Phenotype control and elimination of variables in Boolean networks

Elisa Tonello, Loïc Paulevé

Freie Universitat Berlin
LaBRI, CNRS, Bordeaux, France

Published in Peer Community Journal, 2024 doi:10.24072/pcjournal.452
(recommended by PCI Mathematical & Computational Biology open peer review)

Overview

Motivation

- Model reduction is common practice, including in Boolean modeling community
  - ➔ especially for dimension reduction (variable elimination/lumping)
  - ➔ usual fallback strategy when initial model is too large
- Control/reprogramming is a prominent application of Boolean networks
  - ➔ predict mutations/interventions to enforce long-term properties
- Question: can we do both at the same time?

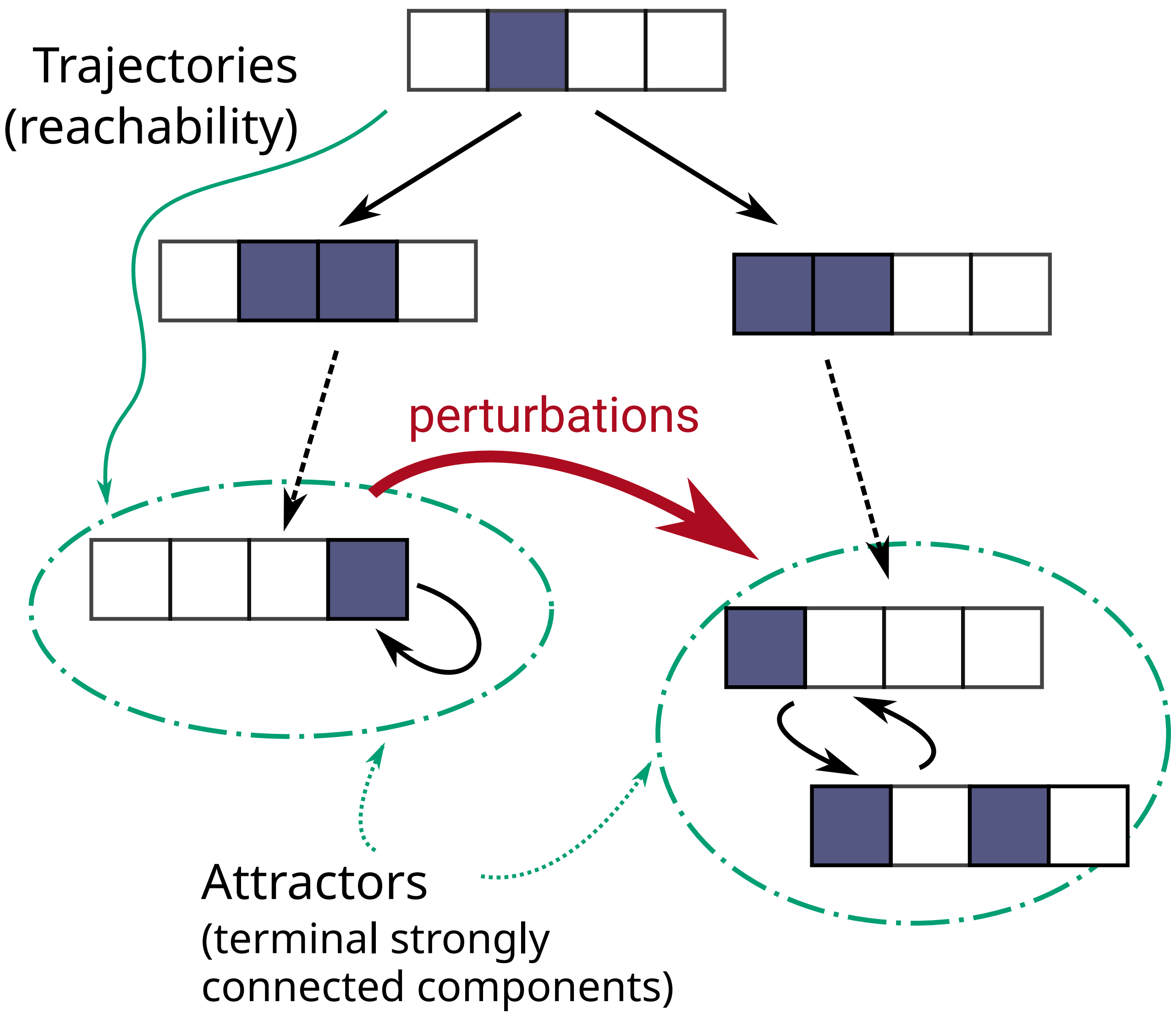
This talk

- Theoretical results on robustness of control predictions to a usual model reduction
- Spoiler: mostly negative results 😥 EXCEPT for a few settings 😅

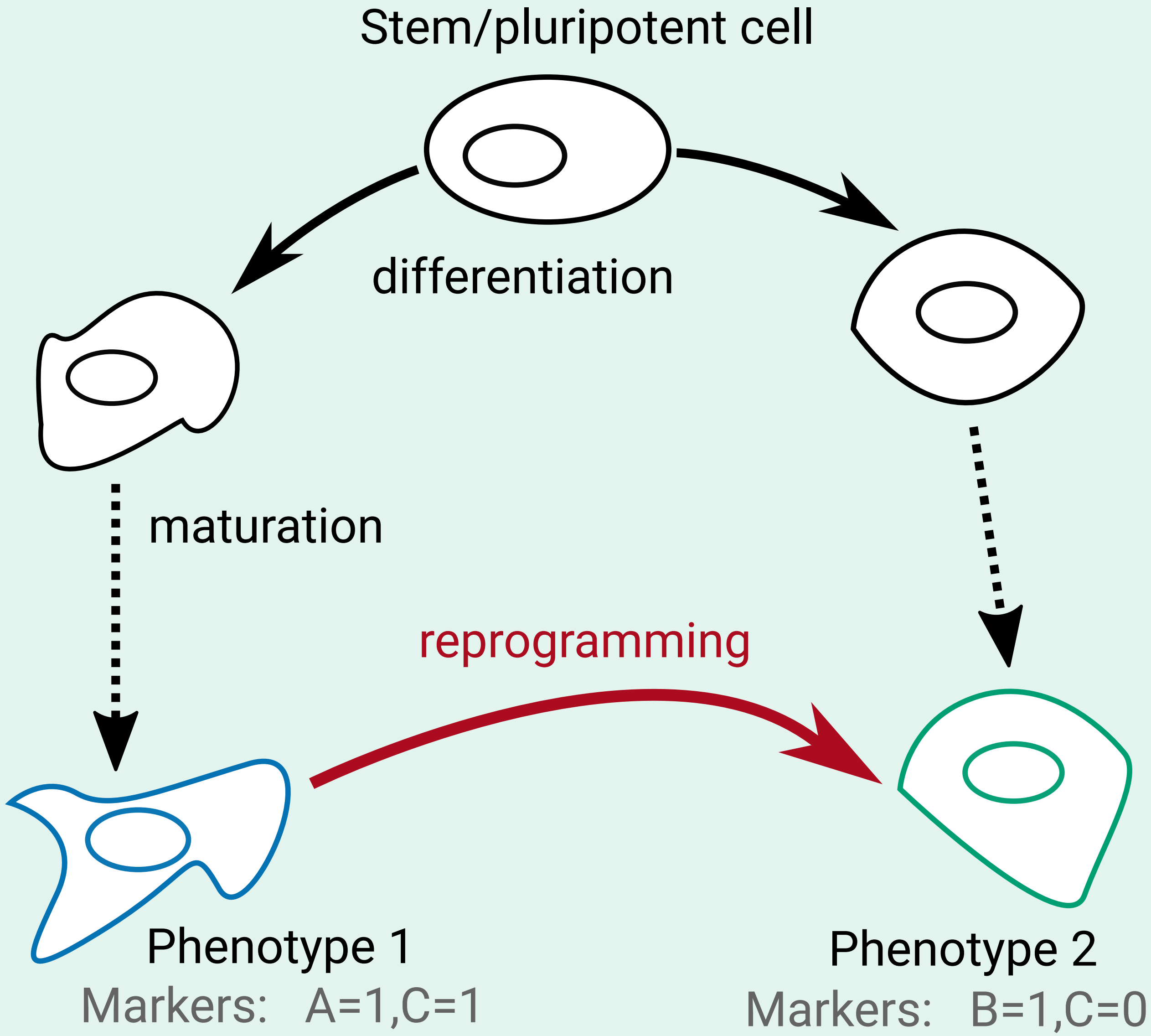
Boolean networks as models of biological processes

Boolean network dynamics

$f : \mathbb{B}^n \to \mathbb{B}^n$ + update mode



Cellular differentiation/fate decision



Boolean networks: definition, dynamics

Function $f : \mathbb{B}^n \to \mathbb{B}^n$ with $\mathbb{B} = \{0, 1\}$

$f_i : \mathbb{B}^n \to \mathbb{B}$ is the local function of component $i \in \{1, ..., n\}$

Configuration: $x \in \mathbb{B}^n$ x_i is the state of component i

+ semantics (update mode) for computing next configurations

= discrete dynamical system

Example with $n = 3$

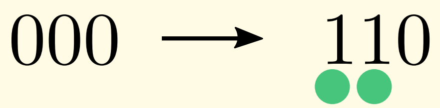
$$f_1(x) = \neg x_2$$

$$f_2(x) = \neg x_1$$

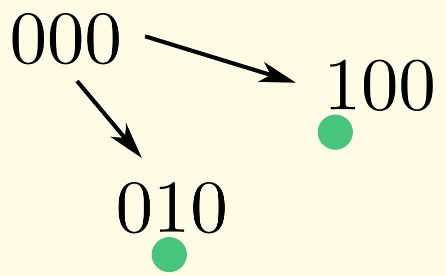
$$f_3(x) = \neg x_1 \wedge x_2$$

$$f(000) = 110$$

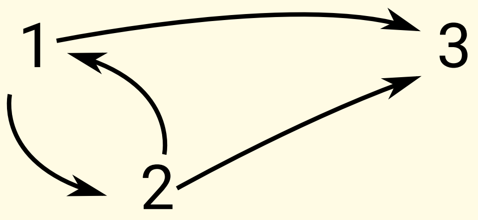
synchronous (parallel) transition



fully asynchronous transition



Influence graph



Attractors

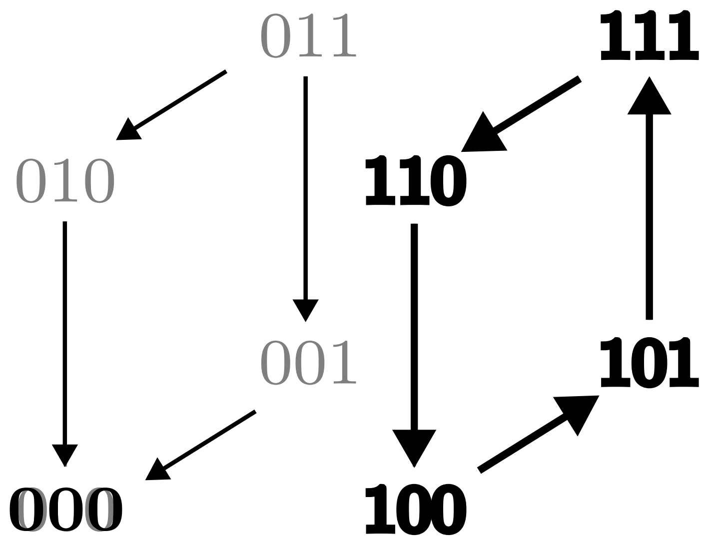
Property of dynamics of f

Definition

Subset-smallest set of configurations closed by transitions

- ➥ terminal strongly connected components of transition graph
- ➥ fixed points: attractors with a single configuration
- ➥ highly dependent on the update mode

$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$



Fully asynchronous

$$f(110) = 100$$
$$f(100) = 101$$
$$f(101) = 111$$
$$f(111) = 110$$

Trap spaces of Boolean networks

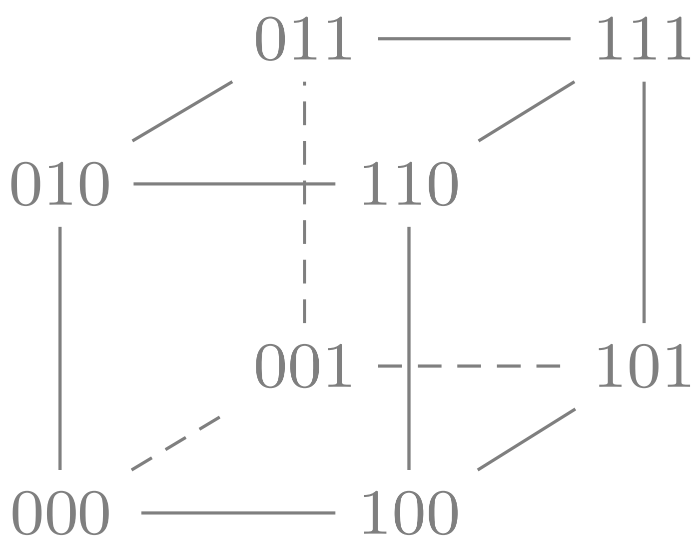
Property of f — independant of the update mode

Definition

A trap space is a subcube of $\mathbb{B}^n$ closed by f

➥ subcube can be characterized as a vector in $\{0,1,*\}^n$

➥ subcube T is a trap space of f iff $\forall \mathbf{x} \in \mathrm{T},\ f(\mathbf{x}) \in \mathrm{T}$



$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$

Minimal trap spaces:

Smallest trap spaces by vertices inclusion

➥ Fixed points of f are minimal trap spaces

➥ Each minimal trap space encloses at least one attractor with any update mode

- Minimal trap spaces are exactly the attractors of the Most Permissive update mode (which accounts for quantitative refinements of the BNs) [P et al., Nature Comm 2020] also 1-1 with update mode capturing single-threshold Multivalued networks [Naldi et al., Natural Comp. 2023]

  ➥ **relevant and robust feature for reasoning on long-term dynamics**

Trap spaces of Boolean networks

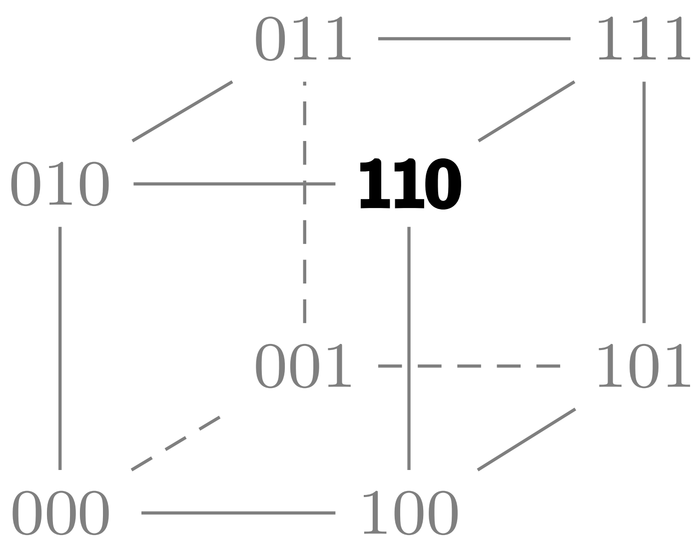
Property of f — independant of the update mode

Definition

A trap space is a subcube of $\mathbb{B}^n$ closed by f

➥ subcube can be characterized as a vector in $\{0,1,*\}^n$

➥ subcube T is a trap space of f iff $\forall \mathbf{x} \in \mathrm{T},\ f(\mathbf{x}) \in \mathrm{T}$



$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$

$$f(110) = 100$$
$$f(100) = 101$$
$$f(101) = 111$$
$$f(111) = 110$$

Minimal trap spaces:

Smallest trap spaces by vertices inclusion

➥ Fixed points of f are minimal trap spaces

➥ Each minimal trap space encloses at least one attractor with any update mode

- Minimal trap spaces are exactly the attractors of the Most Permissive update mode (which accounts for quantitative refinements of the BNs) [P et al., Nature Comm 2020] also 1-1 with update mode capturing single-threshold Multivalued networks [Naldi et al., Natural Comp. 2023]

  ➥ **relevant and robust feature for reasoning on long-term dynamics**

Trap spaces of Boolean networks

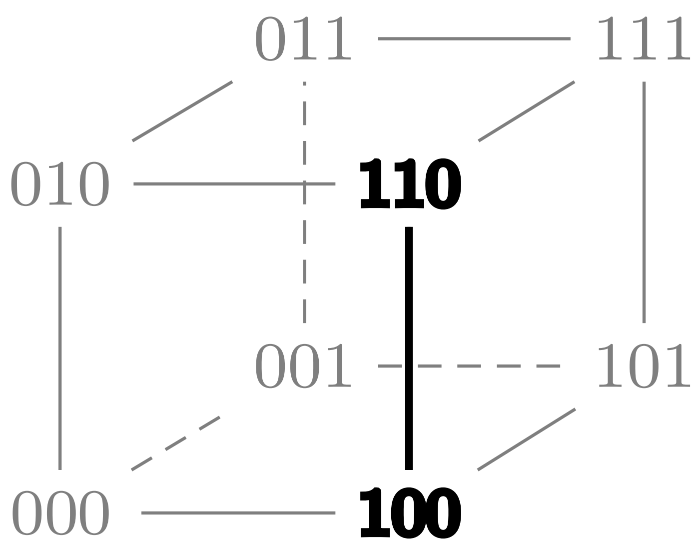
Property of f — independant of the update mode

Definition

A trap space is a subcube of $\mathbb{B}^n$ closed by f

➥ subcube can be characterized as a vector in $\{0,1,*\}^n$

➥ subcube T is a trap space of f iff $\forall \mathbf{x} \in \mathrm{T},\ f(\mathbf{x}) \in \mathrm{T}$



$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$

$$f(110) = 100$$
$$f(100) = 101$$
$$f(101) = 111$$
$$f(111) = 110$$

Minimal trap spaces:

Smallest trap spaces by vertices inclusion

➥ Fixed points of f are minimal trap spaces

➥ Each minimal trap space encloses at least one attractor with any update mode

- Minimal trap spaces are exactly the attractors of the Most Permissive update mode (which accounts for quantitative refinements of the BNs) [P et al., Nature Comm 2020]
  also 1-1 with update mode capturing single-threshold Multivalued networks [Naldi et al., Natural Comp. 2023]

  ➥ **relevant and robust feature for reasoning on long-term dynamics**

Trap spaces of Boolean networks

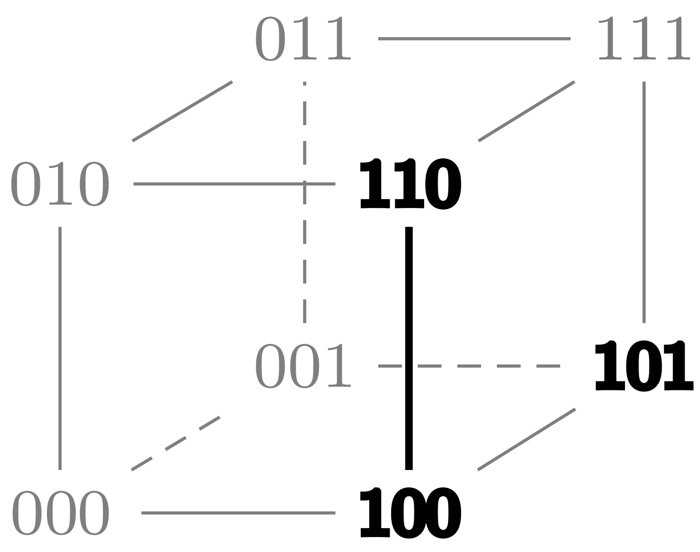
Property of f — independant of the update mode

Definition

A trap space is a subcube of $\mathbb{B}^n$ closed by f

➥ subcube can be characterized as a vector in $\{0,1,*\}^n$

➥ subcube T is a trap space of f iff $\forall \mathbf{x} \in \mathrm{T},\ f(\mathbf{x}) \in \mathrm{T}$



$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$

$$f(110) = 100$$
$$f(100) = 101$$
$$f(101) = 111$$
$$f(111) = 110$$

Minimal trap spaces:

Smallest trap spaces by vertices inclusion

➥ Fixed points of f are minimal trap spaces

➥ Each minimal trap space encloses at least one attractor with any update mode

- Minimal trap spaces are exactly the attractors of the Most Permissive update mode (which accounts for quantitative refinements of the BNs) [P et al., Nature Comm 2020] also 1-1 with update mode capturing single-threshold Multivalued networks [Naldi et al., Natural Comp. 2023]

  ➥ **relevant and robust feature for reasoning on long-term dynamics**

Trap spaces of Boolean networks

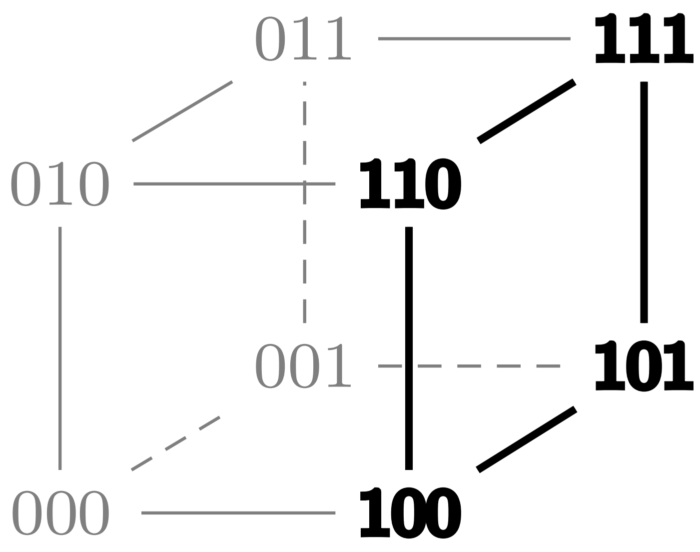
Property of f — independant of the update mode

Definition

A trap space is a subcube of $\mathbb{B}^n$ closed by f

➥ subcube can be characterized as a vector in $\{0,1,*\}^n$

➥ subcube T is a trap space of f iff $\forall \mathbf{x} \in \mathrm{T},\ f(\mathbf{x}) \in \mathrm{T}$



$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$

$$f(110) = 100$$
$$f(100) = 101$$
$$f(101) = 111$$
$$f(111) = 110$$

Minimal trap spaces:

Smallest trap spaces by vertices inclusion

➥ Fixed points of f are minimal trap spaces

➥ Each minimal trap space encloses at least one attractor with any update mode

- Minimal trap spaces are exactly the attractors of the Most Permissive update mode (which accounts for quantitative refinements of the BNs) [P et al., Nature Comm 2020] also 1-1 with update mode capturing single-threshold Multivalued networks [Naldi et al., Natural Comp. 2023]

  ➥ **relevant and robust feature for reasoning on long-term dynamics**

Trap spaces of Boolean networks

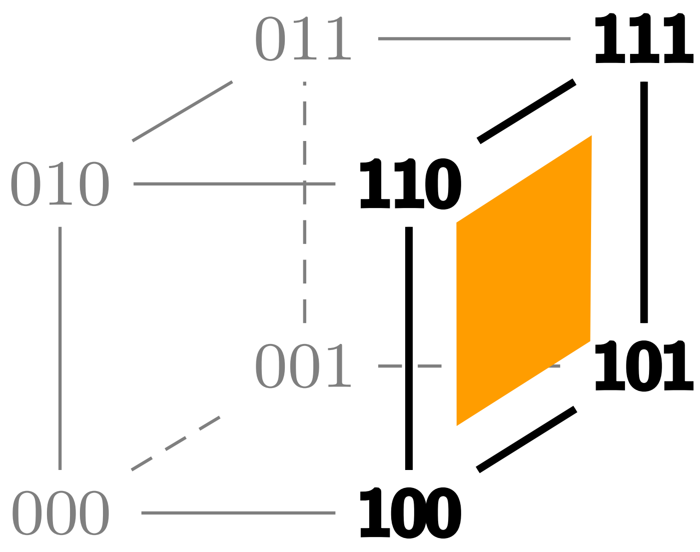
Property of f — independant of the update mode

Definition

A trap space is a subcube of $\mathbb{B}^n$ closed by f

➥ subcube can be characterized as a vector in $\{0,1,*\}^n$

➥ subcube T is a trap space of f iff $\forall \mathbf{x} \in \mathrm{T},\ f(\mathbf{x}) \in \mathrm{T}$



$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$

$$f(110) = 100$$
$$f(100) = 101$$
$$f(101) = 111$$
$$f(111) = 110$$

Minimal trap spaces:

Smallest trap spaces by vertices inclusion

➥ Fixed points of f are minimal trap spaces

➥ Each minimal trap space encloses at least one attractor with any update mode

- Minimal trap spaces are exactly the attractors of the Most Permissive update mode (which accounts for quantitative refinements of the BNs) [P et al., Nature Comm 2020] also 1-1 with update mode capturing single-threshold Multivalued networks [Naldi et al., Natural Comp. 2023]

➥ **relevant and robust feature for reasoning on long-term dynamics**

Trap spaces of Boolean networks

Property of f — independant of the update mode

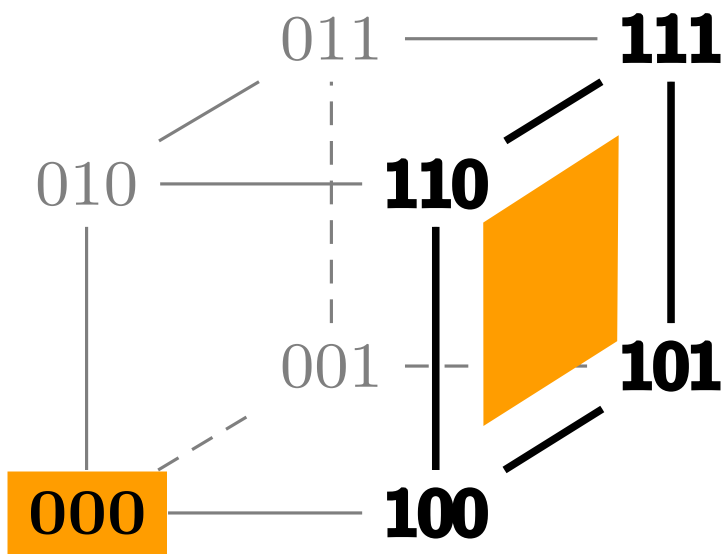
Definition

A trap space is a subcube of $\mathbb{B}^n$ closed by f

➥ subcube can be characterized as a vector in $\{0,1,*\}^n$

➥ subcube T is a trap space of f iff $\forall \mathbf{x} \in \mathrm{T},\ f(\mathbf{x}) \in \mathrm{T}$

$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$



$$f(110) = 100$$
$$f(100) = 101$$
$$f(101) = 111$$
$$f(111) = 110$$

Minimal trap spaces:

Smallest trap spaces by vertices inclusion

➥ Fixed points of f are minimal trap spaces

➥ Each minimal trap space encloses at least one attractor with any update mode

- Minimal trap spaces are exactly the attractors of the Most Permissive update mode (which accounts for quantitative refinements of the BNs) [P et al., Nature Comm 2020] also 1-1 with update mode capturing single-threshold Multivalued networks [Naldi et al., Natural Comp. 2023]

➥ relevant and robust feature for reasoning on long-term dynamics

Trap spaces of Boolean networks

Property of f — independant of the update mode

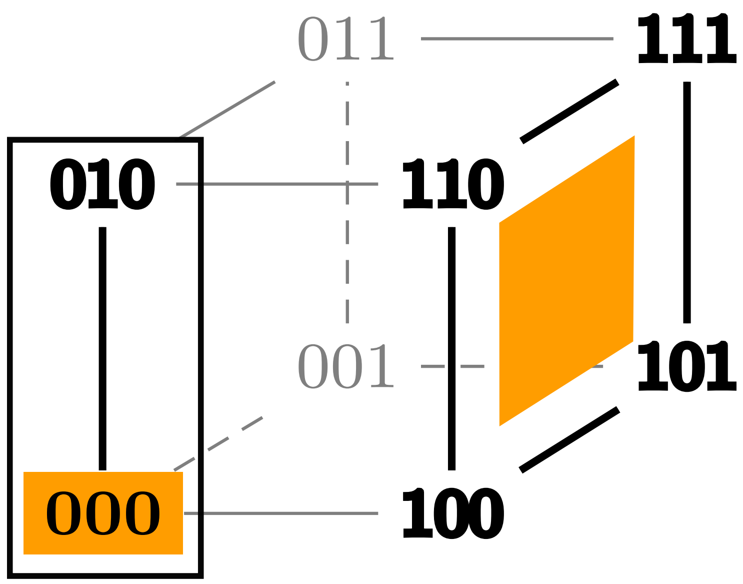
Definition

A trap space is a subcube of $\mathbb{B}^n$ closed by f

➥ subcube can be characterized as a vector in $\{0,1,*\}^n$

➥ subcube T is a trap space of f iff $\forall \mathbf{x} \in \mathrm{T},\ f(\mathbf{x}) \in \mathrm{T}$

$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$



$$f(110) = 100$$
$$f(100) = 101$$
$$f(101) = 111$$
$$f(111) = 110$$

Minimal trap spaces:

Smallest trap spaces by vertices inclusion

➥ Fixed points of f are minimal trap spaces

➥ Each minimal trap space encloses at least one attractor with any update mode

- Minimal trap spaces are exactly the attractors of the Most Permissive update mode (which accounts for quantitative refinements of the BNs) [P et al., Nature Comm 2020] also 1-1 with update mode capturing single-threshold Multivalued networks [Naldi et al., Natural Comp. 2023]

  ➥ **relevant and robust feature for reasoning on long-term dynamics**

Phenotype control / reprogramming

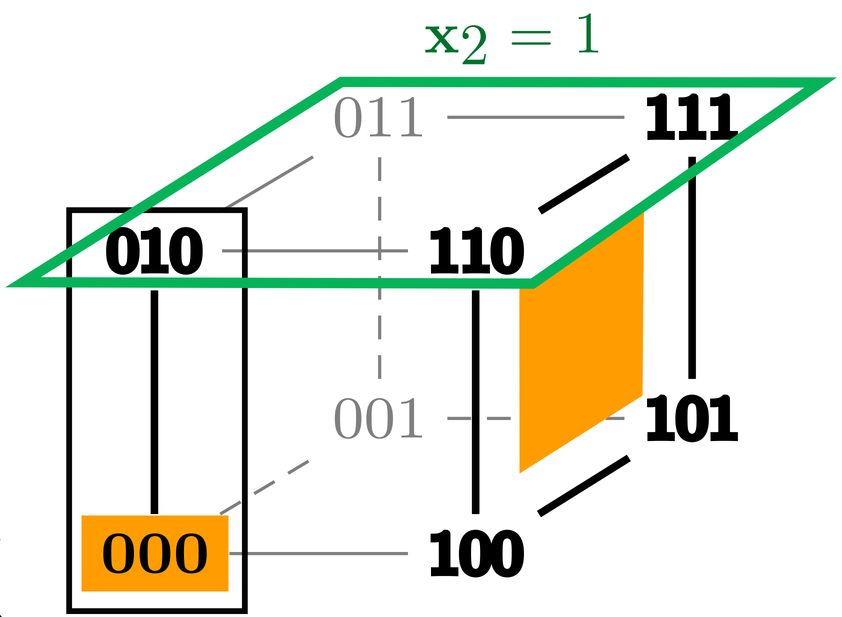
Phenotype: subcube P - fixed components are markers of phenotype

Control strategy: subcube S - freeze some components (different than phenotype) to a fixed value

We note C(f,S) the controled BN

- control of attractors (for a given update mode)
  ➥ all attractors of C(f,S) are within P
- control of minimal trap spaces (= attractors of MP)
  ➥ all minimal trap spaces of C(f,S) are within P
- control by value propagation only
  ➥ constant propagation of C(f,S) falls within P

Example with $\mathrm{P} = {*}1{*}$ and $\mathrm{S} = {*}{*}1$



$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1$$

Phenotype control / reprogramming

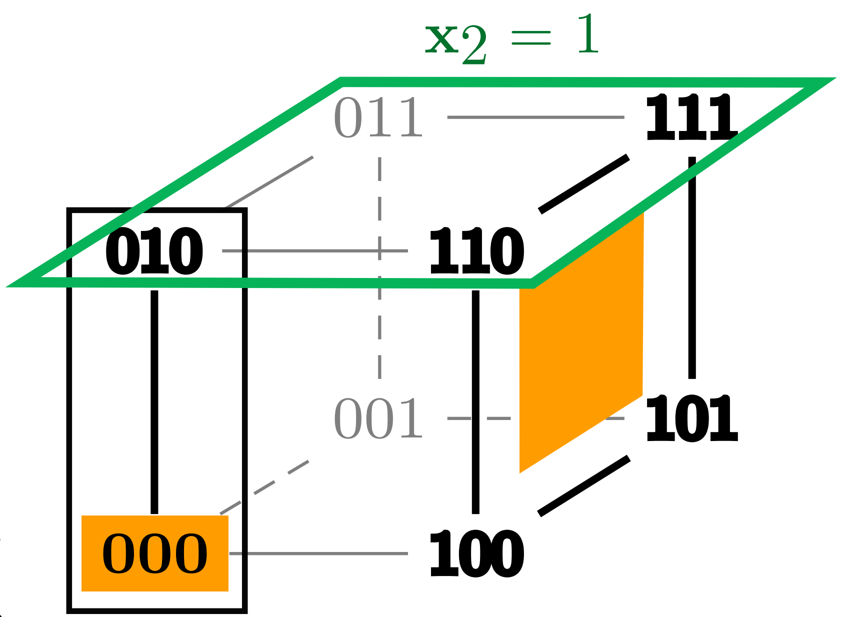
Phenotype: subcube P - fixed components are markers of phenotype

Control strategy: subcube S - freeze some components (different than phenotype) to a fixed value

We note C(f,S) the controled BN

- control of attractors (for a given update mode)
  ➥ all attractors of C(f,S) are within P
- control of minimal trap spaces (= attractors of MP)
  ➥ all minimal trap spaces of C(f,S) are within P
- control by value propagation only
  ➥ constant propagation of C(f,S) falls within P

Example with $\mathrm{P} = {*}1{*}$ and $\mathrm{S} = {*}{*}1$



$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \sout{\text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1}\ 1$$

Phenotype control / reprogramming

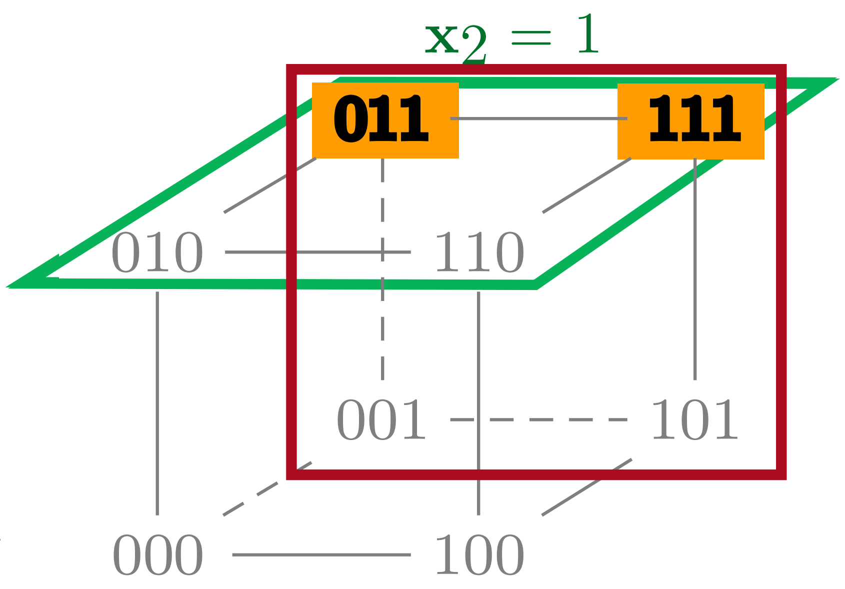
Phenotype: subcube P - fixed components are markers of phenotype

Control strategy: subcube S - freeze some components (different than phenotype) to a fixed value

We note C(f,S) the controled BN

- control of attractors (for a given update mode)
  ➥ all attractors of C(f,S) are within P
- control of minimal trap spaces (= attractors of MP)
  ➥ all minimal trap spaces of C(f,S) are within P
- control by value propagation only
  ➥ constant propagation of C(f,S) falls within P

Example with $\mathrm{P} = {*}1{*}$ and $\mathrm{S} = {*}{*}1$



$$f_1(\mathbf{x}) = \mathbf{x}_1$$
$$f_2(\mathbf{x}) = \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \text{not } \mathbf{x}_2 \text{ and } \mathbf{x}_1 \;\; 1$$

Reduction of Boolean networks by variable elimination

- Introduced for Boolean networks by Naldi et al in 2009
- General idea: remove a non-autoregulated component (variable)

  \+ replace all occurences of the component by its function

Exemple:

...

$f_4(x) = x_3 \text{ or } x_5$

$f_5(x) = x_4 \text{ and } x_2$

➡

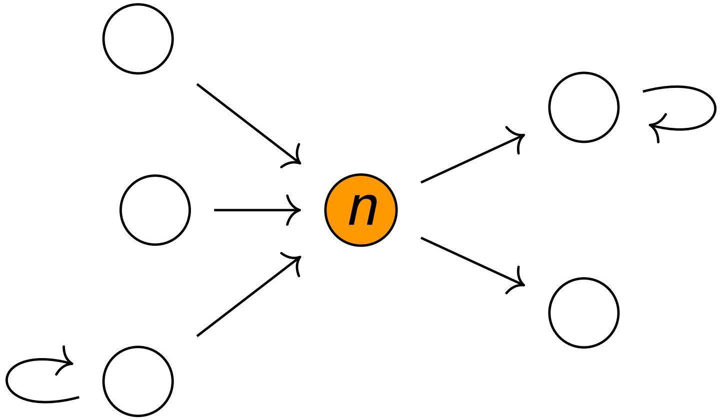
...

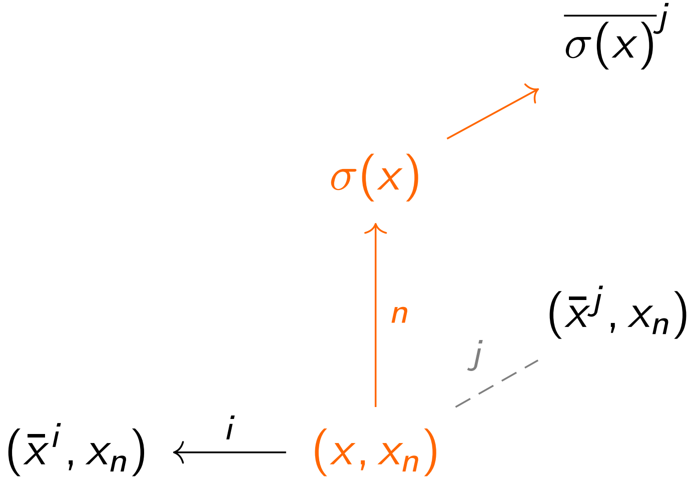
$f_4(x) = x_3 \text{ or } (x_4 \text{ and } x_2)$

- Reduces dimension, and can have a strong effect on the local functions of the other components
- Compatible interpretation: always update the component before the others

Reduction of Boolean networks by variable elimination: effect

$$f = \begin{cases} f_1(x, x_n) \\ f_2(x, x_n) \\ \vdots \\ f_{n-1}(x, x_n) \\ f_n(x, 0) = f_n(x, 1) \end{cases}$$



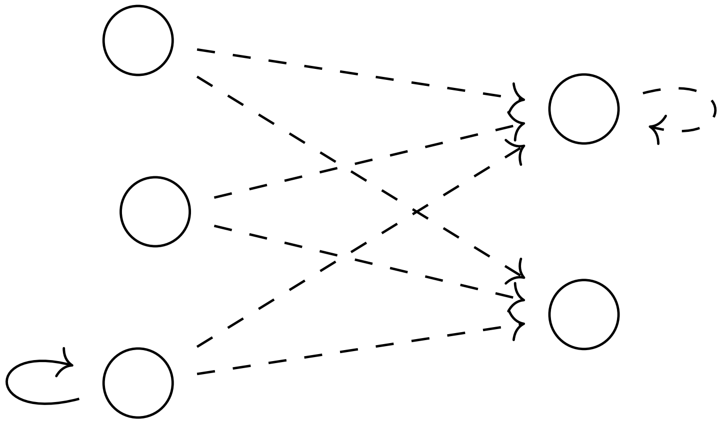


$\downarrow$ elimination of n

$\downarrow$ elimination of n

$\downarrow$ elimination of n

$$\rho(f) = \begin{cases} f_1(x, f_n(x, 0)) \\ f_2(x, f_n(x, 0)) \\ \vdots \\ f_{n-1}(x, f_n(x, 0)) \end{cases}$$



Reduction of Boolean networks by variable elimination: some properties

- If there is a transition in the reduced BN, there is a corresponding transition in the initial network
  ➥ can cut transitions, and thus trajectories, and thus impact attractors

- Preserves fixed points (one to one correspondence)

- Prior work [Tonello & P @CMSB23]: computation of attractors of a BN from its reduction

Relation of control strategies between BN and reduced version

Main motivation: can we deduce control strategies of the BN from its reduced version?

Related question: does the reduction preserves the control strategies?

● Can the reduction introduce control strategies? (that do not work in the initial network)

↪ in general, **yes**. We show cases where the initial has no possible strategy, but reduced has one.

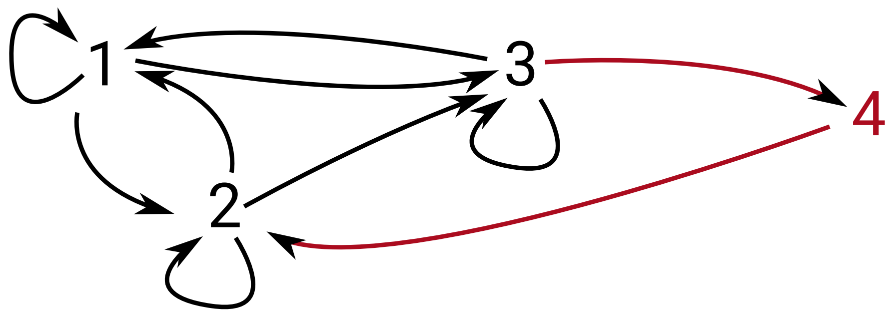
● Can the reduction lose control strategies?

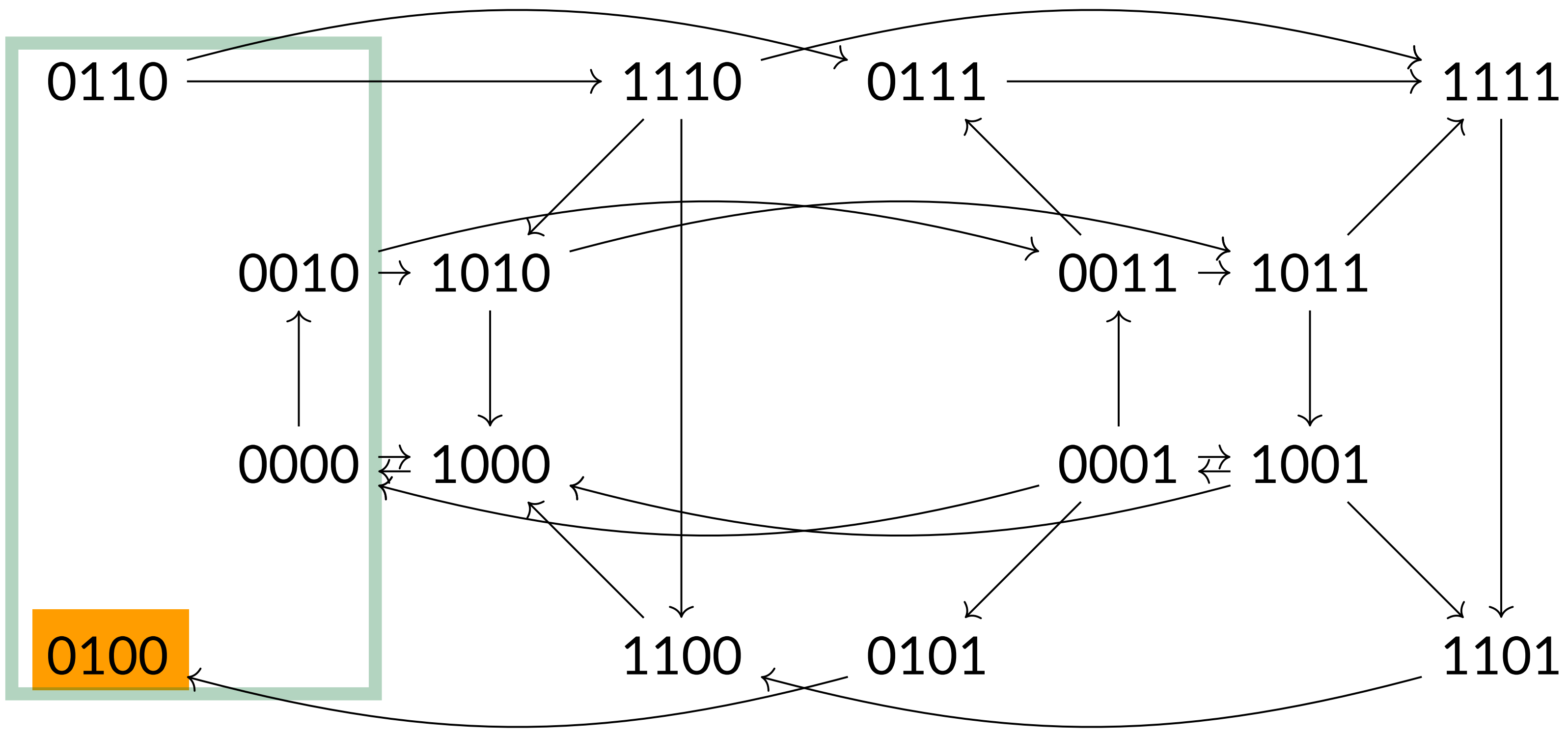
↪ in general **yes**. We show cases where the reduced has no strategy, but the initial has one.

BUT for a class of BNs we demonstate the minimal trap spaces are preserved

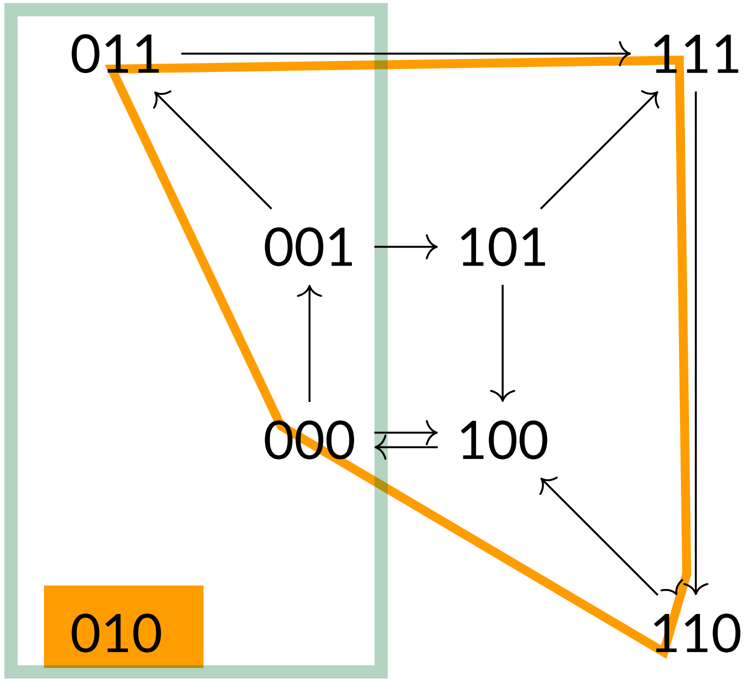
Note: in this talk, we focus on the case where the eliminated component is not fixed in the phenotype
(we have similar results in the paper for the other case)

Reduction can lose attractor-control strategies





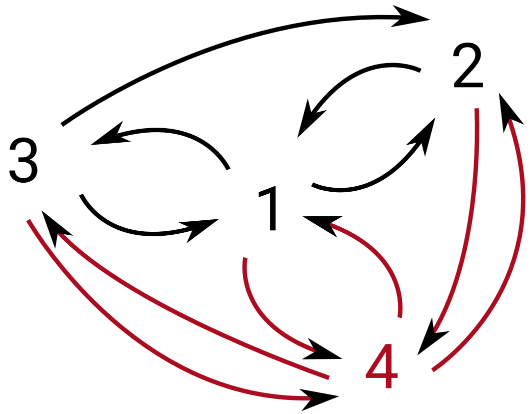
S = ****   is a control strategy (single attractor in P)



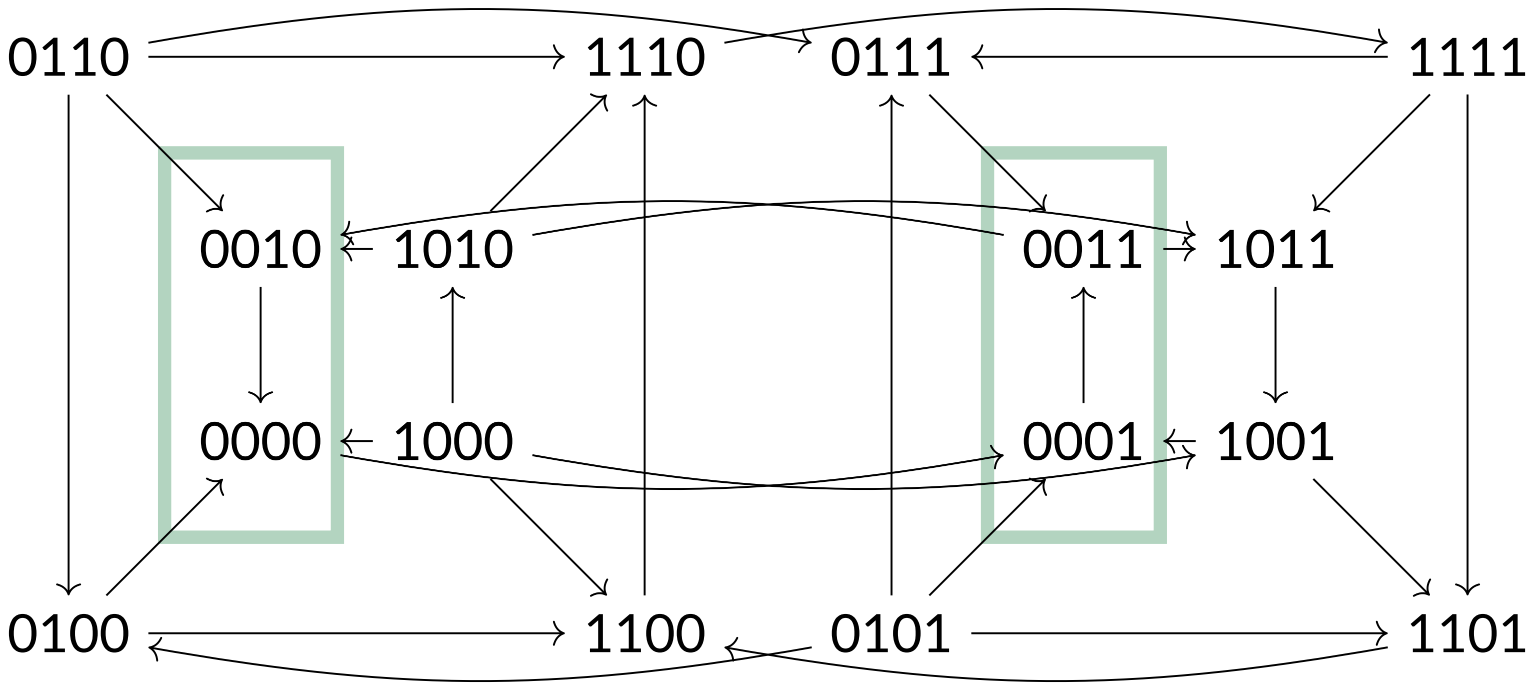
No (a)sync-attractor control strategy

(but S = *** is an MTS-control strategy)

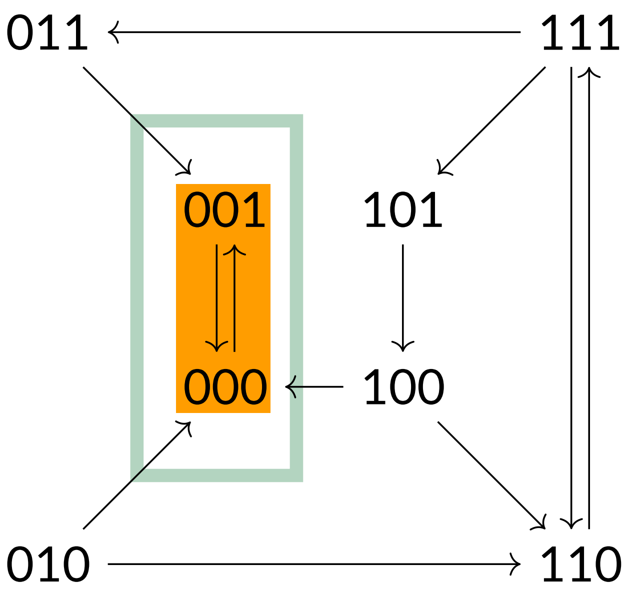
Reduction can introduce control strategies



P = 00**



No attractor/MTS-control strategy

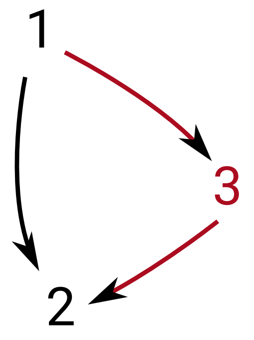


S = *** is an attractor-control and MTS-control strategy

The case of mediator components

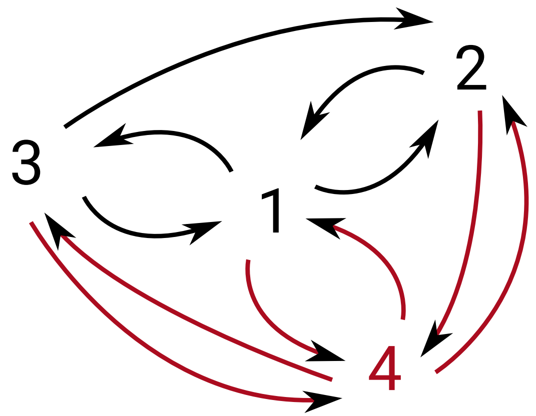
Mediator component: no regulator regulates a target of the component

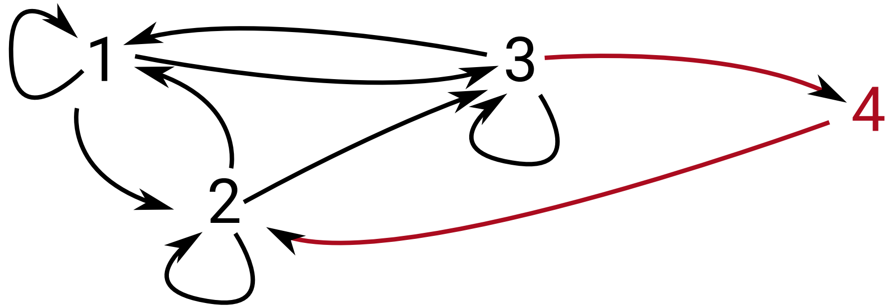
➥ prevents function simplification after reduction



$f_2(x) = x_1$ and x_3
$f_3(x) = !x_1$
=>
$f_2(x) = 0$

(no variable that will appear in the new target functions is already in the function)





Theorem 3.3. *Suppose that no regulator of n regulates a target of n. Then the minimal trap spaces of f are strictly preserved by the elimination of n.*

Theorem 4.3. *Consider a Boolean network f and a phenotype P. Suppose that no regulator of n regulates a target of n.*

(i) If S is an MTS-control strategy S for (f, P) with $S_n = \star$, then $S_{[n-1]}$ is an MTS-control strategy for $(\rho(f), P_{[n-1]})$.

(ii) If S is an MTS-control strategy S for $(\rho(f), P_{[n-1]})$ and $P_n = \star$, then the subspace $S^\star$ is an MTS-control strategy for (f, P).

Additional results in the paper

- Control (ensured) by value propagation: quite robust (but also quite specific)
- Case whenever the variable eliminated is fixed in the phenotype: even worst.

<table>
<tr><th></th><th colspan="2">$\exists$ CS for $(f, P) \Rightarrow$
$\exists$ CS for $(\rho(f), P_{[n-1]})$</th><th colspan="2">$\exists$ CS for $(\rho(f), P_{[n-1]})$
$\Rightarrow \exists$ CS for (f, P)</th></tr>
<tr><th></th><th></th><th>$I \to n \to J$</th><th></th><th>$I \to n \to J$</th></tr>
<tr><td>AD</td><td rowspan="4">✗Ex. 4.10</td><td rowspan="3">✗Ex. 4.11</td><td rowspan="5">✗Ex. 4.12</td><td rowspan="5">✗Ex. 4.13</td></tr>
<tr><td>GD</td></tr>
<tr><td>SD</td></tr>
<tr><td>MTS</td><td>✓Thm. 4.3</td></tr>
<tr><td>VP</td><td colspan="2">✓Thm. 4.6</td></tr>
</table>

(a) n fixed in P

Behind the scenes:

- many counter-examples have been found using automatic BN synthesis with exhaustive search using ASP. It also gaved us ideas of theorems :-) (link to the code in the paper)

<table>
<tr><th></th><th colspan="2">$\exists$ CS for $(f, P) \Rightarrow$
$\exists$ CS for $(\rho(f), P_{[n-1]})$</th><th colspan="2">$\exists$ CS for $(\rho(f), P_{[n-1]})$
$\Rightarrow \exists$ CS for (f, P)</th></tr>
<tr><th></th><th></th><th>$I \to n \to J$</th><th></th><th>$I \to n \to J$</th></tr>
<tr><td>AD</td><td rowspan="4">✗Ex. 4.14</td><td rowspan="3">✗Ex. 4.15</td><td colspan="2">✗Ex. 4.18</td></tr>
<tr><td>GD</td><td colspan="2">✗Ex. 4.18</td></tr>
<tr><td>SD</td><td colspan="2">✗Ex.4.18</td></tr>
<tr><td>MTS</td><td>✓Thm. 4.3</td><td>✗Ex. 4.17</td><td>✓Thm. 4.3</td></tr>
<tr><td>VP</td><td colspan="2">✓Thm. 4.6</td><td>✗ VP, SD Ex. 4.7,4.16
✓ AD, GD Thm. 4.8</td><td>✓Thm. 4.9</td></tr>
</table>

(b) n free in P

Wrap-up

Variable elimination in Boolean networks can have a strong impact on prediction of control strategies

- Elimination of mediator nodes does not preserve attractors (in general)
- Elimination of mediator nodes does preserve minimal trap spaces (because they are cool)

- Minimal trap spaces of Boolean networks are cool

➥ more robust to synchronism

➥ more robust to reduction

Future work

- Patterns preserving async attractors? Other patterns for MTSs

Thanks!

Peer Community In (PCI)

& Peer Community Journal

A **free** way to have your **preprints** peer-reviewed, validated and published in open access

https://peercommunityin.org/

- 17 PCI, inc. **Mathematical & Computational Biology**
  https://mcb.peercommunityin.org/
- preprint recommendations
- diamond open access
- supported by > 150 institutions and 22 countries