1\}\\\{0,1\}\end{pmatrix}
\end{tikzcd}$$

fixed-point of the BN transition graph is over-approximated by a fixed-point in the meta-state semantics

Steps overwiew

- what is a boolean network. Concretely...? (easy-ish)
- retrieve the list of components (super archi easy)
- retrieve the influences between component (easy : use the Theorems of Fages 2008)
- retrieve the binarized behavior of the system (easy : just solve deterministically the equa diff then apply a binarization threshold)
- overapproximate the dynamic using the reachability (mediumly hard...)
- encode this in ASP (OMG, haaaaaaaaaard)

ASP = Answer Set Programming

- declarative language
- rules are written in the form of clauses:
  `H :- B1; ...; Bn.`
- that correspond to logical implications:
  `H if B1 and ...  and Bn.`
- "generate then filter" approach
- a solution is a True / False instensiation of atoms that respect the rules
- ASP program can generates "answer sets" = the set of all the solutions that would work

Crazy hard to sum the element of a table. But 6 lines are needed to solve a sudoku...

sum the elements of a table in ASP

```
val(1;3;18;23;42;67;72).

first(X):- val(X) ; not val(Y): val(Y), Y<X.

next(X,Y):- val(X) ; val(Y) ; X<Y ; not val(Z): val(Z), X<Z, Z<Y.

sum(1,F,F):- first(F).
sum(N+1,W,T+W):- sum(N,V,T) ; next(V,W).
step(S):- sum(S,_,_).

sum(S):- sum(N,_,S) ; not sum(M,_,_): step(M), N<M.
#show sum/1.  % Pouloulou !
```

Conclusion : “on en bave des ronds de chapeau”

sudoku solving in ASP

```
x(1..9). y(1..9). n(1..9).

{sudoku(X,Y,N): n(N)}=1 :- x(X);y(Y).

subgrid(X,Y,A,B) :- x(X); x(A); y(Y); y(B);
                    (X-1)/3==(A-1)/3; (Y-1)/3==(B-1)/3.

:- sudoku(X,Y,N); sudoku(A,Y,N); X!=A.
:- sudoku(X,Y,N); sudoku(X,B,N); Y!=B.
:- sudoku(X,Y,V); sudoku(A,B,V); subgrid(X,Y,A,B); X != A; Y != B.

#show sudoku/3.
```

Our problem in ASP = TODO

Ask ASP what are the logical rules of a given topology and that behave such that their dynamic fullfill the constraints about the reachability between the configurations?

For the next tea time... ;-)

Thanks for your attention...
Any questions ? :)

References I

Extra slides

sum element of a table in ASP

```
% Values to be sum up :
val(1;3;18;23;42;67;72).
% Find the first one :
first(X):- val(X) ; not val(Y): val(Y), Y<X.
% Find the "next" one following an increasing order
next(X,Y):- val(X) ; val(Y) ; X<Y ; not val(Z): val(Z),
% Add the thing in the list, one after the other
sum(1,F,F):- first(F).  % At first step, this corres. to
sum(N+1,W,T+W):- sum(N,V,T) ; next(V,W).  % At step n, i
step(S):- sum(S,_,_).
% The total is the last sum at the last step
sum(S):- sum(N,_,S) ; not sum(M,_,_): step(M), N<M.
#show sum/1.  % Pouloulou !
```

Conclusion : "on en bave des ronds de chapeau"

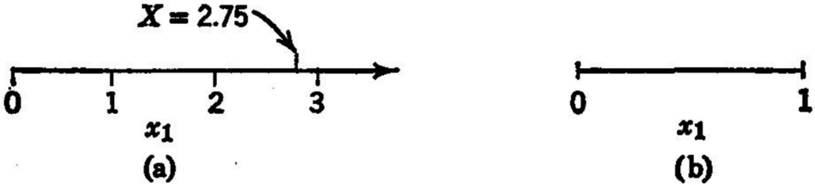
influence graph — Fages 2008

influence graph: a directed graph with labeled edges ($+$: positive influence; $-$: negative influence).

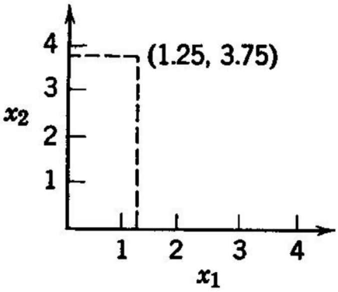
Formally, 2 definitions (Fages et al. 2008):

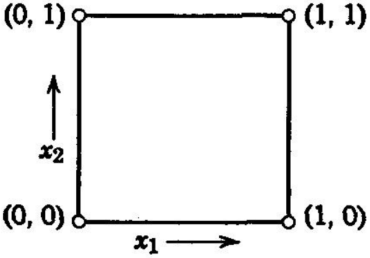
- Synthactical Influence Graph (SIG), from the stoichiometric coefficients given in the rules.
- Differential Influence Graph (DIG), from the jacobian matrix J of the ODE model formed of the partial derivatives $J_{i,j} = \frac{\partial x}{\partial j}$.

hypercubical representation

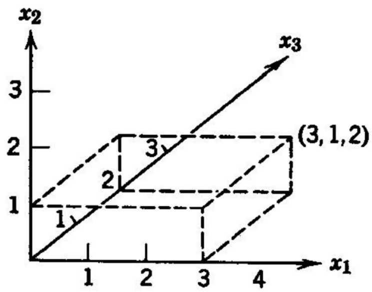


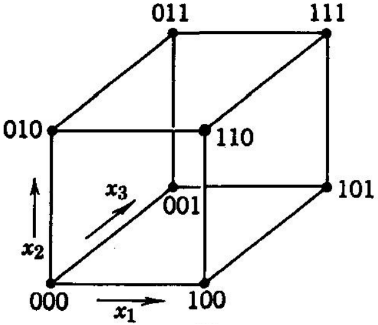
hypercubical representation





hypercubical representation





Reaction model

Reactions:

$R_1 : e_1$ `for` A`=[  ]=>`B

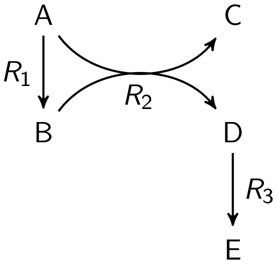
$R_2 : e_2$ `for` $A + B$`=[  ]=>`$C + D$

$R_3 : e_3$ `for` D`=[  ]=>`E

Stoichiometrix matrix:

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccc} R_1 & R_2 & R_3 \end{array} \\ \begin{pmatrix} -1 & -1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

Hypergraph:



citation: Klamt `https://journals.plos.org/ploscompbiol/article/file?id=10.1371/journal.pcbi.1000385&type=printable`
citation Fages 2018

Reaction model with stoichiometry

Reactions:

$R_1 : e_1$ `for` $2A$`=[ ]=>`B

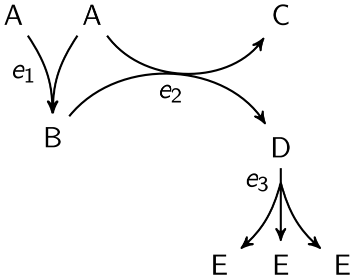
$R_2 : e_2$ `for` $A + B$`=[ ]=>`$C + D$

$R_3 : e_3$ `for` D`=[ ]=>`$3E$

Stoichiometrix matrix:

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccc} R_1 & R_2 & R_3 \end{array} \\ \begin{pmatrix} -2 & -1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 3 \end{pmatrix} \end{array}$$

Hypergraph:



citattion: Klamt `https://journals.plos.org/ploscompbiol/article/file?id=10.1371/journal.pcbi.1000385&type=printable`
citation Fages 2018

Reaction model with modifier

Reaction:

$$R : e \ \texttt{for} \ A + M \texttt{=[} \quad \texttt{]=>} B + M$$

$$R : e \ \texttt{for} \ A \texttt{=[} \ M \ \texttt{]=>} B$$

$$R : e \ \texttt{for} \ A \texttt{=[} \ M/M \ \texttt{]=>} B$$

Hypergraph :

$$\begin{array}{c} A \\ \Big\downarrow e \\ B \end{array}$$

Stoichiometrix matrix:

$$\begin{array}{cc} & R \\ \begin{array}{c} A \\ B \\ M \end{array} & \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \end{array}$$

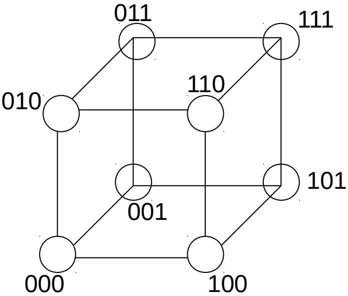
citattion : Klamt `https://journals.plos.org/ploscompbiol/article/file?id=10.1371/journal.pcbi.1000385&type=printable`
citation Fages 2018

meta-states are hypercubes

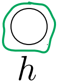
set of associated configurations :

$$c(h) = \{x \in \mathbb{B}^n | \forall i \in \{1, ..., n\}, h_i \neq * \Rightarrow x_i = h_i\}$$

Configurations in
concrete semantic
$\mathbb{B}^n$



Meta-configurations in
meta semantics
$\mathbb{M}^n$

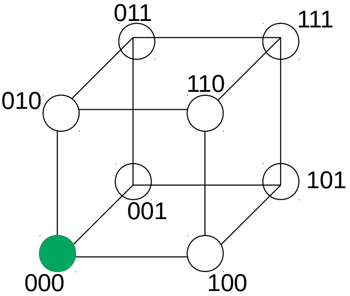


meta-states are hypercubes

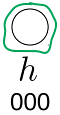
set of associated configurations :

$$c(h) = \{x \in \mathbb{B}^n | \forall i \in \{1, ..., n\}, h_i \neq * \Rightarrow x_i = h_i\}$$

```
Configurations in
concrete semantic
```
$\mathbb{B}^n$




```
Meta-configurations in
meta semantics
```
$\mathbb{M}^n$

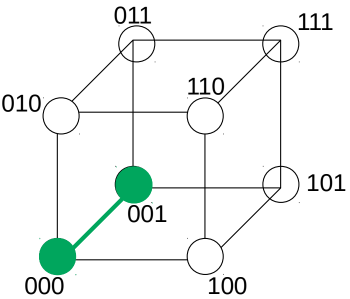


meta-states are hypercubes

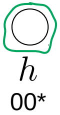
set of associated configurations :

$$c(h) = \{x \in \mathbb{B}^n | \forall i \in \{1, ..., n\}, h_i \neq * \Rightarrow x_i = h_i\}$$

Configurations in
concrete semantic
$\mathbb{B}^n$



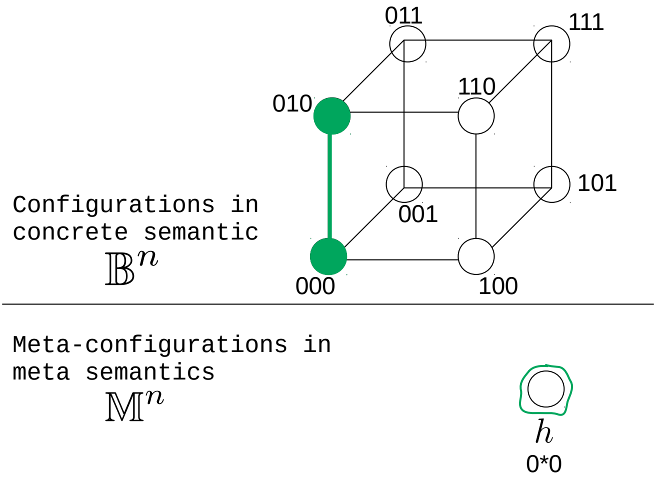
Meta-configurations in
meta semantics
$\mathbb{M}^n$



meta-states are hypercubes

set of associated configurations :

$$c(h) = \{x \in \mathbb{B}^n | \forall i \in \{1, ..., n\}, h_i \neq * \Rightarrow x_i = h_i\}$$

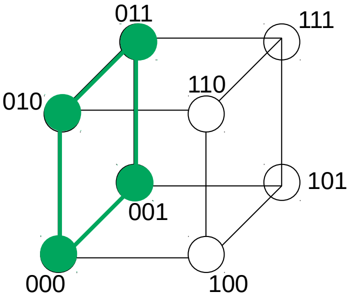


meta-states are hypercubes

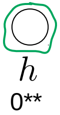
set of associated configurations :

$$c(h) = \{x \in \mathbb{B}^n | \forall i \in \{1, ..., n\}, h_i \neq * \Rightarrow x_i = h_i\}$$

```
Configurations in
concrete semantic
```
$\mathbb{B}^n$




```
Meta-configurations in
meta semantics
```
$\mathbb{M}^n$

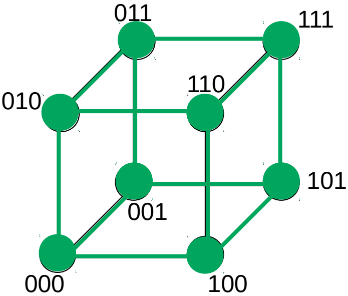


meta-states are hypercubes

set of associated configurations :

$$c(h) = \{x \in \mathbb{B}^n | \forall i \in \{1, ..., n\}, h_i \neq * \Rightarrow x_i = h_i\}$$

```
Configurations in
concrete semantic
```
$\mathbb{B}^n$




```
Meta-configurations in
meta semantics
```
$\mathbb{M}^n$