Non-deterministic updates of Boolean networks

Loïc Paulevé[1] and Sylvain Sené[2]

[1] CNRS/LaBRI, Bordeaux, France

[2] Université publique, Marseille, France

AUTOMATA 2021, Marseille

Boolean (automata) networks

A BN of dimension n is specified by a function $f : \mathbb{B}^n \to \mathbb{B}^n$ with $\mathbb{B} := \{0, 1\}$

For $i \in \{1, \cdots, n\}$, $f_i : \mathbb{B}^n \to \mathbb{B}$ is the *local function* of the *automaton* i

The Boolean vectors of $x \in \mathbb{B}^n$ are called *configurations*,
x_i is the *state* of automaton i in the configuration x

Example BN f of dimension 3:

$$f_1(x) = \neg x_2$$
$$f_2(x) = \neg x_1$$
$$f_3(x) = \neg x_1 \land x_2$$

Boolean (automata) networks

A BN of dimension n is specified by a function $f : \mathbb{B}^n \to \mathbb{B}^n$ with $\mathbb{B} := \{0, 1\}$

For $i \in \{1, \cdots, n\}$, $f_i : \mathbb{B}^n \to \mathbb{B}$ is the *local function* of the *automaton* i

The Boolean vectors of $x \in \mathbb{B}^n$ are called *configurations*,
x_i is the *state* of automaton i in the configuration x

Example BN f of dimension 3:

$$f_1(x) = \neg x_2$$
$$f_2(x) = \neg x_1$$
$$f_3(x) = \neg x_1 \wedge x_2$$

$$f(000) = 110$$

➡️ different ways for *updating* 000 according to f: **updating mode**

(different modeling hypothesis)

Updating modes of Boolean networks

Updating mode μ: how to compute *next* configurations

Dynamical system (f, μ):

- binary *transition relation* $\rightarrow_{(f,\mu)} \subseteq \mathbb{B}^n \times \mathbb{B}^n$
- can be represented as a digraph $\mathcal{D}_{(f,\mu)} = (\mathbb{B}^n, \rightarrow_{(f,\mu)})$

Updating modes of Boolean networks

Updating mode μ: how to compute *next* configurations

Dynamical system (f, μ):

- binary *transition relation* $\rightarrow_{(f,\mu)} \subseteq \mathbb{B}^n \times \mathbb{B}^n$
- can be represented as a digraph $\mathcal{D}_{(f,\mu)} = (\mathbb{B}^n, \rightarrow_{(f,\mu)})$

Example: *Synchronous* (or *parallel*) updating mode:

$$x \rightarrow_{(f,\mathsf{s})} y \iff y = f(x)$$

Updating modes of Boolean networks

Updating mode μ: how to compute *next* configurations

Dynamical system (f, μ):

- binary *transition relation* $\rightarrow_{(f,\mu)} \subseteq \mathbb{B}^n \times \mathbb{B}^n$
- can be represented as a digraph $\mathcal{D}_{(f,\mu)} = (\mathbb{B}^n, \rightarrow_{(f,\mu)})$

Example: *Synchronous* (or *parallel*) updating mode:

$$x \rightarrow_{(f,\mathsf{s})} y \iff y = f(x)$$

Example: *Fully asynchronous* updating mode:

$$x \rightarrow_{(f,\mathsf{fa})} y \iff \exists i \in \{1, \ldots, n\} : y_i = f_i(x), \forall j \neq i, y_j = x_j$$

Determistic *elementary* updates of configurations

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

Given a set of automata $W \subseteq \{1, \cdots, n\}$, and a configuration $x \in \mathbb{B}^n$:

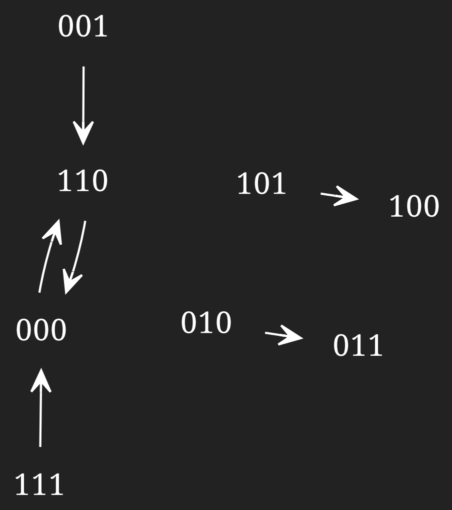
$$\forall i \in \{1, \cdots, n\}, \quad \phi_W(x)_i = \begin{cases} f_i(x) & \text{if } i \in W, \\ x_i & \text{otherwise} \end{cases}$$

Example with $f(000) = 110$

- $\phi_{\{1,\cdots n\}}(000) = 110$
- $\phi_{\{1\}}(000) = \phi_1(000) = 100 = \phi_{\{1,3\}}(000)$
- $\phi_{\{1,2\}}(000) = 110$

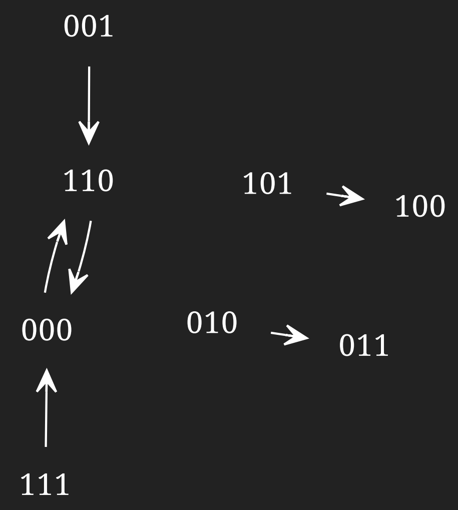
Elementary updating modes

Synchronous: $x \rightarrow_{(f,\mathsf{s})} \phi_{\{1,\cdots,n\}}(x)$



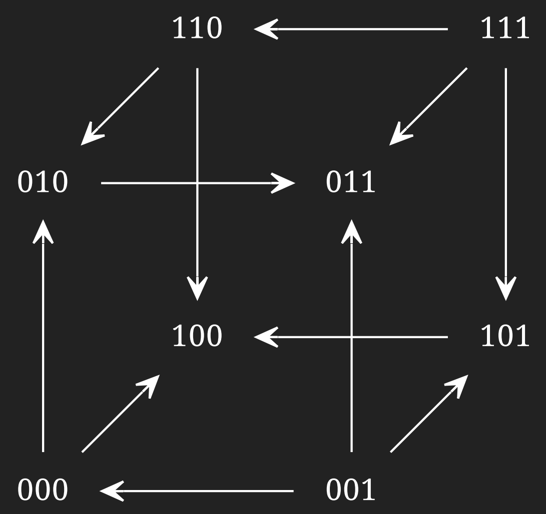
Elementary updating modes

Synchronous: $x \rightarrow_{(f,\mathsf{s})} \phi_{\{1,\cdots,n\}}(x)$



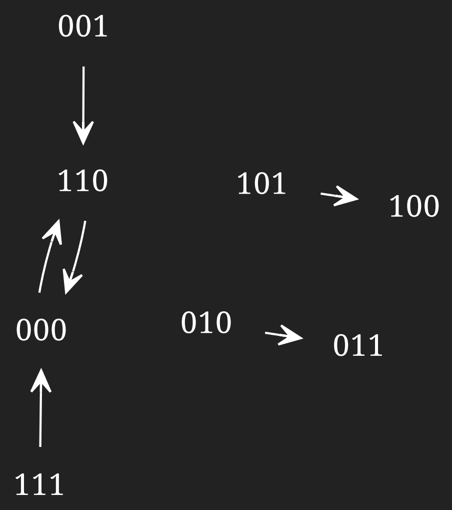
Fully asynchronous: $x \rightarrow_{(f,\mathsf{fa})} y$

$$\exists i \in \{1,\cdots,n\} : y = \phi_{\{i\}}(x)$$



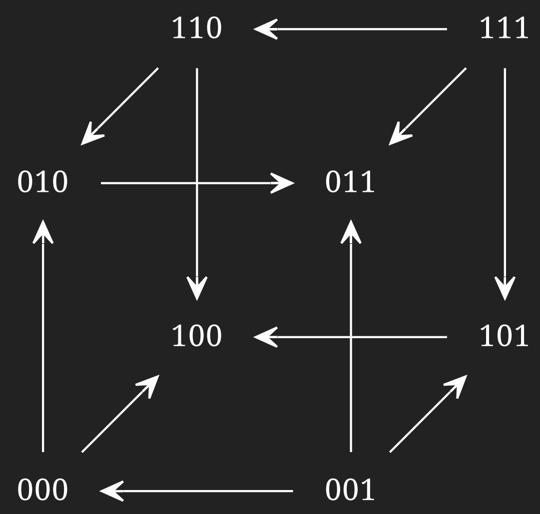
Elementary updating modes

Synchronous: $x \rightarrow_{(f,\mathsf{s})} \phi_{\{1,\cdots,n\}}(x)$



Fully asynchronous: $x \rightarrow_{(f,\mathsf{fa})} y$

$\exists i \in \{1,\cdots,n\} : y = \phi_{\{i\}}(x)$



Asynchronous: $x \rightarrow_{(f,\mathsf{a})} y$ iff $\exists W \subseteq \{1,\cdots,n\}, W \neq \emptyset : y = \phi_W(x)$

$$\rightarrow_{(f,\mathsf{a})} = \rightarrow_{(f,\mathsf{s})} \cup \rightarrow_{(f,\mathsf{fa})} \cup \cdots$$

Updating modes from composition of updates (*non-elementary updates*)

- *Sequential*: $\mu = (w_1, \cdots, w_n)$ with $\{w_1, \cdots, w_n\} = \{1, \cdots, n\}$

$$x \rightarrow_{(f,\mu)} \phi_{w_n} \circ \cdots \circ \phi_{w_1}(x)$$

- *Block sequential*: $\mu = (W_1, \cdots, W_p)$ ordered partition of $\{1, \cdots, n\}$

$$x \rightarrow_{(f,\mu)} \phi_{W_p} \circ \cdots \circ \phi_{W_1}(x)$$

- Block parallel, local clocks, periodic, etc.

Remark: these are all deterministic updating modes

Deterministic updates ϕ_W

- Building block for updating modes: *unifying view*
- Can generate both deterministic and non-deterministic updating modes
- Non-determinism results from a *choice* of deterministic updates

Deterministic updates ϕ_W

- Building block for updating modes: *unifying view*
- Can generate both deterministic and non-deterministic updating modes
- Non-determinism results from a *choice* of deterministic updates

🤔 do they cover all introduced updating modes?

Deterministic updates ϕ_W

- Building block for updating modes: *unifying view*
- Can generate both deterministic and non-deterministic updating modes
- Non-determinism results from a *choice* of deterministic updates

🤔 do they cover all introduced updating modes?

First remark: non-deterministic updating modes require the transition relation "$\longrightarrow$"; deterministic ones do not (next configuration is the application of a function) 😔

Exotic updating modes

Recently, motivated by modeling applications in systems biology:

- *Memory BNs* [Goles et al. AUTOMATA'19]
  deterministic *discrete* dynamics generated from a BN f (and parameters M)
  ↪️ Boolean projection select specific transitions within the asynchronous dynamics
- *Interval BNs* [Chatain et al. AUTOMATA'18]
  projection of fully-asynchronous dynamics of a BN encoded from the given BN f
- *Most Permissive BNs* [Paulevé et al. Nature Comms 2020]
  introduction of pseudo dynamic states
  ↪️ generate transitions that are neither elementary nor non-elementary

🤔 can we consider them as updating modes of BNs?

Most Permissive Boolean networks

[Paulevé et al, Nature Communications 2020]

Modeling perspective: automata match with components of the system

🤔 Are Boolean networks *abstractions* of quantitative systems?

Deterministic updates implies that $\mathbb{B}(0) \leftrightarrow \mathbb{R}(0)$ and $\mathbb{B}(1) \leftrightarrow \mathbb{R}(> 0)$
i.e., an automaton is either in state 0 or non-0.

Most Permissive Boolean networks

[Paulevé et al, Nature Communications 2020]

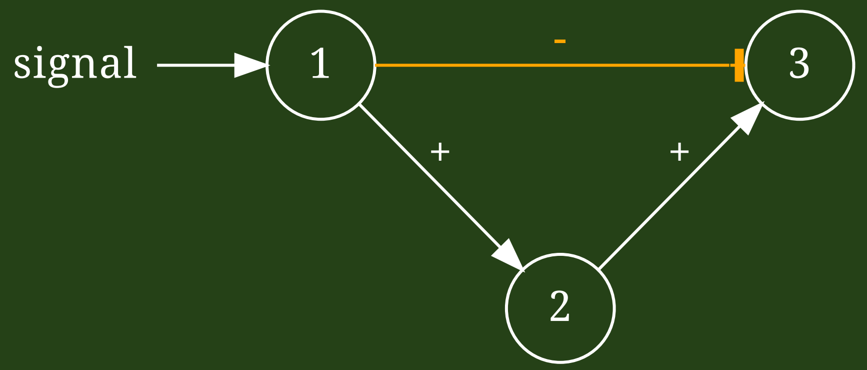
Modeling perspective: automata match with components of the system

🤔 Are Boolean networks *abstractions* of quantitative systems?

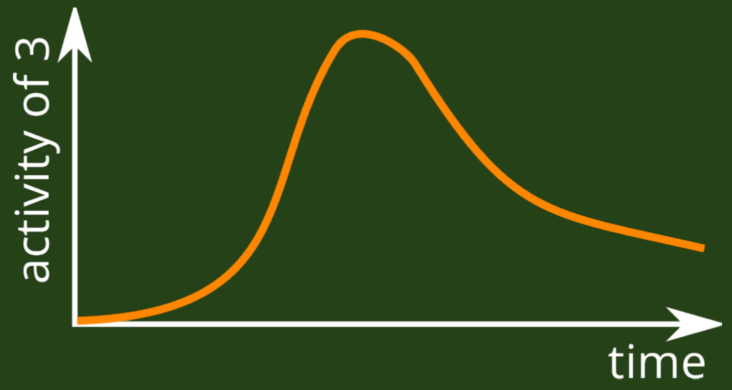
Deterministic updates implies that $\mathbb{B}(0) \leftrightarrow \mathbb{R}(0)$ and $\mathbb{B}(1) \leftrightarrow \mathbb{R}(>0)$
i.e., an automaton is either in state 0 or non-0.

⚠️ Some trajectories of quantitative systems cannot be captured by BNs with neither elementary nor non-elementary updates

Incoherent feed-forward loop of type 3 (I3-FFL)

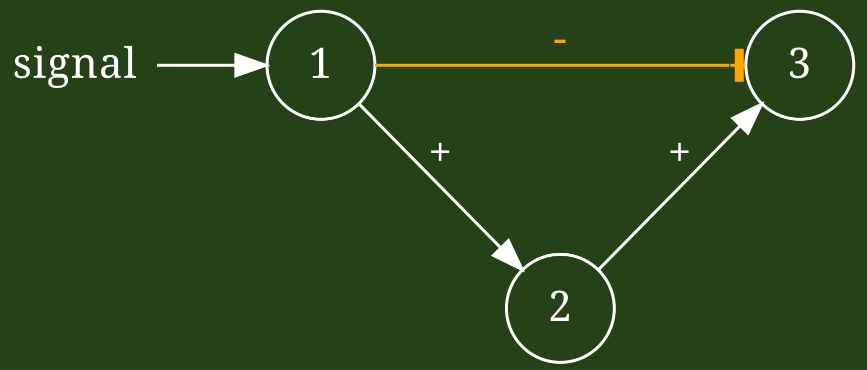


Possible output (automata 3)

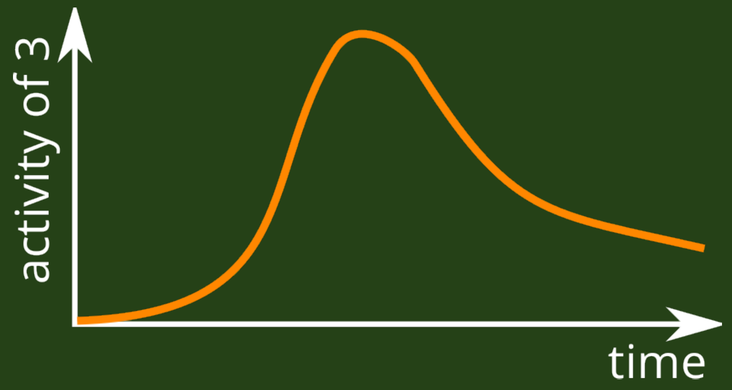


with quantitative models/synthetic DNA

Incoherent feed-forward loop of type 3 (I3-FFL)



Possible output (automata 3)



with quantitative models/synthetic DNA

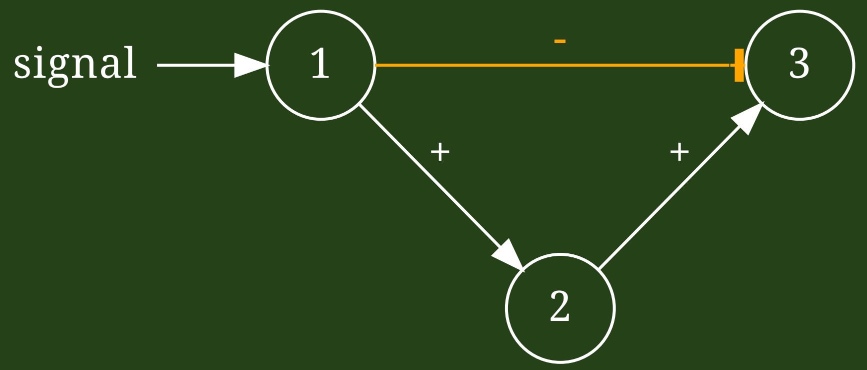
Boolean network

$$f_1(x) = 1$$
$$f_2(x) = x_1$$
$$f_3(x) = \neg x_1 \wedge x_2$$

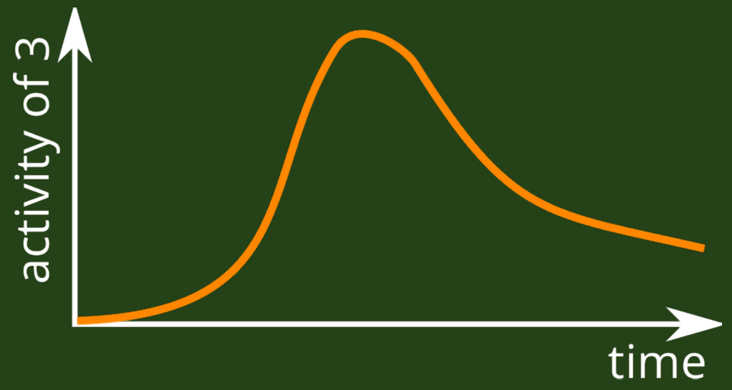
(A)synchronous dynamics from 000:

$$000 \rightarrow_{(f,\mathrm{a})} 100 \rightarrow_{(f,\mathrm{a})} 110 \circlearrowleft$$

Incoherent feed-forward loop of type 3 (I3-FFL)



Possible output (automata 3)



with quantitative models/synthetic DNA

Boolean network

$$f_1(x) = 1$$
$$f_2(x) = x_1$$
$$f_3(x) = \neg x_1 \wedge x_2$$

(A)synchronous dynamics from 000:

$$000 \to_{(f,\text{a})} 100 \to_{(f,\text{a})} 110 \circlearrowleft$$

❌ **fails to capture time when 1 is high enough to activate 2, but not high enough to repress 3**

Most Permissive (MP) Boolean networks

Automata are either in state 0 or in state *maximal* (=1)

💡 when changing, states of automata can be read differently (non-deterministic) by other automata (threshold abstraction)

Most Permissive (MP) Boolean networks

Automata are either in state 0 or in state *maximal* (=1)

💡 when changing, states of automata can be read differently (non-deterministic) by other automata (threshold abstraction)

Consider 4 states: $P = \{0, \nearrow, \searrow, 1\}$
In a configuration $z \in P^n$, each automaton i can change of state as follows:

- from 0 to $\nearrow$ only if $\exists x \in \gamma(z) : f_i(x) = 1$
- from 1 to $\searrow$ only if $\exists x \in \gamma(z) : f_i(x) = 0$
- from $\nearrow$ to 1 and from $\searrow$ to 0 w/o condition

where $\gamma(z) = \{x \in \mathbb{B}^n \mid \forall i \in \{1, \cdots, n\}, z_i \in \mathbb{B} \implies x_i = z_i\}$
(reading possibilities of changing automata)

Most Permissive (MP) Boolean networks

Automata are either in state 0 or in state *maximal* (=1)

💡 when changing, states of automata can be read differently (non-deterministic) by other automata (threshold abstraction)

Consider 4 states: $P = \{0, \nearrow, \searrow, 1\}$
In a configuration $z \in P^n$, each automaton i can change of state as follows:

- from 0 (or $\searrow$) to $\nearrow$ only if $\exists x \in \gamma(z) : f_i(x) = 1$
- from 1 (or $\nearrow$) to $\searrow$ only if $\exists x \in \gamma(z) : f_i(x) = 0$
- from $\nearrow$ to 1 and from $\searrow$ to 0 w/o condition

where $\gamma(z) = \{x \in \mathbb{B}^n \mid \forall i \in \{1, \cdots, n\}, z_i \in \mathbb{B} \implies x_i = z_i\}$
(reading possibilities of changing automata)

Example

$$f_1(x) = 1$$
$$f_2(x) = x_1$$
$$f_3(x) = \neg x_1 \wedge x_2$$

$$000 \rightarrow_{\mathsf{MP}} \nearrow 00$$

Example

$$f_1(x) = 1$$
$$f_2(x) = x_1$$
$$f_3(x) = \neg x_1 \wedge x_2$$

$$000 \rightarrow_{\text{MP}} \nearrow 00 \rightarrow_{\text{MP}} \nearrow\nearrow 0$$

Example

$$f_1(x) = 1$$
$$f_2(x) = x_1$$
$$f_3(x) = \neg x_1 \wedge x_2$$

$$000 \to_{\text{MP}} \nearrow 00 \to_{\text{MP}} \nearrow\nearrow 0 \to_{\text{MP}} \nearrow\nearrow\nearrow$$

Example

$$f_1(x) = 1$$
$$f_2(x) = x_1$$
$$f_3(x) = \neg x_1 \wedge x_2$$

$$000 \to_{\mathsf{MP}} \nearrow 00 \to_{\mathsf{MP}} \nearrow \nearrow 0 \to_{\mathsf{MP}} \nearrow \nearrow \nearrow \to_{\mathsf{MP}} 111$$

Example

$$f_1(x) = 1$$
$$f_2(x) = x_1$$
$$f_3(x) = \neg x_1 \wedge x_2$$

$$000 \to_{\text{MP}} \nearrow 00 \to_{\text{MP}} \nearrow\nearrow 0 \to_{\text{MP}} \nearrow\nearrow\nearrow \to_{\text{MP}} 111 \cdots \to_{\text{MP}} 110$$

Example

$$f_1(x) = 1$$
$$f_2(x) = x_1$$
$$f_3(x) = \neg x_1 \wedge x_2$$

$$000 \rightarrow_{\mathsf{MP}} \nearrow\ 00 \rightarrow_{\mathsf{MP}} \nearrow\nearrow\ 0 \rightarrow_{\mathsf{MP}} \nearrow\nearrow\nearrow \rightarrow_{\mathsf{MP}} 111 \cdots \rightarrow_{\mathsf{MP}} 110$$

(there is no path from 000 to 111 with elementary and non-elementary transitions)

Properties of Most Permissive BNs

[Paulevé et al, Nature Communications 2020]

Formal abstraction of trajectories of quantitative systems

- guarantees not to miss any transition of any *quantitative refinement* of the BN
- *minimal abstraction* with this guarantee
- preserves fixed points of f; *limit sets* are the *minimal trap spaces* of f

Complexity:

- *reachability*: **P** with locally-monotone BNs/$\mathbf{P}^{\mathbf{NP}}$ otherwise
- *limit configurations*: **coNP** / $\mathbf{coNP}^{\mathbf{coNP}}$
  (both problems are PSPACE-complete w/ (a)synchronous)

Properties of Most Permissive BNs

[Paulevé et al, Nature Communications 2020]

Formal abstraction of trajectories of quantitative systems

- guarantees not to miss any transition of any *quantitative refinement* of the BN
- *minimal abstraction* with this guarantee
- preserves fixed points of f; *limit sets* are the *minimal trap spaces* of f

Complexity:

- *reachability*: **P** with locally-monotone BNs/$\mathbf{P}^{\mathbf{NP}}$ otherwise
- *limit configurations*: **coNP** / $\mathbf{coNP}^{\mathbf{coNP}}$
  (both problems are PSPACE-complete w/ (a)synchronous)

🤔 formalize Most Permissive as an updating mode

$f_1(x) = 1$, $f_2(x) = x_1$, $f_3(x) = \neg x_1 \wedge x_2$, and $W = \{1, 2, 3\}$

$$f_1(x) = 1,\ f_2(x) = x_1,\ f_3(x) = \neg x_1 \wedge x_2, \text{ and } W = \{1, 2, 3\}$$

000

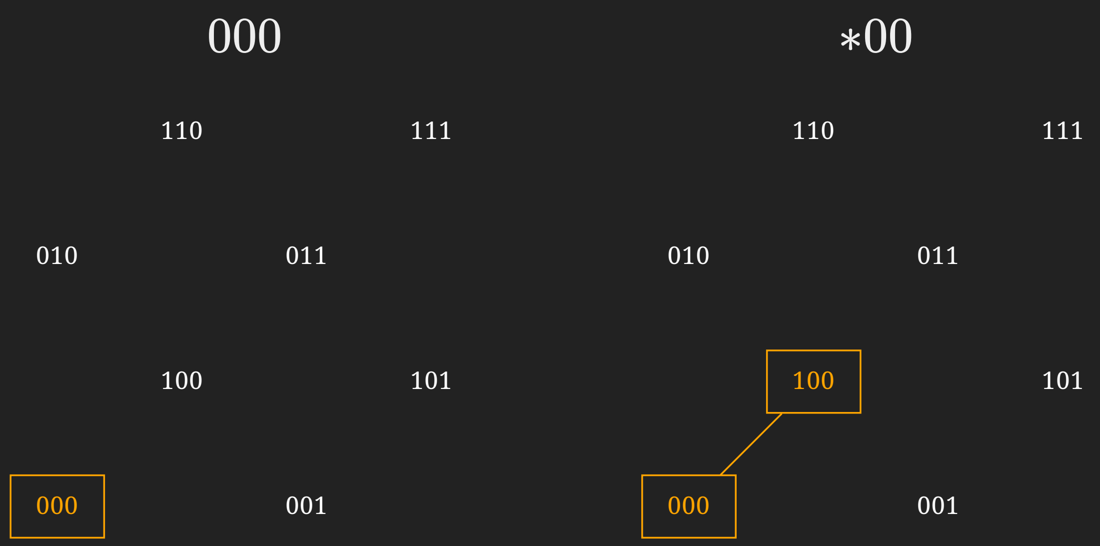
110 111

010 011

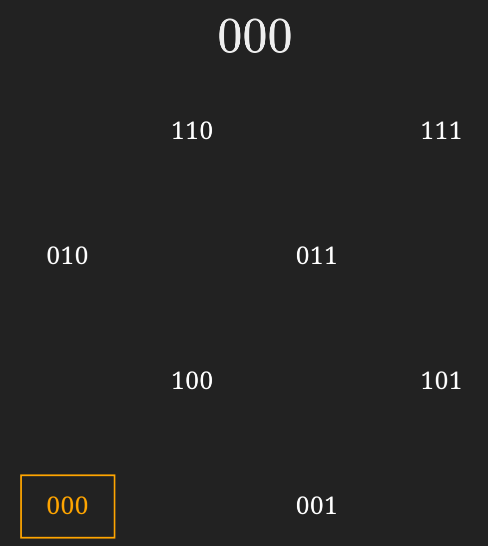
100 101

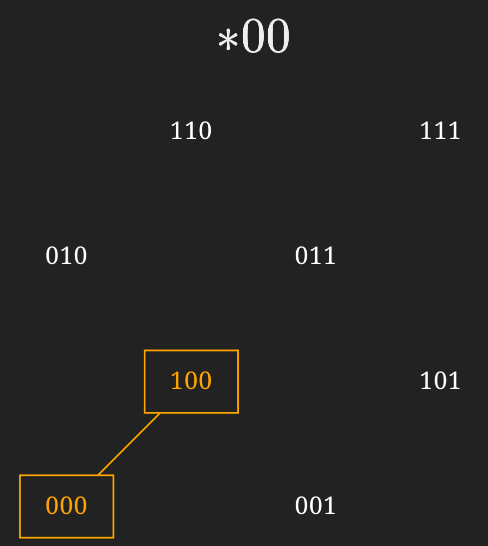
000 001

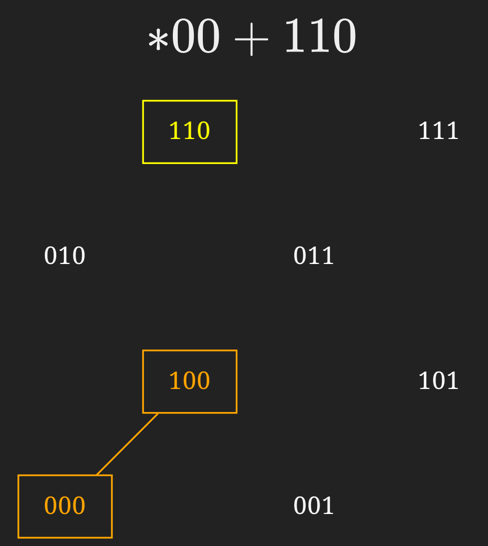
$$f_1(x) = 1,\ f_2(x) = x_1,\ f_3(x) = \neg x_1 \wedge x_2, \text{ and } W = \{1, 2, 3\}$$



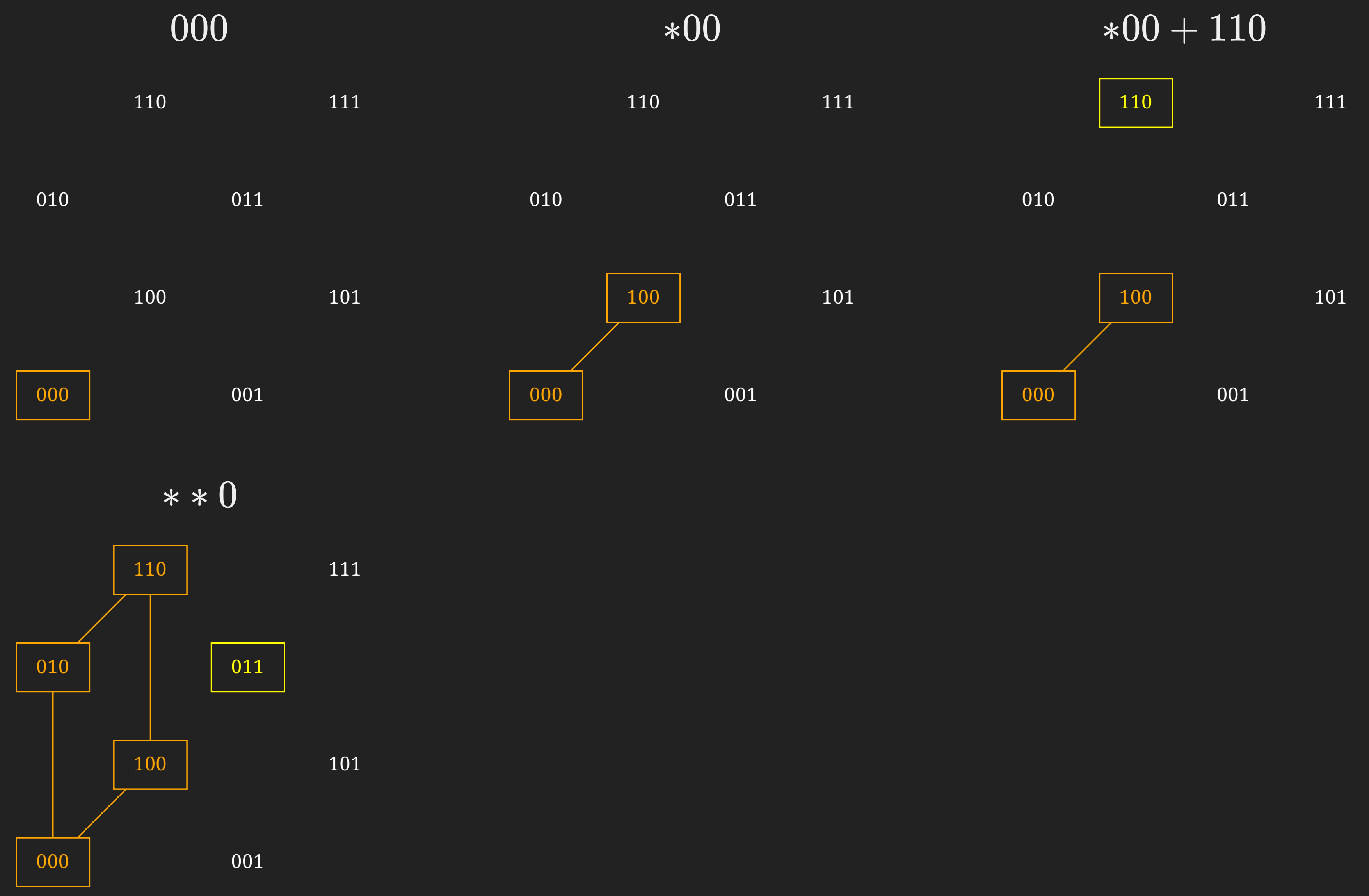
$f_1(x) = 1$, $f_2(x) = x_1$, $f_3(x) = \neg x_1 \wedge x_2$, and $W = \{1, 2, 3\}$



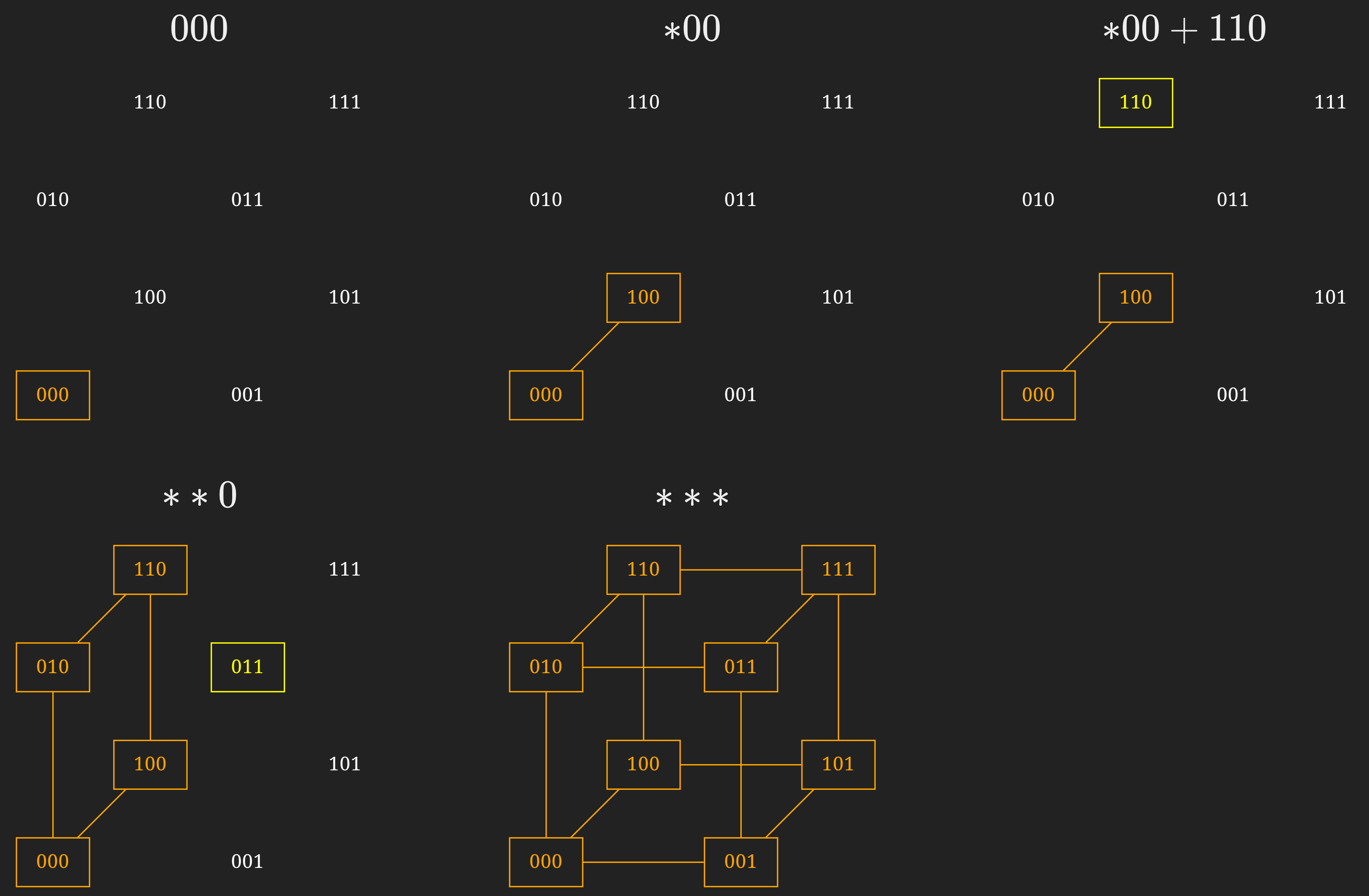




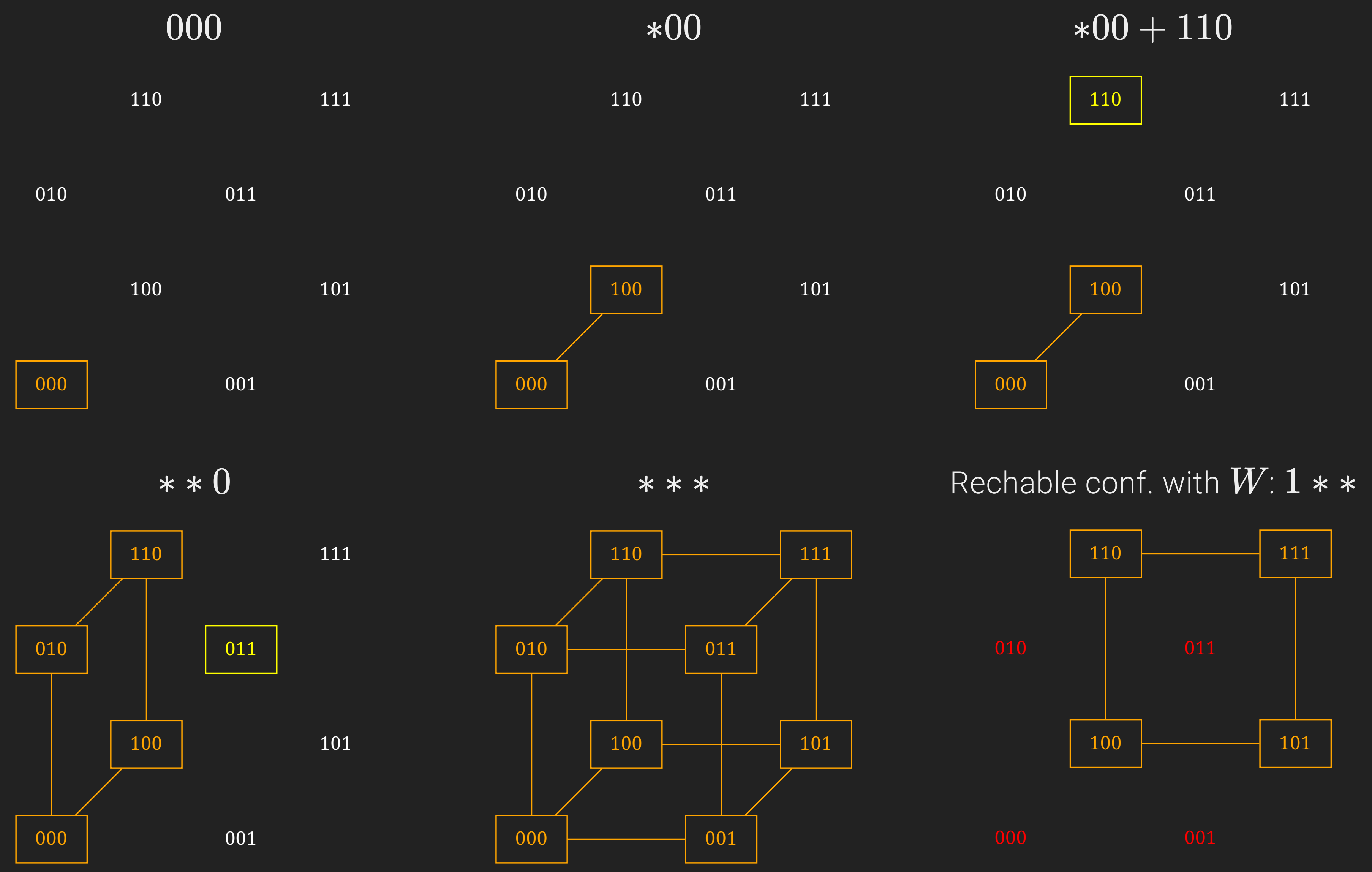
$f_1(x) = 1$, $f_2(x) = x_1$, $f_3(x) = \neg x_1 \wedge x_2$, and $W = \{1, 2, 3\}$



$f_1(x) = 1$, $f_2(x) = x_1$, $f_3(x) = \neg x_1 \wedge x_2$, and $W = \{1, 2, 3\}$



$f_1(x) = 1$, $f_2(x) = x_1$, $f_3(x) = \neg x_1 \wedge x_2$, and $W = \{1, 2, 3\}$



Non-deterministic updates

Configuration set update: $\Phi : 2^{\mathbb{B}^n} \to 2^{\mathbb{B}^n}$

We assume that $\forall X \subseteq \mathbb{B}^n$, $\Phi(X) = \bigcup_{x \in X} \Phi(\{x\})$

Translate into transitions: $y \in \Phi(\{x\}) \iff x \to y$

- *Asynchronous* updating mode: given a set of configurations $X \subseteq \mathbb{B}^n$,

$$\Phi_{\mathsf{a}}(\{x\}) = \{\phi_W(x) \mid emptyset \neq W \subseteq \{1, \cdots, n\}\}$$

- *Fully asynchronous* updating mode:

$$\Phi_{\mathsf{fa}}(\{x\}) = \{\phi_i(x) \mid i \in \{1, \cdots, n\}\}$$

Most permissive updating mode

$$\Phi_{\mathsf{MP}}(X) = \bigcup_{W\subseteq\{1,\cdots,n\}} \Lambda_W \circ \Phi^{\omega}_{W,\nabla}(X)$$

Widening:

$$\Phi_{W,\nabla}(X) = \prod_{i=1}^{n} \bigcup_{x\in X} \{x_i, f_i(x)\}$$

Narrowing:

$$\Lambda_W(X) = \{x \in X \mid \forall i \in W, \exists y \in X, x_i = f_i(y)\}$$

Discussion

Non-deterministic updates with configuration set updates

- *Unifying framework* for expressing complex updating modes
  - in the paper: memory, interval, most permissive updating modes
  - avoid using auxiliary objects/encodings
  - may give a better understanding of the computations
- Enable *envisioning new updating modes*
  - restrictions of the asynchronous (invariant preservation, …)
  - restrictions of the most permissive (monotony enforcement, …)

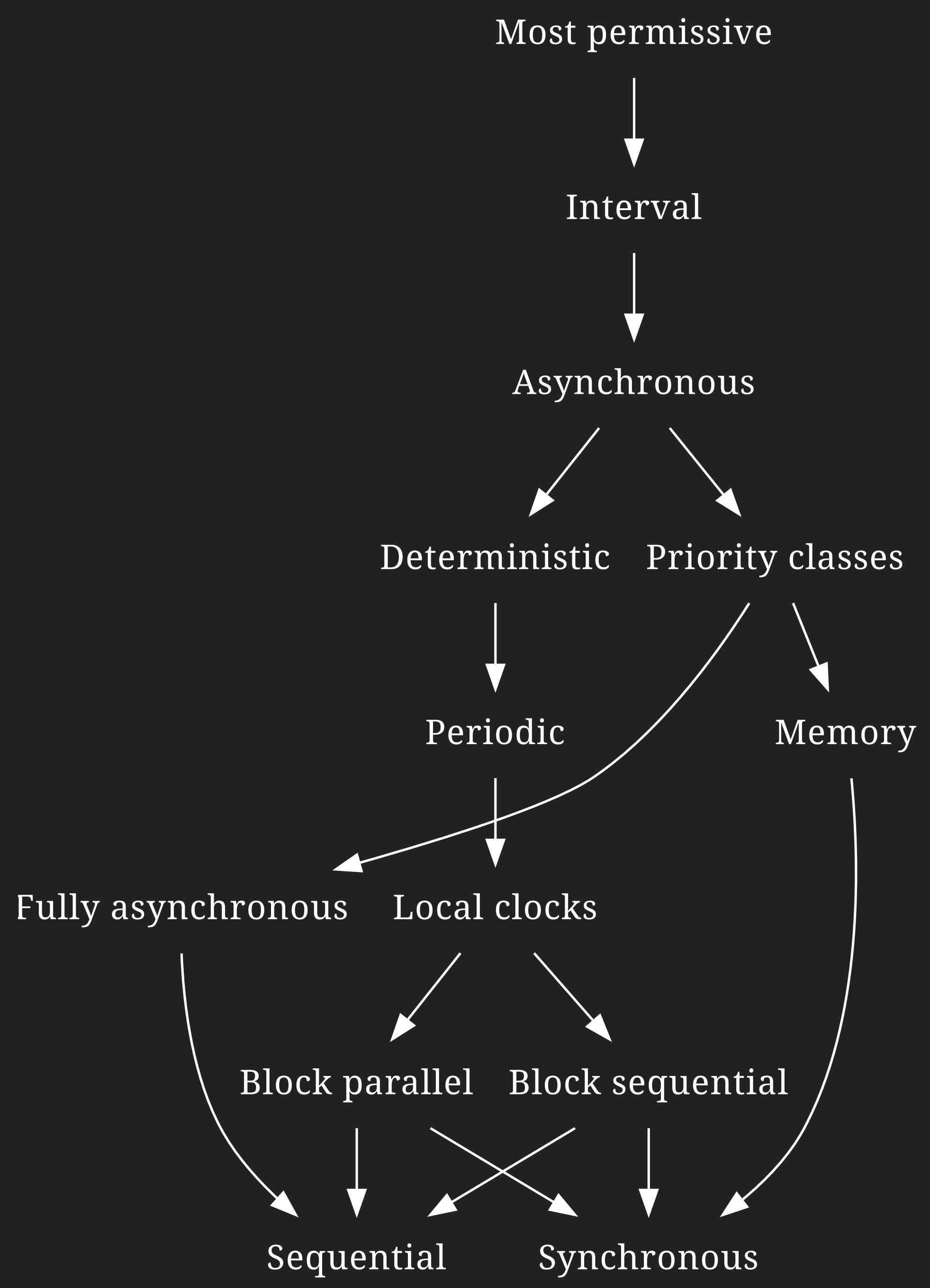
Discussion

Non-deterministic updates with configuration set updates

- *Unifying framework* for expressing complex updating modes
  - in the paper: memory, interval, most permissive updating modes
  - avoid using auxiliary objects/encodings
  - may give a better understanding of the computations
- Enable *envisioning new updating modes*
  - restrictions of the asynchronous (invariant preservation, …)
  - restrictions of the most permissive (monotony enforcement, …)

Reasonable non-deterministic updates? (from a modeling p.o.v.)

- E.g., $\Phi(X) = \mathbb{B}^n$ (independent of f)
- Is Most Permissive the *largest reasonable* set update?
  more configurations implies having states of automata computed without f!



Simulations

Updating modes hierarchy

$$C \preceq C' \iff \forall \mu \in C, \exists \mu' \in C' :$$
$$\forall f \forall x \forall y, x \rightarrow^*_{(f,\mu)} y \implies x \rightarrow^*_{(f,\mu')} y$$

Intrinsic simulations

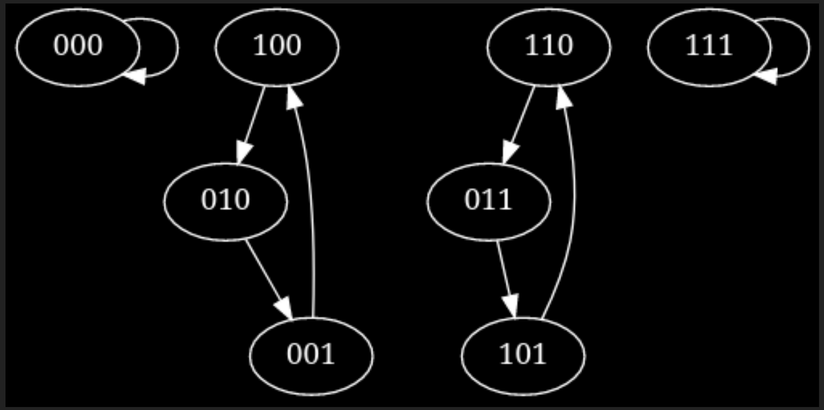
Simulate (f, μ) with (f', μ')

E.g., can we simulate MP with asynchronous? (at which cost?)

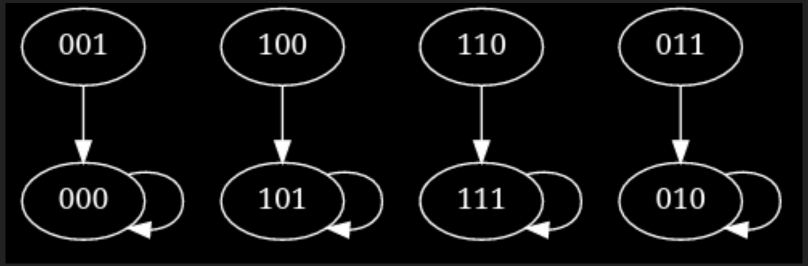
Code

github.com/pauleve/updating-modes-notebooks

```
from colomoto.minibn import *
f = BooleanNetwork({
    "x1": "x3",
    "x2": "x1",
    "x3": "x2"
})
ds = SynchronousDynamics(f).dynamics()
ds
```




```
dp = PeriodicDynamics(
    ({"x2", "x3"}, {"x1", "x3"}, {"x1", "x2"}),
    f).dynamics()
dp
```



includes all mentioned in the article (memory, interval, etc.)