Universal sequential Boolean automata networks

Maximilien Gadouleau
Durham University

Joint work with Alonso Castillo-Ramirez

St Andrews, April 08 2015

Outline

Sequential Boolean automata networks

Minimum size and time for universal networks

Complete networks

The need for asynchronism

Outline

Sequential BANs

A Boolean Automata Network (BAN) of size m is a collection of m automata, each having a Boolean state $x_i \in \{0,1\}$ and an update function $f_i : \{0,1\}^m \to \{0,1\}$.
The state of the network is $x = (x_1, \ldots, x_m) \in \{0,1\}^m$ and the network is viewed as

$$f = (f_1, \ldots, f_m) \in \mathrm{Tran}(\{0,1\}^m).$$

A BAN is sequential if at each time step, only one automaton updates its state. We denote this update by

$$xF^{(i)} = (x_1, \ldots, xf_i, \ldots, x_m).$$

As such, we are interested in the semigroup

$$S_f := \langle F^{(1)}, \ldots, F^{(m)} \rangle \leq \mathrm{Tran}(\{0,1\}^m).$$

(Henceforth, "network" refers to a sequential BAN.)

Why study BANs?

1. Many real-life networks can be modelled as BANs (gene networks, neural networks, epidemics...).
   The dynamics of these systems are difficult to predict, especially when updates are asynchronous.
2. A lot of work has been done on computational models based on similar ideas (Cellular automata, Automata networks from Tchuente or Dőmősi and Nehaniv, etc.).
   We want to understand what can be computed and how.
3. There is more than meets the eye.

Computing by sequential BANs

Let $n \leq m$ and $g \in \mathrm{Tran}(\{0,1\}^n)$.

The network f computes (simulates) g if there exists $h \in S_f$ such that

$$h\mathrm{pr}_{1..n} = \mathrm{pr}_{1..n}g,$$

or in other words, if for all $x \in \{0,1\}^m$,

$$(xh)_{1..n} = x_{1..n}g$$

The network f is n-universal if it can compute any $g \in \mathrm{Tran}(\{0,1\}^n)$.

Universal network

Theorem

For any $n \geq 1$, there exists an n-universal network.

Proof.

Construction: Take $x_{1..n}$, copy it into $x_{n+1..2n}$, and describe the transformation g.

Basic ingredient for the description: the one-bit switch. It has two automata s and t, where

$$xf_t = x_s$$
$$xf_s = x_s + 1.$$

The first line initialises the switch to OFF, while the second line flicks it ON and OFF. □

Outline

Minimum size

Theorem

For any $n \geq 1$, there is no n-universal network of size n.
Conversely, there is an n-universal network of size $n+2$.

For size n, there are two main refinements:

1. Unless $n = 2$, there is an network of size n which can compute any permutation of $\{0,1\}^n$ (i.e. $\mathrm{Sym}(\{0,1\}^n) \leq S_f$). Similar results hold for $\mathrm{Alt}(\{0,1\}^n)$ and $\mathrm{GL}(2,n)$ [Cameron, Fairbairn, G. 14a + 14b].
2. There is no network of size n which can compute all singular transformations (i.e. $\mathrm{Sing}(\{0,1\}^n) \not\leq S_f$).

The construction of size $n+2$ is based on memoryless computation (MC).

A bit on memoryless computation

In MC we have n registers, each gets updated one at a time.
Main difference with networks: the update functions
$f_i : \{0,1\}^n \to \{0,1\}$ can change over time.

Main results:

1. Any $g \in \mathrm{Tran}(\{0,1\}^n)$ can be computed in MC in time at most $4n-3$ [Burckel, Gioan, Thomé 09 + 14, G. and Riis 14].
   Conjecture: time $2n-1$.
2. Any $g \in \mathrm{Sym}(\{0,1\}^n)$ can be computed in MC in time at most $2n-1$ (tight) [Burckel, Gioan, Thomé 09, G. and Riis 14].
3. Any $M \in \mathrm{GLS}(2,n)$ can be computed in MC in time at most $\lfloor 3n/2 \rfloor$ (tight) [Cameron, Fairbairn, G. 14b].

n-universal network of size $n+2$

Lemma

Any transformation can be computed in MC using only these updates in time at most $(3+o(1))n2^n$*:*

$$\begin{aligned} xf_1' &= x_1 + \mathbb{1}\{x_{-1} = 0\ldots0\} \\ xf_k &= x_k + \mathbb{1}\{x_{-k} \neq 0\ldots0\} \qquad k = 1,\ldots,n \\ xf_2' &= x_2 + \mathbb{1}\{x = 0\ldots0\}. \end{aligned}$$

Corollary

There exists an n*-universal network of size* $n+2$ *and time at most* $(3+o(1))n2^n$.

Minimum time

Updating one automaton provides one bit of information.
We need $n2^n$ bits to describe g.
Thus, the time is at least $n2^n$...or is it?

Theorem

There exists an n-universal network with time $2^n + O(n)$.

Proof.

Construction composed of $x_{1..n}$, its copy in $x_{n+1..2n}$, and $2^{2^n} + (2^n + 1)$ automata y, z to describe $g_1, g_2, \ldots, g_n$ sequentially.
Encode $y \in \mathrm{GF}(2)^{2^{2^n}}$ using a $(2^{2^n} + 2^n + 1, 2^{2^n}, 3)$-shortened Hamming code to get (y, z).
We can then describe g_1 as such: If g_1 is the a-th in the ordering of Boolean functions, then flick the a-th bit of y. □

Minimum time

Conjecture

Any n-universal network has time at least 2^n.

Proof.

(For networks that work by describing $g_1, g_2, \ldots, g_n$ sequentially.)
Let $l^{(a)}$ denote the time to describe $g_1^{(a)}$. When describing $g_1^{(a)}$, say the transformation computed by automata $m+1..n$ has rank $r^{(a)}$. We then have $r^{(a)} \geq 2^{m-n-l^{(a)}}$ and $\sum_a r^{(a)} \leq 2^{m-n}$, thus

$$\sum_a 2^{-l^{(a)}} \leq 2^{n-m} \sum_a r^{(a)} \leq 1$$

and $\max_a l^{(a)} \geq 2^n$. □

(That was Kraft's inequality for prefix codes.)

Outline

Complete networks

An network is n-complete if it can compute any sequence of transformations (with possible gaps in between).

Formally, if for any sequence $g^{(1)}, \ldots, g^{(\ell)} \in \mathrm{Tran}(\{0,1\}^n)$, there exist $h^{(1)}, \ldots, h^{(\ell)} \in S_f$ such that for all $1 \leq i \leq \ell$,

$$\mathrm{pr}_{1..n} g^{(i)} = h^{(1)} \cdots h^{(i)} \mathrm{pr}_{1..n}.$$

The first example of an n-universal network was actually n-complete.

Minimum size for complete networks

Theorem

Any n-complete network has size at least $2n$.
Conversely, there exists an n-complete network of size $2n+4$.

Proof.

Let f be an n-complete network. It computes the following in order: $g^{(a_1)}, g^{(a_2)}, \ldots, g^{(a_{2^n})}$ where $a_i \in \{0,1\}^n$ and

$$x g^{(a_i)} = \begin{cases} (1,0,\ldots,0) & \text{if } x = a_i \\ (0,0,\ldots,0) & \text{otherwise.} \end{cases}$$

Denote the state of the automata of f when $g^{(a_i)}$ is computed as $h^{(a_i)} = (g^{(a_i)}, l^{(a_i)})$. Then $\operatorname{rank}(l^{(a_1)}) \geq 2^n - 1$ and there are at least $\lceil \log(2^n - 1) \rceil = n$ additional automata.

The construction of size $2n+4$ is the universal network of size $n+2$ with a copy of $x_{1..n}$ and a copy-back switch. □

Minimum time for complete networks

The minimum time is defined over any ordering $g^{(1)}, \ldots, g^{(2^{n2^n})}$ of $\mathrm{Tran}(2^n)$.

Theorem

Any n-complete network has minimum time at least 2^{n2^n}. Conversely, there exists an n-complete network with minimum time $(1+o(1))2^{n2^n}$.

Proof.

(For time $(2+o(1))2^{n2^n}$.)

The construction counts $\mathrm{Tran}(\{0,1\}^n)$ using two Gray codes:

1. An $(n, 2^{2^n})$-Gray code to make sure that computing the next g only requires updating one automaton in $1..n$.
2. An $(n2^n, 2)$-Gray code for the automata that implement the counter.

□

Outline

Parallel computation

Let us consider parallel BANs, where all automata update their states at the same time. This time, we are interested in

$$\langle f \rangle = \{\mathrm{id}, f, f^2, \ldots, f^k\}.$$

Theorem

For any $n \geq 1$, there is no n-universal parallel BAN.

Proof.

If f is n-universal parallel, there exist $k_a < k_b$ such that for all $x \in \{0,1\}^m$, $xf^{k_a}\mathrm{pr}_{1..n} = a$ and $xf^{k_b}\mathrm{pr}_{1..n} = b$ for some $a \neq b \in \{0,1\}^n$. Then

$$b = xf^{k_b}\mathrm{pr}_{1..n} = (xf^{k_b-k_a})f^{k_a}\mathrm{pr}_{1..n} = a.$$

□

Therefore, asynchronism is compulsory for BANs!

Quasi-parallel computation

How much asynchronism do we need?

We consider quasi-parallel computation, where automata 1 to $m-1$ operate in parallel, and automaton m is only updated once.

Theorem

For any $n \geq 1$, there exists an n-complete quasi-parallel network.

Proof.

The construction is again based on a counter C with a reset switch. The second half of the switch is the m-th automaton. We also need $2^{n2^n}+1$ copies $x^d_{1..n}$ of $x_{1..n} = x^0_{1..n}$. □

Quasi-parallel construction

3 steps of Initialisation and then Computation:

$$x_{1..n} \leftarrow x_{1..n}^{C} g^{(C)}$$
$$x_{1..n}^{d} \leftarrow x_{1..n}^{d-1} \qquad d = 1, \ldots, 2^{n2^n} + 1 \qquad \text{(original } x_{1..n} \text{ in } x_{1..n}^{1}\text{)}$$
$$C \leftarrow ?$$

$$x_{m-1} \leftarrow x_m;$$
$$x_m \leftarrow x_m + 1; \qquad \text{(Switch is ON: reset)}$$
$$x_{1..n} \leftarrow x_{1..n}^{C} g^{(C)}$$
$$x_{1..n}^{d} \leftarrow x_{1..n}^{d-1} \qquad d = 1, \ldots, 2^{n2^n} + 1 \qquad \text{(original } x_{1..n} \text{ in } x_{1..n}^{2}\text{)}$$
$$C \leftarrow 2$$

$$x_{m-1} \leftarrow x_m; \qquad \text{(Switch is OFF: no reset)}$$
$$x_{1..n} \leftarrow x_{1..n}^{C} g^{(C)} \qquad \text{(original } x_{1..n} g^{(C)}\text{)}$$
$$x_{1..n}^{d} \leftarrow x_{1..n}^{d-1} \qquad d = 1, \ldots, 2^{n2^n} + 1$$
$$C \leftarrow C + 1 \mod (2^{n2^n} + 2)$$

$$x_{m-1} \leftarrow x_m;$$