Nat Comput
DOI 10.1007/s11047-013-9374-9

Solving the parity problem in one-dimensional cellular automata

Heather Betel · Pedro P. B. de Oliveira · Paola Flocchini

Abstract The parity problem is a well-known benchmark task in various areas of computer science. Here we consider its version for one-dimensional, binary cellular automata, with periodic boundary conditions: if the initial configuration contains an odd number of 1s, the lattice should converge to all 1s; otherwise, it should converge to all 0s. Since the problem is ill-defined for even-sized lattices (which, by definition, would never be able to converge to 1), it suffices to account for odd-sized lattices only. We are interested in determining the minimal neighbourhood size that allows the problem to be solvable for any arbitrary initial configuration. On the one hand, we show that radius 2 is not sufficient, proving that there exists no radius 2 rule that can solve the parity problem, even in the simpler case of prime-sized lattices. On the other hand, we design a radius 4 rule that converges correctly for any initial configuration and formally prove its correctness. Whether or not there exists a radius 3 rule that solves the parity problem remains an open problem; however, we review recent data against a solution in radius 3, thus providing strong empirical evidence that there may not exist a radius 3 solution even for prime-sized lattices only, contrary to a recent conjecture in the literature.

Keywords Elementary cellular automata · Emergent computation · Parity problem · Density classification · De Bruijn graphs

H. Betel (✉) · P. Flocchini
School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Avenue, Ottawa, ON K1N 6N5, Canada
e-mail: hbetel@site.uottawa.ca

P. Flocchini
e-mail: flocchin@site.uottawa.ca

P. P. B. de Oliveira
Faculdade de Computação e Informática, Universidade Presbiteriana Mackenzie, Rua da Consolação 896, Consolação, São Paulo, SP 01302-907, Brazil
e-mail: pedrob@mackenzie.br

1 Introduction

Cellular automata (CAs) are dynamical systems discrete in time and space, whose dynamics has been extensively studied across a variety of disciplines from different perspectives (e.g., see Acerbi et al. 2009; Betel and Flocchini 2011; Boccara and Cheong 1993; Cattaneo et al. 2004; Dennunzio et al. 2009; Kurka 2003; Langton 1986; Voorhees 2009).

Understanding the nature of computations within cellular automata remains however an elusive problem. In fact, in spite of their long-proclaimed ability to perform computations, very little is still known as to how we should design the local state transitions towards achieving a given global behaviour. As examples are designed or found by search, it is inevitable to try to understand their underlying programming language; but the truth is, to this date, every attempt along these lines has fallen into the strenuous effort of trying to tame local state patterns towards the global state target, or trying to make sense of the latter in terms of the former (Griffeath and Moore 2003).

On the other hand, studying how to employ local actions to achieve desirable global behaviours is of utmost importance and extensively investigated in many other evolving systems (e.g., distributed systems, mobile robots, population protocols). In such systems, in fact, understanding the limitations and the power of local interactions

Published online: 18 May 2013

to solve global computations has immediate implications for the design of efficient and scalable solutions (e.g., see Angluin 1980; Angluin et al. 2007; Lenzen et al. 2008; Peleg 2000). CAs are the simplest possible evolving systems, and understanding the impact that neighbourhood size has on computability could have consequences for more complex systems based on local interactions.

This paper is aligned with these efforts. Here, we concentrate on the one-dimensional *parity problem*, which has essentially the objective of figuring out the parity of an arbitrary binary string, by means of a one-dimensional, binary cellular automaton (Sipper 1998). The parity problem is a well-known benchmark task in various areas of computer science, typically camouflaged under the XOR operation on a binary input, as in artificial neural networks (Haykin 2008), but it also lends itself to the context of CAs, as a typical case of a global problem that has to be solved by purely local processing. The problem is formulated under periodic boundary conditions and arbitrary finite lattice size, so that, if the parity of the global configuration is odd, the lattice is supposed to lead to an homogeneous configuration with only 1s; otherwise, it should converge to all 0s (Lee et al. 2001).

The notion of parity has appeared quite often in the CA literature, even if implicitly, as it bears relevance to the related notion of additivity of CA rules (Chaudhuri et al. 1997; Dennunzio et al. 2009; Voorhees 2009). However, the parity problem per se has not been extensively studied, particularly in comparison with the well-known benchmark CA task of *density classification*, where the aim is to determine the most frequent bit in the initial configuration of an odd-sized lattice, also by reaching an homogeneous configuration. The density classification problem, in fact, has been extensively investigated and is fully understood in odd-sized lattices. In particular, it has been shown that there exists no single rule able to solve the problem for any arbitrary initial configuration. Combinations of rules have been devised, however, as well as probabilistic solutions to the problem (e.g., see Fatès 2011, Fukś 1997, de Oliveira et al. 2006, Wolz and de Oliveira 2008).

An advantage in favour of the parity problem is that, from the perspective of automata theory, it is simpler than its kin, insofar as the notion of parity can be handled by finite automata, whereas the ability to compare arbitrarily variable quantities (which is inherent to density classification) requires at least a pushdown automaton (Hopcroft et al. 2006). In fact, the increased simplicity of the parity problem is reflected in the fact that it is easier to find good rules for it, by searching, than for density classification (Wolz and de Oliveira 2008). Therefore, there are strong reasons for considering the parity problem generally more tractable and amenable to analysis, which makes it a serious candidate for case studies that might help the understanding of the nature of computation in CAs in general.

The parity problem is ill-defined for even-sized lattices (by definition, an all 1 configuration converges to an all 0 configuration making it impossible for any rule to converge to 1). Modifying the definition of the problem to allow the target homogeneous configuration to be achieved only once, and not as a fixed point, the problem becomes solvable also for even lattices. In fact, by relying on this variation, it can be perfectly solved by a carefully engineered sequence of rule applications, quite surprisingly, of elementary CAs (Martins and de Oliveira 2009). However, if we do not want to change the definition of the problem, it is then necessary to restrict the study to odd-sized lattices. We then say that a CA rule is *perfect* if it solves the parity problem for arbitrary initial odd-sized configurations.

Unlike the density classification problem, we show that the parity problem can, indeed, be solved by a single rule. Besides being interested in its general solvability, we are also interested in determining the minimal neighbourhood that allows the construction of a perfect rule. With this goal in mind, we first prove that radius 2 is not sufficient for a perfect rule to exist. We first identify several constraints to which such a perfect rule is subject and we show that no rule is feasible with all of these constraints. We then show that the problem becomes solvable when CAs have radius 4: our proof is constructive as we design a perfect rule and we prove its correctness. We leave open the case of whether or not there exists a radius 3 rule that solves the parity problem; however, by reviewing recent data against a solution in radius 3, we provide strong empirical evidence that no such rule exists, even restricting to prime-sized lattices, contrary to the conjecture in (Wolz and de Oliveira 2008).

2 Notation and basic facts

We consider one-dimensional, binary CA on finite lattices with periodic boundary conditions, also called circular CAs. Let $f:\{0,1\}^{2r+1} \to \{0,1\}$ denote the local rule of a CA with radius r. The global dynamics of a one-dimensional cellular automaton composed of n cells and of radius r is then defined by the global rule (or transition function): $F:\{0,1\}^n \to \{0,1\}^n$ *s.t.* $\forall \mathbf{X} \in \{0,1\}^n, \forall i \in \{0,\ldots,n-1\}$, $F(X)_i = f(x_{i-r} \ldots, x_i, \ldots, x_{i+r})$, where all operations on indices are modulo n.

We will use the word *configuration* to refer to an element of $\{0,1\}^n$, and will describe it as having *size* n. Shorter sequences of bits will be called *blocks*. We will again refer to the number of bits in a block as its size.

Furthermore, we will often refer to an instance of the local rule with specified input and resulting output as a *transition*. If the output differs from the central value of the input, so that the local rule applied to a cell having the neighbourhood described would result in the cell changing values, we will call this an *active transition*.

A *fixed point* $\mathbf{P} \in \{0,1\}^n$ of a circular CA with global transition rule F is a configuration $\mathbf{P}$ such that $F(\mathbf{P}) = \mathbf{P}$.

We say that a cellular automaton *converges to a configuration* P from configuration X^0 if P is a fixed point and if for some finite n, $F^n(X^0) = P$ where F^n is the n^{th} iteration of F. We are particularly interested in the homogeneous configurations as fixed points and will refer to these as the *0-configuration* and the *1-configuration*.

For an arbitrary configuration, we say that it has *odd (resp. even) parity* if it contains an odd (resp. even) number of 1s.

We now recall the definition of de Bruijn graphs, which are useful tools for representing CA rules and which will be used in the subsequent sections. The de Bruijn graph of a local rule of radius r is a directed graph on 2^{2r} nodes, one for each value in the set $\{0,1\}^{2r}$. There is an edge from node $x_0 \ldots x_{2r-1}$ to node $y_1 \ldots y_{2r}$ if $x_i = y_i$ for all i from 1 to $2r - 1$. These edges are labelled with the value of the local function at $(x_0, \ldots, x_{2r-1}, y_{2r})$, $f(x_0, \ldots, x_{2r-1}, y_{2r}) = f(x_0, y_1, \ldots, y_{2r})$. Note that the shape of the de Bruijn graph for a local rule of a given family (i.e., those with the same neighbourhood and states) is fixed, only the edge labels change. For example, Fig. 1 shows the de Bruijn graph for a radius 1 rule.

We say that a local rule solves the *parity problem* if, starting from an arbitrary initial configuration, on an arbitrarily sized lattice, the cellular automaton converges to the 0-configuration, if and only if the initial configuration contains an even number of 1s, and converges to the 1-configuration otherwise.

Since a rule solving the parity problem must converge to the homogeneous configurations, we have our first two simple properties of perfect rules.

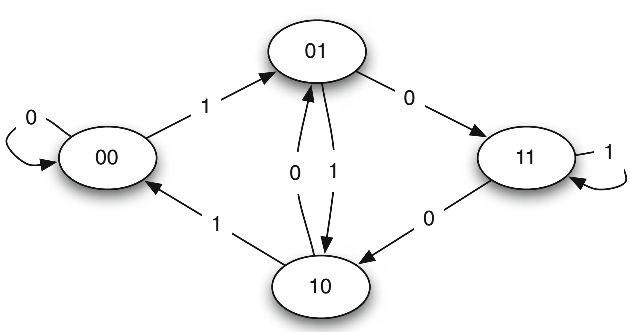

Fig. 1 De Bruijn graph for the local parity rule (150)

Property 1 *If* f *solves the parity problem, then* $f(0, \ldots, 0) = 0$ and $f(1, \cdots, 1) = 1$.

It is immediately obvious that, by definition, no solution exists for even-sized lattices.

Theorem 1 *Consider circular CAs with radius r and even size n. There exists no rule that works correctly from any initial configuration.*

Proof Trivially $f(1\ldots1) = 0$ otherwise the configuration with all 1s would incorrectly converge to 1. Since $(11111\ldots111)$ is not a fixed point, it follows that no initial configuration can ever converge to the 1-configuration. □

For this reason, from now on, we consider only *odd*-sized lattices and we call a rule *perfect* if it solves the parity problem for any odd-sized lattice, starting from any initial configuration.

The following is a necessary condition for parity preservation.

Property 2 *In order for a rule to preserve the parity of a configuration, the number of active transitions must be even. That is, given a local rule f of radius r, and any configuration* $(x_0, \ldots, x_{n-1})$*, the number of times that* $f(x_{i-r}, \ldots, x_i, \ldots, x_{i+r}) \neq x_i$ *is even.*

It is also very simple to see that no solution exists for elementary circular CAs (i.e., with radius 1).

Theorem 2 *There exists no perfect rule for elementary CAs.*

Proof From Property 1, for any perfect rule, we must have $f(000) = 0$ and $f(111) = 1$. Now consider a configuration containing a single 1. In order to both maintain parity and move towards convergence, we must have $f(100) = f(010) = f(001) = 1$. Similarly, from the singleton 0, we must have $f(110) = f(101) = f(011) = 0$. So the only possible perfect rule is the local parity check (rule 150). However, it is easy to see that such a rule does not solve the parity problem for infinitely many initial configurations: for example, the configuration (00100) leads to cyclic behaviour. □

3 Impossibility with radius 2

In this section we show that with radius 2 it is impossible to construct a perfect parity rule.

Our aim is to show, first, several necessary transitions for a perfect rule, and second, the existence of a limited set of feasible pre-images for the two final homogeneous configurations. Each possible pair of feasible pre-images further induces necessary transitions for a perfect rule, significantly reducing the space of possible perfect rules.

We conclude the proof by showing that the few remaining rules have non-homogeneous fixed points. We begin with a series of lemmata that force certain transitions to 0 or 1.

Lemma 1 *Given a perfect rule, three or five of the following must transition to 1:* (10000), (01000), (00100), (00010), (00001).

Proof A configuration consisting of a single 1 must eventually converge to all 1s, hence the number of 1s in the configuration must increase. Furthermore, in order to maintain parity, it must increase to an odd number. The five blocks of size 5 above are the only ones occurring at the local level that are not all 0s, and therefore 3 or 5 of them must map to 1. □

Similarly,

Lemma 2 *Given a perfect parity rule, three or five of the following must transition to 0:* (01111), (10111), (11011), (11101), (11110).

Consider the de Bruijn graph for radius 2 rules. We equate the parity of a cycle in the graph with the parity of the set of its edges and we consider the set of all possible pre-images of the final 0- and 1- configurations. For a given global rule, a pre-image of a configuration is any configuration that is mapped to it by that rule. Any pre-image of the final 0-configuration must correspond to a cycle of odd size and even parity, while any pre-image of the final 1-configuration is a cycle of odd size and odd parity. Let $\mathcal{B}_0$ be the subgraph containing only the edges corresponding to transitions to 0 and $\mathcal{B}_1$ the subgraph containing the edges corresponding to transitions to 1.

Lemma 3 *Neither $\mathcal{B}_0$ nor $\mathcal{B}_1$ can contain a cycle of even size and odd parity.*

Proof A cycle of even size in either $\mathcal{B}_0$ or $\mathcal{B}_1$ will become a sequence having even parity at the next iteration since it will be either all 0s or all 1s , so this cycle itself will have changed parity. Assume that the de Bruijn graph of a rule F admits such a cycle, let C be such a cycle and let P be a cycle of odd size passing through a node of C. For the rule to be perfect, $F(P)$ must have the same parity as P. Now consider a new cycle P' formed from P by adding the cycle C where P passes through it. Since C has even size, P' has odd size. Since C has odd parity, P and P' have different parity. However, since $F(C)$ has even parity, $F(P') = F(P)$, hence the parity of P' has changed and F cannot be perfect. □

Lemma 4 *Let f be a perfect parity rule, then either: i)* $f(10101) = 1$ *and* $f(01010) = 0$, *or ii)* $f(10101) = 0$ *and* $f(01010) = 1$.

Proof This is a direct consequence of Lemma 3 since $f(10101) = f(01010)$ would imply the existence of the even cycle with odd parity (1010, 0101, 1010, 0101, 1010, 0101) either in $\mathcal{B}_0$ or in $\mathcal{B}_1$. □

Lemma 5 *Given a perfect parity rule, it is impossible to have four or more consecutive 0s in a pre-image of the 0-configuration.*

Proof Let the block (0000) be present in a pre-image P of the 0-configuration. Then the following neighbourhood blocks must also be present and must be transitioning to 0: (10000) and (00001). By Lemma 1, we must then have the following blocks transitioning to 1, so they may not occur in P: (01000), (00100), (00010). Hence our group of four 0s must be both preceded and followed by at least two 1s, thus entailing that we have the must have the following transitions $f(11000) = f(00011) = f(10000) = f(00001) = 0$. Consider now an initial configuration of size greater than 7 containing a single 1 surrounded by 0s. From S, we have that the subsequence 0001000 can only grow to 0011100, but, again from S, we have that, from 0011100, no growth is possible anymore, which is a contradiction. □

Analogously, we have:

Lemma 6 *Given a perfect parity rule, it is impossible to have four or more consecutive 1s in a pre-image of the 1-configuration.*

From Lemma 3, any feasible pre-image of the 0-configuration (resp. 1-configuration) corresponds to either a simple odd cycle c with even (resp. odd) parity, or the composition of cycles not containing any even cycle of odd parity.

So, to identify feasible pre-images for final configurations for lattice size n in the de Bruijn graph, we have to find at least one cycle of size n to be labeled 0 and one to be labeled 1, having the property that they do not include:

(i) the self-loops (which are forbidden by Lemmata 5 and 6),
(ii) the 2-cycle (0101,1010) (which is forbidden by Lemma 4); and
(iii) an even cycle with odd parity (forbidden by Lemma 3).

By inspecting all cycles of size 5, we obtain that:

Lemma 7 *In a perfect rule at least one of these three cycles in the de Bruijn graph must transition to 1:*

$B_1^5 =$ (0011, 0111, 1110, 1100, 1001) (corresponding to configuration: 00111)
$B_2^5 =$ (0000, 0001, 0010, 0100, 1000) (corresponding to configuration: 00001)

$B_3^5 = (0101, 1011, 0110, 1101, 1010)$ (corresponding to configuration: 01011)

and one of these must transition to 0:

$W_1^5 = (0001, 0011, 0110, 1100, 1000)$ (corresponding to configuration: 00011)
$W_2^5 = (0111, 1111, 1110, 1101, 1011)$ (corresponding to configuration: 01111)
$W_3^5 = (0010, 0101, 1010, 0100, 1001)$ (corresponding to configuration: 00101)

Proof B_1^5, B_2^5 and B_3^5 (resp. W_1^5, W_2^5 and W_3^5) are the only cycles corresponding to feasible pre-images for the 1-configuration (resp. 0-configuration) for lattices of size 5, which do not violate Lemmata 4, 5, and 6. □

Consider, now, lattices of size 7. All cycles of size 7 have been enumerated and the only cycles that do not contradict Lemmata 3, 4, 5, and 6 and correspond to feasible pre-images of the 1-configuration are:

$B_1^7 = (0000, 0001, 0011, 0111, 1110, 1100, 1000)$ (configuration: 0000111)
$B_2^7 = (0001, 0011, 0110, 1101, 1010, 0100, 1000)$ (configuration: 0001101)
$B_3^7 = (0001, 0010, 0101, 1011, 0110, 1100, 1000)$ (configuration: 0001011)
$B_4^7 = (1001, 0011, 0110, 1100, 1001, 0010, 0100)$ (configuration: 1001100)

Analogously, the only cycles which do not violate Lemmata 3, 4, 5, and 6 and correspond to feasible pre-images of the 0-configuration are:

$W_1^7 = (0001, 0011, 0111, 1111, 1110, 1100, 1000)$ (configuration: 0001111)
$W_2^7 = (0010, 0101, 1011, 0111, 1110, 1100, 1001)$ (configuration: 0010111)
$W_3^7 = (0011, 0111, 1110, 1101, 1010, 0100, 1001)$ (configuration: 0011101)
$W_4^7 = (0110, 1100, 1001, 0011, 0110, 1101, 1011)$ (configuration: 0110011)

From simple observation, we can rule out some of these cycles and combinations of cycles.

Lemma 8 *A perfect rule of radius 2 cannot have* W_1^5 *as a pre-image of the 0-configuration.*

Proof Cycle W_1^5 shares at least one transition in common with each of the possible pre-images of the 1-configuration of size 7. For example, W_1^5, B_1^7 and B_2^7 all share the edge (0001, 0011) in the de Bruijn graph. Cycles W_1^5 and B_3^7 share (0010, 0101), and W_1^5 shares (0011,0110) with B_4^7. □

Lemma 9 *A perfect rule of radius 2 cannot have* B_2^7 *as a pre-image of the 1-configuration.*

Proof First, if B_2^7 is a pre-image of the 1-configuration, then W_2^5 is a pre-image of the 0-configuration of size 5 since B_2^7 and W_3^5 share (1010, 0100). Cycle B_2^7 also has transitions in common with W_1^7, W_3^7, W_4^7. It has no common transitions with W_2^7, however, W_2^7 and W_2^5 together form the cycle (0111, 1110, 1101, 1011), in violation of Lemma 3. □

Similarly,

Lemma 10 *we can show, A perfect rule of radius 2 cannot have* B_3^7 *as a pre-image of the 1-configuration.*

In fact, we can now restrict to very well defined possible cases.

Lemma 11 *A perfect rule of radius 2 must have* W_2^5 *as a pre-image of the 0-configuration,* B_2^5 *as a pre-image of the 1-configuration and either*

- B_1^7 *as a pre-image of the 1-configuration and* W_4^7 *as a pre-image of the 0-configuration, or*
- B_4^7 *as a pre-image of the 1-configuration and* W_1^7 *as a pre-image of the 0-configuration.*

Proof From the lemmata above, we know that the only possible pre-images of the 1-configuration of size 7 are B_1^7 and B_4^7. Cycle B_1^7 has transitions in common with all possible pre-images of the 0-configuration except W_4^7. Of the 5-cycle pre-images of the 1-configuration, only B_2^5 is compatible with W_4^7. Of the possible pre-images of the 0-configuration of size 5, neither W_2^5 nor W_3^5 poses any conflict, however, W_3^5 makes it impossible to have any cycles of size 3 going to the 1-configuration, so that all configurations having a period of size 3, (i.e. configurations of the form $(001001001\cdots001)$) will fail to converge. The proof is analogous beginning with cycle B_4^7. □

We can finally conclude:

Theorem 3 *There is no perfect parity rule of radius 2.*

Proof From the lemmata in this section it follows that for a perfect parity rule of radius 2, we must have one of the de Bruijn graphs shown in Figs. 2 or 3. However, from observation of Fig. 2 we see that (00111) is a fixed point, while (00011) is a fixed point for Fig. 3. □

A final note concerns lattices of prime size. It has been conjectured, in the case of radius 3, that there may exist rules with the desired behaviour on arbitrary lattices of prime size (Wolz and de Oliveira 2008). While the case of radius 3 is still open, we can show that the impossibility result for radius 2 holds even if we restrict the discussion to prime-sized lattices.

Theorem 4 *There is no radius 2 rule that always solves the parity problem even restricting to lattices of prime size.*

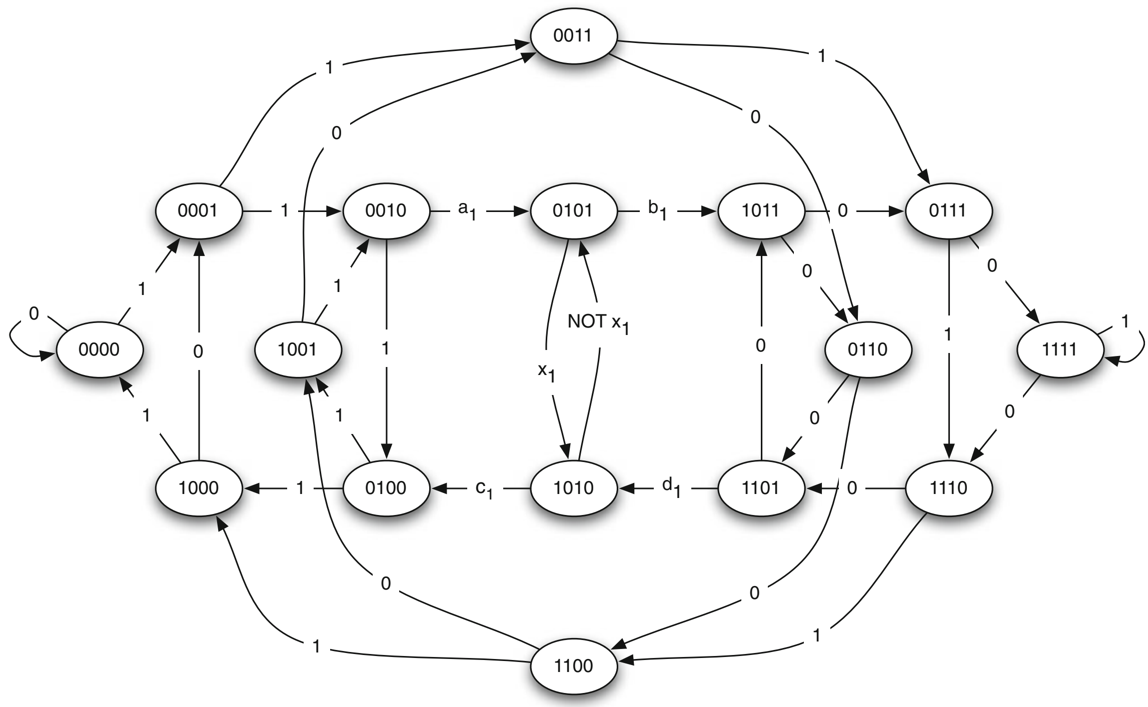

Fig. 2 Possible perfect rule for radius 2, with B_1^7 and W_4^7

Proof Restricting to prime-sized lattices, we can no longer use Lemma 11. This introduces only a few possible extra cases using B_3^5 or W_3^5. Consider rules containing B_3^5. As before, we can eliminate W_1^5, B_2^7 and B_3^7 as pre-images of the 0- and 1- configurations. In addition, B_3^5 conflicts with W_2^7 and W_3^7 on the edge from 0101 to 1011 and with W_4^7 on the edge from 0110 to 1101. Hence, we must have W_1^7 as the 7-cycle pre-image of the 0-configuration. Now, B_1^7 conflicts with W_1^7, so we are left with B_4^7 as the 7-cycle pre-image of the 1-configuration. Since B_4^7 conflicts with W_3^5 on the edge from 0100 to 1001, we are left with W_2^5 as the 5-cycle pre-image of the 0-configuration. These results are illustrated in the graph of Fig. 4. Proposition 2 dictates that the edges labeled a_3 must be the same, as will be the edges labeled b_3. Now consider rules containing W_3^5. Similar analysis shows that must have B_2^5, W_4^7 and B_1^7, as illustrated in Fig. 5. As before, these rules have fixed points at (00011) and (00111), respectively, for the rules given by Figs. 4 and 5. □

4 A perfect rule with radius 4

We now describe the construction of a rule with radius 4 having the desired properties: parity preservation and convergence to a homogeneous configuration. The intention is to first give the reader an intuitive understanding of how the rule works and how it was developed. Formal proofs will follow in the subsequent section.

4.1 Rule BFO

The most compact representation of rule BFO that we propose for solving the parity problem in any lattice of odd size is given in Fig. 6 and corresponds to rule number: 127660195799278877488283087836321251372089486295714341994- 04394002671695991869267727072917454377539194754200976283425175983876- 539715064584172642413634846720 in Wolfram's lexicographic ordering scheme. The figure shows all active transitions (i.e., transitions that change the current state). However, it is often easier to explain why and how the rule works using a less compact form, where pairs of rules can be made explicit. This form of the rule is given in Fig. 7; we will be referring to this representation in the remainder of this section.

We now describe the intended behaviour of the rule before proving its correctness. Consider an initial configuration X^0 as being formed by blocks b_i of consecutive 1s separated by blocks w_i of consecutive 0s: $X^0 = (b_1^0, w_1^0, b_2^0, w_2^0, \cdots, b_k^0, w_k^0 \cdots)$. The idea of our construction is to have a block b_i of 1s propagate to the right, two cells per iteration, until a stopping condition or convergence has been reached. Such propagation *might* result in merging the block with the next b_{i+1} (if the corresponding w_i is of even size). When the merger does not occur (because $|w_i|$ is odd or due to some other condition), there will be a propagation of 0s to the left, led by a block of the form (01). Such

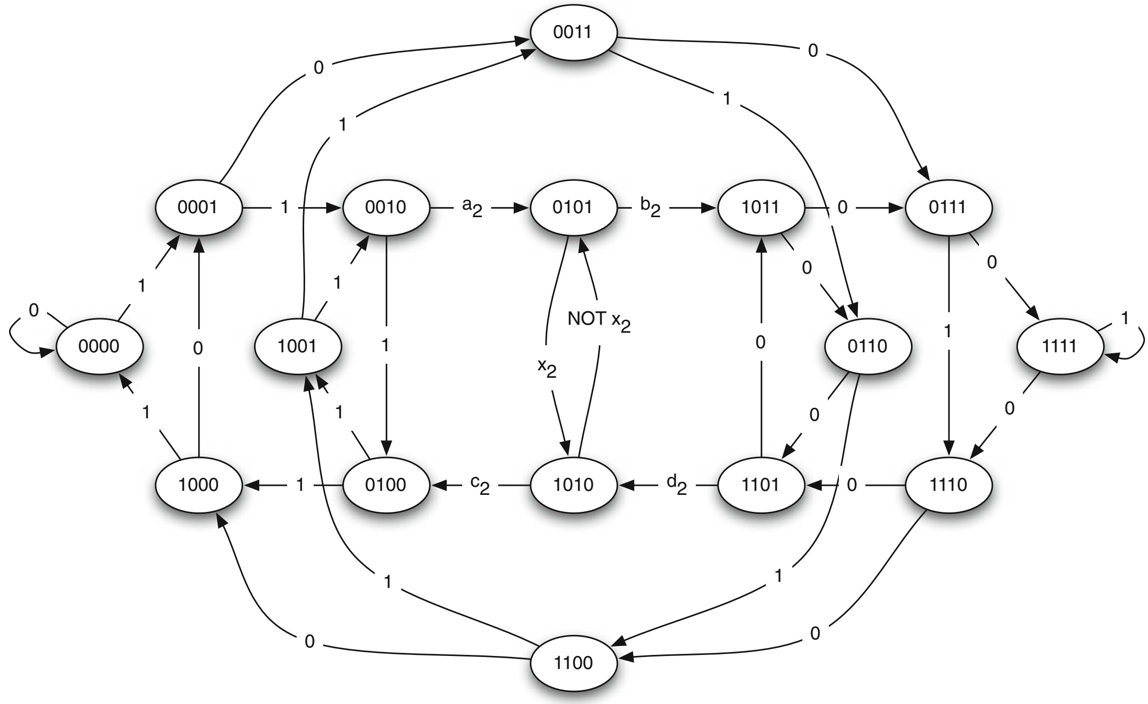

Fig. 3 Possible perfect rule for radius 2, with B_4^7 and W_1^7

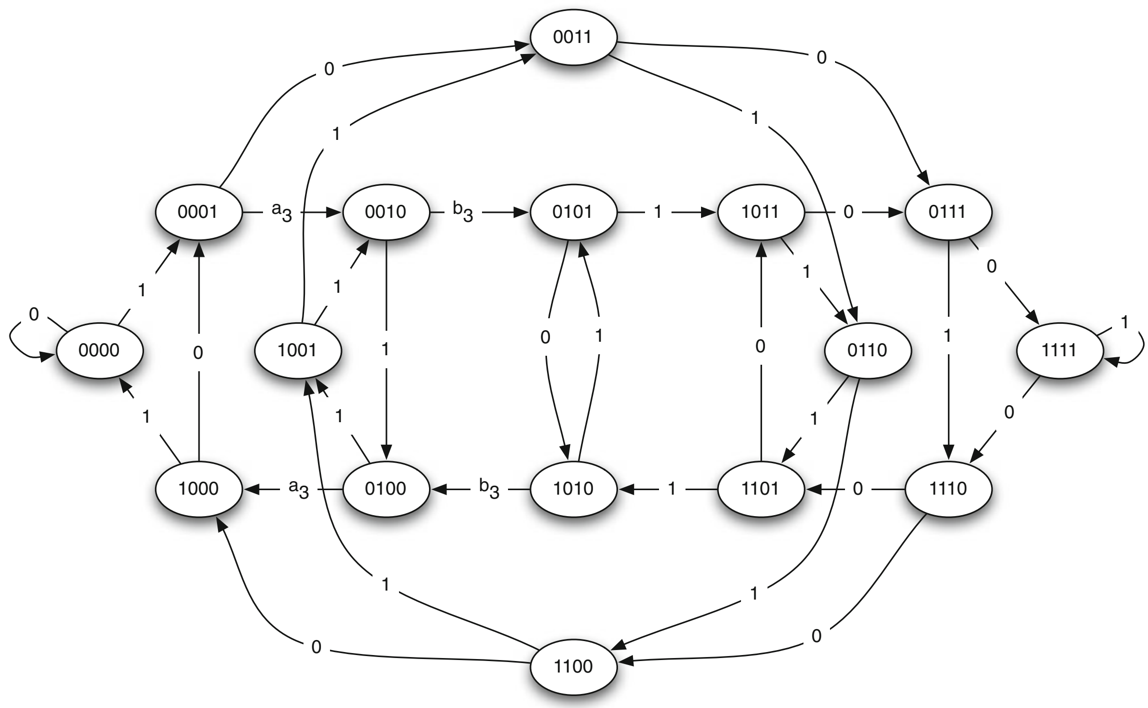

Fig. 4 Additional possible perfect rules for radius 2 on prime lattices with W_2^5

counter-propagation *might* result in the total annihilation of the block of 1s. Otherwise, it will result in the creation of a single 1 surrounded by 0s, which will start propagating to the right again. We will show that such behaviour reduces the number of blocks, eventually converging to an homogenous configuration.

We now describe some properties of the rule that can easily be derived by construction and that give an intuition

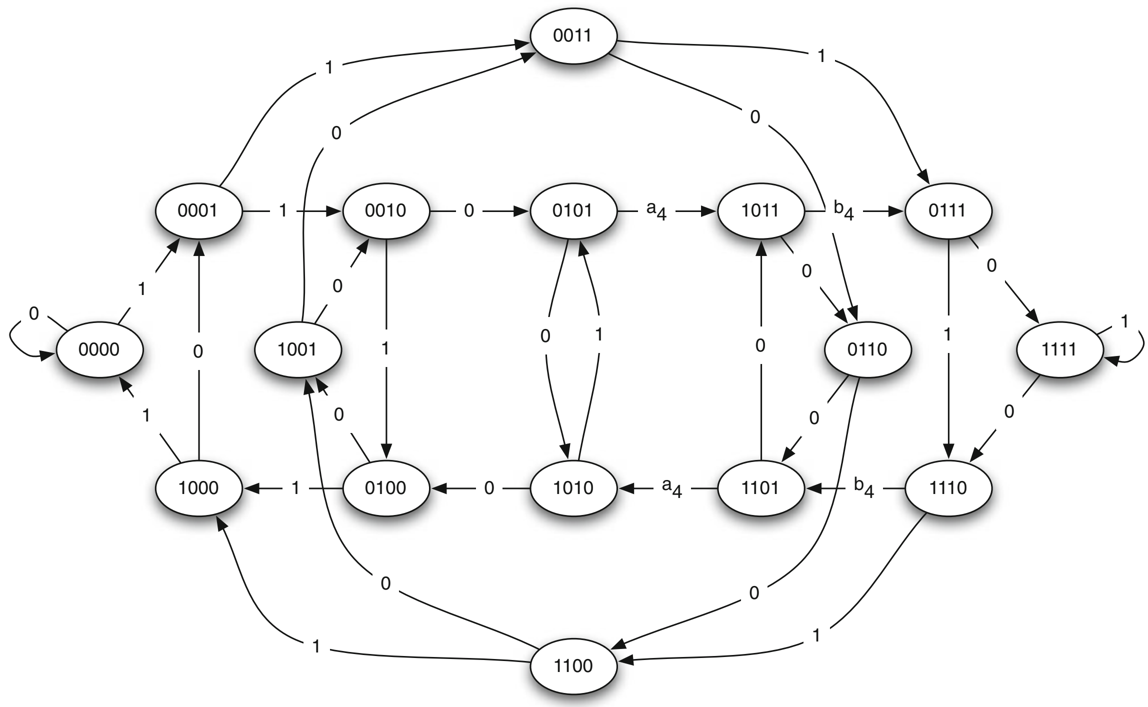

Fig. 5 Additional possible perfect rules for radius 2 on prime lattices with B_2^5

for the reasons for the behaviour described above. Note that, by construction, the rule's transitions always occur in pairs; in other words, whenever a transition occurs in a cell, another transition occurs in its neighbourhood. It is useful to describe the behaviour of each pair and we will also use these pairs to prove that parity is being preserved.

– *Rightward growth of 1-blocks* a singleton 1 grows to the right, by two 1s at each step, if it is preceded and followed by at least two 0s, as prescribed by transitions T_3 and T_4. A block of three or more 1s grows to the right if it is followed (on the right) by at least two 0s. This behaviour is created by the pair of transitions T_1 and T_2.
– *Annihilation of pairs of 1s* as a consequence of transitions T_5 and T_6, an isolated pair of 1s is always eliminated.
– *Leftward growth of 0-blocks* a (01) block moves to the left, leading a growing block of 0s (at a growth rate of two 0s per step) if there are at least three 1s to its left and one of the following: (*i*) at least three 1s to the right of the 0 (the growth is obtained by the pair of transitions, (T_9,T_{11}); or (*ii*) at least one 0 to its right (due to T_9 and T_{10}). Note that the pair (T_9,T_{11}) *starts* the growth of a 0-block, while the pair (T_9,T_{10}) *continues* the growth as far as possible.
– *Local shift* a (101) block is transformed into (110) if there are a 0 on its left and at least two 0s on its right (combination of transitions T_7 and T_8).

Neighbourhood configurations	Output bit
$*1110\mathbf{0}***$	1
$1110\mathbf{0}****$	1
$*0010\mathbf{0}***$	1
$0010\mathbf{0}****$	1
$**01\mathbf{0}100*$	1
$1110\mathbf{1}****$	0
$*010\mathbf{1}*0**$	0
$**01\mathbf{1}0***$	0
$***1\mathbf{1}0110$	0
$***0\mathbf{1}10**$	0
$****\mathbf{1}101*$	0

Fig. 6 Minimised rule BFO (the asterisk * refers to any value)

– *Local adjustment* finally, if a (0110) block is preceded by at least three 1s that is, (...1110110...) occurs, in order to avoid parity errors due to the annihilation of the pair of 1s, we force the creation of a solid block of 0s to the right of the existing block of 1s with transition pair (T_9, T_{12}), so that (...1110110...) becomes (...1000000...).

Examples of the evolution of the rule are given in Fig. 8.

4.2 Correctness

In order to show that the rule we have constructed (or in fact any rule) performs a perfect parity check, we must

Fig. 7 Active rule transitions (*left*) and behaviour of combinations of rules (*right*)

Name	Neighbourhood configurations	Output bit
T_1:	∗11100∗∗∗	1
T_2:	11100∗∗∗∗	1
T_3:	∗00100∗∗∗	1
T_4:	00100∗∗∗∗	1
T_7:	∗∗010100∗	1
T_5:	∗∗∗0110∗∗	0
T_6:	∗∗0110∗∗∗	0
T_8:	∗010100∗∗	0
T_9:	∗∗∗11101∗	0
T_{10}:	111010∗∗∗	0
T_{11}:	1110111∗∗	0
T_{12}:	∗∗1110110	0

Pair	Behaviour
T_1, T_2	Rightward growth of 1-blocks
T_3, T_4	Rightward growth of 1-blocks
T_5, T_6	Annihilation of 11
T_7, T_8	Local shift
T_9, T_{10}	Leftward growth of 0-blocks
T_9, T_{11}	Start of 0-growth
T_9, T_{12}	Local adjustment

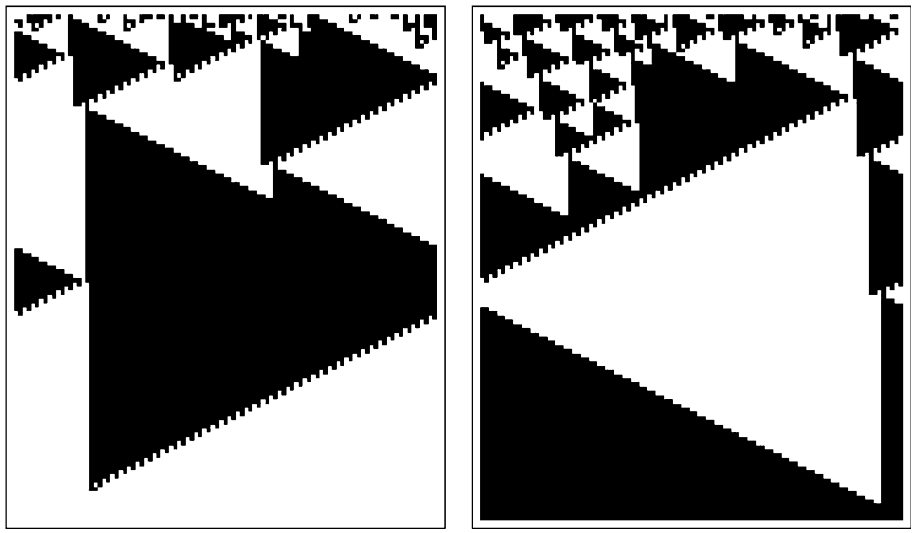

Fig. 8 Evolution of rule BFO for even parity (*left*) and odd parity (*right*). A *black cell* corresponds to 1, a *white cell* corresponds to 0. The initial configuration is at the top and time goes downward

prove that it preserves parity at every iteration, and that it always converges in finite time to an homogeneous configuration. We begin with the proof of parity conservation.

4.2.1 Parity preservation

A rule preserves the parity of a configuration if active transitions always come in pairs. That is, given a local rule f of radius r, and any configuration $(x_0, \ldots, x_{n-1})$, the number of times that $f(x_{i-r}, \ldots, x_i, \ldots, x_{i+r}) \neq x_i$ is even. To show that our rule does indeed have this property, we will use a modification of the de Bruijn graph.

Given a configuration $X = (x_1, \ldots x_n)$, one can determine the next iteration, $F(X)$ by reading the edge labels as one traverses the graph from $(x_{i-4} \ldots x_i \ldots x_{i+3})$ to $(x_{i-3} \ldots x_i \ldots x_{i+4})$. Since we are considering circular configurations, the traversal of the de Bruijn graph will result in a closed loop. For our purposes, we are not interested in what the actual output is, only if an active transition has occurred. In keeping with that, we are, in fact, only interested in the parts of the graph where such transitions occur. Since a de Bruijn graph for a function of radius 4 can be quite unwieldy, we define a *reduced transitional de Bruijn graph* modifying the standard de Bruijn graph as follows. First, we label the edges with T_i (indicating one of our 12 active state transitions), or N, meaning no transition. Second, we draw only those parts of the graph connected to transitions, reducing the rest of the graph to a single node, denoted by an asterisk. Also, where there is no conflict, several nodes leading to or from the same transition are depicted as one, using the notation of the previous section; for example, there is an edge from node (*11100**) to (11100***), because of the various state transitions entailed by T_1, given all possible values for the * symbols. Furthermore, we ensure that all nodes in the reduced transitional de Bruijn graph are distinct for all values of the * symbol. Finally, for any nodes occurring explicitly in the graph, all adjacent edges are represented, whether they correspond to a transition or not. It is easy to see that parity preservation can be detected from the reduced transitional de Bruijn graph for a given rule; more precisely:

Lemma 12 *A rule is parity preserving if and only if any cycle in its reduced transitional de Bruijn graph contains an even number of edges labeled with some transition* T_i.

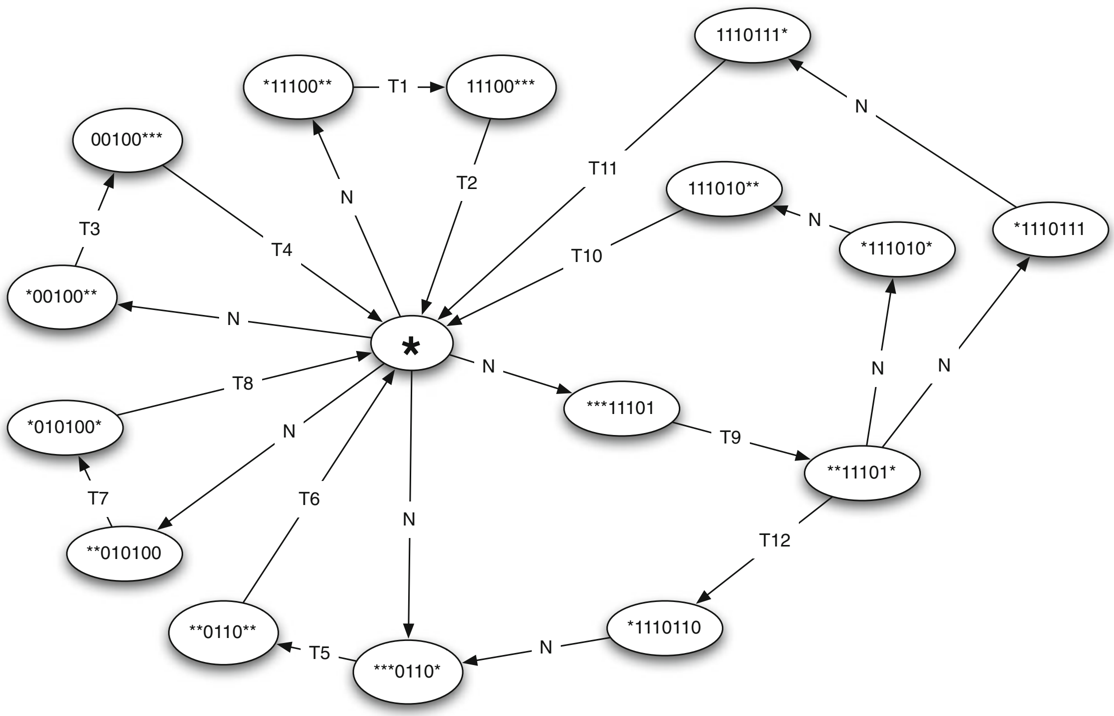

Fig. 9 Reduced transitional de Bruijn graph for rule BFO

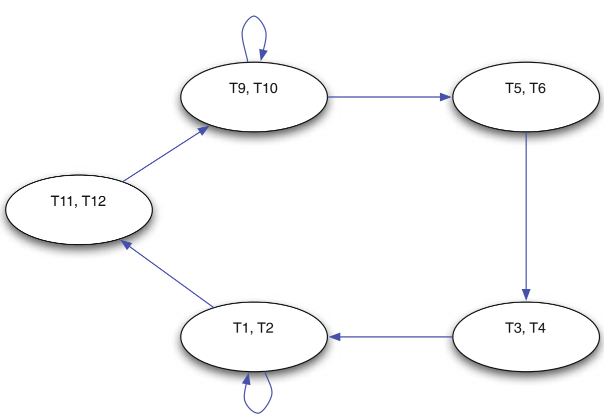

Fig. 10 Accordion loop

Proof Since the output of a circular CA is given by the edge labels of a cycle in its de Bruijn graph, we need only count the transitions to verify parity preservation. Furthermore, the reduced graph compresses only parts of the graphs where no transitions occur. Hence, if the are no cycles containing an odd number of transitions in the reduced graph, there can be no configurations leading to an odd number of transitions and vice versa. □

By inspecting the transitional de Bruijn graph for rule BFO (Fig. 9), and by noticing that there are no cycles containing an odd number of transitions, we then have:

Theorem 5 *Rule* BFO *is parity preserving.*

4.2.2 Convergence

We now turn our attention to the more challenging problem of proving that this rule will converge under any condition. We can think of any CA configuration as an alternating sequence of blocks of 0s and blocks of 1s of varying sizes. We will show that BFO eventually converges by showing that its only fixed points are the homogeneous configurations and, furthermore, that any change in the configuration will lead, in a finite number of iterations, to a reduction in the overall number of blocks. Our first lemma shows that every non-homogeneous configuration is changing.

Lemma 13 *The only fixed points of rule* BFO *are the homogeneous configurations.*

Proof A non-homogeneous fixed-point configuration cannot contain pairs of 1s, since rule pair (T_5,T_6) would apply. It cannot contain a block of three or more 1s since (T_1,T_2) would apply if it is followed by two or more 0s, and a transition pair containing T_9 would apply if it is followed by only one 0. Therefore, a non-homogeneous fixed-point configuration could only contain isolated 1s but the odd size of the configuration would imply that we must have at least two consecutive 0s, hence the sub-configuration 0100 must occur. If this is preceded by a 0, (T_3,T_4) apply. If it is preceded by a 01, (T_7,T_8) apply. □

We now show that every transition pair will eventually lead to a reduction in the total number of blocks. We begin with the transition pairs for which this is immediate.

Lemma 14 *Transition pairs* (T_5,T_6), (T_7,T_8) *and quadruplet* (T_9, T_{12}, T_5, T_6) *lead to block reduction in a single step*.

Lemma 15 *Transition pair* (T_9,T_{10}) *leads to block reduction in a finite number of steps.*

Proof Rule pair T_9:*f*(***1**1** 101*) = 0 and T_{10}:*f*(1110 **1** 0***) = 0 causes the leftward growth of 0-blocks. While a single application of this pair maintains the number of blocks, it leads (possibly through a repeated application of the pair) to an eventual block reduction through either an annihilation or the creation of a single 1:

$$\begin{aligned} &\rightsquigarrow 00001000\cdots \text{ by rules } T_5 \text{ and} T_6\\ 0111010\cdots &\rightsquigarrow 0101000\cdots \text{ by rules } T_9 \text{ and } T_{10}\\ &\rightsquigarrow 0110000\cdots \text{ by rules } T_7 \text{ and } T_8\\ &\rightsquigarrow 0000000\cdots \text{ by rules } T_5 \text{ and } T_6 \end{aligned}$$

Notice that (T_9,T_{10}) leads to reducing the number of blocks by either two or four, depending on the parity of the block of 1s on which it is acting. Also note that, even though it is possible for the leading block of 1s to have shrunk from the left side, while the (T_9, T_{10}) pair is reducing it from the right, one of these two situations will still be reached, since, on the left, the 1s can only be eliminated one at a time. □

Lemma 16 *Transition pairs* (T_1,T_2), *and* (T_3,T_4) *lead to either reduction or maintenance of the number of blocks.*

Proof Both pairs T_1:*f*(*111 **0** 0***) = 1, T_2:*f*(1110 **0** ****) = 1, and T_3:*f*(*001 **0** 0***) = 1, T_4:*f*(0010 **0** ****) = 1 are responsible for the rightward growth of 1-blocks. In fact, they grow a block of 1s until it either merges with the next block or an isolated 0 preceded by three or more 1s is created. At this point, one of the following transition sets will apply: (T_9,T_{10}) if the isolated 0 is followed by 10, (T_9, T_{12}, T_5, T_6) if it is followed by 110, and (T_9,T_{11}) if it is followed by three or more 1s. We have already seen that the first two cases lead to block reduction, only the latter case can maintain block numbers. □

Lemma 17 *Transition pair* (T_9,T_{11}) *leads to reduction or maintenance of the number of blocks in a finite number of steps.*

Proof The pair T_9:*f*(***1**1** 101*) = 0, T_{11}:*f*(***1 **1** 1011) = 0 is responsible for the start of 0-growth. This is the only rule pair that initially increases the number of blocks. It is the beginning of the growth of 0s from an isolated 0 surround by three or more 1s on either side. Once this growth has begun, it is continued by the (T_9,T_{10}) pair until the number of blocks has been returned to its original size or is reduced by two. □

Note that in the proof above, if the original number of blocks is maintained, it is because the transitions have produced an isolated 1 with two or more 0s on either side which then begins to grow to the right either merging with the block on the right (and thus reducing the total number of blocks) or creating an isolated 0 which then begins to grow left. What we wish to avoid is a CA which evolves to a periodic configuration of 1s growing until only an isolated 0 remains and then shrinking back to an isolated 1 which then regrows. We call this block of growing and shrinking *the accordion effect*. In the next lemma, we show that the accordion effect cannot occur on lattices of odd size.

Lemma 18 *The accordion effect can only occur on lattices of even size.*

Proof Figure 10 shows the transition pairs involved in a cycle of growing and shrinking. If we think of this loop as starting from the (T_9, T_{11}) node, a block of 1s is shrinking

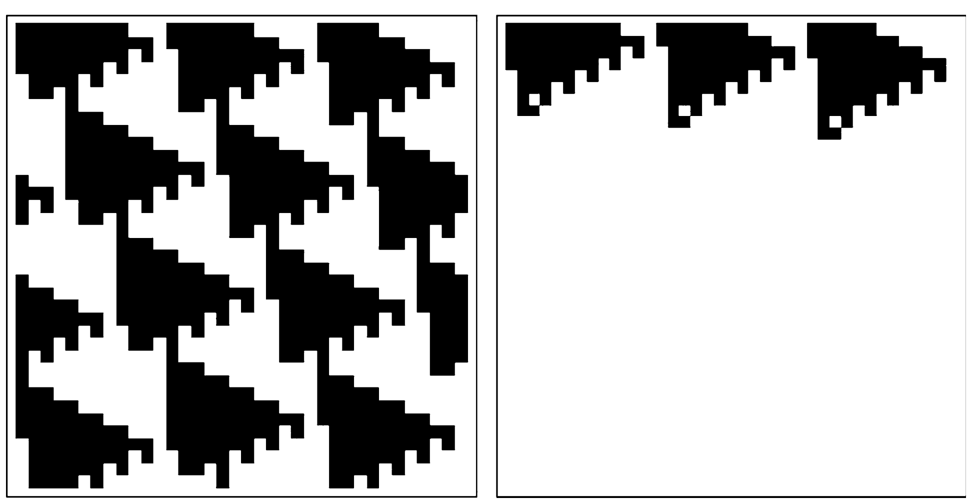

Fig. 11 The accordion effect on an even-sized lattice (*left*) and the successful resolution of an odd-sized lattice (*right*)

Index	Rule Number	Index	Rule Number
1	328447672826993550020983459564344832408	51	296641291816561632723600280019903963920
2	297492748577089511288345839143552794896	52	296475381706932217088987437103489999632
3	297494046666159425466677605435168768272	53	296475057188107759278668089718071091984
4	297473276211074685927935207853055525136	54	296808337742714963527092405349712388880
5	296474976058488678040367253460474913040	55	296641291817219288369471768588866020112
6	296808256613095882288791569092116209936	56	296474976058450001858872564260317160208
7	296475057188069064207707451447140671760	57	296808256613057206107296879891958457104
8	296808337742676268456131767078781968656	58	296475625099271138830182306581427774224
9	296641291817180593298511130317935599888	59	296475057187488789259024273680748241680
10	296474976060964558118992057205935825168	60	296474976057831031839228748222994309904
11	296474976057869708020723437423152062736	61	296661975438027476540473486768714404176
12	296475057187450094188063635409817821456	62	296661894308331033494450293177369874768
13	296641291816561623278867314280612749584	63	296661894308369719120677948116823036240
14	296474976058411306787911969969856045328	64	296661975437988790914245814237079392592
15	296808256613018511036336285601497342224	65	328589536630495703136085988472858282244
16	296475057188107749833935123978779877648	66	328964340348163983266751953117489596676
17	296808337742714954082359439610421174544	67	328631059793556779018363666282867263748
18	296641291817219278924738802849574805776	68	328589617760076089303426186459524040964
19	296474976058449992414139598521025945872	69	328964421477744369434092151104155355396
20	296808256613057196662563914152667242768	70	328631140923137165185703864269533022468
21	296475057187488779814291307941457027344	71	328964345418780655171180791515735591684
22	296474976057831022394495782483703095568	72	328631064864173450922792504681113258756
23	296474976057869783578587163337425154320	73	328964426548361041338520989502401350404
24	296475057187450169745927361324090913040	74	328964340348178480931854363546770934532
25	296641291816561698836731040194885841168	75	328631059793571276683466076712148601604
26	296474976058488753598230979374815113488	76	328964344151130069110050838696687504132
27	296808256613095957846655295006456410384	77	328964425280710455277391036683353262852
28	296475057188069139765571177361480872208	78	328964420210108281038064608714388605700
29	296641291817180668856374856232275800336	79	328631139655501076789676321879766272772
30	296641129557981210895466809717539729680	80	328631063596508508867554627934657653508
31	296474976060964633676855783120276025616	81	327614459766465259907200645449039558212
32	296475057190545019844195981106941784336	82	327614459768941139985789420397481506372
33	296641291816600384462958712726525047056	83	327281179214333935737401133562859173444
34	296474976057831097952359508397976187152	84	327281098082234148161930000576236303940
35	296641129558019896521694464656992891152	85	327614459768974989908738616614073480772
36	296475057188107825391798849893120078096	86	327281179214367785660350329779451147844
37	296641291817219354482602528763915006224	87	327614378639317232488943091156319548996
38	296474976058450067972003324435366146320	88	327281098084710028240554804321697216068
39	296808256613057272220427640067007443216	89	327281098082272833788157673107875509828
40	296474976060925948050628128180827058448	90	327614378636880038036581988739516806724
41	297494047933810020972540523993508070160	91	327281179211853219955497871094541268548
42	296474976058488687485100219199766127376	92	327614459766460424203922186726182565444
43	296808256613095891733524534831407424272	93	327614378639355918115170763687958754884
44	296475625099309824456409961520876741392	94	327281098084748713866782476853336421956
45	296475057188069073652440417186431886096	95	327614459768936304282510961674624513604
46	296808337742676277900864732818073182992	96	327281179214329100034122674840002180676
47	296641291817180602743244096057226814224	97	297244119599433449400320668614539593636
48	296475381706313247069345872865913725712	98	297244160164257492406939981622812854180
49	2964749760578697174654564031624432771072	99	297244200729091206820116207266435155876
50	296475057187450103632796601149109035792		

Fig. 12 Good radius-3 candidates to solve the parity problem from prime-sized lattices

until the (T_5, T_6) node is reached and then beginning with the (T_3, T_4) pair, a block of 1s begins to grow (Fig. 11). Let us assume that the entire lattice is perpetually in some stage of this cycle. We make several observations:

- The 1-blocks start off having odd size.
- The 0-blocks start off having even size.
- In order for a block of 1s to regrow, it must have even size at the end of the shrinking process.
- In order for a 0-block to grow, it must have odd size at the end of the growth of 1s.

Since the entire configuration is in this process, it is made up of sub-configurations of two forms: $b_i\ w_i$, a block of 1s followed by a block of 0s when the 1s are growing; or b_i01w_i a block of 1s separated by a 01 from a block of 0s when the block of 1s is shrinking. Consider a block of 0s, w_1 followed by a block of 1s, b_2. If w_1 has odd size, then it must not change sizes again before the growth of 0s restarts. Otherwise when the block of 1s on its left, b_1 grows, it will merge with b_2. This means that (T_9,T_{11}) must be applied on the left before (T_5,T_6) can be applied on the right. Hence b_2 will shrink by 1 before the 1s have finished shrinking on the right. Since we will need to have an even number of 1s at that time, we must now have an odd number of 1s. In other words, a block of 0s of odd size is always followed by a block of 1s of odd size. Now assume that b_1 has even size. In this case it must change sizes before the regrowth of 1s is complete, so (T_5,T_6) must be applied on the right before (T_9,T_{11}) is applied on the left again. Hence w_2 must already have even size. Taken together, we see that for the accordion effect to endure in perpetuity, the CA must have even size. □

Finally, taking these various lemmata together, we have the heart of our convergence proof.

Lemma 19 *From any non-homogeneous configuration, the total number of blocks decreases in finite time.*

Proof From the previous lemmata, we see that only rule pair (T_9, T_{11}) increases the number of blocks and that within a finite number of steps, this increase is resolved. Our contention is that the accordion effect is the only way to maintain block numbers. Consider a (T_9, T_{11}) pair occurring anywhere in our CA. If the leading block of 1s is odd at the end of the execution of the (T_9, T_{10}) pairs, then the number of blocks had decreased by two. If it is even, the 1s will begin to grow right. If the 0-block created by the execution of (T_9, T_{11}) is unchanged from the right, then it has even size and the regrowth of 1s will result in a merger of 1s and a reduction in the number of blocks. Now the only way for the 0-block to have changed on the right is if the 1-block on its right had shrunk due to the application of (T_9, T_{10}) or (T_9, T_{11}). If the reduction in the 1-block was not initiated by the (T_9, T_{11}) pair, then real reduction in the number of blocks has occurred. If it was initiated by (T_9, T_{11}), then block reduction can only be prevented if another (T_9, T_{11}) pair is being executed to its right. Arguing in this way, we see that reduction in total block number can only be avoided if the CA is experiencing the accordion effect which can only occur in even-sized lattices. □

From Theorem 5 and Lemmata 13 and 19, we obtain:

Theorem 6 *Given a CA of odd size, rule* BFO *converges to all 1s if the initial configuration has odd parity and to all 0s if it has even parity.*

5 Empirical evidence against a solution for radius 3

According to (Wolz and de Oliveira 2008), no perfect rule was found for all odd initial configuration (IC) sizes less than 25, but about 250 perfect rules were found for all prime-sized ICs in the range of 11 to 31. This was checked by completely enumerating all possible ICs in the range. These rules were then tested against 2 sequential and independent sets, each one with 400 billion random odd-sized ICs in the range from 37 to 149; as a result, 103 rules survived the first test, and 98 survived the second. Adding another, found later, resulted in a set of 99 very good rules for the parity problem, all with radius 3 (all of them in individual classes of dynamical equivalence, each class having exactly 4 members). Their decimal numbers are listed in Fig. 12 (in Wolfram's lexicographic order), with the most recently added rule showing as the first one in the list. Only four of them had been shown in (Wolz and de Oliveira 2008), all the others are only now being unveiled.

Further to the intensive testing the rules above underwent, as described, it should be remarked that they were originally found in (Wolz and de Oliveira 2008) after very intensive searches in the entire radius 3 space, carried out through the most sophisticated evolutionary algorithm described so far in the literature, for CAs rule spaces. This is strong empirical evidence that a perfect rule might not exist for odd-sized lattices, but that it might for prime sizes, the 99 rules above being strong candidates.

However, by studying their failure behaviour, we were able to construct specific prime-sized ICs for 79 of them, thus gathering evidence that a prime perfect rule might not exist either. A summary of the corresponding data is presented next. For what follows, the rules are referred to by their order index in the table.

First of all, our own analysis with respect to details of their failure showed that rule 1 fails for any IC with size a multiple of 3. And, according to (Wolz and de Oliveira 2008), rules 2 through 99 would fail for some IC of size 25. In keeping with these findings, we verified that this is indeed the case; but, in fact, most rules fail even for size 15. The details are as follows:

- Only the following rules {42, 45, 47, 48, 49, 50, 51, 52, 53, 55, 56, 59, 60} do not fail on all ICs of size 15.
- Out of these 13 rules, only rule 52 fails for some IC of size 21, thus leaving the other 12 unscathed.
- These remaining 12 rules {42, 45, 47, 48, 49, 50, 51, 53, 55, 56, 59, 60} eventually fail for some IC size 25.

The ICs that make the latter 12 rules fail lead them to cyclic regimes, with periods 50 (most often) or 25. Based upon these failing ICs, the prime number sized IC (size 83) given by

$$0^1 1^1 0^8 1^7 0^3 1^5 0^1 1^1 0^4 1^3 0^7 1^9 0^1 1^3 0^1 1^1 0^6 1^5 0^5 1^7 0^1 1^1 0^2$$

was constructed. Here, the exponents indicated the number of consecutive 1s or 0s. This IC made 55 out of the 99 rules fail, thus leaving the following 44 unscathed: {1, 2, 3, 4, 41, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99}.

However, from this set, 24 failed on further specially constructed prime-sized ICs, conceived by direct analysis of the specific periodic patterns that appear in the rules, as they failed for size 25. The conclusion was as follows:

1. Rules {61, 62, 63, 64} failed on an IC of size 83, constructed from observing the two configurations, periods 50 and 25. The IC was:

   $0^1 1^1 0^{10} 1^6 0^1 1^1 0^2 1^4 0^7 1^2 0^1 1^1 0^6 1^8 0^1 1^3 0^9 1^4 0^1 1^1 0^4 1^6 0^3$.

2. {65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80} failed on an (yet another) IC of size 83, constructed from observing two other configurations, also with periods 50 and 25. The failing IC was

   $0^6 1^1 0^3 1^1 0^1 1^6 0^5 1^5 0^7 1^1 0^1 1^3 0^1 1^8 0^7 1^3 0^5 1^1 0^1 1^4 0^3 1^7 0^3$.

3. {1, 97, 98, 99} failed on an IC of size 157, constructed from observing two configurations, periods 225 and 10. This time the failing IC was

   $0^2 1^1 0^{31} 1^{19} 0^1 1^2 0^{13} 1^7 0^{19} 1^{10} 0^1 1^1 0^{23} 1^{25} 0^2$.

In summary, the data above shows that, out of the original 99 rules, 20 of them remain that have not yet failed on any prime-sized IC: {84, 85, 86, 87, 88}, which happen to fail for size 25, and {2, 3, 4, 41, 81, 82, 83, 89, 90, 91, 92, 93, 94, 95, 96}, which do not fail for size 25. Therefore, although the problem has not yet been settled, all in all the data makes a strong empirical case against a solution of the parity problem for radius 3, even considering only prime-sized lattices.

6 Concluding remarks

In this paper, we have established upper and lower bounds on the radius of rules that solve the parity problem by showing that there exists a rule of radius 4 which converges to all 1s if the initial configuration is odd, and to all 0s if it is even and, further, by proving that this problem is unsolvable by rules of radius 2, even with the less strict condition of prime-sized lattices.

The corresponding questions for radius 3 remain open, in spite of the strong empirical evidence we provided against the existence of a solution, even in the simpler case of prime-sized lattices only. We believe that the tools developed in the paper should be helpful to resolve this issue as well.

It is clear by now, how painstaking the task of designing a CA rule can be, let alone the formal proof of its correctness. To some extent, the process reminds us of similar programming efforts on simple, pre-modern computational models, such as Turing machines. And in this sense, we are still indeed at this point in history, when programming CAs.

Since our main motivation for addressing the parity problem is not conscribed to it, one may ask how generalisable our experience herein could be to related problems, including the parity problem for radius 3, as well as other related computational problems for CAs. It is tempting to think of the possibility of implementing a high-level programming approach that would automatically generate the state transitions of a CA rule, given the kinds of notions we have used, such as the growth of blocks of a given size in a given direction, the annihilation of blocks of given kind, etc. Even if this form of programming, so to speak, *by patterns*, does not solve the problem of designing a rule (the target algorithm), at least it would help its high-level conception, and its implementation in terms of the required state transitions.

As a methodological note, it is worth mentioning that it was demanding in practice to resort to computational aids to complement the formal efforts. After all, the details involved in rule design are so many that it is quite easy to overlook some of them. This turned out to be essential in the present case for fine tuning our design in its origin. Such an interplay between formal and computational methods also came into play (apart from all the obvious efforts in the context of radius 3) for devising the most compact representation of the BFO rule, as shown in the paper, for enumerating all cycles of size 7 in the de Bruijn graph of radius 2, and for the evaluation of the radius 2 rules that had retained potential for being perfect solvers of the parity problem, by not violating the constraints derived in the proof, at their various stages of development.

Acknowledgements This work came out of a sabbatical period of P. P.B. de Oliveira at the University of Ottawa, thanks to a grant provided by MackPesquisa – Fundo Mackenzie de Pesquisa, and also benefitted from travel grant 2012/15804-5 provided to him by FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo. This work was also supported in part by the Natural Sciences and Engineering Research Council of Canada (Discovery Grant) and by Dr. Flocchini's University Research Chair.

References

Acerbi L, Dennunzio A, Formenti E (2009) Conservation of some dynamical properties for operations on cellular automata. Theoret Comput Sci 410(38–40):3685–3693

Angluin D (1980) Local and global properties in networks of processors. In: Proceedings of the 28th Symposium on Theoretical Aspects of Computer Science (STACS), p 82–93

Angluin D, Aspnes J, Eisenstat D, Ruppert E (2007) The computational power of population protocols. Distrib Comput 20(4): 279–304

Betel H, Flocchini P (2011) On the relationship between fuzzy and boolean cellular automata. Theoret Comput Sci 412(8-10): 703–713

Boccara N, Cheong K (1993) Automata network epidemic models. In BoccaraN etal.(eds)Cellularautomataandcooperativesystems, vol. 396, Kluwer Academic, Dordrecht, p 29–44.

Cattaneo G, Dennunzio A, Margara L (2004) Solution of some conjectures about topological properties of linear cellular automata. Theoret Comput Sci 325(2):249–271

Chaudhuri P, Chowdhury D, S SN, Chattopadhyay S (1997) Additive cellular automata: Theory and applications, vol. 1. IEEE Computer Society Press, Los Alamitos

Dennunzio A, Di Lena P, Formenti E, Margara L (2009) On the directional dynamics of additive cellular automata. Theoret Comput Sci 410(47–49):4823–4833

Fatès, N (2011) Stochastic cellular automata solve the density classification problem with an arbitrary precision. In: Proceedings of the 28th Symposium on Theoretical Aspects of Computer Science (STACS), p 284–295

Fukś H (1997) Solution of the density classification problem with two cellular automata rules. Phys Rev E 55(3):R2081–R2084

Griffeath D, Moore C (eds) (2003) New constructions in cellular automata, (Santa Fe institute Sstudies on the sciences of complexity). Oxford University Press, New York

Haykin S (2008) Neural networks and learning machines. 3rd edn. Prentice Hall, Englewood Cliffs

Hopcroft J, Motwani R, Ullman J (2006) Introduction to Automata Theory, Languages, and Computation. 3rd edn. Addison-Wesley, Reading

Kurka P (2003) Topological and Symbolic Dynamics. vol 11, 3rd edn. IEEE Computer Society Press, Los Alamitos

Langton CG (1986) Studying artificial life with cellular automata. In: Evolution, Games, and Learning. North Holland

Lee K, Xu H, Chau H (2001) Parity problem with a cellular automaton solution. Phys Rev E 64(2):026,702/1–026,702/4

Lenzen C, Oswald Y, Wattenhofer R (2008) What can be approximated locally?: case study: dominating sets in planar graphs. In: Proc. 20th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), p 46–54

Martins C, de Oliveira P (2009) Improvement of a result on sequencing elementary cellular automata rules for solving the parity problem. Electron Notes Theoret Comput Sci 252:103–119

de Oliveira P, Bortot J, Oliveira G (2006) The best currently known class of dynamically equivalent cellular automata rules for density classification. Neurocomputing 70(1-3):35–43

Peleg D (2000) Distributed computing: a locality-sensitive approach. SIAM Monographs on discrete mathematics and applications

Sipper M (1998) Computing with cellular automata: Three cases for nonuniformity. Physical Review E 57(3):3589–3592

Voorhees B (2009) Additive cellular automata. In: Meyers R (ed.) Encyclopedia of Complexity and Systems Science. Elsevier, New York, p 101, 17

Wolz D, de Oliveira P (2008) Very effective evolutionary techniques for searching cellular automata rule spaces. J Cell Autom 3(4): 289–312