Vorlesung 2 [Fr 25.10.2019]

Boolean Networks

Motivation

- often very limited information in biological systems

Example

- gene states only qualitative as on/off (active/inactive genes)
- complex regulatory networks with information only on nodes (i.e. genes) and weather they interact

Modeling approach: Boolean networks

- simple representation of dynamic biological networks
- **directed graph** consisting of nodes with **boolean state** updated by rules based on inputs (edges)
- boolean networks are **discrete in space and time**
  - diskreter Zustandsraum
  - diskrete Zeitschritte

Nodes & states

- players in network, either true/false (1/0)

Rules

- **Boolean rules** using operators **and, or**, **not** and combinations thereof
- rules are constant over time (not adaptive)

Update schema

- synchronous: concerted (synchronous) update of all states
- no memory: states do not know anything about previous trajectory

Examples

~~$x_t = x(t)$~~

~~$x_{t+1} = f(x_t)$~~

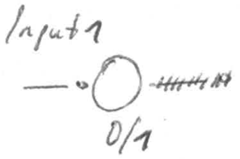
$$\vec{X}_t = \vec{X}(t) = \underset{t=0}{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}} \xrightarrow{f} \underset{t=1}{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}} \to \underset{t=2}{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}} \to \underset{t=3}{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}}$$

$$X_{t+1} = f(X_t)$$

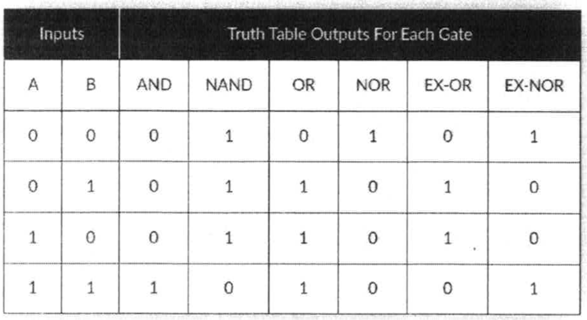
Boolean tables

1 dimensional: single input (4 possible rules)

0	1	
0	1	identity
1	0	inverse
0	0	zero
1	1	one



2 dimensional: two inputs (16 possible rules)

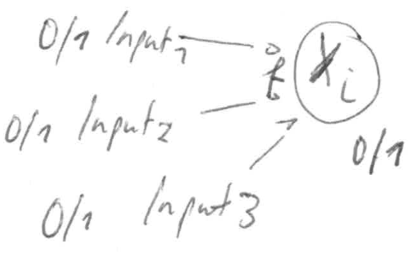


Inputs		Truth Table Outputs For Each Gate					
A	B	AND	NAND	OR	NOR	EX-OR	EX-NOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

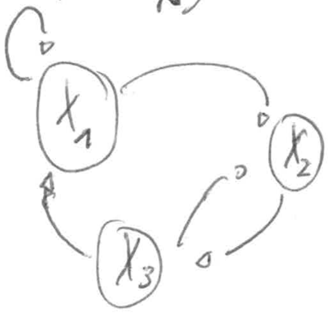
in general: rule has to specify the output for all possible inputs

- for a 2 input node 4 different input states are possible (2^2): [0,0], [0,1], [1,0], [0,1]
- for a N input node 2^N different input states are possible and 2^N^2 rules

Networks



$X_t = f(X_1, \ldots, X_N)$



$X_1(t+1) = f_1(X_1(t), X_3(t))$
$X_2(t+1) = f_2(X_1(t), X_3(t))$
$X_3(t+1) = f_3(X_2(t))$

Important terms

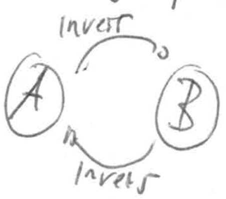
- **trajectory**: **sequence of states** through state space
- **attractor:** since number of states is finite, number of possible transitions finite → each trajectory will lead either to a **steady state** or a **steady cycle**
- **transient states:** states not belonging to attractor
- **basin of attraction**: all states that lead to same attractor
- **cycle length:** number of states on the cycle (steady state is a cycle with length 1)
- **path length**: number of states between initial state and attractor
- **average path length:** in [1, N]
- **number of attractors:** in {1, .., N}; median attractors & cycle length ~ $\sqrt{N}$

Analysis

- **study attractors & attractor graph** (especially cycles of interest → cell cycle models)
- **perturbation simulations**
    - change of node state, follow dynamics of system
    - change of rules of individual notes: study change in attractors
      ⇒ **robustness**

~~More complex example~~

Simple example:



$A(t+1) = f_A(B_t)$

$B(t+1) = f_B(A_t)$

f_A

B_t	0	1
A_t	1	0

f_B

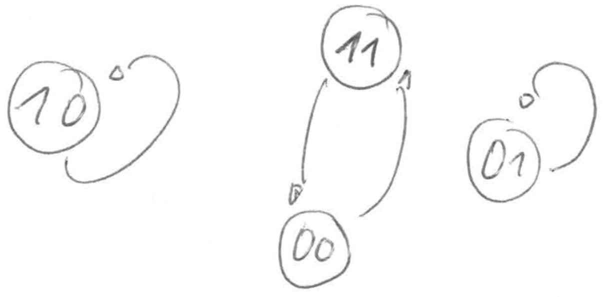
A_t	0	1
B_t	1	0

$\begin{pmatrix} A_0 \\ B_0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \overset{?}{\longrightarrow} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ locked: steady state

$\begin{pmatrix} A_0 \\ B_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ locked steady state but

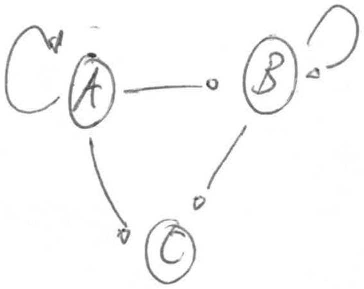
$$\begin{pmatrix}A\\B\end{pmatrix} = \begin{pmatrix}1\\1\end{pmatrix} \to \begin{pmatrix}0\\0\end{pmatrix} \to \begin{pmatrix}1\\1\end{pmatrix}$$ steady cycle

Zustandsraum



3 attractors

More complex



$A_{t+1} = f_A(A_t)$ NOT

A_t	0	1
A_t	1	0

$B_{t+1} = f_B(A_t, B_t)$

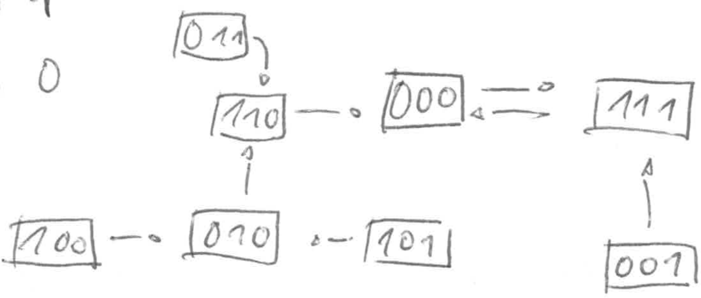
$(AB)_t$	00	01	10	11
B_t	1	1	1	0

$C_{t+1} = f_C(A_t, B_t)$

$(AB)_t$	00	01	10	11
C_{t+1}	1	0	0	0

t	ABC
0	000
1	111
2	000

total 8 states



global connected network
One periodic attractor

ABC
101
010

Advanced Types of boolean networks

asynchronous Boolean networks

- asynchronous update schemas: a random node is selected in each time point and updated
- repeated simulation of the same network with identical start conditions can provide an average behavior of the network

multiple responses per state

- **discrete levels: 0,1, 2,3**
- more complicated rules but the principles remain the same

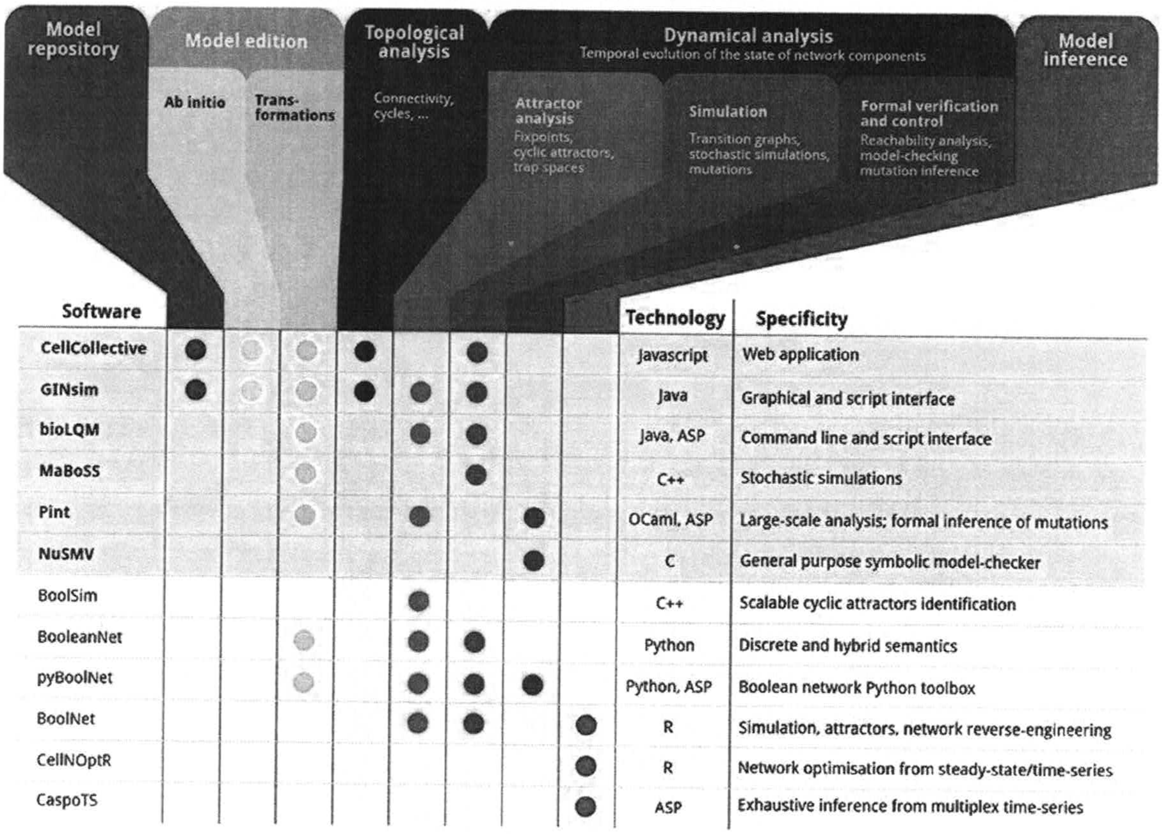
random Boolean networks

- generalization of boolean networks
- update rules are choosen randomly during construction (remain constant over time)

probabilistic boolean networks

- assign with a certain probability update rules to nodes at each time step

Software



Software	Model repository	Model edition: Ab initio	Model edition: Trans-formations	Topological analysis (Connectivity, cycles, ...)	Dynamical analysis: Attractor analysis (Fixpoints, cyclic attractors, trap spaces)	Dynamical analysis: Simulation (Transition graphs, stochastic simulations, mutations)	Dynamical analysis: Formal verification and control (Reachability analysis, model-checking mutation inference)	Model inference	Technology	Specificity
CellCollective	●	●	●	●		●			Javascript	Web application
GINsim	●	●	●	●	●	●			Java	Graphical and script interface
bioLQM			●		●	●			Java, ASP	Command line and script interface
MaBoSS			●			●			C++	Stochastic simulations
Pint			●		●		●		OCaml, ASP	Large-scale analysis; formal inference of mutations
NuSMV							●		C	General purpose symbolic model-checker
BoolSim					●				C++	Scalable cyclic attractors identification
BooleanNet			●		●	●			Python	Discrete and hybrid semantics
pyBoolNet			●		●	●	●		Python, ASP	Boolean network Python toolbox
BoolNet					●	●		●	R	Simulation, attractors, network reverse-engineering
CellNOptR								●	R	Network optimisation from steady-state/time-series
CaspoTS								●	ASP	Exhaustive inference from multiplex time-series

- CellCollective
  https://cellcollective.org/#
- BooleSim Systems biology tool: Boolean network simulator
  https://github.com/matthiasbock/BooleSim
- CANA: A Python Package for Quantifying Control and Canalization in Boolean Networks
  https://github.com/rionbr/CANA

- Python booleannet
  https://github.com/ialbert/booleannet
- GinSim
  http://ginsim.org/
- PyBoolNet
  https://github.com/hklarner/PyBoolNet

References

- *Systems Biology*, A textbook, Second edition, Edda Klipp, Wolfram Liebermeister, Christoph Wierling and Axel Kowald, Section 7.1 Discrete Models

Application

Boolean networks have been applied to biological processes such as regulation of gene expression in the framework of **Kauffman**'s NK Boolen Networks

- gene expression by 2 states: on/off, 1/0, expressed/not expressed
- network has N elements (nodes) with binary state
- each element has K inputs (regulatory interactions) and 1 output (that is its state)
- since every node can be in one of the 2 states 2^N different states are possible
- an N dimensional vector can describe the state at time t
- values are updated in discrete time steps