Boolean Networks

Juri Kolčák

Friday 7th November, 2025

Boolean Abstraction

Boolean networks constitute a "top-down" abstraction, allowing us to model systems which are not understood in full detail.

In general, Boolean networks are well suited for modelling systems with the following properties:

- Made up of interacting entities.
- Each entity is characterised by a variable quantity.
- The events (and their mechanisms) are not directly observable, only their complete consequences.

Boolean Networks – Syntax

A Boolean network f of dimension $n \in \mathbb{N}$ (on n variables) is a collection of n-ary Boolean functions $f_1, \ldots, f_n$:

$$f = (f_1, \ldots, f_n)$$

For each $i \in \{1, \ldots, n\}$, $f_i \colon \mathbb{B}^n \to \mathbb{B}$ is the **local function** of the i-th variable.

The Boolean network itself can be understood as a Boolean vector function $f \colon \mathbb{B}^n \to \mathbb{B}^n$, defined as $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x}))$.

Example:

$$\begin{aligned} f_1(\mathbf{x}) &= \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2) \\ f_2(\mathbf{x}) &= \mathbf{x}_1 \wedge \mathbf{x}_3 \\ f_3(\mathbf{x}) &= \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3 \end{aligned}$$

Boolean Networks – Semantics

A Boolean network defines a finite dynamical system with discrete states (**transition system**). A transition system $(S, \rightarrow)$ is composed of:

- A finite set of states S;
- An irreflexive transition relation $\rightarrow \subseteq S \times S$;

We use the natural infix notation for the transition relation, $\mathbf{x} \rightarrow \mathbf{y}$ instead of $(\mathbf{x}, \mathbf{y}) \in \rightarrow$.

For a Boolean network of dimension n, the set of states is the set of all Boolean vectors of length n, $S = \mathbb{B}^n$.

EXAMPLE:
For dimension $n = 3$, $S = \{000, 001, 010, 100, 011, 101, 110, 111\}$.

The exact shape of the transition relation is determined by the chosen **update mode**.
Multiple different update modes are commonly employed.

Update Modes

Given a configuration $\mathbf{x} \in \mathbb{B}^n$, then for each variable $i \in \{1, \ldots, n\}$, $f_i(\mathbf{x})$ is the "next value" of the i-th variable.

In particular a variable such that $\mathbf{x}_i \neq f_i(\mathbf{x})$ is called **frustrated** (in the configuration $\mathbf{x}$).

There are multiple ways of upgrading configurations:

- All variables update value simultaneously;
- Variables update value sequentially;
- Variables are split into blocks and each block updates values simultaneously;
- . . . each block updates values sequentially;
- Internal clocks;
- . . .

Update Functions

Update functions are a family of functions which are restating the local functions $f_1, \ldots, f_n$ as direct modification of the configurations, $\Phi\colon \mathbb{B}^n \to \mathbb{B}^n$.

For any subset of variables $W \subseteq \{1, \ldots, n\}$, the update function $\Phi_W\colon \mathbb{B}^n \to \mathbb{B}^n$ is defined as $\Phi_W\colon \mathbf{x} \mapsto \mathbf{y}$ where for each $i \in \{1, \ldots, n\}$:

$$\mathbf{y}_i \triangleq \begin{cases} f_i(\mathbf{x}) & \text{if } i \in W \\ \mathbf{x}_i & \text{if } i \notin W \end{cases}$$

EXAMPLES:

$\Phi_\emptyset = id\colon \mathbf{x} \mapsto \mathbf{x}$

$\Phi_{\{1,\ldots,n\}}\colon \mathbf{x} \mapsto (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x}))$ $\quad \phi_{\{1,\ldots,n\}} = f$

$\forall i \in \{1, \ldots, n\}, \Phi_{\{i\}} = \Phi_i\colon \mathbf{x} \mapsto (\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, f_i(\mathbf{x}), \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n)$

A transition of the form $\mathbf{x} \to \Phi_W(\mathbf{x})$ for some $W \subseteq \{1, \ldots, n\}$ is called **elementary**.

A transition of the form $x \to \phi_{W_1} \circ \phi_{W_2} \circ \ldots \circ \phi_{W_k}(x)$ for some $W_1, \ldots W_k \subseteq \{1, \ldots, n\}$ is called non-elementary.

Synchronous Semantics

"All variables have their value updated simultaneously."

Synchronous semantics only uses one update function, $\Phi_{\{1,\dots,n\}} : \mathbf{x} \mapsto (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{sync} \mathbf{y} \overset{\Delta}{\iff} \mathbf{y} = \Phi_{\{1,\dots,n\}}(\mathbf{x})$$

Synchronous Semantics

"All variables have their value updated simultaneously."

Synchronous semantics only uses one update function, $\Phi_{\{1,\dots,n\}} : \mathbf{x} \mapsto (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{sync} \mathbf{y} \overset{\Delta}{\iff} \mathbf{y} = \Phi_{\{1,\dots,n\}}(\mathbf{x})$$

Deterministic – Each configuration has at most one successor.

Synchronous Semantics

"All variables have their value updated simultaneously."

Synchronous semantics only uses one update function, $\Phi_{\{1,\dots,n\}} \colon \mathbf{x} \mapsto (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{sync} \mathbf{y} \overset{\Delta}{\iff} \mathbf{y} = \Phi_{\{1,\dots,n\}}(\mathbf{x})$$

Deterministic – Each configuration has at most one successor.

EXAMPLE:

$$f_1(\mathbf{x}) = \mathbf{x}_3 \land (\neg \mathbf{x}_1 \lor \neg \mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \land \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \lor \mathbf{x}_2 \lor \mathbf{x}_3$$

Synchronous Semantics

"All variables have their value updated simultaneously."

Synchronous semantics only uses one update function, $\Phi_{\{1,\dots,n\}} : \mathbf{x} \mapsto (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{sync} \mathbf{y} \overset{\Delta}{\iff} \mathbf{y} = \Phi_{\{1,\dots,n\}}(\mathbf{x})$$

Deterministic – Each configuration has at most one successor.

Example:

$$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$$

$$001 \xrightarrow{sync} 101$$
$$100 \xrightarrow{sync} 001 \qquad 101 \xrightarrow{sync} 111$$
$$010 \xrightarrow{sync} 001 \qquad 011 \xrightarrow{sync} 101$$
$$110 \xrightarrow{sync} 001 \qquad 111 \xrightarrow{sync} 011$$

Fully Asynchronous Semantics

"Only one variable updates value in each transition."

Fully asynchronous semantics use all singleton update functions, $\Phi_i : \mathbf{x} \mapsto (\mathbf{x}_1, \dots \mathbf{x}_{i-1}, f_i(\mathbf{x}), \mathbf{x}_{i+1}, \dots, \mathbf{x}_n)$, $i \in \{1, \dots, n\}$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{async} \mathbf{y} \overset{\Delta}{\iff} \exists i \in \{1, \dots, n\}, \mathbf{y} = \Phi_i(\mathbf{x})$$

Fully Asynchronous Semantics

"Only one variable updates value in each transition."

Fully asynchronous semantics use all singleton update functions, $\Phi_i : \mathbf{x} \mapsto (\mathbf{x}_1, \dots \mathbf{x}_{i-1}, f_i(\mathbf{x}), \mathbf{x}_{i+1}, \dots, \mathbf{x}_n)$, $i \in \{1, \dots, n\}$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{async} \mathbf{y} \overset{\Delta}{\iff} \exists i \in \{1, \dots, n\}, \mathbf{y} = \Phi_i(\mathbf{x})$$

EXAMPLE:

$$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$$

Fully Asynchronous Semantics

"Only one variable updates value in each transition."

Fully asynchronous semantics use all singleton update functions, $\Phi_i : \mathbf{x} \mapsto (\mathbf{x}_1, \ldots \mathbf{x}_{i-1}, f_i(\mathbf{x}), \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n)$, $i \in \{1, \ldots, n\}$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{async} \mathbf{y} \overset{\Delta}{\iff} \exists i \in \{1, \ldots, n\}, \mathbf{y} = \Phi_i(\mathbf{x})$$

EXAMPLE:

$$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$$

$100 \xrightarrow{async} 000$ $\quad$ $010 \xrightarrow{async} 000$

$100 \xrightarrow{async} 101$ $\quad$ $010 \xrightarrow{async} 011$

$110 \xrightarrow{async} 010$ $\quad$ $001 \xrightarrow{async} 101$

$110 \xrightarrow{async} 100$ $\quad$ $101 \xrightarrow{async} 111$

$110 \xrightarrow{async} 111$ $\quad$ $011 \xrightarrow{async} 111$

$111 \xrightarrow{async} 011$ $\quad$ $011 \xrightarrow{async} 001$

Generalised Asynchronous Semantics

"Any subset of variables has their values updated simultaneously."

Generalised asynchronous semantics use all update functions, Φ_W for $W \subseteq \{1, \ldots, n\}$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{gen} \mathbf{y} \overset{\Delta}{\iff} \exists W \subseteq \{1, \ldots, n\}, \mathbf{y} = \Phi_W(\mathbf{x})$$

Generalised Asynchronous Semantics

"Any subset of variables has their values updated simultaneously."

Generalised asynchronous semantics use all update functions, Φ_W for $W \subseteq \{1, \ldots, n\}$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{gen} \mathbf{y} \overset{\Delta}{\iff} \exists W \subseteq \{1, \ldots, n\}, \mathbf{y} = \Phi_W(\mathbf{x})$$

$\xrightarrow{sync} \subseteq \xrightarrow{gen}$ and $\xrightarrow{async} \subseteq \xrightarrow{gen}$.

Generalised Asynchronous Semantics

"Any subset of variables has their values updated simultaneously."

Generalised asynchronous semantics use all update functions, Φ_W for $W \subseteq \{1, \ldots, n\}$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{gen} \mathbf{y} \overset{\Delta}{\iff} \exists W \subseteq \{1, \ldots, n\}, \mathbf{y} = \Phi_W(\mathbf{x})$$

$\xrightarrow{sync} \subseteq \xrightarrow{gen}$ and $\xrightarrow{async} \subseteq \xrightarrow{gen}$.

EXAMPLE:

$$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$$

Generalised Asynchronous Semantics

"Any subset of variables has their values updated simultaneously."

Generalised asynchronous semantics use all update functions, Φ_W for $W \subseteq \{1, \ldots, n\}$. Formally:

$$\forall \mathbf{x} \neq \mathbf{y} \in \mathbb{B}^n, \mathbf{x} \xrightarrow{gen} \mathbf{y} \overset{\Delta}{\iff} \exists W \subseteq \{1, \ldots, n\}, \mathbf{y} = \Phi_W(\mathbf{x})$$

$\xrightarrow{sync} \subseteq \xrightarrow{gen}$ and $\xrightarrow{async} \subseteq \xrightarrow{gen}$.

EXAMPLE:

$$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$$

$100 \xrightarrow{gen} 001$	$110 \xrightarrow{gen} 001$	$001 \xrightarrow{gen} 101$
$100 \xrightarrow{gen} 000$	$110 \xrightarrow{gen} 010$	$101 \xrightarrow{gen} 111$
$100 \xrightarrow{gen} 101$	$110 \xrightarrow{gen} 100$	$011 \xrightarrow{gen} 101$
$010 \xrightarrow{gen} 001$	$110 \xrightarrow{gen} 111$	$011 \xrightarrow{gen} 111$
$010 \xrightarrow{gen} 000$	$110 \xrightarrow{gen} 000$	$011 \xrightarrow{gen} 001$
$010 \xrightarrow{gen} 011$	$110 \xrightarrow{gen} 011$	$111 \xrightarrow{gen} 011$
	$110 \xrightarrow{gen} 101$	

Multivalued Extensions

A maximum value for each variable, $m_1, \ldots, m_n \geq 1$, giving us the following state space $S = \{0, \ldots, m_1\} \times \{0, \ldots, m_2\} \times \cdots \times \{0, \ldots, m_n\}$.

Multivalued Extensions

A maximum value for each variable, $m_1, \ldots, m_n \geq 1$, giving us the following state space $S = \{0, \ldots, m_1\} \times \{0, \ldots, m_2\} \times \cdots \times \{0, \ldots, m_n\}$.

THOMAS NETWORKS (Discrete multivalued networks with thresholds)

The network definition remains unchanged $f = (f_1, \ldots, f_n) \colon \mathbb{B}^n \to \mathbb{B}^n$, but additionally an $n \times n$ matrix T of thresholds has to be specified.

Multivalued Extensions

A maximum value for each variable, $m_1, \ldots, m_n \geq 1$, giving us the following state space $S = \{0, \ldots, m_1\} \times \{0, \ldots, m_2\} \times \cdots \times \{0, \ldots, m_n\}$.

Thomas networks (Discrete multivalued networks with thresholds)

The network definition remains unchanged $f = (f_1, \ldots, f_n)\colon \mathbb{B}^n \to \mathbb{B}^n$, but additionally an $n \times n$ matrix T of thresholds has to be specified.

Example:

$$
\begin{aligned}
f_1(\mathbf{x}) &= \delta_1(\mathbf{x})_3 \wedge (\neg\delta_1(\mathbf{x})_1 \vee \neg\delta_1(\mathbf{x})_2) \\
f_2(\mathbf{x}) &= \delta_2(\mathbf{x})_1 \wedge \delta_2(\mathbf{x})_3 \\
f_3(\mathbf{x}) &= \delta_3(\mathbf{x})_1 \vee \delta_3(\mathbf{x})_2 \vee \delta_3(\mathbf{x})_3
\end{aligned}
$$

Where:

$\delta_i(\mathbf{x}) = (T_{1,i} \leq \mathbf{x}_1, \ldots, T_{n,i} \leq \mathbf{x}_n)$

Multivalued Extensions

A maximum value for each variable, $m_1, \ldots, m_n \geq 1$, giving us the following state space $S = \{0, \ldots, m_1\} \times \{0, \ldots, m_2\} \times \cdots \times \{0, \ldots, m_n\}$.

THOMAS NETWORKS (Discrete multivalued networks with thresholds)

The network definition remains unchanged $f = (f_1, \ldots, f_n)\colon \mathbb{B}^n \to \mathbb{B}^n$, but additionally an $n \times n$ matrix T of thresholds has to be specified.

EXAMPLE:

$$f_1(\mathbf{x}) = \delta_1(\mathbf{x})_3 \wedge (\neg\delta_1(\mathbf{x})_1 \vee \neg\delta_1(\mathbf{x})_2)$$
$$f_2(\mathbf{x}) = \delta_2(\mathbf{x})_1 \wedge \delta_2(\mathbf{x})_3$$
$$f_3(\mathbf{x}) = \delta_3(\mathbf{x})_1 \vee \delta_3(\mathbf{x})_2 \vee \delta_3(\mathbf{x})_3$$

Where:
$\delta_i(\mathbf{x}) = (T_{1,i} \leq \mathbf{x}_1, \ldots, T_{n,i} \leq \mathbf{x}_n)$

$W \subseteq \{1, \ldots, n\}$
$\Phi_W\colon S \to S$
$\Phi_W\colon \mathbf{x} \mapsto \mathbf{y}$ where for $i \in \{1, \ldots, n\}$:

$$\mathbf{y}_i \triangleq \begin{cases} max(0, \mathbf{x}_i - 1) & \text{if } i \in W \\ & \text{and } f_i(\mathbf{x}) = 0 \\ min(m_i, \mathbf{x}_i + 1) & \text{if } i \in W \\ & \text{and } f_i(\mathbf{x}) = 1 \\ \mathbf{x}_i & \text{if } i \notin W \end{cases}$$