A 6-state Universal Semi-totalistic Cellular Automaton on Kite and Dart Penrose Tilings

Katsunobu Imai, Takahiro Hatsuda, Victor Poupet, Kota Sato

▶ **To cite this version:**

Katsunobu Imai, Takahiro Hatsuda, Victor Poupet, Kota Sato. A 6-state Universal Semi-totalistic Cellular Automaton on Kite and Dart Penrose Tilings. Fundamenta Informaticae, 2013, Cellular Automata and Models of Computation, 126 (2-3), pp.247-261. 10.3233/FI-2013-880 . lirmm-01476788

HAL Id: lirmm-01476788

https://hal-lirmm.ccsd.cnrs.fr/lirmm-01476788

Submitted on 25 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

A 6-state Universal Semi-totalistic Cellular Automaton on Kite and Dart Penrose Tilings

Katsunobu Imai
Graduate School of Engineering, Hiroshima University
imai@iec.hiroshima-u.ac.jp

Takahiro Hatsuda
Graduate School of Engineering, Hiroshima University
thatsuda@iec.hiroshima-u.ac.jp

Victor Poupet
LIRMM, Université Montpellier 2, CNRS
victor.poupet@lirmm.fr

Kota Sato
Graduate School of Engineering, Hiroshima University
ksatou@iec.hiroshima-u.ac.jp

In this paper we investigate certain properties of semi-totalistic cellular automata (CA) on the well known quasi-periodic *kite and dart* two dimensional tiling of the plane presented by Roger Penrose.

We show that, despite the irregularity of the underlying grid, it is possible to devise a semi-totalistic CA capable of simulating any boolean circuit and any Turing machine on this aperiodic tiling.

1 Introduction

Since the discovery of quasi-periodic tilings and the related quasi-cristals, much research work has been done to study the behavior of cellular automata working on the resulting aperiodic structure.

In 2002, McClure designed a 3-states stochastic cellular automaton whose behavior converges to a stable 3-coloring of Penrose tilings [11]. In 2005, Chidyagwai and Reiter showed that the broken symmetry of quasi-periodic tilings, while still retaining a highly organized structure, could be used to simulate the complex growth of snow crystals [1] by studying real-valued CA on such lattices. They could produce global n-fold symmetry models where regular hexagonal grids could only produce 6-fold symmetry models.

Reiter also investigated the behaviors of cyclic CA rules on several quasi-periodic tilings. Cyclic CA are special CA that can be regarded as a model of excitable media [2, 6]. Once a cell enters a specific state, it must go through a fixed cycle of other states before being active again, thus implementing a *refractory period* mechanism. Reiter found that cyclic CA on quasi-periodic tilings, despite having a very simple local behavior, can take advantage of the lattice geometry to exhibit complex and smooth spiral patterns [17]. A variety of Cyclic CA rules are illustrated and commented on Wojtowicz's online CA rules lexicon [19].

Variants of John H. Conway's *Game of Life* [3] have also been studied on Penrose tilings. In particular, Owens and Stepney found many stable and periodic patterns, similar to the ones known in the usual euclidian *Game of Life* [15, 16]. Despite this, no equivalent to the *glider* structures of the euclidian case are known on quasi-periodic variants of the *Game of Life*. Indeed, because of the non-periodic nature of the underlying grid, the very definition of gliders is unclear.

Some other works on CA working on aperiodic structures have also been described in books or web sites [5, 18, 20].

In this paper, we investigate some semi-totalistic CA on Penrose's *kite and dart* aperiodic tiling of the plane. We show in particular that there exists a 6-state semi-totalistic CA capable of simulating any

Submitted to:
SOS 2007

© K. Imai, T. Hatsuda, V. Poupet & K. Sato
This work is licensed under the
Creative Commons Attribution License.

boolean circuit based on our previous 8-state version [9]. The usual difficulty of synchronizing signals, made even more problematic in an aperiodic setting, is avoided by embedding universal asynchronous logical elements. This construction is similar to that employed to construct a universal CA on an hyperbolic space [8]. Once we are able to simulate logical circuits, we use Morita's construction [13] to simulate a Turing machine with simple finite memory logical elements.

2 Semi-totalistic Cellular Automata on Penrose Tilings

2.1 Penrose's *Kite and Dart* Tiling

This famous tiling of the plane was discovered by Roger Penrose as a simplification of his previous 6-tiles aperiodic tilings. It is generated by two quadrilateral proto-tiles : a convex one called *kite* and a non-convex one called *dart*. To avoid periodic arrangements, tiling constraints are added on the tiles [4]. Figures 1 and 2 illustrate the tiles and their possible combinations.

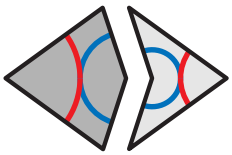

Figure 1: Roger Penrose's *Kite and Dart* tiles.

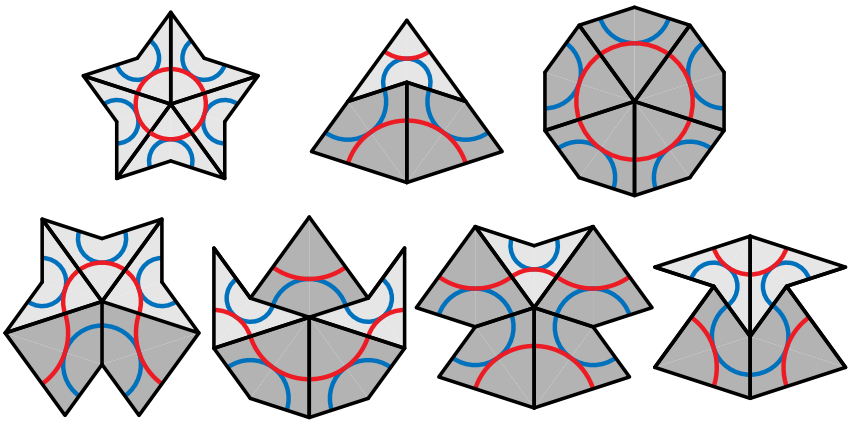

Figure 2: Combination rules of the *Kite and Dart* tiles.

It is known that there exist infinitely many different possible tilings of the plane with the *kite* and *dart* tiles. They are all aperiodic but quasi-periodic, that is to say that there exists a constant $k \in \mathbb{R}$ such that for any $r \in \mathbb{R}$, any finite pattern of size $r \times r$ that appears somewhere on the tiling appears in every square window of size $kr \times kr$.

Much more is known about this tile set and the resulting tilings of the plane but these are the only properties that we will need for this article.

2.2 The Moore Neighborhood on a Penrose Tiling

In order to define a CA on a Penrose tiling, we need to define a neighboring relation between the different tiles. To do so, we extend the usual Moore neighborhood (8 closest neighbors on the regular square grid) by considering that two Penrose tiles are neighbors if they share at least one vertex.

Because all tiles are not identical and not periodically arranged, different tiles might have different neighborhood. In the case of the *kite and dart* tiling, there are 8 different possible neighborhoods [15], illustrated by Figure 3.

It is important to notice that not only these neighborhoods have different shapes, they also have different sizes ranging from a minimum of 9 cells to a maximum of 11. Although this would normally be a problem when defining a cellular automaton (because it would not be possible to define a unified rule for all cells), we will see that the special class of semi-totalistic CA is especially well suited for non-homogeneous neighborhoods.

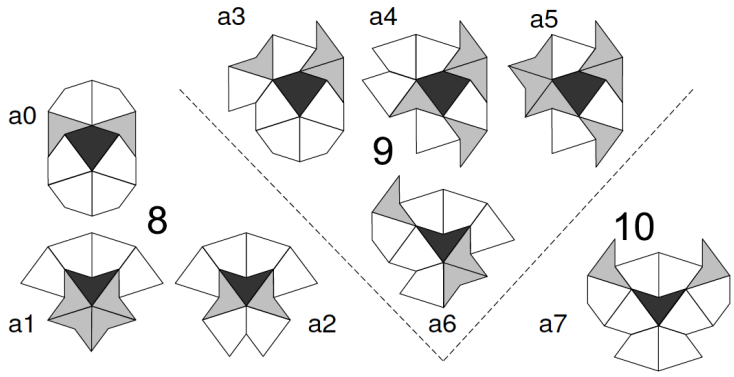

Figure 3: The generalized Moore neighborhoods on a *kite* and *dart* Penrose tiling [15]

2.3 Semi-totalistic Cellular Automata

Because we want to work on aperiodic tilings of the plane, the usual definition of a cellular automaton must be changed in order to be compatible with the fact that different cells might have different neighborhoods. The subclass of semi-totalistic cellular automata is particularly well suited for this as the position of the different neighbors is irrelevant, only the number of neighbors in a given state has an effect on the evolution of a cell.

Definition 1 (Semi-totalistic Cellular Automaton) *A* semi-totalistic cellular automaton *is a pair* $\mathscr{A} = (N, \delta)$ *where :*

- $N \in \mathbb{N}$ *is the number of states of the automaton. The set of states of the automaton is therefore* $\mathscr{Q}_{\mathscr{A}} = \{0, 1, \ldots, N-1\}$ *;*
- $\delta : \mathscr{Q}_{\mathscr{A}} \times \mathbb{N}^N \to \mathscr{Q}_{\mathscr{A}}$ *is the totalistic local transition function of the automaton.*

Definition 2 (Configuration) *A* configuration *of the automaton* $\mathscr{A} = (N, \delta)$ *over a tiling of the plane by Penrose's tiles is a mapping* $\mathfrak{C}$ *that associates a state of* $\mathscr{Q}_{\mathscr{A}}$ *to each tile.*

In accordance with usual cellular automaton terminology, in this context, the tiles of the Penrose tiling will be denoted as *cells*.

Definition 3 (Global Evolution) *Given a configuration $\mathfrak{C}$, the image $\mathfrak{C}'$ of $\mathfrak{C}$ by the automaton $\mathscr{A} = (N, \delta)$ is such that for each cell c, the state $\mathfrak{C}'(c)$ is defined by*

$$\mathfrak{C}'(c) = \delta(q_c, n_0, n_1, \ldots, n_{N-1})$$

where q_c is the current state of c and for every $i \in [\![0, N-1]\!]$, n_i is the number of neighbors of c in state i.

We will denote by $\mathscr{A}(\mathfrak{C})$ the image of a configuration $\mathfrak{C}$ by such a process. The global action of the automaton can be iterated over each new configuration, defining in such a way the infinite evolution of the initial configuration.

In other words, at each time every cell of the automaton updates its state according to the number of neighbors it has in each state.

Remark: In order to be as general as possible, and to be able to define a CA independently of the lattice on which we will consider it, we have defined the local transition function as a function from and infinite set $\mathscr{Q}_{\mathscr{A}} \times \mathbb{N}^N$. However, once the geometric structure of the cells has been decided (in our case the Penrose tiling), it is sufficient to define the local transition function over the tuples that can actually appear. In particular, since we know that in the *kite and dart* tiling all cells have between 9 and 11 neighbors (themselves included), only the input tuples of the form $(q, n_0, n_1, \ldots, n_{N-1})$ where $9 \leq \sum_i n_i \leq 11$ should be considered.

2.4 Previous Studies

Owens and Stepney studied the *Game of Life* on Penrose tilings [15, 16].

Even though the average size of the neighborhoods is larger than that of the standard Moore neighborhood in the regular case, the same parameters for the rule (a cell survives with 2 or 3 live neighbors and a new cell is born when surrounded by 3 live neighbors) exhibits behaviors very similar to the classical *Game of Life* and many similar stable and periodic patterns were found.

However, no *glider* or similarly propagating pattern has been found so far, possibly because of the strong irregularity of the encountered geometry as the signal moves. *Gliders* being the heart of the computational and intrinsic universality proofs of the *Game of Life* in the classical case, this approach cannot be used in the quasi-periodic case.

Another noteworthy difference between the Penrose case and the classical case is that long string-like (closed or not) connected alive cells can be stable on a Penrose tiling. Such patterns, called *rings* and *snakes* have been studied in [16].

3 A Universal CA on *Kite and Dart* Penrose Tilings

In this section we present the construction of a semi-totalistic CA working on Penrose's *kite and dart* tilings that is logically universal in the sense that it can simulate the behavior of any boolean circuit.

3.1 Universal Reversible Logical Elements

In order to construct a universal CA, we will use very simple reversible logical elements with 1-bit memory.

The usual difficulty when embedding circuits in a larger structure (such as a cellular automaton) is to synchronize the different signals. In the case of an aperiodic geometry, the problem is even harder

as construction patterns cannot be precisely shifted in a given direction. To solve this problem we use asynchronous elements that can simulate any circuit while only having one signal at a time.

Figure 4 shows every 2-symbol reversible logical elements with 1-bit memory. There are only eight types of such elements. The left part of each illustration represents the element in state 0, the right part being state 1. Each element only accepts one signal from one of two possible inputs (left side) at a time. Lines (full and dotted) represent the output through which the signal will exit the element.

When a signal traverses the element through a full line, the state of the element is changed. If the signal traverses through a dotted line its state is unchanged.

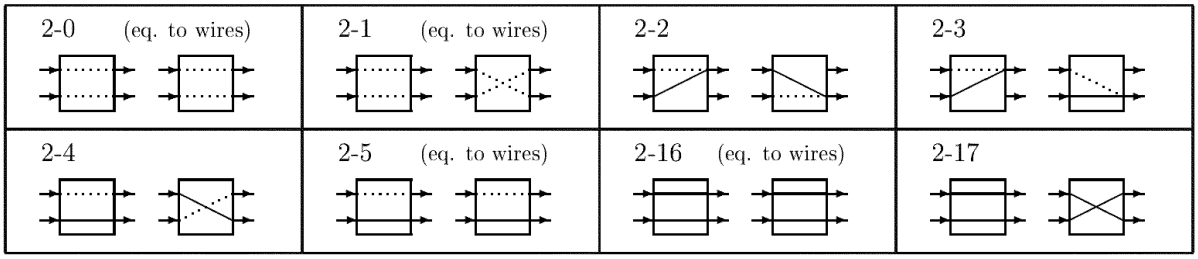

Figure 4: 2-symbol reversible logical elements with 1-bit memory [12].

It has been shown that any reversible Turing machine can be constructed by 4-symbol reversible logical elements (*rotary elements*) [13]. Its computing process is executed by a signal, thus there is no need to synchronize signals. The rotary element can be realized by any combination of two elements amongst 2-3, 2-4, and 2-17 [10, 14]. So the combination of two elements are universal.

In our construction we will use elements 2-3 and 2-17.

3.2 General Overview of the Automaton

The automaton has 6 states :

- State 0 is a *quiescent* state, meaning that a cell in state 0 surrounded by other cells in the same state will not change. This state is therefore used as a "background" for the whole construction and a "filler" in the local transition rule to account for the different sizes of neighborhoods (the number of other states in the neighborhood is significant but the exact number of 0 states is not).
- States 1, 2 and 3 are used to transmit the signal through wire-like structures. Connected cells in state 1 will act as a wire along which the signal moves. The signal itself is made of two states to orient it. State 3 is the head of the signal and state 2 is the refractory state that remains after a cell was in state 3 in order to avoid the signal going back through the wire in the opposite direction.
- Finally, states 4 and 5 are used internally in the logical elements to control the behavior and store the inner state of each element.

Any boolean circuit can be simulated by an asynchronous 1-signal circuit containing only copies of the elements 2-3 and 2-17, connected by wires [10, 14]. Because the circuit is two-dimensional, wires must also be able to cross.

It is therefore enough to show that our automaton can accurately simulate the behavior of the logical elements and that these elements can be connected by wires.

3.3 Simple Wires

Wires can be easily represented by a continuous line of neighboring cells in state 1 surrounded by cells in state 0. It is important that each cell of the wire is connected to exactly two other cells of the same wire. This constraint is not very restrictive though, and it is easy to draw a wire going from any point of the tiling to any other. Simple wires cannot cross, and we will need a specific construction to allow crossings.

A signal on the wire is represented by two consecutive cells in states 3 (front) and 2 (back).

The transition rules needed to describe the behaviors on the wires are quite simple :

- a cell in state 1 that has a neighbor in state 3 (head of the signal) changes to state 3 ;
- a cell that is in state 3 changes to state 2 (refractory state) ;
- a cell in state 2 returns to the simple wire state 1 ;
- finally, cells in state 1 remain in that state in the absence of a neighboring signal.

Figure 5 illustrates a wire on which there is a signal moving towards the right.

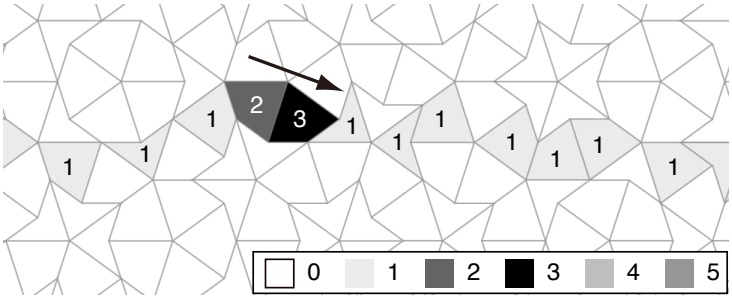

Figure 5: A signal on a wire and their state-color mapping table.

3.4 Logical Elements

The simulation of the elements 2-3 and 2-17 can be done by adding new states to the automaton. It would be very tedious to devise a construction that could behave as a specific logical gate in every possible neighborhood, but we only need to have a pattern that works in a specific geometric neighborhood because the tilings produced by the *kite and dart* tiles are all quasi-periodic.

This means that if we can position the widget at a given position because the surrounding tiles correspond to what we need, there are such appropriate positions in every sufficiently large surface of the tiling.

For instance, Figures 6, 7, 8, 9, and 10 illustrate a widget that behaves as element 2-3. This widget must be located on a neighborhood of type $a7$ (according to the classification of Figure 3) in order to work properly as any other positioning would change the number of neighbors of one of the cells and significantly alter its behavior.

But because of the quasi-periodicity of the tiling, there are such locations in every area of the tiling. We can therefore put different copies of the widget in different areas of the tiling and connect them with wires as needed (the wires do not require specific arrangements of the tiles).

The quasi-periodicity can also be used for larger constructions. If we create part of a circuit in some specific location and want to copy it somewhere else, we know that there exists another appropriate

location not too far from the original (the maximal distance is proportional to the radius of the widget we need to copy). Because of this, it is enough to verify that the widget works as intended in its appropriate environment to ensure that we can use it to build arbitrarily large circuits.

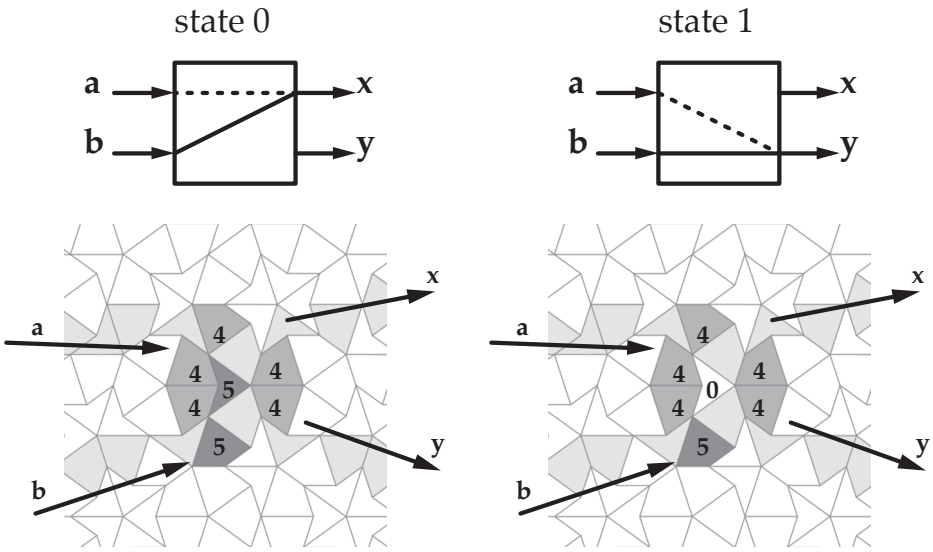

Figure 6: Logical element 2-3 in states 0 (left) and 1 (right). White cells in state 0, light gray cells in state 1.

There are four distinct combinations of an input $\{a,b\}$ and an internal state $\{0,1\}$. For example, Figure 6 shows the two patterns that will simulate the element 2-3 with internal states 0 and 1 respectively. Figure 8 illustrates the evolution of this pattern when a signal enters from its second input (b) and exits through its first output (x) while changing its internal state.

Similarly, Figures 11, 12, and 13 illustrate the pattern used to simulate element 2-17. In this case, the element is symmetric for an input $\{a,b\}$, the behavior of the widget is also symmetric. We only show the two cases, $(\text{input},\text{state}) = (a,0)$ and $(a,1)$.

Because we can control the environment in which the widget evolves, we can precisely set the local transitions up to behave as intended.

Finally, in order to be able to connect all the elements freely, it is necessary to add the possibility of crossing wires. This is achieved by connecting two 2-17 elements as illustrated by Figure 14. Their initial states should have opposite values. Each time a signal pass through, the two elements flip their internal states and the elements act as a crossing module.

Table 3.4 gives a full description of the local transition rule of the automaton.

3.5 Simulation of a Turing Machine

It has been shown in [13] that it is possible to create a circuit that simulates the behavior of a Turing machine cell by arranging eleven copies of a simple logical element with one-bit memory (called *rotary element*). In order to simulate a Turing machine it is then sufficient to connect an infinite line of such

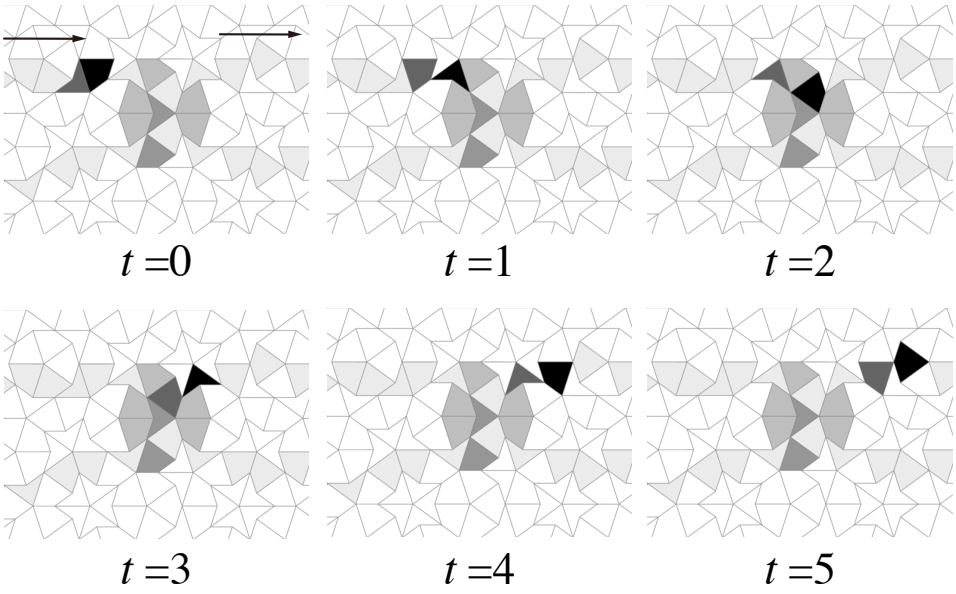

Figure 7: Element 2-3: a transition from state 0, input *a* to state 0, output *x*.

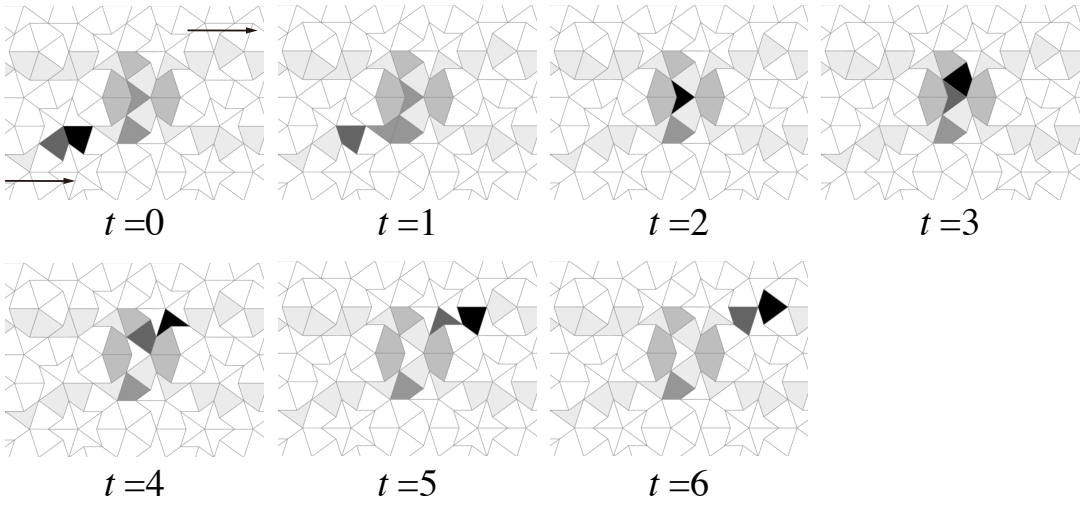

Figure 8: Element 2-3: a transition from state 0, input *b* to state 1, output *x*.

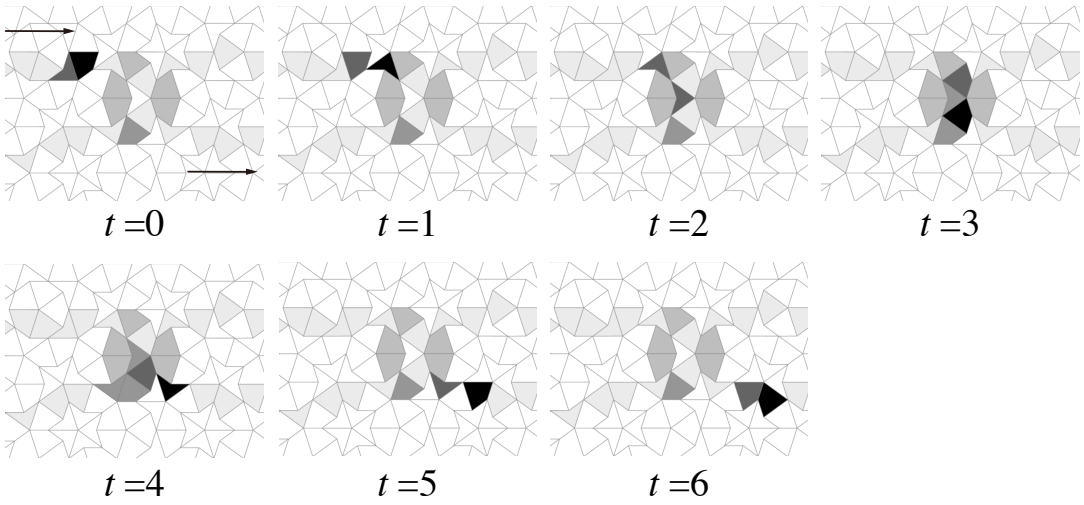

Figure 9: Element 2-3: a transition from state 1, input *a* to state 1, output *y*.

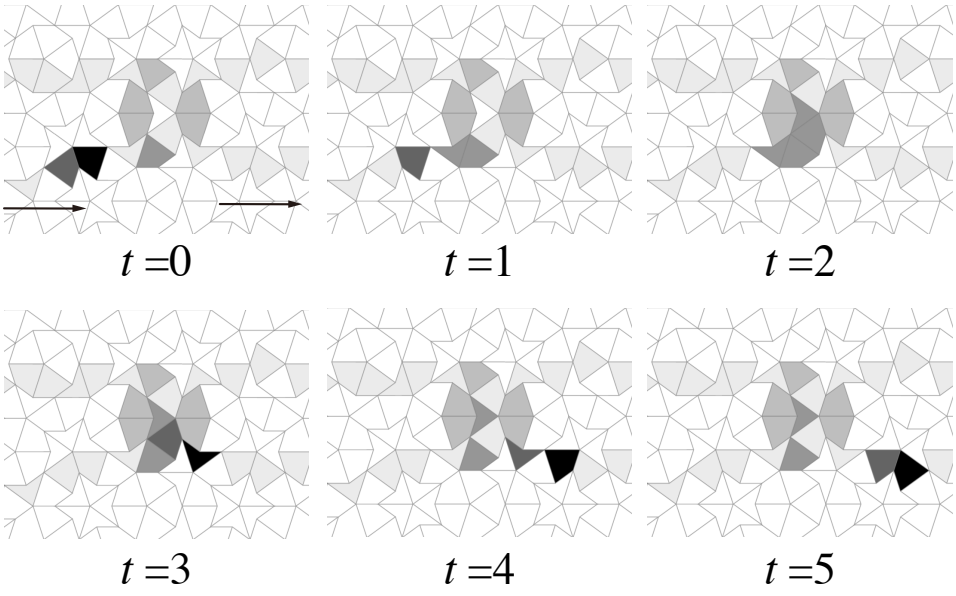

Figure 10: Element 2-3: a transition from state 1, input *b* to state 0, output *y*.

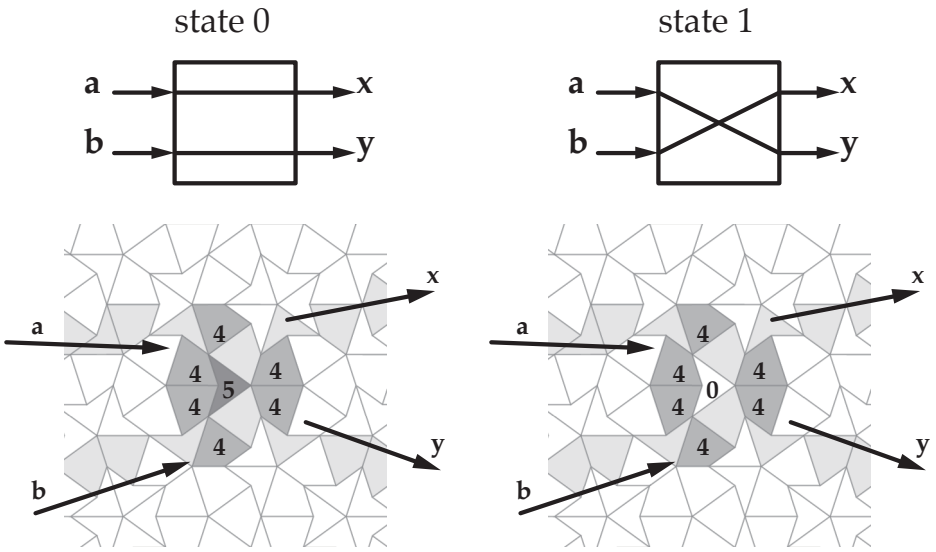

Figure 11: Logical element 2-17 in states 0 (left) and 1 (right). White cells in state 0, light gray cells in state 1.

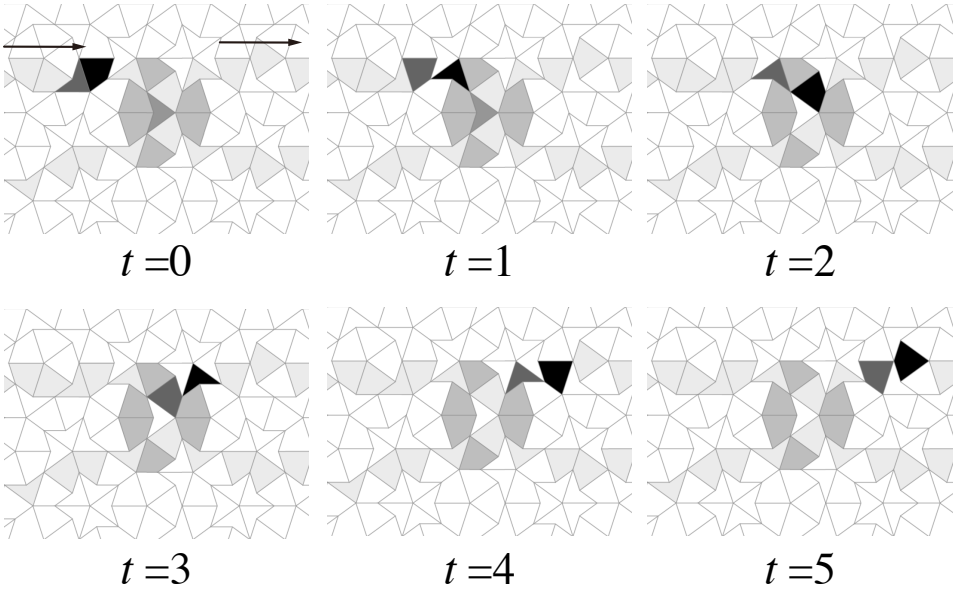

Figure 12: Element 2-17: a transition from state 0, input *a* to state 1, output *x*.

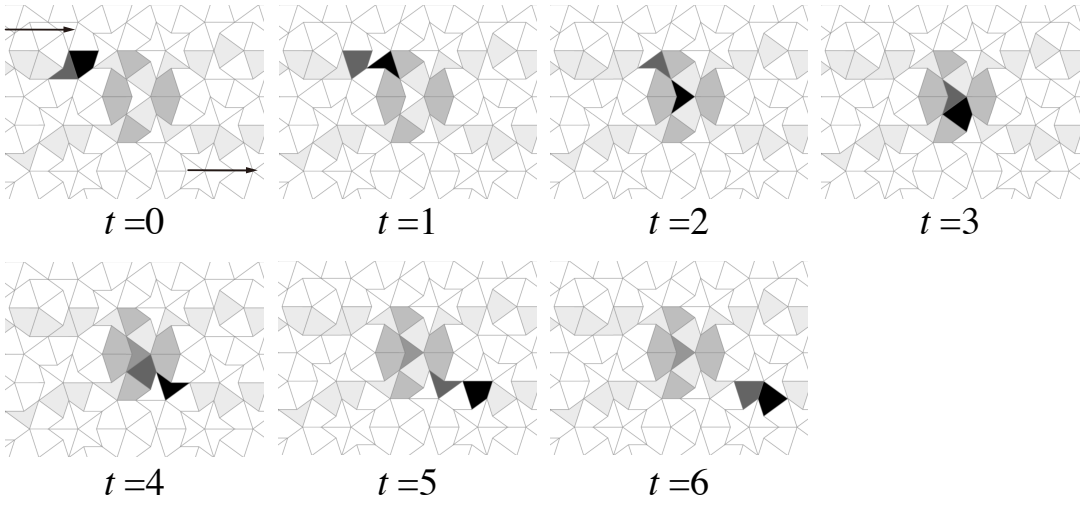

Figure 13: Element 2-17: a transition from state 1, input *a* to state 0, output *y*.

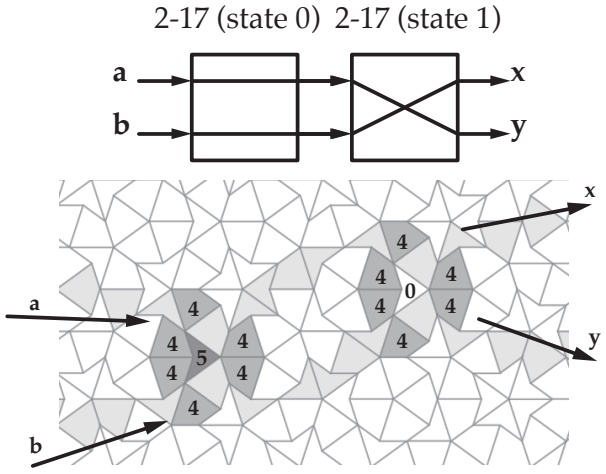

Figure 14: A crossing element. White cells in state 0, light gray cells in state 1.

$3,\ *,1,1,0,0,0 \to 2$	$2,\ *,1,0,1,0,0 \to 1$	$5,\ 0,3,0,1,6,0 \to 0$	$1,\ *,1,0,1,0,0 \to 3$
$1,\ *,1,0,1,2,1 \to 3$	$1,\ *,2,0,1,4,1 \to 3$	$1,\ *,1,0,1,1,0 \to 3$	$1,\ *,1,0,1,2,0 \to 3$
$0,\ 0,3,0,1,6,0 \to 3$	$1,\ *,3,0,1,4,0 \to 3$	$2,\ 0,3,0,1,6,0 \to 5$	$1,\ *,2,0,1,1,0 \to 3$
$1,\ *,2,0,1,2,1 \to 3$	$1,\ *,3,0,1,1,0 \to 3$	$1,\ *,2,0,1,0,0 \to 3$	$1,\ *,0,0,1,2,1 \to 3$
$0,\ 0,2,0,1,6,1 \to 3$	$2,\ 0,2,0,1,6,1 \to 0$	$1,\ *,1,0,1,1,0 \to 3$	$1,\ *,1,0,1,3,0 \to 3$
$1,\ *,2,0,1,3,1 \to 3$	$1,\ *,2,0,1,3,0 \to 3$	$1,\ *,3,0,1,2,1 \to 4$	$1,\ *,3,0,0,2,1 \to 5$
$5,\ *,2,1,0,2,1 \to 1$	$4,\ *,3,1,0,2,1 \to 1$	$1,\ *,3,1,0,2,2 \to 2$	$1,\ *,1,1,0,0,2 \to 3$
$1,\ *,1,0,1,1,2 \to 5$	$5,\ *,1,1,0,1,2 \to 1$	$1,\ *,1,0,1,1,2 \to 5$	$5,\ 0,3,0,0,5,2 \to 3$
$3,\ 0,4,0,0,5,1 \to 2$	$2,\ 0,3,0,1,5,1 \to 0$	$1,\ *,1,0,1,1,1 \to 5$	$1,\ *,2,0,0,3,2 \to 5$
$0,\ 0,3,0,0,5,2 \to 5$	$1,\ *,1,0,0,1,1 \to 3$	$5,\ *,1,0,0,1,3 \to 1$	$5,\ *,2,0,0,3,3 \to 2$
$0,\ 0,3,0,1,5,1 \to 2$	$1,\ *,3,1,0,3,1 \to 3$	$1,\ *,2,2,0,4,0 \to 2$	$2,\ 0,3,1,0,5,1 \to 5$
$5,\ *,1,1,0,1,2 \to 1$	$5,\ 0,2,1,0,5,2 \to 0$	$2,\ *,*,*,*,*,* \to 1$	$3,\ *,*,*,*,*,* \to 2$
$1,\ *,*,*,*,*,* \to 1$	$4,\ *,*,*,*,*,* \to 4$	$5,\ *,*,*,*,*,* \to 5.$	

Table 1: Local transition rule of the 6-state universal cellular automaton.
The symbol '*' is a wildcard that matches any value. In case of multiple matches, the first matching rule is applied (left to right, top to bottom). The rules with no wildcard are only applicable for the $a7$ neighborhood.

circuits. The resulting simulation only requires one active signal to move through the whole infinite circuit.

The rotary element itself can be simulated by elements 2-3 and 2-17 but arranging an infinite line of copies of a given pattern cannot be done periodically on Penrose's tiling. However, since every tiling by Penrose tiles is quasi-periodic, if we can embed a circuit that simulates a Turing machine cell on a certain pattern of the tiling, there exists a constant N such that on every square portion of the plane of size $N \times N$ the pattern appears. This means that if we consider an infinite stripe of the plane divided in squares of size $N \times N$, we can put a Turing machine cell in each square and connect them into an infinite line as illustrated by Figure 15.

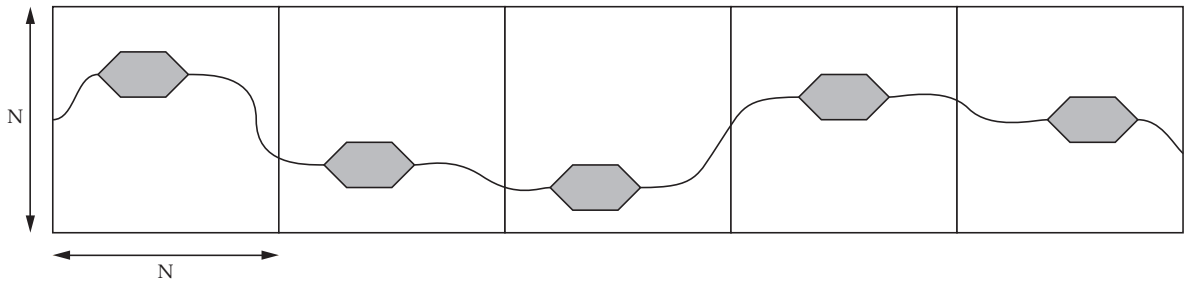

Figure 15: How to simulate the behavior of a Turing machine with copies of a circuit that simulates the behavior of one tape cell.

4 Conclusion

Despite working on an aperiodic structure with heterogeneous neighborhoods, the 6-state cellular automaton that we have described can simulate any logical circuit and even simulate the evolution of a Turing machine, if the infinite circuit is correctly embedded on the tiling (it can be, but in an aperiodic manner). It can work on any valid tiling of the plane by the *kite* and *dart* tiles because all such tilings contain the same finite patterns so any finite configuration on one can be inserted in another.

The number of states of the automaton is likely not minimal for a universal CA. It seems likely that there exist ways to simulate the simple logical elements (or even a different universal combination) with fewer states but possibly larger widgets. It is however very difficult to optimize these constructions as many different environments have to be considered in the search.

Intrinsic universality (the ability to simulate any other cellular automaton) should also be investigated. In that case however, a single-signal asynchronous system cannot work but it might be possible by arranging infinitely many circuits in similar environments, each simulating the behavior of a cell, and connecting them with wires of fixed length to avoid synchronization problems.

Recently gliders are found in the case of Penrose Rhomb tiling [7]. It is also meaningful to investigate the possibility of a glider based universal cellular automaton.

Acknowledgments.

Katsunobu Imai gratefully acknowledges the support of the Japan Society for the Promotion of Science and the Grant-in-Aid for Scientific Research (C) 22500015.

References

[1] Chidyagwai, P., Reiter, CA.: A local cellular model for growth on quasicrystals. *Chaos, Solutions and Fractals* **24** (2005) 803–812.

[2] Dewdney, A.K.: Computer recreations: a cellular universe of debris, droplets, defects and demons. *Scientific American* **261**, August (1989) 102-105.

[3] Gardner, M.: Mathematical Games - The fantastic combinations of John Conway's new solitaire game "life" (1970) 223. pp. 120123.

[4] Gardner, M.: Extraordinary nonperiodic tiling that enriches the theory of tiles. *Mathematical Games, Scientific American*, January, 1977, p. 110-121.

[5] Griffeath, D.: A CA run on Penrose Tiles, http://psoup.math.wisc.edu/archive/recipe39.html.

[6] Fisch, R.: Cyclic cellular automata and related processes. *Physica D* **45** (1990) 19–25.

[7] Goucher, A. P.: Gliders in Cellular Automata on Penrose Tilings. *Journal of Cellular Automata* (to Appear).

[8] Imai, K., Iwamoto, C., and Morita, K.: A Five-State von Neumann Neighbor Universal Hyperbolic Cellular Automaton. *Journal of Cellular Automata* **1** 4 (2006) 275–297.

[9] Imai, K., Hatsuda, T., Poupet, V. and Sato, K.: A Universal Semi-totalistic Cellular Automaton on Kite and Dart Penrose Tilings. *Proc 18th international workshop on Cellular Automata and Discrete Complex Systems and 3rd international symposium Journes Automates Cellulaires (AUTOMATA&JAC 2012), Electronic Proceedings in Theoretical Computer Science* **90** (2012) 267-278.

[10] Lee, J., Peper, F., Adachi, S., and Morita, K.: An asynchronous cellular automaton implementing 2-state 2-input 2-output reversed-twin reversible elements. In *Proc. ACRI 2008, LNCS 5191* (2008) 67–76.

[11] McClure, M.: A Stochastic Cellular Automaton for Three-Coloring Penrose Tiles *Computers & Graphics* **26** 3 (2002) 519–524.

[12] Morita, K., Ogiro, T., and Alhazov, A.: Non-degenerate 2-state reversible logic elements with three or more symbols are all universal. *Multiple-Valued Logic and Soft Computing* **18** 1 (2012) 37-54.

[13] Morita, K.: Constructing a reversible turing machine by a rotary element, a reversible logic element with memory. *In Hiroshima University Institutional Repository* http://ir.lib.hiroshima-u.ac.jp/00029224.

[14] Mukai, Y., Morita, K.: Universality of 2-symbol reversible logic elements with memory. *Resume for LA Symposium Summer* (2011), **S16** 1–15 (in Japanese).

[15] Owens, N., Stepney, S.: Investigation of the Game of Life cellular automata rules on Penrose tilings: lifetime, ash and oscillator statistics. *Journal of Cellular Automata* **5** 3 (2010) 207-225.

[16] Owens, N., Stepney, S.: The Game of Life rules on Penrose tilings: still life and oscillators. In: A. Adamatzky (Ed.) *Game of Life Cellular Automata*, Springer-Verlag London (2010) 331–378.

[17] Reiter, CA.: Medley of spirals from cyclic cellular automata. *Computers & Graphics* **34** (2010) 72–76.

[18] Weeks, E.: Cellular automata on quasicrystals, http://www.physics.emory.edu/˜weeks/pics/qvote.html.

[19] Wójtowicz, M.: Cellular Automata rules lexicon, http://www.mirekw.com/ca/ca_rules.html.

[20] Wolfram, S.: in *New Kind of Science* (2002) Wolfram Media, 929–929.