Some Results on Invertible Cellular Automata

A. Clementi

Dept. of Comp. Science
University of Rome "La Sapienza"
00198 - Rome, Italy

P. Mentrasti, P. Pierini

Dept. of Mathematics
University of Rome "La Sapienza"
00185 - Rome, Italy

Abstract

*We address certain questions concerning invertible cellular automata, and we present new results in this area. Specifically, we explicitly construct a cellular automaton in a class (*residual class*) previously known not to be empty only via a nonconstructive existence proof. This class contains cellular automata that are invertible on every finite support but not on an infinite lattice. Moreover, we show a class that contains invertible cellular automata having bounded neighborhood, but whose inverses constitute a class of cellular automata for which there isn't any recursive function bounding all the neighborhood.*

1 Introduction

Computational models satisfying physical laws are the object of several recent studies [8, 3]; of particular interest are *invertible* models [5, 8]. Cellular automata represent one of the best models of parallel computation; the study of invertibility in cellular automata is of great interest in modelling physics.

Several theoretical results concerning invertibility in cellular automata have been presented ([2, 9, 10, 12, 13, 15, 18]), some leading to open questions.

- In [18], the existence of a peculiar class (*residual class*) of cellular automata had been predicted but, until now, no such cellular automata had been exhibited. Here we explicitly construct a cellular automaton in this class, i.e., a cellular automaton that is invertible on every finite support but is not invertible on an infinite support.
- It is known [18] that for the class of all invertible cellular automata, an upper bound to the radius of the inverses cannot be found. We investigate the meaning of this constraint, exhibting a class of invertible cellular automata whose inverse local maps have neighborhood that cannot be bounded by any recursive function.

We construct these cellular automata starting from a space tiling technique introduced by Robinson in [14]. More precisely, using a variant of Robinson's technique, we discuss a particular set of local maps which had first been presented by Kari in [9, 10] and we prove that this set has the above mentioned properties.

2 Infinite cellular automata

2.1 Cellular automata

A *cellular automaton* is a set of identical finite automata (also called *cells*) locally connected to each other in a uniform way.

In this paper, we consider two kinds of cellular automata, depending on their *support*, that is, the grid containing the cells:

- if the cells are located in the infinite d-dimensional lattice (that is, $\mathcal{Z}^d$), we have a *proper* cellular automaton;
- if the support is a d-dimensional toroidal array (*torus*) of period (or *size*) n along all dimensions (that is, $\mathcal{Z}_n^d$), we have a *toroidal* cellular automaton.

The state q of each cell varies according to a uniform, deterministic *local function* defined on the set of neighborhood states. The *neighborhood*, a set N of displacements, specifies the relative positions (with respect to the cell to be updated) of the cells used by the local function. By *radius* of a cellular automaton, we mean the radius of its neighborhood. In this paper, we use the *Moore neighborhood*, consisting of a cell and the eight adjacent cells in a two-dimensional grid.

Hence, a complete description of a cellular automaton (both for the infinite and the finite case) can be given by defining:

- the support space,
- the state set Q of the cells,
- the neighborhood N,
- the local function $f : Q^{|N|} \rightarrow Q$.

The pair $\lambda = \langle$neighborhood, local function$\rangle$ will be called *local map* or *rule*.

The cells change their states in a parallel, synchronous way. The local function determines a global function F acting on the space Σ of all possible configurations.

2.2 Invertibility

A cellular automaton is *invertible* if its global function is bijective. The invertibility of a cellular automaton is an important issue in modelling reversible physical phenomena.

Here we give a brief summary of the main results about cellular automata invertibility.

The invertibility of a cellular automaton is a property of its global function, while the cellular automaton itself is described in terms of a local map; Richardson proved that the bijectivity (i.e., invertibility) of a cellular automaton's global function implies the existence of an *inverse local map.*

Theorem 2.1 [13] *If the global map of a cellular automaton is injective, then it is invertible, and its inverse is the global map of a cellular automaton as well.*

In other words, if the global map is injective then it is also surjective and the inverse global process can be described in local terms. Richardson's proof is not constructive: it does not give any procedure for finding the inverse local map. However, given two cellular automata it is possible to decide if they are one the inverse of the other:

Lemma 2.1 [18] *There is an effective procedure for deciding, for any two locals maps λ and λ' defined on the same set of configurations, whether the corresponding global maps F and F' are the inverses of one another.*

Early investigators conjectured that invertible cellular automata could not be computational universal [1, 4]. Toffoli [15] proved the existence of universal invertible d-dimensional cellular automata when $d > 1$; Morita and Harao [12] proved the existence of computation universal cellular automata in the one-dimensional case.

For many years a major challenge has been deciding whether or not a given cellular automaton is invertible. For the one-dimensional case Amoroso and Patt proved that

Theorem 2.2 [2] *There is an effective procedure for deciding whether or not an arbitrary one-dimensional cellular automaton, given in terms of a local map, is invertible.*

In other words, the class of invertible one-dimensional cellular automata is recursive. Concerning multidimensional cellular automata, the class of invertible cellular automata is recursively enumerable (see [18] for a proof); however recently Kari proved that, for d greater than one, the class of invertible d-dimensional cellular automata is not recursive:

Theorem 2.3 [9, 10] *There is no effective procedure for deciding whether or not an arbitrary two-dimensional cellular automaton, given in terms of local map, is invertible. Thus, in general, the invertibility of a cellular automaton is undecidable.*

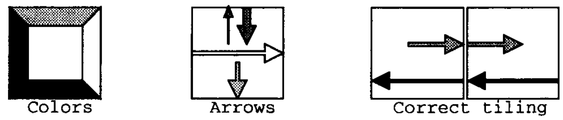

Figure 1: Colors are replaced by *arrows.*

The proof is based on transformation from another undecidable problem — the tiling problem on the infinite two-dimensional lattice [14].

The invertibility of a cellular automaton considered on toroidal finite supports has been proved to be co-NP-complete [6]; the same result of completeness, arise also in the average-case complexity theory [7].

Alongside these theoretical results, there have been technological, architectural and algorithmic developments (see [11, 16, 17, 19]); as a result cellular automata have become a very productive tool for modelling and carrying out parallel computation.

3 A Tiling technique

In the present section we summarize some results and definitions used in [10] to prove Theorem 2.3. In particular, since it is used for proving our results, we briefly describe a finite set of tiles having the following properties: they cover an infinite two-dimensional grid, and they define a path through all the elements of this grid. Moreover we prove that this set of tiles cannot be used to tile a finite toroidal support of any size.

A *tile* is a square with colored edges. Formally, given a finite set C of colors, a *tile set* is a subset $\tau \subset C^4$ and a τ-*tile* is any ordered quadruple t of C^4.

A *tiling* (denoted as τ-tiling) of a fixed grid (*support space*) using the set τ, is a mapping from the sites of the grid to the set of tiles.

By *correct* tiling we mean a tiling such that edges of adjacent tiles have the same color.

Using colors, labels, or numbers to distinguish different kinds of tile edges is just a matter of convention. In what follows we adapt the concept of edge color as done in [10].

We replace each color with one or more *arrows* pointing inwards or outwards from an edge. In other words each edge is tagged by the *heads* and *tails* of different arrows (see Figure 1). Two adjacent edges match correctly if each head meets, in the adjacent tile, the tail of an identical arrow.

Let us further generalize the concept of color by assigning labels to the corners as well. If we call the four *corners* of a tile NE, NW, SW and SE, a *passage* is a pair $\langle a_{in}, a_{out}\rangle$, where both a_{in} and a_{out} (which denote, respectively an *inward direction* and an *outward direction*) belong to the set {NE,NW,SW,SE} (see Figure 2).

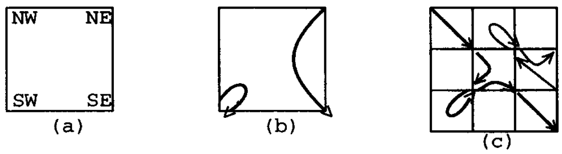

Figure 2: (a) Corners, (b) $\langle SW, SW \rangle$ and $\langle NE, SE \rangle$ passages, (c) Paths.

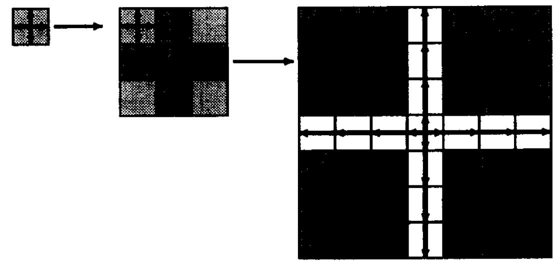

Figure 3: Construction of a correct tiling for a square of dimension 7.

The corner NW is naturally called *opposite* to SE, NE opposite to SW.

With this formulation, the concept of "tile color" has been generalized to one of "arrows and passages" According to these definitions, a tiling is correct if

- each arrow head meets an equal arrow tail.
- for each passage $\langle a_{in}, a_{out} \rangle$ the neighbour tile in the a_{out} direction is "colored" with a passage whose inward direction is the opposite of a_{out}.

If one of these two conditions doesn't arise, we say that a *tiling error* occurs.

We define a *path* as a sequence (possibly cyclic) of consecutive passages $p_1, \ldots, p_i, \ldots$ associated with neighbouring cells, such that the outward direction of p_i is equal to the opposite of the inward direction of p_{i+1} (see Figure 2c).

A complete description of the set τ_K of 160 tiles defined in [10] is given in Appendix A. This set has three very interesting properties expressed by the following lemmas.

Lemma 3.1 [10] *An arbitrarly large square grid can be correctly tiled with the set* τ_K. *From Koenig's infinity lemma, one can then correctly tile the entire plane* $\mathcal{Z}^2$.

The correct tiling of arbitrarly large square grids (*squares* for simplicity) is obtained by a recursive construction that, given the correct tiling of a square of side $2^n - 1$, determines the correct tiling of a square of side $2^{n+1} - 1$. A schematic representation of this construction is shown in Figure 3.

Lemma 3.2 [10] *In every tiling of the plane from the set* τ_K, *only two types of path may arise:*

- *Either the path has a tile for which there is a tiling error in its neighborhood,*
- *or the path visits all the tiles of an arbitrary large square.*

The property of tiling arbitrarly large squares is trivial to achieve; it is the property expressed in 3.2 that makes this set of tiles really interesting. In other words, whatever square we tile with the set τ_K, either this tiling induces a continuous path trough all the tiles, or the fact that the path will not cover the entire square is locally recognizable through a tiling error. The path induced by the passages has the shape of the Hilbert curve (see Figure 11). We remark that the set of tiles τ_K is independent on the size of the square that ones wants to tile.

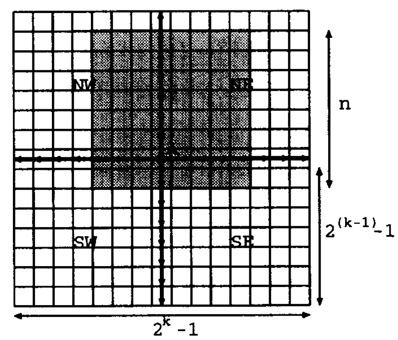

Figure 4: A correct tiling for a $2^{k-1} \times 2^{k-1}$ square including the planar representation of a torus (shaded).

The set τ_K resembles that presented by Robinson [14] as an example of tiles that permit only non-periodic tiling: neither Robinson's tiles nor the set τ_K can be used to tile correctly a torus. In fact, refering to Figure 3, a tiling error must occur when the space is wrapped around by joining together opposite edges since some arrow heads meet other heads instead of tails. In the following lemma we formalize this result, using the construction of $(2^n - 1) \times (2^n - 1)$ squares with the set τ_K given in Appendix A.

Lemma 3.3 *The set of tiles* τ_K *does not permit correct tiling for tori of any size.*

Proof. (By contradiction) Suppose that there is a correct tiling for a torus of size n. This tiling is equivalent to a correct *periodic* tiling of period n for the infinite lattice. Let t be a single cross of the torus (which always exists, since in every correct tiling each 2×2

square must have a single cross). From Lemma A.1, t is in a XY-(2^m-1)-square constructed as explained in Appendix A; since the entire plane is tiled correctly, m is as large as we want. Thus we can find the smaller k such that there exists a XY-(2^k-1)-square including our planar representation of the torus (see Figure 4). Since $n > 2^{k-1}-1$, the torus must include the central cross of the square (A in figure) as well as part of the arms leaving from it; thus, when the space is wrapped around, a tiling error is encountered. □

4 A cellular automaton in the residual class

Here we exhibit a first constructive example of cellular automaton in the residual class.

As described in [18] a cellular automaton on a infinite support can be:

- injective and surjective (*invertible*);
- not injective and not surjective (*nonsurjective*);
- surjective but not injective (*properly surjective*).

It is impossible for a cellular automaton to be injective but not surjective [13]. When the local map is considered on a finite support, the three classes listed above have the following behaviours (see Figure 5):

- invertible cellular automata remain invertible;
- nonsurjective cellular automata remain nonsurjective;
- properly surjective cellular automata can yield either invertible or nonsurjective finite cellular automata.

The class of cellular automata that are invertible on every finite support but noninvertible on an infinite support is called *residual class.* The residual class cannot be empty [18] but no examples of cellular automaton in this class had been shown until now.

4.1 A specific noninvertible cellular automaton

Using the set of tiles defined in §3, we construct a cellular automaton in the residual class.

We tag each passage of the tiles τ_K with a binary digit; let us consider the cellular automaton having as states these modified tiles and performing, at each step, the XOR between bits of adjacent cells along the path induced by the passages. Formally:

Definition 4.1 *The two-dimensional cellular automaton A_K is defined by:*
States: Each state consists of two components

- *A tile $t \in \tau^K$ (tile component)*
- *One bit (0 or 1) for each passage of t (bit components)*

Neighborhood: Moore.
Local function: The local function does not change the tile component.

- *If there is no tiling error in the neighborhood of a cell, then the function performs the XOR between the bit of each passage of the cell and the bit of the corresponding next passage in the path (see Figure 6).*
- *If there is a tiling error, the bits remain unchanged.*

A_K is noninvertible and has an interesting property:

Lemma 4.1 [10]
The cellular automaton A_K

1) *is noninvertible on the infinite support $\mathcal{Z}^2$;*
2) *for each pair of distinct configurations c_1 and c_2 having the same image under the update function ($F(c_1) = F(c_2)$), the tile component of both c_1 and c_2 constitute a correct tiling of the plane.*[1]

Proof. From Lemma 3.1 one can choose two configurations c_1 and c_2 with the tile components constituting a correct tiling of the plane; thus the local function performes the XOR between each bit and the next bit in the space covering path. If the bit components are set to 0 in c_0 and to 1 in c_1, the images of both these two configurations under the update function coincide with c_0 itself; hence A_K is not invertible.

If $c_1 \neq c_2$ and $F(c_1) = F(c_2)$, there must be a cell $\vec{x}$ whose states q_1 and q_2, respectively in c_1 and c_2, are different. The local function f must change at least one of q_1 and q_2; however, the tile components of these states must be the same, since they are not changed by f. According to Definition 4.1, there cannot be a tiling error in the neighborhood of $\vec{x}$ (otherwise the bits would remain unchanged), f computes the XOR between bits of adjacents passages, thus also the bits in the cell that follows $\vec{x}$ along the path, must have different states. By iterating such a process, since a correct path visits every cell (Lemma 3.2), the absert is proved. □

4.2 From infinite to finite support

Here we prove that the noninvertible cellular automaton A_K defined above becomes invertible when considered on finite toroidal supports. Thus A_K is a constructive example of a cellular automaton in the *residual class.*

The noninvertibility of A_K on an infinite support comes from the noninjectivity of the XOR operator; if the path, defined by the passages, is an infinite one, we can't know the predecessor of a configuration (see Figure 6). Nonetheless if we could know the predecessor of at least one cell then we could also determine the predecessor of the cell that follows in the path, and iteratively determine all the other cell predecessors; thus the cellular automaton becomes invertible.

The invertibility of A_K on a finite support follows from the fact that, on a torus, the set of tiles τ_K always

[1] This property has been called *almost injectivity* in [10].

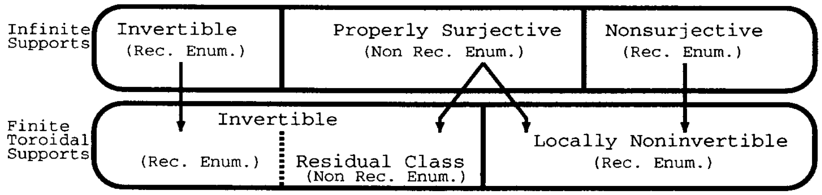

Figure 5: Invertibility for CA.

leads to a tiling error (Lemma 3.3): from a given configuration, we can reconstruct the previous one starting from a cell (there is at least one) where the local function doesn't change the state since the tiling is not correct. Formally:

Theorem 4.1 *The two-dimensional cellular automaton A_K is in the residual class.*

Proof. We still call A_K the toroidal cellular automaton obtained from the local map of A_K defined in Definition 4.1. From Lemma 3.3, for any n, for any configuration of A_K on the toroidal support $\mathcal{Z}_n{}^2$ there must be a tiling error due to the tile component of the states. From the second property of Lemma 4.1, given two configuration c_1 and c_2 such that $F(c_1) = F(c_2)$, it must be that $c_1 = c_2$, otherwise the tiling would be correct. Thus A_K is injective on any toroidal support. □

The proper surjectivity of A_K (see Figure 5) follows from the the fact that A_K is noninvertible on infinite support and becomes invertible on finite supports. Indipendently of this result, it can be proved that:

Theorem 4.2 *The two-dimensional cellular automaton A_K is properly surjective.*

Proof. By lemma 4.1, A_K is not injective on the infinite support $\mathcal{Z}^2$. However, A_K is surjective; indeed, let c be a configuration of A_K and n be a positive integer; on any path $p = \langle p_1, \ldots, p_n \rangle$ of length n (passing through the cells $i = 1, \ldots, n$) we denote as $\langle x_1^t, \ldots, x_n^t \rangle$ the bits associated with the passages in p at time t; then, all possible situations can be easly reduced to the following two cases:

1. The cells $i = 1, \ldots, n-1$ have no tiling errors and cell n has a tiling error; then, in the predecessor of c, x_n must have the same value of that in c (see Definition 4.1). Moreover, for any $t > 0$ and $i = 1, \ldots, n-1$, in the XOR function

$$x_i^t = x_i^{t-1} \oplus x_{i+1}^{t-1}$$

   the value x_i^{t-1} is uniquely determined by x_i^t and x_{i+1}^{t-1}. From these facts, by a backward-iterative procedure, it is easy to correctly define all the bit values of p in the predecessor of c.

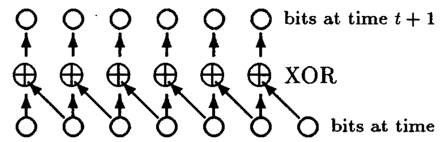

Figure 6: Bits along a path.

2. None of the cells $i = 1, \ldots, n$ have tiling errors; then, by Theorem 3.2, the path p continues into a cell $n+1$ having a bit x_{n+1} associated with the passage p_{n+1}. Then, for any value of x_{n+1}, it is not hard to determine all bit values of p in the predecessor of c by making use of the same procedure mentioned in the first case.

The proof is complete by observing that the local function f^K (and thence the global one) is different from the identity only on the bits of passages. □

5 Unbounded neighborhood

Here we define a class of invertible cellular automata for which the neighborhood of the inverse is not bounded by any recursive function (*nonreciprocal* property).

Toffoli and Margolus observed that:

Theorem 5.1 [18] *There cannot exist a recursive function $f(\lambda)$ defined on the local maps of the two-dimensional cellular automata and bounding the neighborhoods of all the inverse cellular automata.*

Proof. If this function existed, given a local map λ we could sequentially generate all the local maps with a radius bounded by $f(\lambda)$, and by Lemma 2.1 we could check if one of these maps is the inverse of λ. On reaching $f(\lambda)$, either we found an inverse or we can conclude that λ is not invertible. But this contradicts the undecidability of cellular automata invertibility (Theorem 2.3). □

This result can be easily extended to any class of cellular automata for which is undecidable whether a cellular automaton is invertible. Thus, the class of cellular automata used in [10] to prove Theorem 2.3 has the nonreciprocal property.

Showing a class that is "small" and still has the nonreciprocal property is useful in understanding the nature of the theoretical result of Theorem 5.1. We give a different proof, without using the result in Theorem 2.3, of the non reciprocal property for the class introduced in [10], emphasizing the reason of this property.

We modify the cellular automaton A_K (see Definition 4.1), redefining the tile components in such a way that a tiling error is always encountered. In this way, we force the local function to be the identity for at least one cell.

Definition 5.1 *Given a set of tiles τ_{error} that does not tile correctly the two-dimensional space, we consider the cellular automaton A_{INV} defined by:*
States: Each state is a pair $\langle t, q \rangle$ where $t \in \tau_{error}$ and q is an element of the state set defined in 4.1.
Neighborhood: Moore.
Local function: The local function operates as the function of A_K except that, when checking for tiling errors, it also consider the state component given by the set τ_{error}.

It can be easily proved, as done in [10] for different goals, that

Lemma 5.1 *The cellular automaton A_{INV} is invertible.*

Proof. If we suppose, by contradiction, that A_{INV} is noninvertible, by the same reasoning used for proving the second statement of Lemma 4.1 we can prove that τ_{error} admits a correct tiling of the plane. But this is false, and thus the lemma is proved. □

Thus, A_{INV} is invertible. However, in order to explicitly obtain the predecessor of a cell $\vec{x}$ we must follow the path originating from $\vec{x}$ until a tiling error is encountered. When we find a tiling error in a cell $\vec{y}$, since the local function in $\vec{y}$ is the identity, we know the predecessor of $\vec{y}$; by going backwards along the path, we can find the predecessor of $\vec{x}$. This is the only way to construct the inverse; it follows that the radius of the neighborhood must be large enough to recognize the nearest tiling error.

For each set of tiles that does not admit a correct tiling of the plane, we can define a cellular automaton in a way similar to what used for A_{INV}; thus we obtain a class of cellular automata, denoted by INV.

Since we can't predict where a tiling error eventually occurs, the radii of the inverses of the cellular automata in INV cannot be bounded by any recursive function.

The following formalizes this result.

Theorem 5.2 *The radius ($|N^{-1}|$) of the inverses of the cellular automata in INV cannot be bounded by any recursive function.*

Proof. Let us denote by f^{-1} the local function of the inverse of a cellular automaton in INV.

The neighborhood N^{-1} must contain at least one tiling error; that is, every cell, in the backward evolution, must have at least one tiling error in its neighborhood in order to compute its next state. Indeed, let

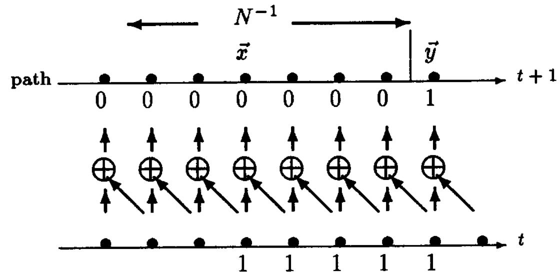

Figure 7: A tiling error in the neighborhood of the inverse.

us assume by contradiction that N^{-1} does not always contain a tiling error (with respect to the τ_K component or the τ_{error} component). Thus, there exists a configuration in which a cell $\vec{x}$ has a correct tiling (for both τ_K and τ_{error} components) in its neighborhood. For simplicity we call N^{-1} the neighborhood of $\vec{x}$. By Lemma 3.2, the path passing on $\vec{x}$ must leave this neighborhood and reach a cell $\vec{y}$ which is adjacent to N^{-1}. Without loss of generality, let us suppose that there is a tiling error (there must always exist one) in the cell $\vec{y}$ (more precisely, in the part of its neighborhood which is outside N^{-1}); thus the bits in $\vec{y}$ are not changed by the local function. Let us consider the configuration in which all bits in N^{-1} are 0. The predecessor of $\vec{x}$ in the forward evolution of the cellular automaton is $f^{-1}(N^{-1})$; let us suppose that this value is 0 and consider the configuration in which the bit in $\vec{y}$ is 1 (see Figure 7). Under these conditions, all the bits that preceed $\vec{y}$ in the path that goes from $\vec{y}$ to $\vec{x}$ must be 1 (see Figure 7), also bit in $\vec{x}$ must be 1, but this is a contradiction.

Similar arguments can be applied if we suppose $f^{-1}(N^{-1}) = 1$.

Finally, since the set of tiles τ_K admits a correct tiling of the plane and the tiling problem is undecidable on $\mathcal{Z}^2$, the thesis is proved. □

6 Conclusions

The existence of families of invertible toroidal cellular automaton having an inverse local map with large and complex interactions could determine a set of one-way functions having practical applications in cryptography. Indeed, knowledge of the direct local map (the cryptor) gives no sufficient informations (to the cryptoanalist) on the inverse local map (the decryptor) (see also [7]). In terms of dynamical system theory, the results shown in this paper imply the existence of reversible dynamical systems having local and simple interactions but whose inverses have almost-global interactions.

Acknowledgments We thank Tommaso Toffoli for many advices, comments and helpful suggestions on this paper.

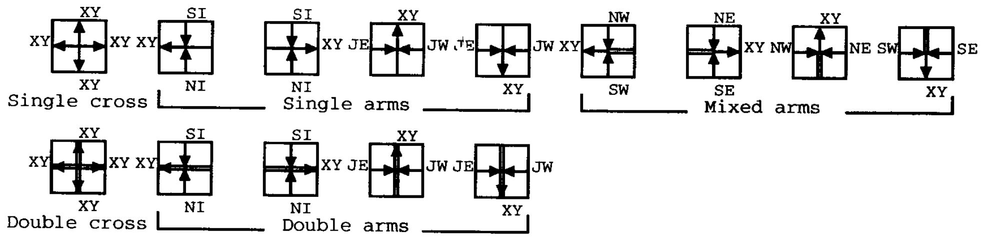

Figure 10: Labelled arrows.

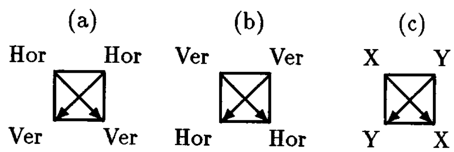

Figure 8: The diagonal arrows on (a) horizontal arms, (b) vertical arms, (c) crosses ($X, Y \in$ {Hor,Ver}).

A Kari's tiling technique

Here we present in detail the set τ_K of tiles defined in [10] and used in our results.

<u>Arrows</u> Different kind of arrows (see §3) are distinguished by drawing them in different ways and by labelling them with different tags. Thus we have the set of *labelled arrows* shown in Figure 10. This set permits the recursive construction of correct tilings for arbitrarly large squares (see Figure 9). Each of these tiles also has two *diagonal arrows* as shown in Figure 8. The diagonal arrows force the horizontal and vertical arms to alternate on each diagonal row of tiles.

<u>Passages</u> If we denote a tile by the label of its arrow; we have that:

- Double crosses must have the passages {(NW,NE),(NE,SE), (SE,SW),(SW,NW)};
- Single crosses must have one of the following six passage sets: {(XY,XY)} such that XY ∈ {NE,NW,SE,SW}; {(NW,SE),(SE,NW)}; {(NE,SW), (SW,NE)}.
- no passage for any type of arms.

In Figure 11 the path induced by a correct tiling on a square of dimension 7 is shown.

In a $(2^n - 1) \times (2^n - 1)$ square correctly tiled by the recursive costruction sketched in Figure 3, the tile in the middle is always a *double cross* (see Figure 3 and

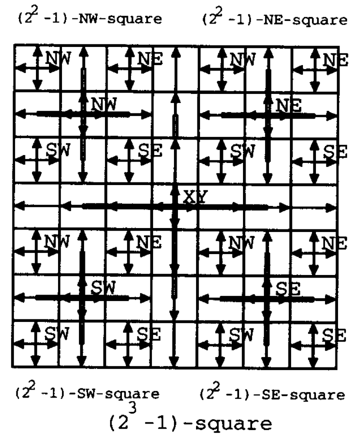

Figure 9: Correct tiling: Arms are drawn without "secondary" arrows and without labels.

Figure 9); we denote this square by $(2^n - 1)$-XY-square where XY is the label of the central double cross.

The set τ_K is such that, given a $(2^n - 1)$-square tiled correctly, the tiles immediately outside the square are the ones that allow the correct tiling to be extended to a $(2^{n+1} - 1)$-square.

In a correct tiling every 2×2 block of tiles contains a single cross. Thus, if we consider each *final* tile consisting of one of all possible correct 2×2 blocks of elementary tiles, we obtain a path visiting the entire plane (Lemma 3.1); the following technical lemma proves this result.

Lemma A.1 [10] *Let t be a single cross on the plane. Consider the path that goes via t. Suppose that there are no tiling errors in any of the 4^n tiles that precede and the 4^n tiles that follow t on this path. Then t belongs to a XY-$(2^n - 1)$-square (XY can be as usual, NE, NW, SE or SW) whose single crosses are all visited by the path.*

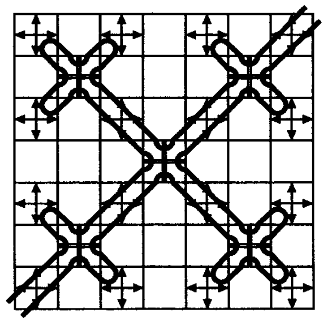

Figure 11: A correct path: the path is drawn without direction.

References

[1] V. Aladyev, "Computability in homogeneous structures", *Izv. Akad. Nauk. Estonian SSR*, Fiz-Mat., 21, 80–83, 1972.

[2] S. Amoroso, Y. N. Patt, "Decision procedures for surjectivity and injectivity of parallel maps for tesselation structures", *J. Comp. Syst. Sci*, 6, 448–464, 1972.

[3] E. Bernstein, U. Vazirani, "Quantum Complexity Theory", *Proc. of the* 25^{th} *ACM-STOC*, 11–20, 1993.

[4] A. Burks, *Essays on Cellular Automata*, University of Illinois Press, 1970.

[5] C. H. Bennett, P. Gacs, M. Li, P. Vitanyi, W. Zurek, "Thermodynamics of Computation and Information Distance", *Proc. of the* 25^{th} *ACM-STOC*, 21–30, 1993.

[6] A. Clementi, P. Pierini, "Computational complexity of the finite cellular automata reversibility problem", *Proceedings of the* 4^{th} *Italian Conference on Theoretical Computer Science*, L'Aquila, Italy, 165–178, 1992.

[7] A. Clementi, P. Pierini, R. Impagliazzo, "On the average-case complexity of the reversibility problem for finite cellular automata", *Proc. of PhysComp '94*, Dallas, TX, 1994.

[8] G. Jacopini, G. Sontacchi, "Reversible Parallel Computation: an evolving space model", *Theoret. Comput. Sci.*, 73,1–46, 1990.

[9] J. Kari, "Reversibility of 2D cellular automata is undecidable", *Physica D 45*, 379–385, 1990.

[10] J. Kari, "Reversibility and surjectivity problems of cellular automata", *J. Comp. Syst. Sci.*, 48, 149–182, 1994.

[11] N. Margolus, "Cellular Automata Machines: a new kind of computer", to appear in the *Proceedings of the* 6^{th} *International Conference on Computational Physics*, Bejing, China, 1993.

[12] K. Morita, M. Harao, "Computation Universality of One-Dim. reversible CA", *Trans. IEICE E*, 72, 758–762, 1989.

[13] D. Richardson, "Tesselation with local transformations", *J. Comp. Syst. Sci.*, 6, 373–388, 1972.

[14] R. M. Robinson, "Undecidability and non periodicity for tilings of the plane", *Inven. Math. 12*, 177–209, 1971.

[15] T. Toffoli, "Computation and Construction Universality of Reversible Cellular Automata", *J. Comp. Syst. Sci.*, 15, 213–231, 1977.

[16] T. Toffoli, "A high performance cellular automata machine", *Physica D*, 10, 195–204,1984.

[17] T. Toffoli, N. Margolus, *cellular automata Machines - A New Environment for Modeling*, MIT Press, Cambridge, MA, 1987.

[18] T. Toffoli, N. Margolus, "Invertible cellular automata: a review", *Physica D*, 45, 229–253,1990.

[19] T. Toffoli, N. Margolus, "Programmable Matter: Concepts and realization", *Physica D*, 47, 263–272, 1991.