THE COMPLEXITY OF SOME AUTOMATA NETWORKS

ERIC GOLES
UNIVERSIDAD ADOLFO IBANEZ
SANTIAGO, CHILE
ANTONIO.CHACC@GMAIL.COM

Let V a finite set of sites or vertices. An *Automata Network* defined on V is a triple $\mathcal{A} = (G, Q, (f_i : i \in V))$, where $G = (V, E)$ is a simple undirected graph, Q is the set of states, which is assumed to be finite, and $f_i : Q^{|V|} \to Q$ is the transition function associated to the vertex i. The set $Q^{|V|}$ is called the set of configurations, and the automaton's global transition function $F : Q^{|V|} \to Q^{|V|}$, is constructed with the local transition functions $(G, Q, (f_i : i \in V))$ and with some kind of updating rule, for instance a synchronous or a sequential one. We will be only interested in the case where $Q = \{0, 1\}$.

Consider the following problem: given an initial configuration $x(0) \in \{0, 1\}^{|V|}$ and a vertex $v \in V$, initially passive ($x_v(0) = 0$), determine if there exists a time $T > 0$ such that v is active ($x_v(T) = 1$), where $x(t) = F(x(t-1))$ and F is some synchronous global transition function (for example, the strict majority rule). We call this decision problem **PER**.

First I present the complexity of **PER** when the global transition function is bootstrap percolation, this is, the rule given by the local functions:

$$f_i(x) = \begin{cases} 1 & if \quad x_i = 1 \\ 1 & if \quad \sum_{j\in N(i)} x_j > \frac{|N(i)|}{2} \quad and \quad x_i = 0 \\ 0 & if \quad \sum_{j\in N(i)} x_j \leq \frac{|N(i)|}{2} \quad and \quad x_i = 0 \end{cases}$$

In other words, a passive vertex becomes active if the strict majority of its neighbors are active, and thereafter never changes its state. For this rule we show that the problem is in **NC** if the network belongs to the family of graphs with degree less or equals than 4, and over this threshold the problem is $P - Complete$, and we leave open the case of planar networks. The P-Completeness proves were done using simulating a monotone circuit, while the membership in **NC** was established with an time $\mathcal{O}(log^2(n)$ algorithm on a PRAM, using $\mathcal{O}(n^4)$ processors.

Consider now a little different rule: the simply majority function:

$$f_i(x) = \begin{cases} 1 & if \quad \sum_{j\in N(i)} x_j > \frac{|N(i)|}{2} \\ 0 & if \quad \sum_{j\in N(i)} x_j < \frac{|N(i)|}{2} \\ x_i & if \quad \sum_{j\in N(i)} x_j = \frac{|N(i)|}{2} \end{cases}$$

For this rule we have that the problem is $P-Complete$ even if the network belongs to the family of planar graphs. This was proven simulating a non-necessarily planar circuit with a planar graph, using a gadget that allows us to 'cross cables' without mixing information using some 'traffic lights'.

Finally, I present the complexity of the problem **PER** with some fixed rule, when we change the iteration mode. That is to say, in this situation the complexity is associated with the update schedule of the network. The usual iteration mode (as we considered above) is the parallel one (every site is updated at the same time). Other one is the sequential mode, sites are updated one by one in a prescribed periodic order. Between this two ones, there are a huge number of updating modes (some sites in parallel, other ones sequentially).

It is direct to notice that the complexity of **PER** with the bootstrap percolation rule is invariant over changes in the iteration mode.

Consider now the rule when the vertices have by local transition function f_i one of the following (arbitrary chosen and fixed):

$$AND(x) = \begin{cases} 1 & if \quad \forall j \in N(i)\ ,\ x(i) = 1 \\ 0 & if \quad \exists j \in N(i)\ ,\ x(i) = 0 \end{cases}$$

$$OR(x) = \begin{cases} 1 & if \quad \exists j \in N(i)\ ,\ x(i) = 1 \\ 0 & if \quad \forall j \in N(i)\ ,\ x(i) = 0 \end{cases}$$

And the sets $\overline{OR} = \{v \in V\ :\ f_v = OR\}$, $\overline{AND} = \{v \in V\ :\ f_v = AND\}$

And the restriction that in the initial configuration the active vertices (initially in state 1) can be only vertices in $\overline{OR}$.

In this case we have: For a iteration mode given by a word ω

(1) If $|\omega| > n$ then **PER** is P-Complete.
(2) If $|\omega| = n$
    (a) If we have that $\forall v \in \overline{AND}$, $\omega(v) \leq \omega(u)$ or $\omega(v) \geq \omega(u)$ $\forall u \in \overline{OR} \cap N(v)$, this is, if every AND vertex is updated after or before all its OR neighbors, then the problem is in **NC** (Notice that this case includes the synchronous iteration mode).
    (b) Otherwise, the problem is $P-complete$

Then we see that the complexity of **PER** with this rule highly depends on the iteration mode.

REFERENCES

E.Goles, P. Montealegre, I. Todinca, The Complexity of bootstrap percolation in arbitrary graphs, to appear in Theoretical Computer Science (2012).
P. Montealegre, Thesis of Mathematical Engineering, Universidad de Chile, Santiago, 2012.