Necessary and Sufficient Conditions for Reversibility in One Dimensional Cellular Automata

José Alberto Baptista Tomé

IST/INESC

Rua Alves Redol nº9 1000 Lisboa Portugal

Abstract

In this work it is proved that necessary and sufficient conditions for binary, regular, non-bounded, one dimensional cellular automata do exist. These conditions are derived from the definition of sufficiency for non reversibility and using a well known property of Boolean algebra. These conditions are dependent on the relative neighbourhood of the considered automaton and a graph search algorithm is derived for the establishment of these conditions and the associated listing of solutions. It is shown, however, that this general solution has an exponential complexity nature, depending on the number of cell neighbours. A simple example is also presented.

1: Introduction

One dimensional binary cellular automata are considered in this work, that is, with regular structure of cell units, and where each unit is a finite 2-state machine which state function (local map) acts on its neighbour states for the definition of the next cell state.

Regularity is assumed :

a) The local map is the same for every cell.

b) The relative neighbourhood $(V_0,V_1,...V_{n-1})$ is the same for every cell, where n represents the number of neighbours and Vi is a signed integer representing the relative position of neighbour i relatively to the considered cell.

The absolute neighbourhood of a cell Ci is the set of n cells Cj such that $i=j+V_k$ and $k \in [0,1,..n-1]$.

One considers here non-bounded cellular automata, that is, where the number of cells in the one dimensional array is infinite and the state of the cellular automaton is defined by the set of the binary states of every cell in the array, which means that the state space is also infinite. Moreover, one considers that every cell changes its state synchronously with every other cell in the automaton.

The possible applications of cellular automata are increasing and areas such as physical phenomena modulation [3],[4], pattern recognition [1],[2] or differential equations substitution [5] are long ago being reported. The theoretical understanding of these structures can be very useful in judging their computational capabilities and also in other possible more sophisticated architectures. One such theoretical problem is reversibility of cellular automata state space. A cellular automaton is considered reversible if its state space is a partition of cycles, that is, no two different states are direct ancestors of the same given state.

For the one dimensional case, procedures for detecting the reversibility of the cellular automaton, given the local map are known [7], as well as some necessary conditions for reversibility [8] and techniques for generating some reversible automata [9], but the solution for the necessary and sufficient conditions were not yet known, except for the work of the author [11], but in a completely different audience and with no reference to some problems, such as the computational complexity of the solution .

Using the opportunity given by this cellular automata conference, the author presents the established necessary and sufficient reversibility conditions for one-dimensional, non-bounded cellular automata.

2: A Set of Necessary Conditions for reversibility

Let Si be the state i of the cellular automaton state space, Cj the cell j of the automaton, X_j^i the binary state of cell j in state Si,F^π one of the possible local maps (π is the enumeration parameter) and $(x_j,x_{j+1},...,x_{j+n-1})^i$ the set of cell states $x_j,x_{j+1},...,x_{j+n-1}$ in state Si.

F^π is said to be reversible if the resulting state space of the cellular automaton is. That is, F^π is non-reversible if:

$\exists$ S,S1,S2 such that:

$$S1 \neq S2 \text{ and } S1 \rightarrow S \text{ and } S2 \rightarrow S \quad (1)$$

This means that, for every possible absolute neighbourhood (every possible j from -∞ to +∞) one has:

$F^{\pi}(x_j,x_{j+1},...,x_{j+n-1})^1 = F^{\pi}(x_j,x_{j+1},...,x_{j+n-1})^2$ (2)

and so a sufficient condition for non-reversibility is the ANDING of expressions (2) taken from every possible j:

$$\cap_j [F^{\pi}(x_j,x_{j+1},...,x_{j+n-1})^1 = F^{\pi}(x_j,x_{j+1},...,x_{j+n-1})^2] = 1 \quad (3)$$

Considering a pair of n bit wide windows which synchronously shift along S1 and S2 respectively,each of the window pair static position corresponds to an expression as (2). Thus a sufficient condition for non-reversibility is obtained if the windows are shifted from -∞ to +∞, and the final ANDING (3)of the terms is performed.

Now, since:

```
 if:
CONDITION ⇒ NON-REVERSIBILITY
              then:
```

REVERSIBILITY ⇒ $\overline{\text{CONDITION}}$

a necessary condition for reversibility has been obtained.

Take two states S1 and S2 on a 2 nearest neighbouring case (n=2), where S2 is the "all at zero" state and S1 has only two successive 1's with every other cell state 0. The only interesting window pair positions give expressions (2) as follows:

$$f(0,1) = f(0,0)$$
$$f(1,1) = f(0,0)$$
$$f(1,0) = f(0,0)$$

From these two states one gets the following sufficient condition for non-reversibility:

$[f(0,1) = f(0,0) \text{ AND } f(1,1) = f(0,0) \text{ AND } f(1,0) = f(0,0)] = 1$

and a necessary condition for reversibility is:

$[f(0,1) = \overline{f(0,0)} \text{ OR } f(1,1) = \overline{f(0,0)} \text{ OR } f(1,0) = \overline{f(0,0)}] = 1$

3: Necessary and Sufficient Conditions for Reversibility

Let one define:

Configuration is an n-element pattern, obtained from the contents of a window pair position and where each element is a bit pair - one bit from each window .

Set Configuration of a pair of states: the set of different configurations obtained by synchronously shifting (-∞ to + ∞) the windows in that state pair of the automaton state space.

Minimal Set Configuration : A Set Configuration which does not include any other Set Configuration obtained from any other possible pair of states.

Notice that each Configuration of a pair of states is associated to an expression as (2) and, as a consequence, to its complement P_{Ck}, where k is an enumeration index.

To each Set Configuration of a pair of states i is associated a necessary condition for reversibility P_{Ni}:

$$P_{Ni} = P_{C1} \text{ OR } P_{C2} \text{ OR } ... \text{ OR } P_{Cq}$$

Considering now two different pairs of states, i and j, a more restrictive necessary condition is obtained:

$$P_{Ni,j} = P_{Ni} \text{ AND } P_{Nj}$$

If P_{Ni} and P_{Nj} are as follows:

$$P_{Ni} = P_{C1} \text{ OR } P_{C2} \text{ OR..OR } P_{Cq}$$
$$P_{Nj} = P_{C1} \text{ OR } P_{C2} \text{ OR..OR } P_{Cq} \text{ OR } P_{Ch} \text{ ..OR } P_{Ck}$$

it follows that $P_{Ni,j} = P_{Ni}$, since

A+A.B = A if Boolean algebra operations are considered.

As a direct corollary:

Corollary 1

A necessary condition for reversibility of a cellular automaton state space is obtained by ANDING the necessary conditions P_{Ni} obtained from the minimal set configurations, only!

Notice that the number of minimal set configurations is finite, as well as the number of terms P_C in each of them, despite the fact that the number of states in the state space is infinite.

Consider now every possible pair of states. From each one, it has been seen that a sufficient condition for non-reversibility, as in (3), may be derived (as well as a necessary condition for reversibility), which is also the necessary and sufficient condition for S1 and S2 to have the same direct successor (by definition). Then, ORING all of these conditions gives the necessary and sufficient condition for at least one of the possible pair of states to have the same successor, that is, the necessary and sufficient condition for non-reversibility. Complementing one obtains the necessary and sufficient condition for reversibility - an ANDING of necessary conditions P_N . But, as seen above, all terms in that ANDING are eliminated, except for those corresponding to minimal set configurations. ***Thus the necessary condition of corollary 1 is also the necessary and sufficient condition!***

4: Algorithm for minimal set configurations.

Following the necessary and sufficient condition just seen, it is now necessary to derive the minimal set configurations.

By definition, the number of different configurations is 2^{2N} (where N is the number of neighbours of each cell) and, as the windows are shifted one cell at a time , there is a relation between two consecutive configurations. If a configuration is y0 y1 ..yn-1, where yi ∈ [00,01,10,11], after shifting the window pair , the only possible configurations are four: y1 ...yn-1 *yn* with *yn* ∈ [00,01,10,11].

A graph may now be built, with a node per different configuration, and where a branch links node Ci to node Cj if and only if Cj corresponds to a possible configuration after the window pair is right shifted from Ci. In figure 1 it may be observed such a graph for the example of an automaton with nearest neighbouring and N=2.

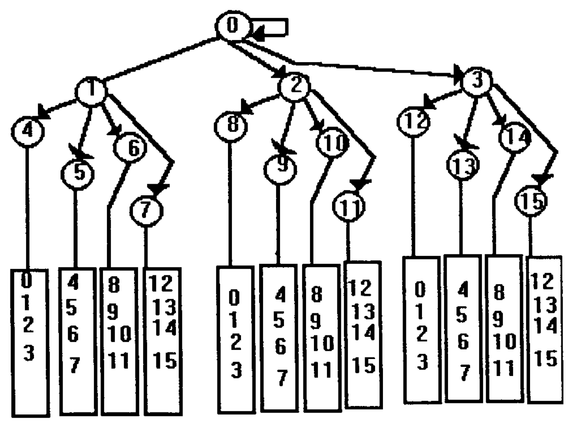

Figure 1

Minimal set configurations correspond to paths (minimal paths) in this graph which nodes do not contain as a sub set the nodes of any other possible set configuration. Moreover there is a condition to which these paths must obey, as follows:

The possible paths are cycles, that is the BEGIN node is a direct successor of the END node.

The proof is direct since the number of nodes in the graph is finite. The problem is then the search for cycles which do not contain other cycles.

An efficient algorithm [11] has been developed for the search of these minimal cycles, which is the same as the search for minimal set configurations.Table 1 shows the cycles for n=2 , the necessary and sufficient condition for reversibility extracted from those cycles and the local maps (cell logical functions) extracted from these conditions. Table 2 shows the number of cycles and number of searches used in the algorithm for n=3.

4.1 Complexity of the search algorithm

Bounds may be established for the search as follows:

Lower Bound: If N is the number of neighbours per cell a lower bound is of the order of 4^N.

A very simple lower bound may be established which shows the exponential nature of the problem. For a neighbourhood of N there are 2^{2N} configurations or nodes. Since the set of minimal cycles (associated to minimal set configurations) must include every one of this states it is proved the exponential nature of the problem.

Upper Bound: Notice that minimal cycles have at the maximum a length of $2^{2N}/4 = 4^{(N-1)}$. The proof is simple . First, each state has 4 direct successors, which means that the state space is a partition of sets of four states - I_m sets, where m is an enumeration index- (notice that 4 states, one from a different set I have the same I_m successor set). Now, a minimal cycle can not contain more than one node of each set Im. If this was true consider A,B ∈Im belonging to the same cycle : X-A-K-Z-B-X , then the cycle A-Z-K-A also exists, which means that the first one was not minimal. This demonstrates the maximum cycle length used above.

As the search is through a tree (each node with 4 leafs) an upper bound on that search is of the order of $4^{MaxLenght} = 16^{N-1}$

TABLE 1

Minimal Cycles (N=2)

1-4 1-6-8 1-6-11-12 1-7-12 1-7-14-8 2-8 2-9-4 2-9-7-12 2-11-12 2-11-13-14 3-12 3-13-4 3-13-6-8 3-14-8 3-14-9-4 6-9 6-11-13 7-13 7-14-9 11-14

Necessary and Sufficient condition from above cycles:

$[f(0,0)=\overline{f(1,0)}$ OR $f(0,0)=\overline{f(0,1)}]$ AND $[f(1,0)=\overline{f(0,1)}]$ AND $[f(1,1)=\overline{f(1,0)}$ OR $f(1,1)=\overline{f(0,1)}]$ AND $[f(1,1)=\overline{f(1,0)}$ OR $f(1,0)=\overline{f(0,1)}$ OR $f(0,1)=\overline{f(1,1)}]$ AND $[f(1,0)=\overline{f(0,1)}$ OR $f(0,0)=\overline{f(1,0)}$ OR $f(0,1)=\overline{f(0,0)}]$ AND $[f(1,1)=\overline{f(1,0)}$ OR $f(1,0)=\overline{f(0,1)}$ OR $f(0,0)=\overline{f(1,0)}]$ AND $[f(1,0)=\overline{f(1,1)}$ OR $f(0,0)=\overline{f(1,0)}]$ AND $[f(1,1)=\overline{f(0,1)}$ OR $f(1,0)=\overline{f(1,1)}$ OR $f(0,0)=\overline{f(1,0)}$ OR $f(0,1)=\overline{f(0,0)}]$ AND $[f(0,1)=\overline{f(1,1)}$ OR $f(0,0)=\overline{f(0,1)}]$ AND $[f(1,0)=\overline{f(0,1)}$ OR $f(0,1)=\overline{f(1,1)}$ OR $f(0,1)=\overline{f(0,0)}] = 1$

The local maps satisfying are 0101 1010 0011 1100 ; where the following input sequence of combinations is used :

[00,10,01,11] for the 2 inputs.

TABLE 2

Total of Minimal Cycles	120534
Total of Searches	1167626
Medium length of cycles	12

5: Conclusions

It has been shown that for one dimensional cellular automata there exists necessary and sufficient conditions for reversibility. For a given cellular automata that condition may be expressed in the form of a Boolean expression, which imposes restrictions on the local map output, and from this expression the local maps can be derived. The Boolean expression is derived from a set of cycles in a neighbourhood dependent graph. Unfortunately it is shown that the computational complexity involved is of exponential nature which implies severe practical restrictions for the determination of such conditions for non-trivial cases. A simple example is also shown in the text.

6: References

[1] J.B.Tomé, "Low order arrays for geometric pattern detection", Proceedings of the Computer Architecture for Pattern Analysis and Image Database Management, Miami, 11985

[2] J.B.Tomé, "Pattern Detection with low order arrays - a safe algorithm" Proceedings of the Second International Conference on Image Processing, London , 1986

[3] N.Margoulous, "Physics-like Models of Computation" - PHYSICA D - Non Linear Phenomena, vol 10D, Jan 1984

[4] G:Vichniac, "Simulating Physics with Cellular Automata" PHYSICA D - Non Linear Phenomena, vol 10D, Jan 1984[5] T.Toffoli, "Cellular Automata as an alternative to differential equations in modelling physics" - Non Linear Phenomena, vol 10D, Jan 1984

[6] S.Wolfram, "Universality and Complexity in Cellular Automata", - Non Linear Phenomena, vol 10D, Jan 1984

[7] S.Amoroso,N.Patt, "Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures", J.Comput.Systems Sci. 6 1972

[8] A.Maruoka,M.Kimura, "Conditions for injectivity of global maps for tessellation automata", Information Contr. 1976

[9] T.Tofolli, "Cellular Automata Mechanics", Technical Report 208, Computer and Comm. Sci. Dept. - The Univ. of Michigan 1977

[10] H.Yamada,S.Amoroso, "Completeness Problem for Pattern generation in Tessellation Automata", J. Comput.Sci. 4, 1970

[11] J.B.Tomé, "Reversibility Conditions for Cellular Automata" - Neural Networks from Models to Applications. Editors: L.Personnaz and G.Dreyfus, I.D.S.E.C., Paris, 1988

[12] P.Sousa, "Minimal Set Configurations Algorithm"- Internal Report INESC , Lisboa, 1990