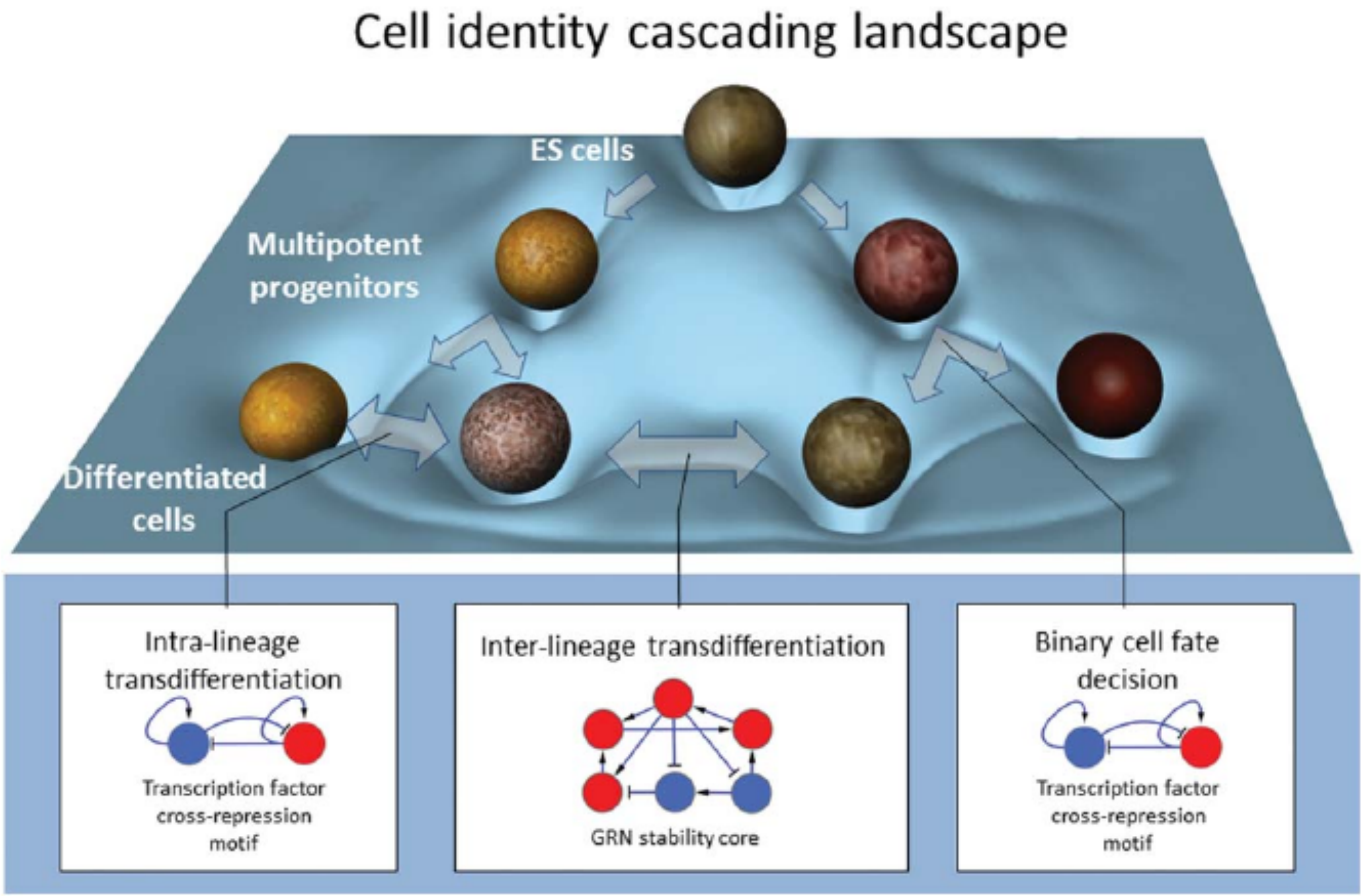
Formal methods for capturing dynamics of biological systems

Loïc Paulevé

CNRS/LaBRI, Bordeaux, France

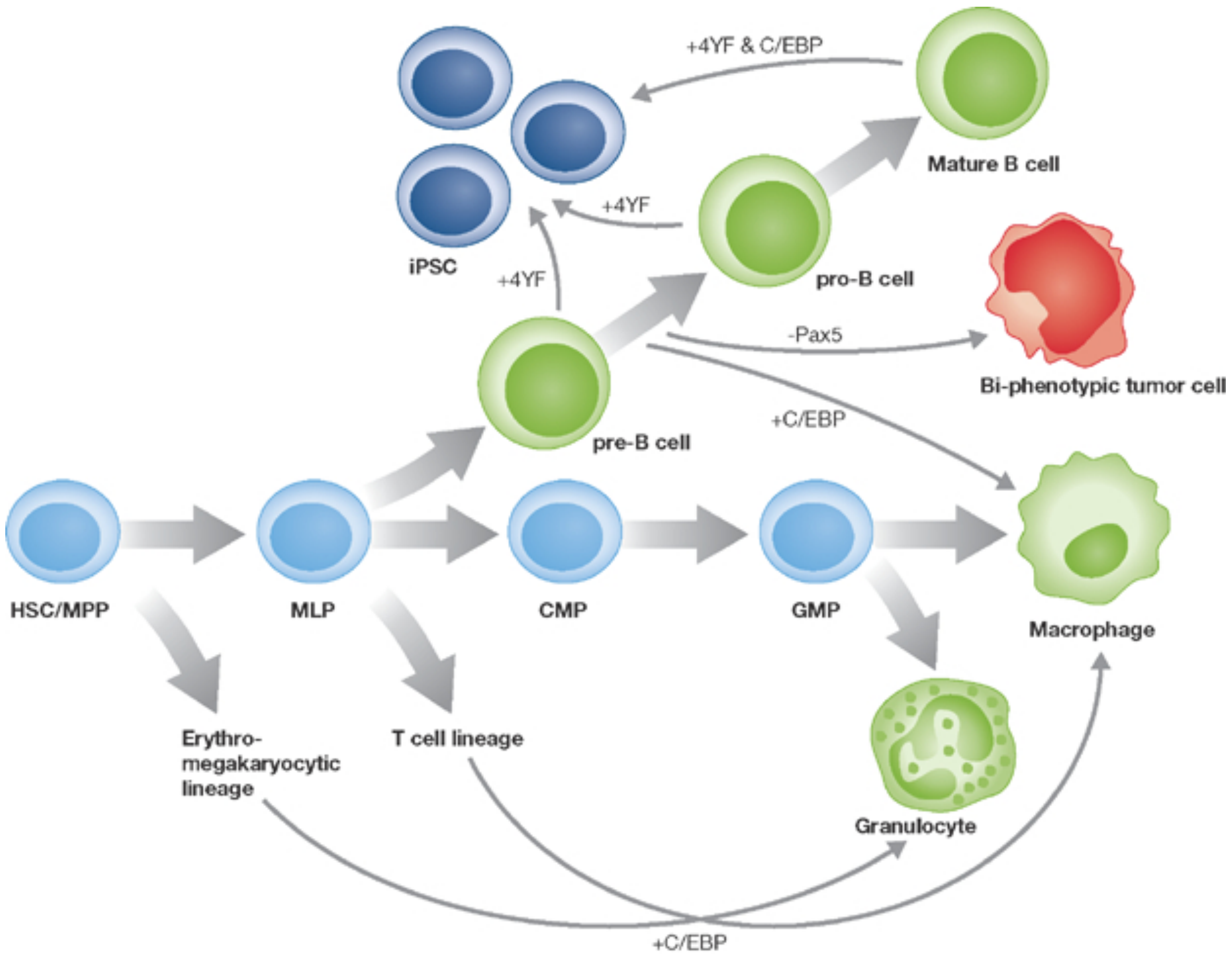
https://loicpauleve.name

Cellular differentiation



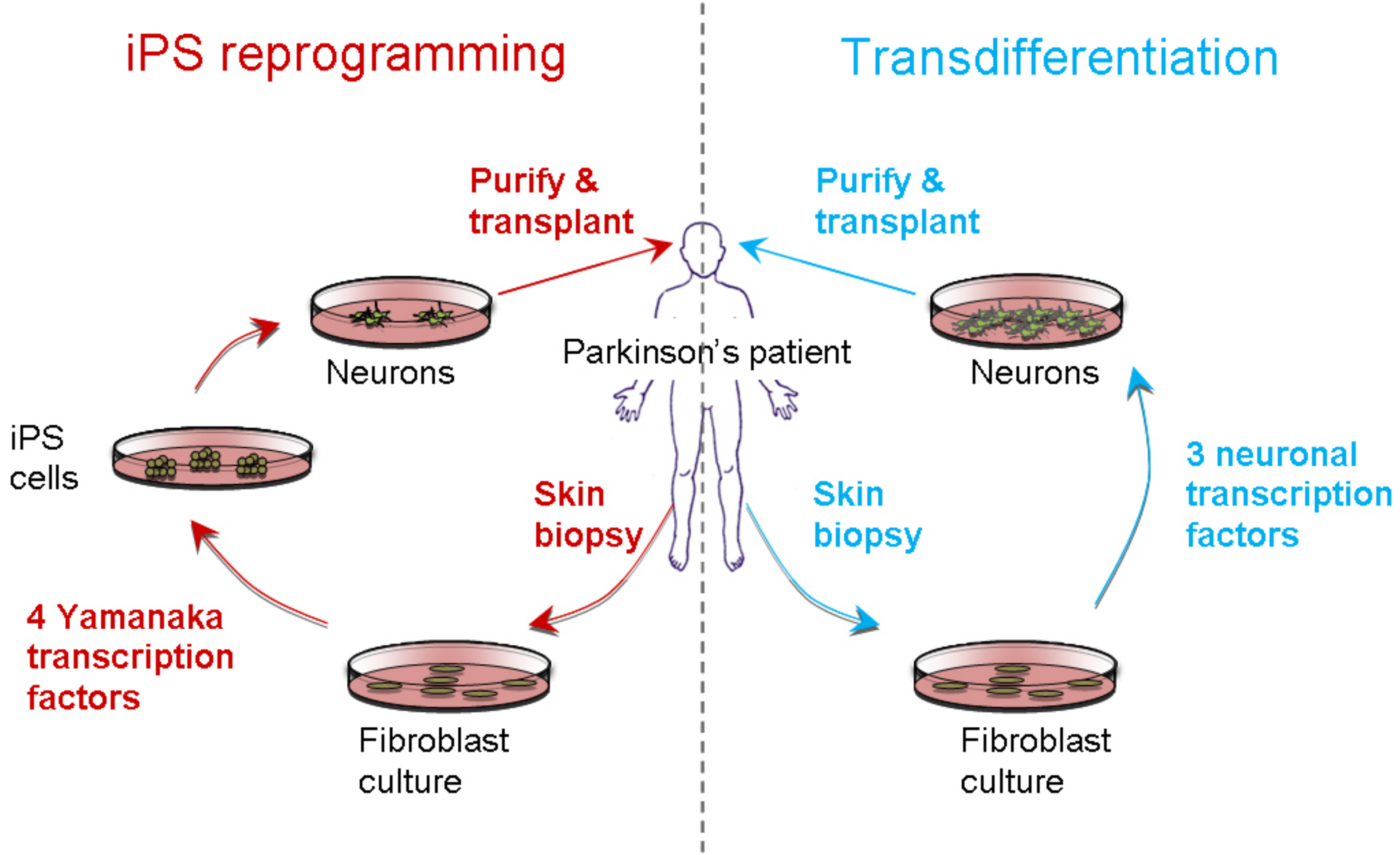
(source : Crespo et al. Stem cells 2013)

Cellular differentiation



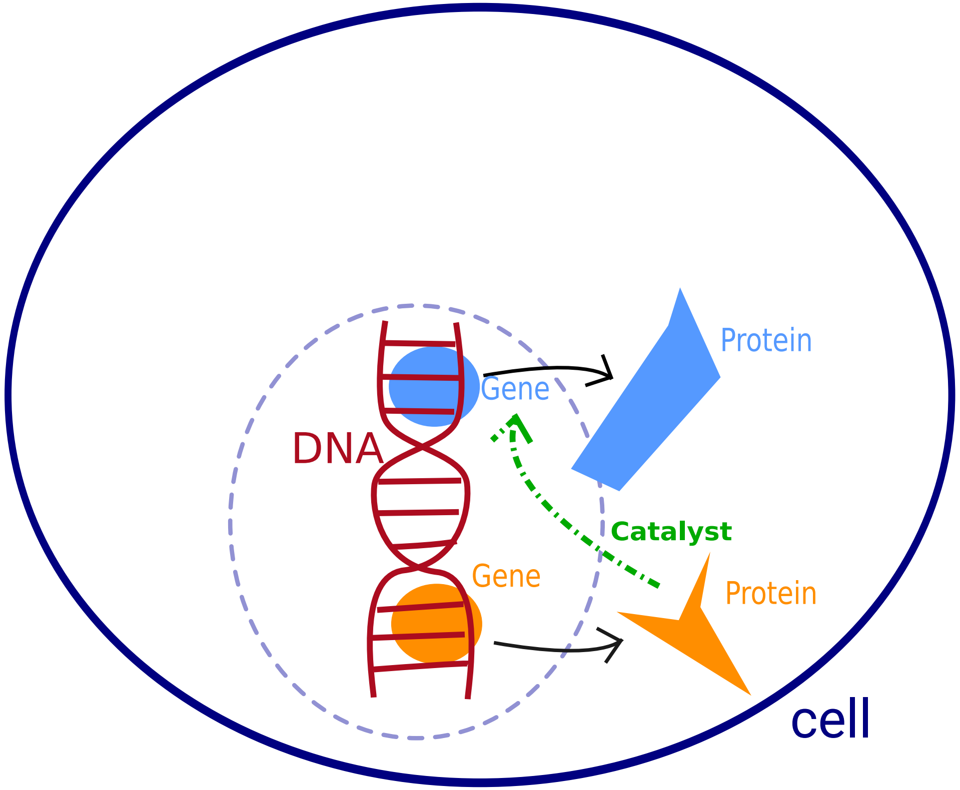
(source : Regalo, Leutz. EMBO Mol Medicine, 2013)

Cellular reprogramming



(credits : Thomas Graf, Centre for Genomic Regulation (Spain))

Modeling focus: gene and signaling networks



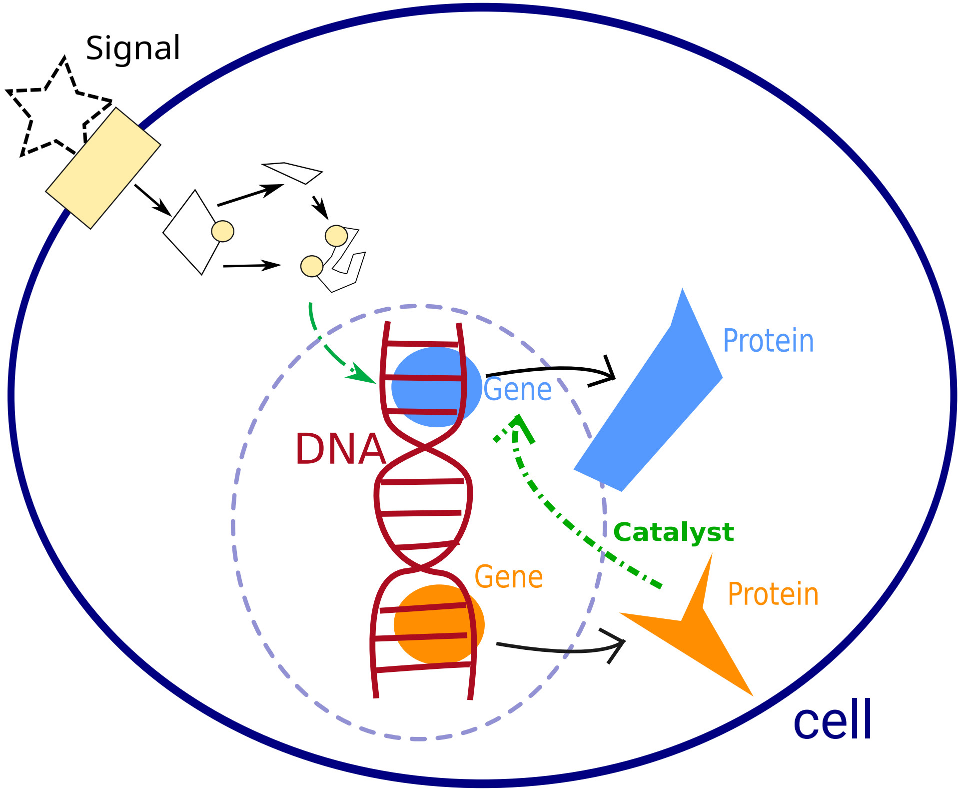
Many features: cell shape, composition (proteins), ion fluxes, gene expression, metabolism, ...
a modeling choice has to be made: hypotheses from experts

This talk: methods related to gene regulatory networks and signalling pathways

Some numbers in human cells:
- ~20,000 genes
- ~1,500-2,000 transcription factors

but not all important for a specific differentiation process!

Modeling focus: gene and signaling networks



Many features: cell shape, composition (proteins), ion fluxes, gene expression, metabolism, ...
a modeling choice has to be made: hypotheses from experts

This talk: methods related to gene regulatory networks and signalling pathways

Some numbers in human cells:

- ~20,000 genes
- ~1,500-2,000 transcription factors

but not all important for a specific differentiation process!

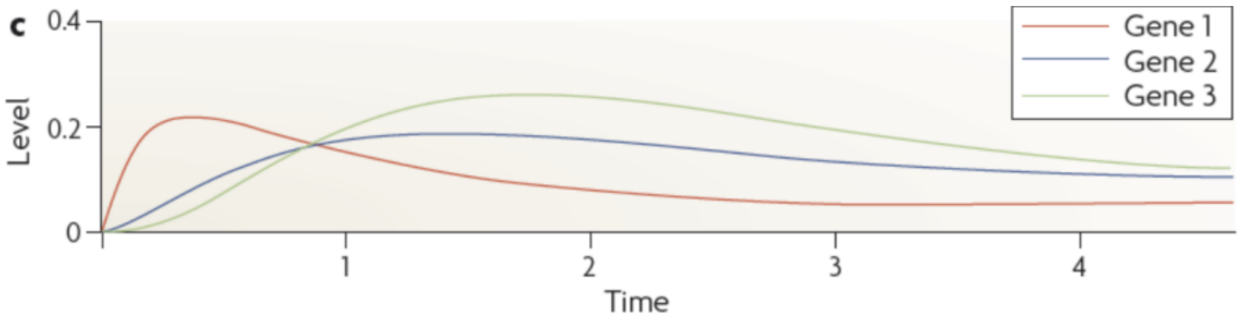
Modeling approaches

● Differential equations: concentration of proteins/gene activity
➥ numerous parameters (speed, quantities, precise function of derivatives...)

$$\frac{d(gene_1)}{dt} = k_{1s} \cdot \frac{1}{1 + k_{13} \cdot gene_3} - k_{1d} \cdot gene_1$$

$$\frac{d(gene_2)}{dt} = k_{2s} \cdot \frac{k_{21} \cdot gene_1}{1 + k_{21} \cdot gene_1} - k_{2d} \cdot gene_2$$

$$\frac{d(gene_3)}{dt} = k_{3s} \cdot \frac{k_{31} \cdot gene_1 \cdot k_{32} \cdot gene_2}{(1 + k_{31} \cdot gene_1) \cdot (1 + k_{32} \cdot gene_2)} - k_{3d} \cdot gene_3$$



● Stochastic models (Markov chains, graph rewriting): copy-number of proteins, gene activity
➥ rougly same type of parameters than ODEs

● Qualitative models: coarse-grain view of activity of genes/proteins
➥ discrete parameters
➥ **Boolean networks** (~1970: Stuart Kaufman; René Thomas)

Boolean networks

definition and basic properties

Boolean networks across scientific communities

Object of study on their own... and a tool for the study of biological processes

Combinatorics

Boolean (automata) networks

Experimental biology

Theory of
dynamical systems

Theoretical biology

Formal methods

Simulation

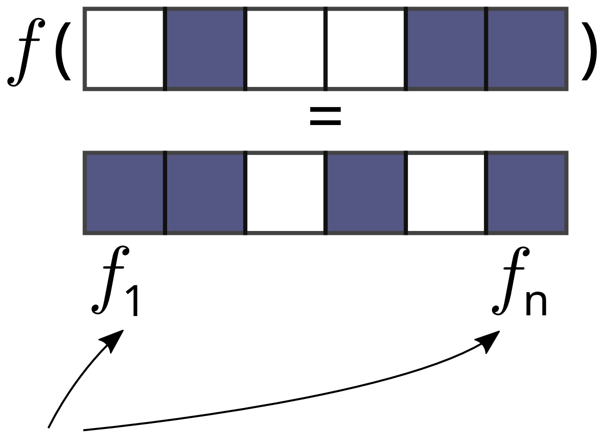
Algorithmics

Software engineering

(non-exhaustive list)

Boolean Network (BN) $f : \mathbb{B}^n \to \mathbb{B}^n$

with $\mathbb{B} = \{0, 1\} = \{\square, \blacksquare\}$



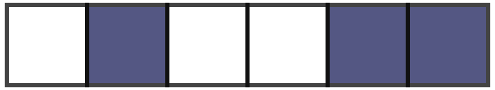
Local function of automaton i

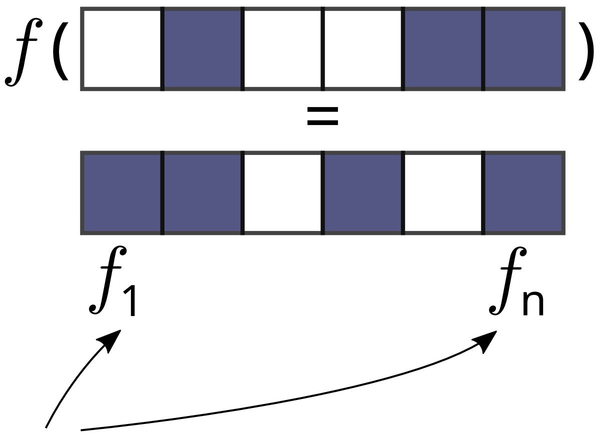
$$f_i : \mathbb{B}^n \to \mathbb{B}$$

Boolean Network (BN) $f : \mathbb{B}^n \to \mathbb{B}^n$

with $\mathbb{B} = \{0, 1\} = \{\square, \blacksquare\}$

Configuration: $\mathbf{x} \in \mathbb{B}^n$ $\mathbf{x}_i$: state of automaton i





Local function of automaton i

$$f_i : \mathbb{B}^n \to \mathbb{B}$$

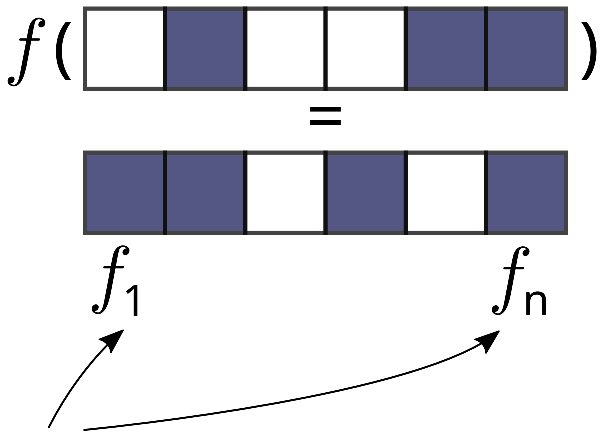
\+ updating mode =

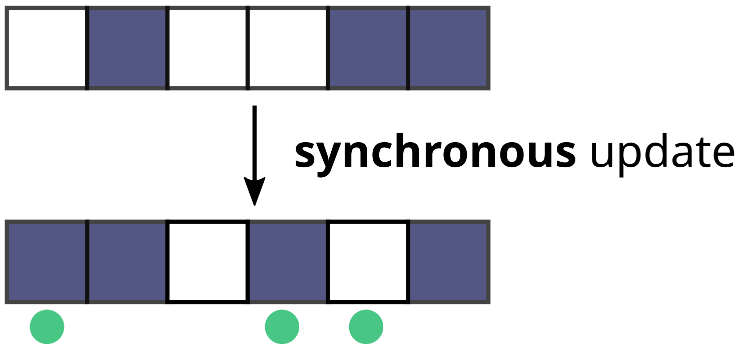
discrete dynamical system

Boolean Network (BN) $f : \mathbb{B}^n \to \mathbb{B}^n$

with $\mathbb{B} = \{0, 1\} = \{\square, \blacksquare\}$

Configuration: $\mathbf{x} \in \mathbb{B}^n$ $\quad \mathbf{x}_i$: state of automaton i





Local function of automaton i

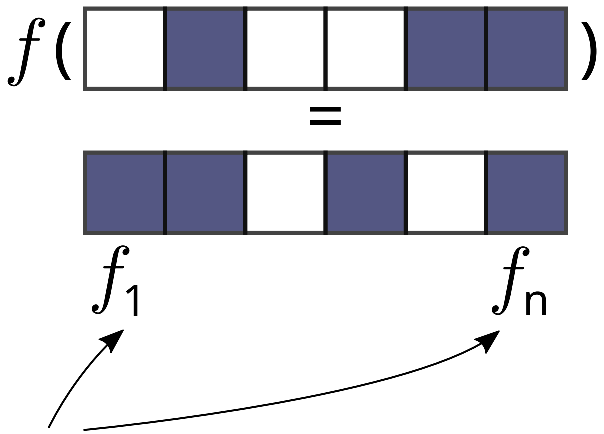
$$f_i : \mathbb{B}^n \to \mathbb{B}$$

\+ updating mode =

discrete dynamical system

Boolean Network (BN) $f : \mathbb{B}^n \to \mathbb{B}^n$

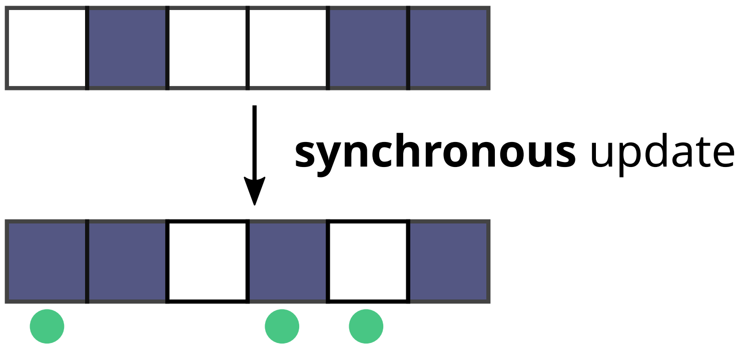
with $\mathbb{B} = \{0, 1\} = \{\square, \blacksquare\}$

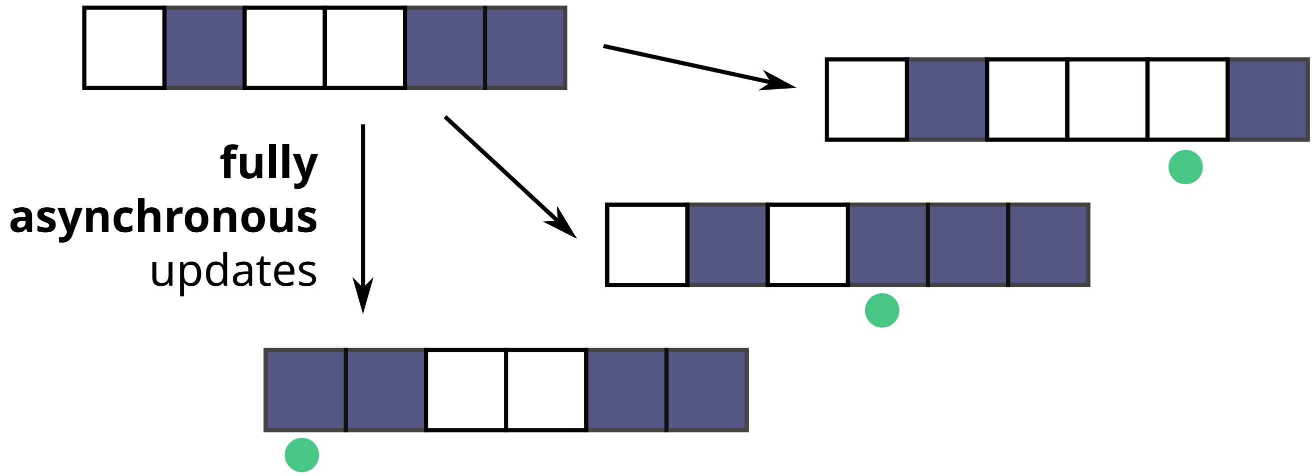


Local function of automaton i

$$f_i : \mathbb{B}^n \to \mathbb{B}$$

Configuration: $\mathbf{x} \in \mathbb{B}^n$ $\quad \mathbf{x}_i$: state of automaton i

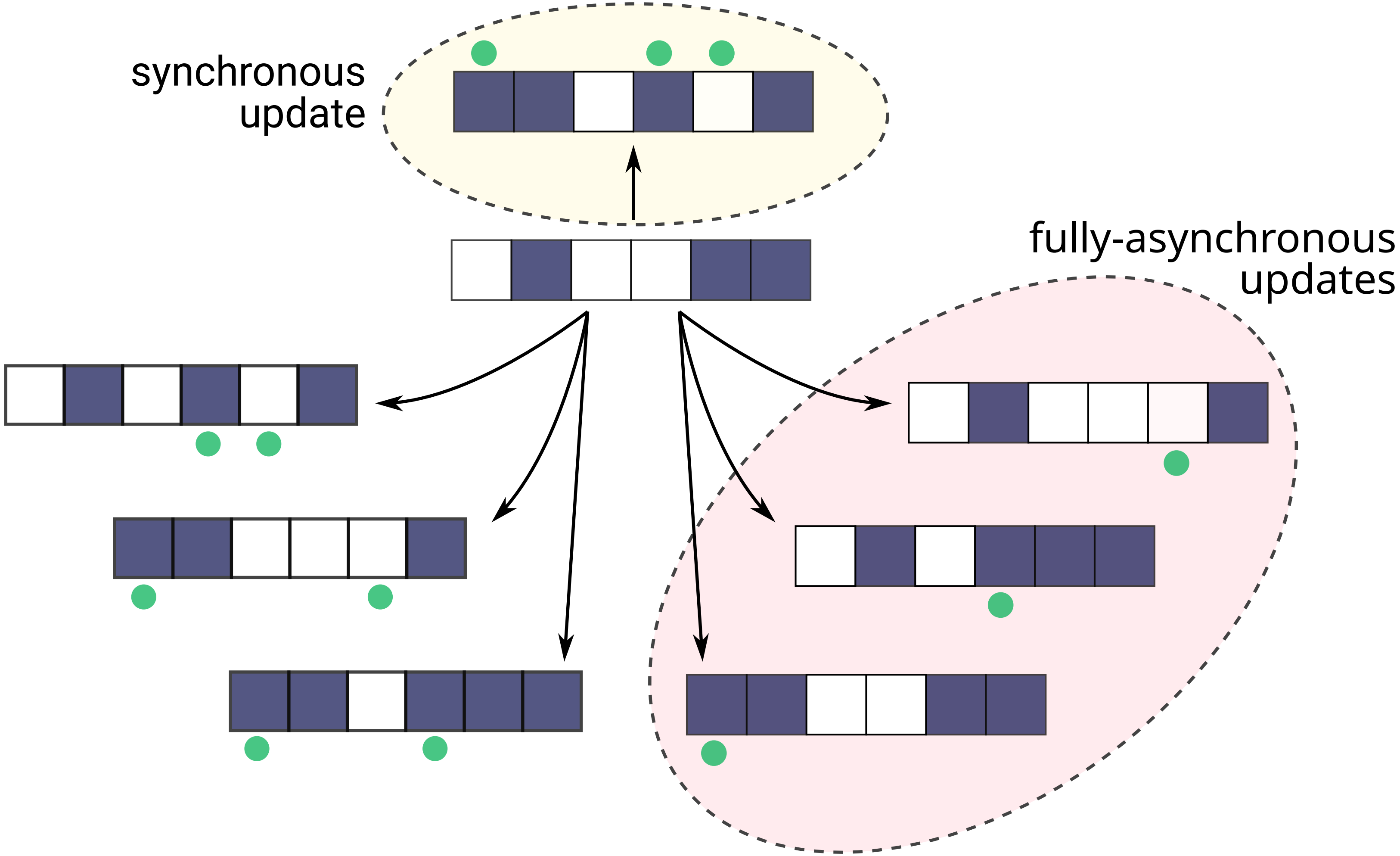




\+ updating mode =

discrete dynamical system

Asynchronous dynamics of BNs



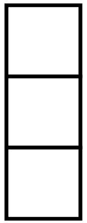
Example with asynchronous updating mode

Boolean network of dimension 3

$$f_1(\mathbf{x}) = \text{not } \mathbf{x}_2$$

$$f_2(\mathbf{x}) = \text{not } \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



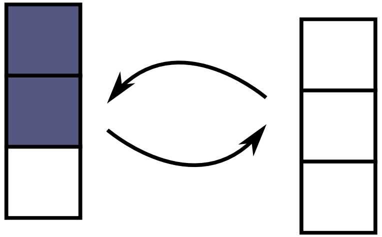
Example with asynchronous updating mode

Boolean network of dimension 3

$$f_1(\mathbf{x}) = \text{not } \mathbf{x}_2$$

$$f_2(\mathbf{x}) = \text{not } \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



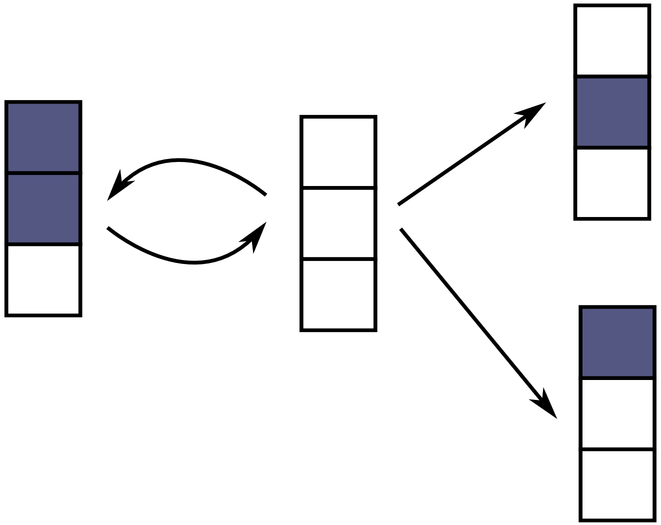
Example with asynchronous updating mode

Boolean network of dimension 3

$$f_1(\mathbf{x}) = \text{not } \mathbf{x}_2$$

$$f_2(\mathbf{x}) = \text{not } \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



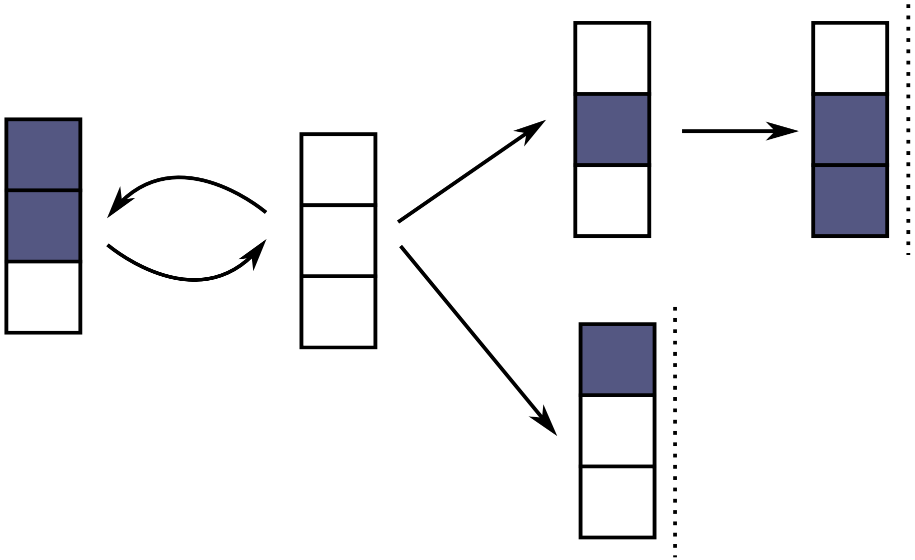
Example with asynchronous updating mode

Boolean network of dimension 3

$$f_1(\mathbf{x}) = \text{not } \mathbf{x}_2$$

$$f_2(\mathbf{x}) = \text{not } \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



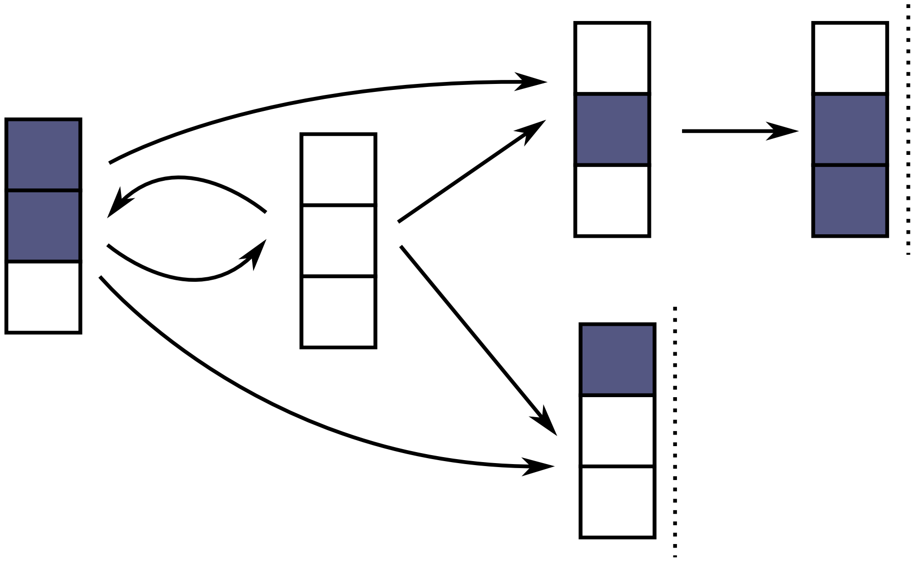
Example with asynchronous updating mode

Boolean network of dimension 3

$$f_1(\mathbf{x}) = \text{not } \mathbf{x}_2$$

$$f_2(\mathbf{x}) = \text{not } \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



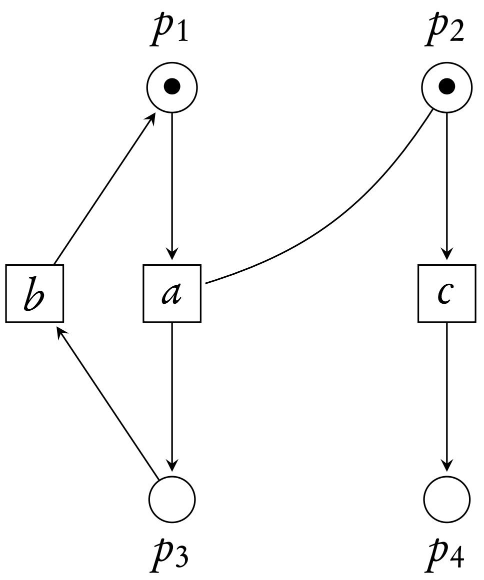
Related formalisms

Cellular automata

- Finite cellular automata are a subclass of BNs

Petri nets

- safe (1-bounded) with read arcs
- explicit conditions for state change
- steps semantics:
  synchronous = max-steps
  fully-asynchronous = atomic
  asynchronous = steps

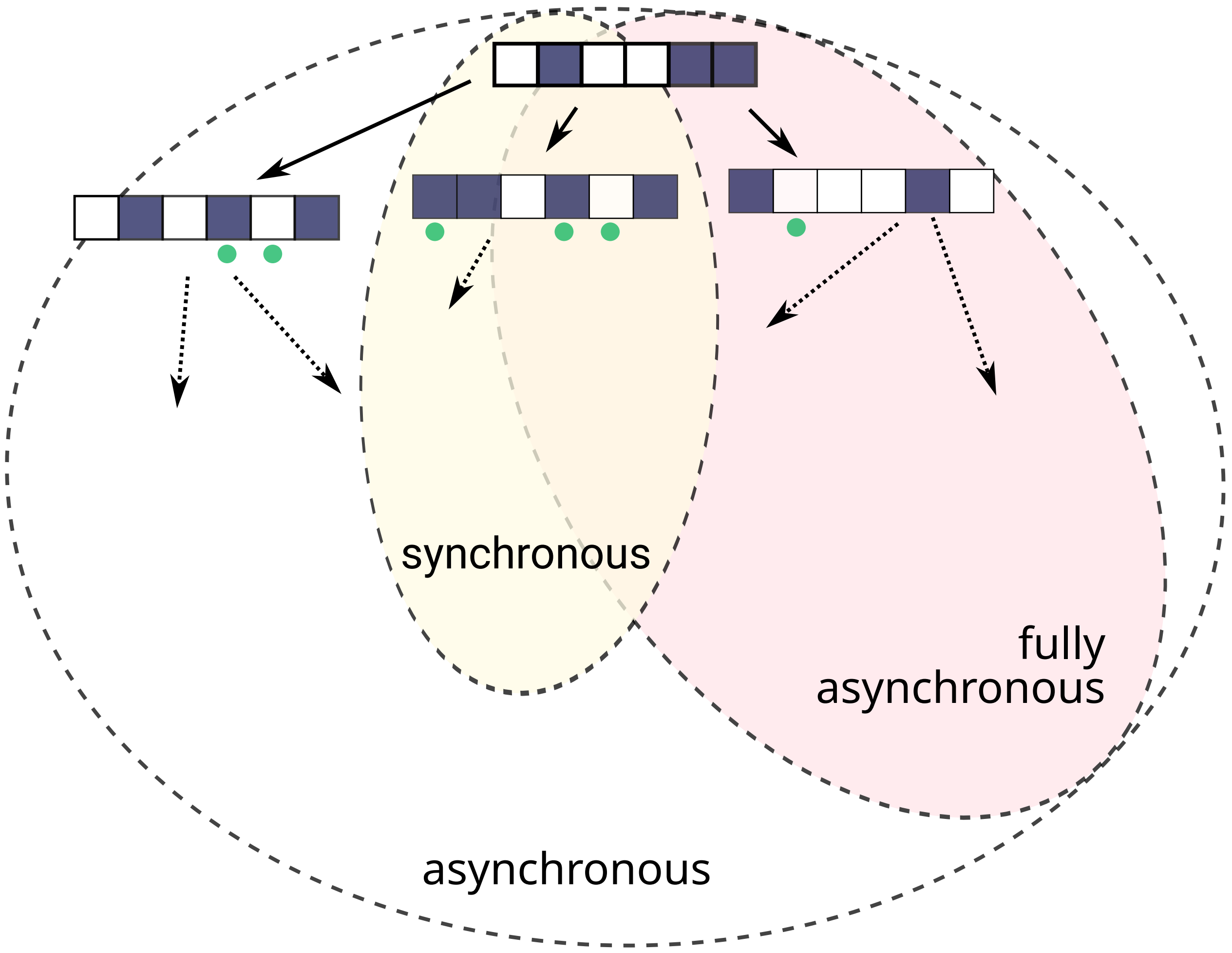


Concurrency in BNs
➥ unfoldings
➥ synchronism sensitivity

PN → BN: linear
BN → PN: requires computing implicants

w/ Haar, Chatain; see Nat. Comp. 2020

Reachable configurations



- Given a BN f and an initial configuration $\mathbf{x}$, an updating mode σ defines

$$\rho_\sigma^f : \mathbb{B}^n \to 2^{\mathbb{B}^n}$$

- For reachability, asynchronous also includes sequential, bloc-sequential, ...

but is it complete?
(w.r.t. what?)

Dynamical properties and complexity

Given a BN f and an updating mode σ...

Reachability problem
given configurations $\mathbf{x}, \mathbf{y} \in \mathbb{B}^n$
decide wether

$$\mathbf{y} \in \rho^f_\sigma(\mathbf{x})$$

with sync/fasync/async:
PSPACE-complete

Fixed point: $\rho^f_\sigma(\mathbf{x}) = \{\mathbf{x}\}$
deciding existence is NP-complete;
equiv with f(x)=x with sync/fasync/async

(f represented using propositional logic;
eval f(x) is linear with size of f)

Attractor
Non-empty set of configurations $A \subseteq \mathbb{B}^n$

$$\forall \mathbf{x} \in A, \rho^f_\sigma(\mathbf{x}) = A$$

(Terminal SCC of transition graph)
(Fixed points are singleton attractors)

In-attractor problem
Given a configuration $\mathbf{x} \in \mathbb{B}^n$
decide wether it belongs to an attractor
with sync/fasync/async:
PSPACE-complete

➥ scale limitation for verification: 50-200 automata

Architecture of a BN: influence graph

Recall that for each automaton $i \in \{1, \cdots, n\}$, $f_i : \mathbb{B}^n \to \mathbb{B}$
... but, f_i likely does not depend on all automata:

$$f_1(\mathbf{x}) = \text{not } \mathbf{x}_2$$

$$f_2(\mathbf{x}) = \text{not } \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$

Architecture of a BN: influence graph

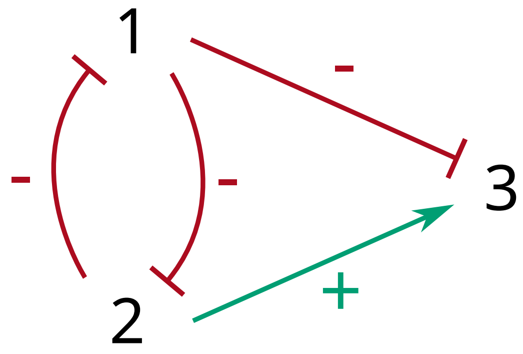
Recall that for each automaton $i \in \{1, \cdots, n\}$, $f_i : \mathbb{B}^n \to \mathbb{B}$

... but, f_i likely does not depend on all automata:

$$f_1(\mathbf{x}) = \text{not } \mathbf{x}_2$$

$$f_2(\mathbf{x}) = \text{not } \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



Influence graph: signed digraph between automata

Architecture of a BN: influence graph

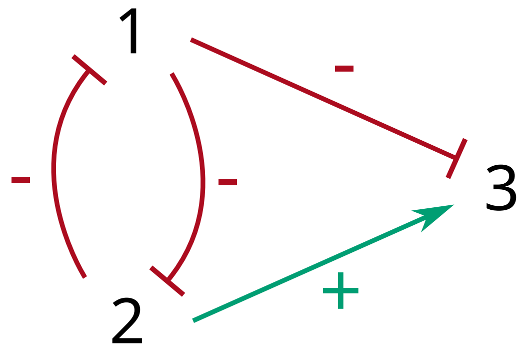
Recall that for each automaton $i \in \{1, \cdots, n\}$, $f_i : \mathbb{B}^n \to \mathbb{B}$

... but, f_i likely does not depend on all automata:

$$f_1(\mathbf{x}) = \text{not } \mathbf{x}_2$$

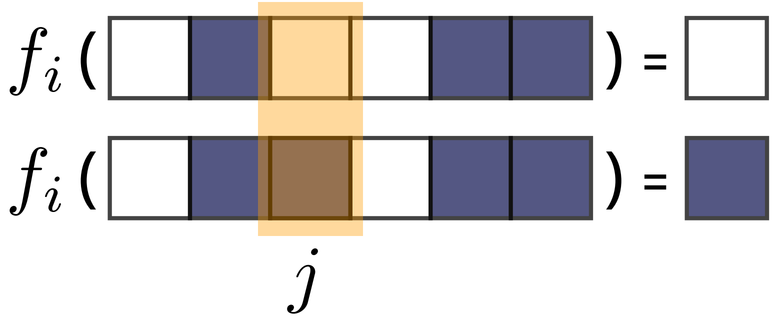
$$f_2(\mathbf{x}) = \text{not } \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



Influence graph: signed digraph between automata

If there exists at least one configuration s.t.:



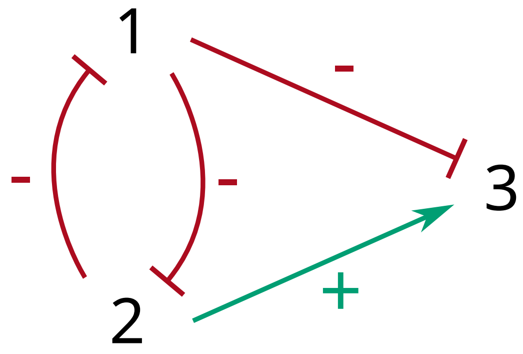
➥ j has a positive influence on i

Architecture of a BN: influence graph

Recall that for each automaton $i \in \{1, \cdots, n\}$, $f_i : \mathbb{B}^n \to \mathbb{B}$

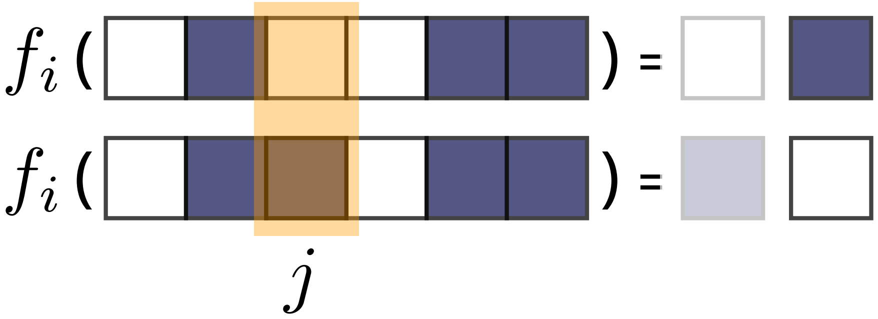
... but, f_i likely does not depend on all automata:

$$f_1(\mathbf{x}) = \text{not } \mathbf{x}_2$$
$$f_2(\mathbf{x}) = \text{not } \mathbf{x}_1$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



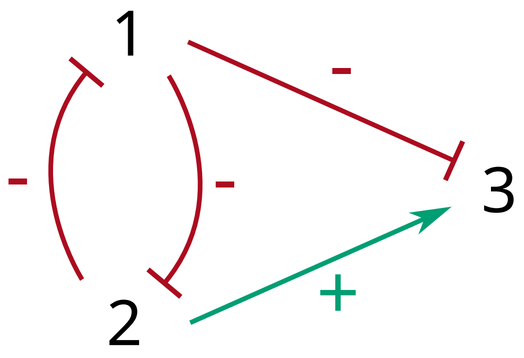
Influence graph: signed digraph between automata

If there exists at least one configuration s.t.:

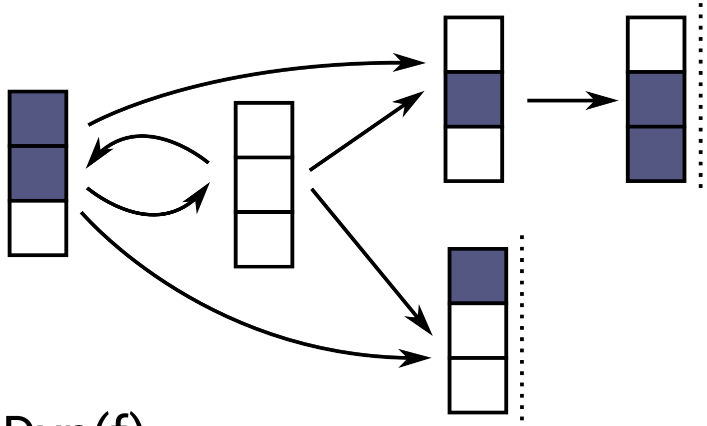


➥ j has a positive influence on i
negative

Influence graph and dynamical properties



G(f)



Dyn(f)

- having multiple attractors requires a positive cycle in G(f)
- acyclic G(f): : unique attractor is a fixed point, reachable in *n* steps
- in (fully-)async, cyclic attractors require a negative cycle
- bounds on the number of attractors, notably influence of intersection of positive cycles on upper bound

Boolean networks in practice in systems biology

are trendy ;-)

- typical queries: reachable attractors from initial conditions

↪ verify if they match with observed phenotypes

↪ compute propensities/probabilities of reaching this or that attractor

- predictions for cellular reprogramming: control of reachable attractors

- most models are designed by experts, 3 to 200 automata

↪ tedious task, complexity of verification is one of the limiting factors

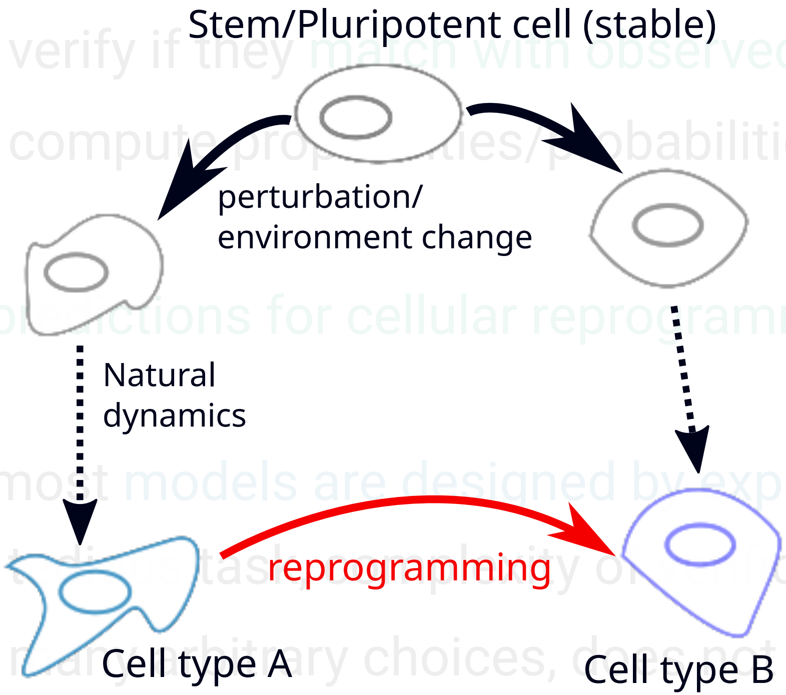
↪ many arbitrary choices, does not scale

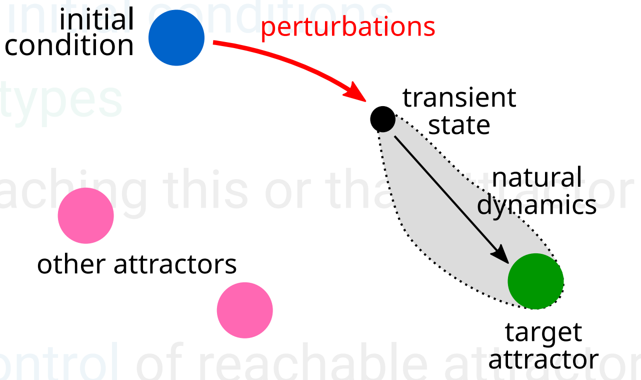
↪ direction: automatic synthesis of BNs from knowledge and observations

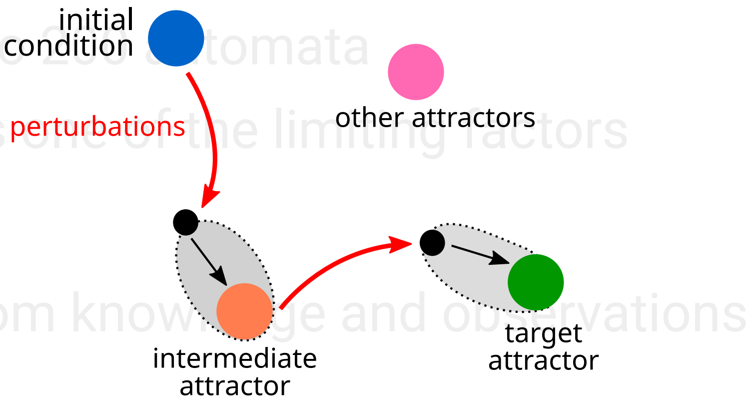
Boolean networks in practice in systems biology

are trendy ;-)

- typical queries: reachable attractors from initial conditions
  - ➥ verify if they match with observed phenotypes
  - ➥ compute probabilities of reaching this or that attractor
- predictions for cellular reprogramming: control of reachable attractors
- most models are designed by experts, 3 to 200 automata
  - ➥ tedious task, complexity of identification is one of the limiting factors
  - ➥ many arbitrary choices, does not scale
  - ➥ direction: automatic synthesis of BNs from knowledge and observations



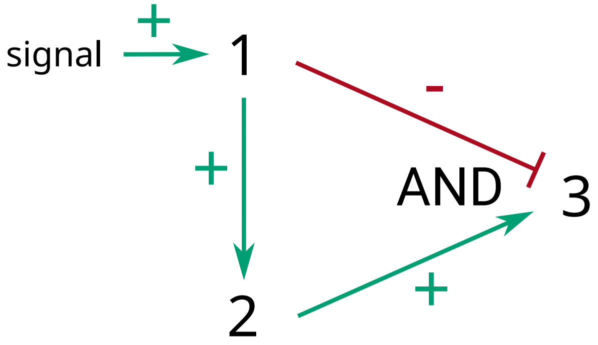




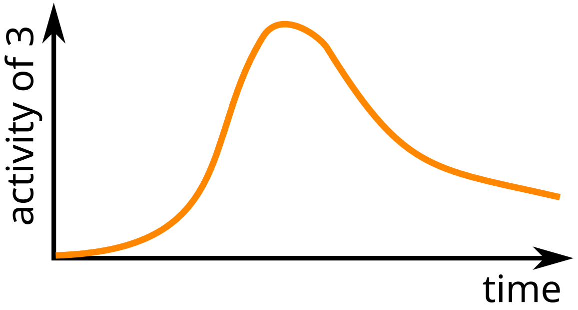
Boolean networks as abstractions of quantitative systems

Boolean modeling of the I3-FFL (Incoherent feed-forward loop)

Regulation motif



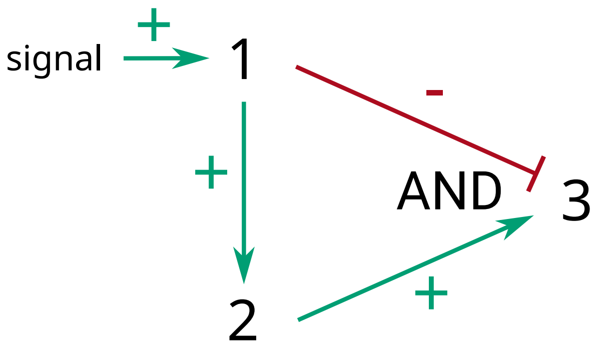
Observed output



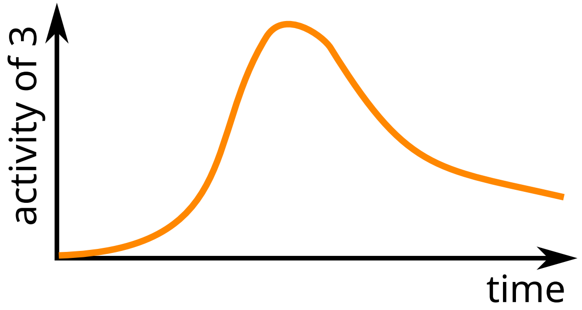
with quantitative models, and
synthetically designed DNA circuits

Boolean modeling of the I3-FFL (Incoherent feed-forward loop)

Regulation motif



Observed output



with quantitative models, and
synthetically designed DNA circuits

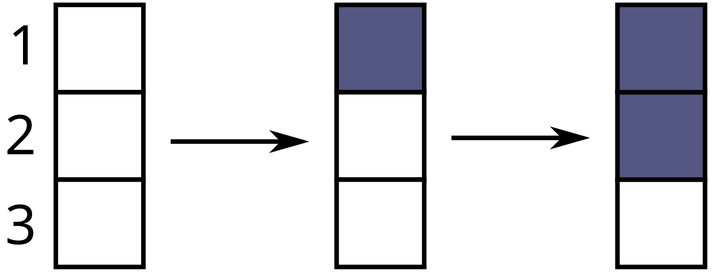
Boolean network

$$f_1(\mathbf{x}) = \text{signal}$$

$$f_2(\mathbf{x}) = \mathbf{x}_1$$

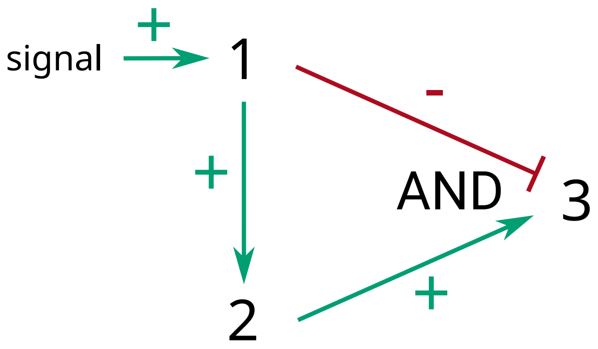
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$

Asynchronous dynamics from 000

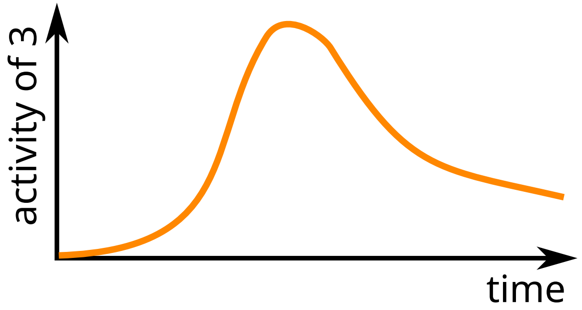


Boolean modeling of the I3-FFL (Incoherent feed-forward loop)

Regulation motif



Observed output



with quantitative models, and
synthetically designed DNA circuits

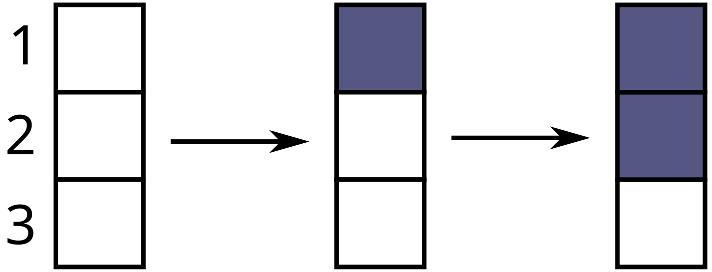
Boolean network

$$f_1(\mathbf{x}) = \text{signal}$$

$$f_2(\mathbf{x}) = \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$

Asynchronous dynamics from 000



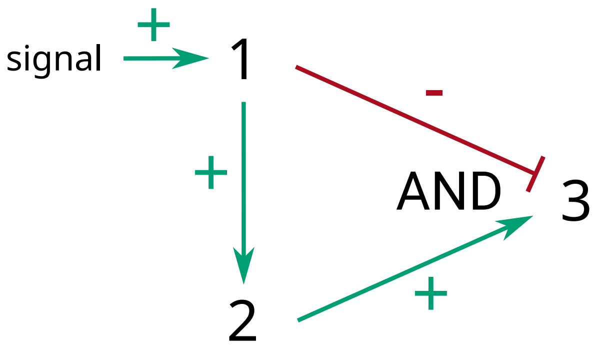
↪ impossible to activate 3...

- model validation fails but the logic is correct!
- no BN matching the motif works..

↪ incoherent abstraction for reachability...

Boolean modeling of the I3-FFL (Incoherent feed-forward loop)

Regulation motif



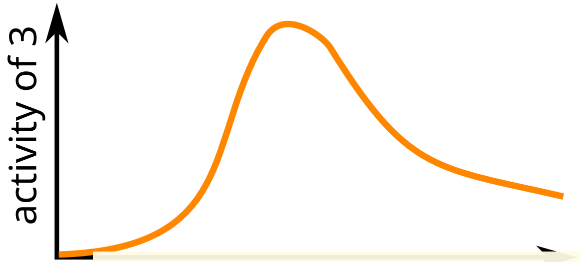
Boolean network

$$f_1(\mathbf{x}) = \mathrm{signal}$$

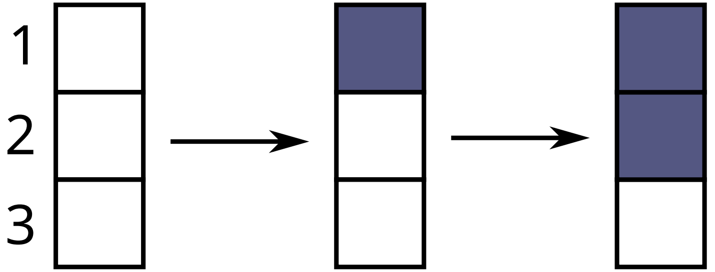
$$f_2(\mathbf{x}) = \mathbf{x}_1$$

$$f_3(\mathbf{x}) = \mathrm{not}\ \mathbf{x}_1\ \mathrm{and}\ \mathbf{x}_2$$

Observed output



Asynchronous dynamics from 000



↪ **impossible to activate 3...**

with q

synthe

correct!

ility...

- Boolean dynamics fails to capture the period when 1 is high enough to activate 2, but not high enough to inhibit 3...
- one can fix the issue with multivalued networks, or delays

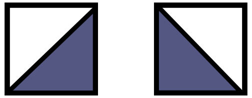
↪ adds many parameters, limiting their general application

Most Permissive Boolean networks

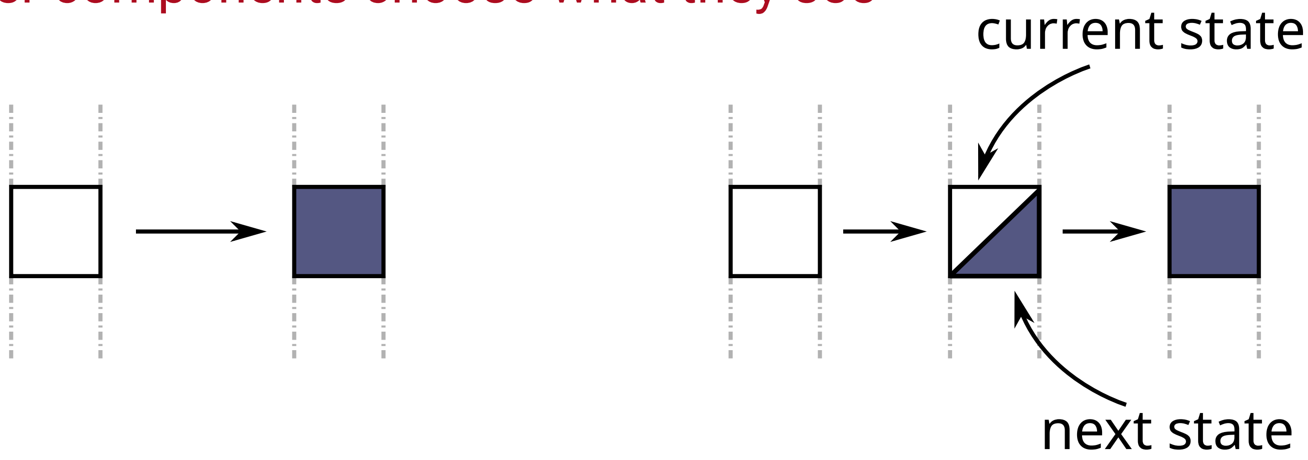
Two key ingredients:

- delay between firing and application of state change

  ↪ allow interleaving other state changes

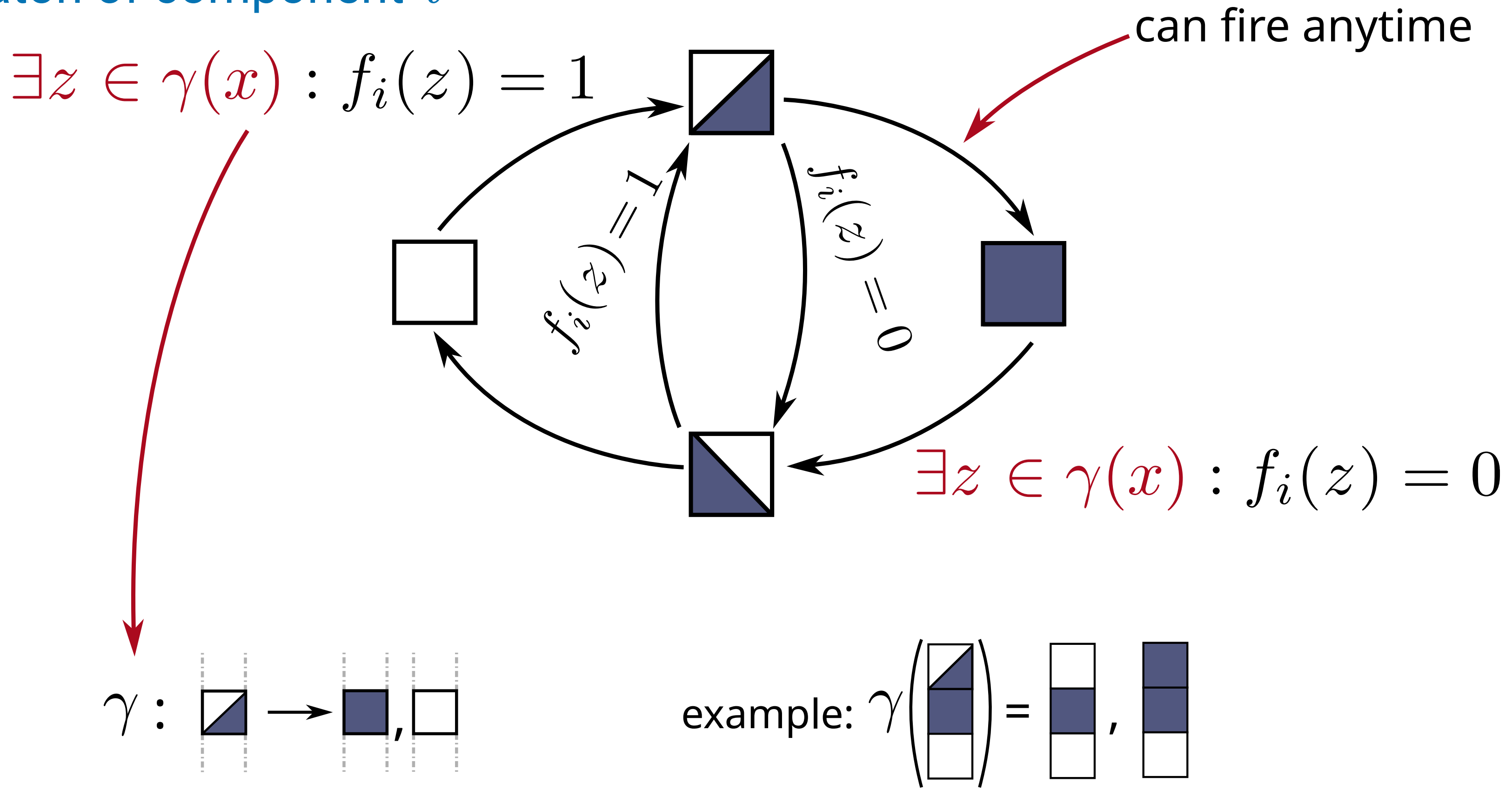
- in pseudo "dynamic" states 

  other components choose what they see



Most Permissive semantics - with pseudo dynamic states

Automaton of component i

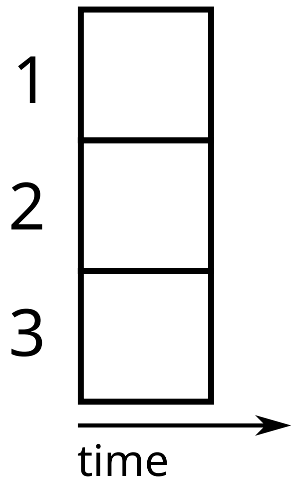


+ full-asynchronous interleaving

$$\rho^f_{\mathrm{mp}}(\mathbf{x}) := \{\mathbf{y} \in \mathbb{B}^n \mid \mathbf{x} \xrightarrow[\mathrm{mp}]{f}{}^* \mathbf{y}\}$$

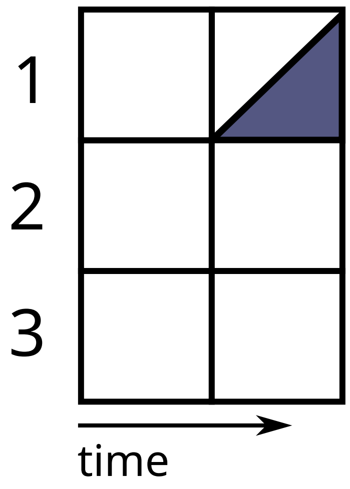
Most Permissive semantics - example of trajectory

$$f_1(\mathbf{x}) = \text{signal}$$
$$f_2(\mathbf{x}) = \mathbf{x}_1$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



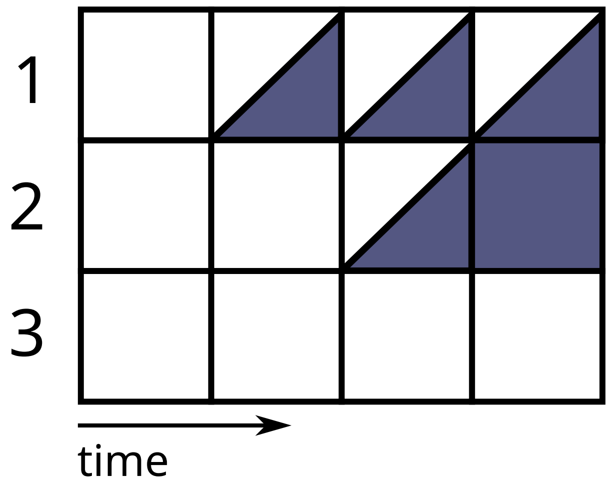
Most Permissive semantics - example of trajectory

$$f_1(\mathbf{x}) = \text{signal}$$
$$f_2(\mathbf{x}) = \mathbf{x}_1$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



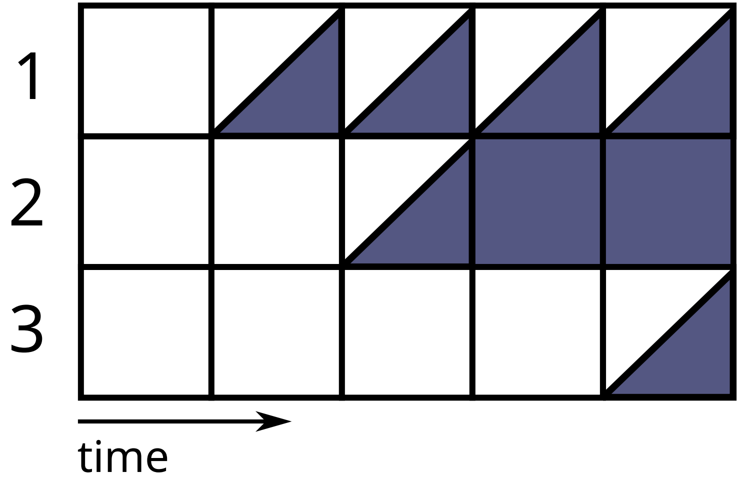
Most Permissive semantics - example of trajectory

$$f_1(\mathbf{x}) = \text{signal}$$
$$f_2(\mathbf{x}) = \mathbf{x}_1$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



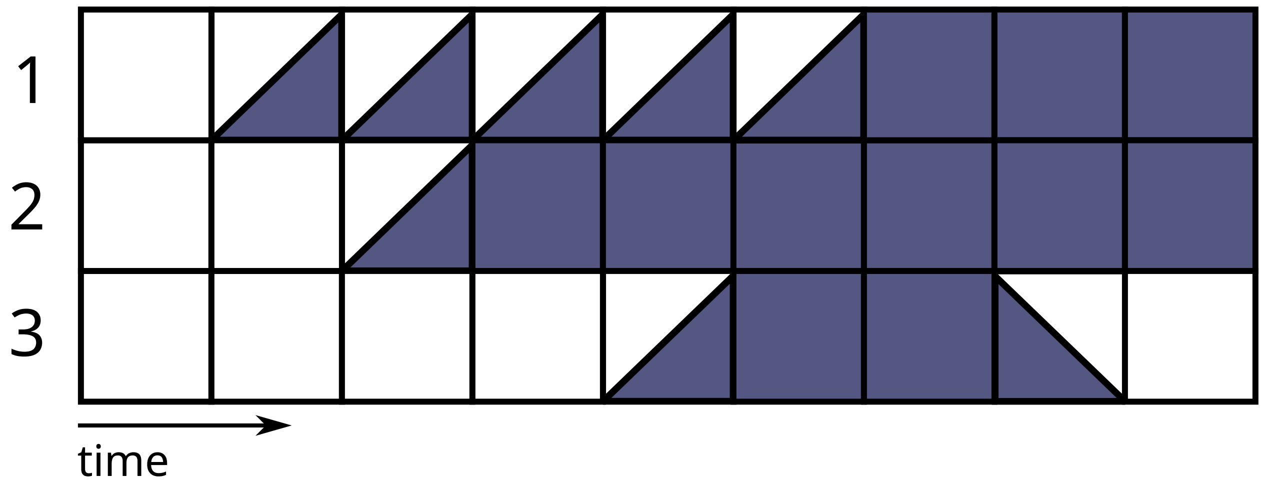
Most Permissive semantics - example of trajectory

$$f_1(\mathbf{x}) = \text{signal}$$
$$f_2(\mathbf{x}) = \mathbf{x}_1$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



Most Permissive semantics - example of trajectory

$$f_1(\mathbf{x}) = \text{signal}$$
$$f_2(\mathbf{x}) = \mathbf{x}_1$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_1 \text{ and } \mathbf{x}_2$$



(A)synchronous Boolean Networks

Bad abstractions of non-binary systems

- can miss behaviors...

... also includes stochastic methods

- impact reachable attractors: one can wrongly conclude an attractor is not reachable

Costly to analyze

- reachability and attractor properties are PSPACE-complete
- usually limited to 50-200 automata then requires approximations..

Most Permissive Boolean Networks (MPBNs)

Paulevé et al, Nature Communications, 2020

Complete abstraction

- guarantees not to miss any behavior achievable by a quantitative model following the same logic
- remains stringent enough to capture differentiation processes

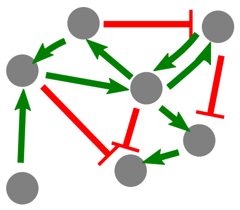
Highly scalable

- reachability: P/P^{NP}; attractor: $coNP/coNP^{coNP}$
- benchmarks with 100,000 automata
- unlocks large-scale BN inference

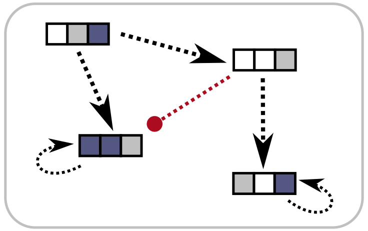
No additional parameters!

Synthesis of ensembles of BNs for reprogramming

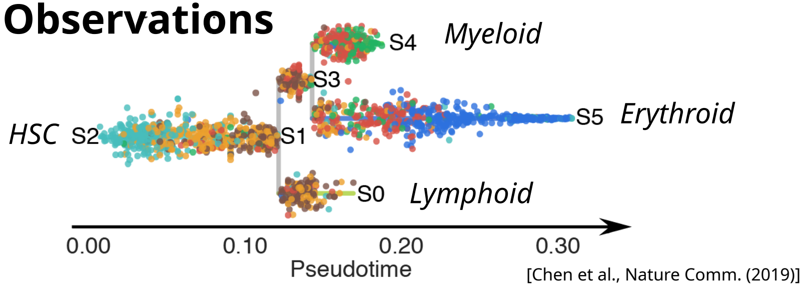
Architecture constraints
(databases, data analysis)



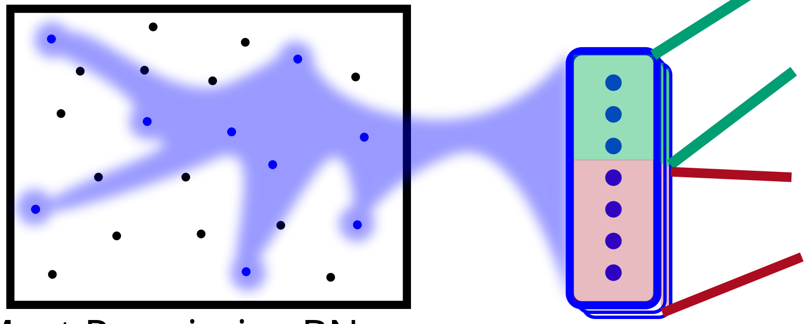
Dynamical properties



Observations



(Implicit) **formal BN space**



Most Permissive BNs verifying architecture and dynamical properties

Sampling
(w/ diversity)

Predictions (control)
on individual models

A++/B--
C--/D++

Scoring
accross different profiles

A++/B-- 51%
C--/D++ 31%

Work in progress
- Logic programming (Answer-Set Programming)
- Sampling with diversity
- Genome-scale

➥ tool BoNesis
github.com/bioasp/bonesis