Turing Universality of Step-Wise and Stage Assembly at Temperature 1

Bahar Behsaz[1], Ján Maňuch[1,2], and Ladislav Stacho[1]

[1] Department of Mathematics, Simon Fraser University, Burnaby, Canada
[2] Department of Computer Science,
University of British Columbia, Vancouver, Canada
{bbehsaz,lstacho}@sfu.ca, jmanuch@cs.ubc.ca

Abstract. In this paper, we investigate the computational power of two variants of Winfree's abstract Tile Assembly Model [14] at temperature 1: the Stage Tile Assembly Model and the Step-wise Tile Assembly Model. In the Stage Tile Assembly Model, the intermediate assemblies are assembled in several "bins" and they can be mixed in prescribed order and attach together to form more complex structures. The Step-wise Tile Assembly Model is a simplified model of stage assembly in which only one bin is used and assembly happens by attaching tiles one by one to the growing structure.

An interesting and still open question is whether the abstract Tile Assembly Model at temperature 1 is Turing Universal, i.e., it can simulate a Turing machine. It is known that various slight modifications of the model are indeed Turing Universal. Namely, deterministic self-assembly in 3D and probabilistic self-assembly in 2D at temperature 1 [3] and self-assembly model at temperature 1 with a single negative glue [10] are known to be able to simulate a Turing machine. In this paper we show that the Step-wise Tile Assembly Model and the Stage Tile Assembly Model are also Turing Universal at temperature 1.

1 Introduction

Informally, self-assembly is a bottom-up process in which a small number of types of components automatically assemble to form a more complex structure.

In 1998, Winfree [14] introduced the (abstract) Tile Assembly Model (TAM) – which utilizes the idea of Wang tiling [13] – as a simplified mathematical model of the DNA self-assembly pioneered by Seeman [12]. Nature provides many examples: Atoms react to form molecules. Molecules react to form crystals and supermolecules. Similarly, self-assembly of cells plays a part in the development of organisms. Simple self-assembly schemes are already widely used in chemical syntheses. It has been suggested that more complicated schemes will ultimately be useful for circuit fabrication, nano-robotics, DNA computation, and amorphous computing [2,15,11,7,6,1]. Several promising experiments as well has been suggested and practiced. In accordance with its practical importance, self-assembly has received increased theoretical attention over the last few years [9,14,2,15,8].

D. Stefanovic and A. Turberfield (Eds.): DNA 18, LNCS 7433, pp. 1–11, 2012.

One of the interesting theoretical questions about this process is how "powerful" the model of Winfree is. In the past years Self Assembly Model has been studied by many researchers from computational aspect of view. Many of these results are focused on the systems with temperature 2 or higher [3,5]. In such models strength of at least two is needed for a tile to permanently bind to an assembly. It is known that at temperatures equal or greater than 2, Self Assembly Systems are quite powerful, in fact it is shown that they are Turing Universal [3]. But for temperature 1, where the system is allowed to place a tile at a position if any positive strength bond exist among the tile and the neighbors at that position, the computational "power" is unknown.

From practical point of view, temperature 1 systems are more well-mannered and easier to control for laboratory experiments. Thus recent attention has been redirected to some variants of temperature 1 Self Assembly Model. It is shown that by adding some constraints and features to the original model more "power" is achievable. Cook et al. [3] proved the Turing Universality for deterministic self-assembly in 3D and probabilistic self-assembly in 2D at temperature 1, and Patitz et al. [10] introduced a self-assembly model at temperature 1 with a single negative glue, and showed it is Turing Universal. All results use the idea of "geometric tiles".

In this paper we study the power of two variants of the Self Assembly Model at temperature 1: "the Stage Tile Assembly Model" and "the Step-wise Tile Assembly Model". These models are defined in Section 2. We reuse the idea of "geometric tiles", and prove that these models are Turing Universal as well. In Section 3, we introduced the Zig-Zag Assembly which can simulate Turing machines at temperature 2. To show our main results it is enough to simulate the Zig-Zag assembly at temperature 1 in these two models. Details of these simulations are presented in Sections 4 and 5, respectively.

2 Abstract Tile Assembly Model

We present a brief description of the tile assembly model based on Rothemund and Winfree, for a more detailed description we refer the reader to [14]. We will be working on a $\mathbb{Z}\times\mathbb{Z}$ grid of unit square locations. The set of directions $D = \{N, E, W, S\}$ is used to indicate relative positions in the grid. Formally, they are functions $\mathbb{Z}\times\mathbb{Z} \to \mathbb{Z}\times\mathbb{Z}$ such that:

$$N(i,j) = (i, j+1),\ E(i,j) = (i+1, j),\ S(i,j) = (i, j-1) \text{ and } W(i,j) = (i-1, j).$$

The inverse directions are defined in a natural way, for example, $N^{-1}(i,j) = S(i,j)$.

Let Σ be a set of *binding domains (glues).* The set Σ contains a special binding domain *null* that represents no glue. A *tile type* t is a 4-tuple $(t_N, t_E, t_S, t_W) \in \Sigma^4$ indicating the associated binding domains on the north, east, south and west sides, respectively. Note that tile types are oriented, thus a rotated version of a tile type is considered to be a different tile type. We denote the set of tile types with T. There is a special tile type *empty* = (*null*, *null*, *null*, *null*) in T which represents an empty space, i.e., no tile has been placed in that position. A configuration is a function C, mapping $\mathbb{Z} \times \mathbb{Z}$ to the set of tile types in T. Hence, $C(i, j)$ is the tile at the position (i, j) in the configuration C. Let a *structure* $S(C)$ of a configuration be the set of positions that are not mapped to *empty* in C. A configuration may contain finite or infinte number of tiles.

Under Tile Assembly Model a tile system is a 5-tuple $\langle \Sigma, T, \phi, g, \tau \rangle$, where T is the finite set of tile types with binding domains from Σ and contains the tile *empty*, ϕ is a set of configurations on T called set of seed configurations, g is a function which determines the strength of glues, and τ is a threshold parameter called temperature. In this paper we will omit parameters Σ and g when they are defined already.

A strength function $s_\Sigma : \Sigma \times \Sigma \rightarrow \mathbb{N}$ measures the strength of interaction between binding domains. For example, for simple strength function at temperature 1 $\tau = 1$, denoted by s_Σ, it must satisfy $s_\Sigma(\sigma, \sigma') = 1$, if $\sigma = \sigma' \neq null$, and $s_\Sigma(\sigma, \sigma') = 0$ otherwise. A tile t can be added to a configuration at position (i, j), if and only if the sum of the interaction strengths of t with its neighbors at position (i, j) reaches or exceeds τ (the temperature). If the assembly process reaches a point when no more attachments are possible, the produced assembly is denoted terminal and is considered the output assembly of the system. The smallest number of tile types and glues required to assemble a given shape are called the tile and glues complexity of the shape, respectively.

2.1 Staged Tile Assembly Model

Erik D. Demaine et al. [4], presented the Staged Tile Assembly Model, a generalization of the tile assembly model that captures the temporal aspect of the laboratory experiment, and enables more flexibility in the design and fabrication of complex shapes using a small tile and glue complexity. The main feature of staged assembly is the ability of gradual addition of specific tiles in a sequence of stages. In addition, any tiles that have not yet attached as part of a growing structure can be washed away and removed (in practice, using a weight-based filter, for example) at the end of the stage. More generally, we can have any number of bins, each containing tiles and/or assemblies that self-assemble as in the standard tile assembly model. During a stage, any collection of operations of two types are allowed: (1) add (arbitrarily many copies of) new tiles to an existing bin; and (2) pour one bin into another bin, mixing the contents of the former bin into the latter bin. In both cases, any tiles that do not assemble into larger structures are removed at the end of the stage. These operations let us build intermediate terminal assemblies in isolation and then combine different terminal assemblies to more complex structures. Two new complexity measures in addition to tile and glue complexity arise: the number of stages, or stage complexity, measures the time required by the operator of the experiment, and the number of bins, or bin complexity, measures the space required for the experiment. (When both of these complexities are 1, the model is equivalent to the regular tile assembly model.)

2.2 Step-Wise Tile Assembly Model

Reif proposed a related, but simpler, model called the Step-wise Assembly Model [11]. This model is the special case of Staged Assembly Model, in which the bin complexity is 1. Step-wise Assembly Model uses several sets of tiles (one for each step). Let T_i be the tile set for step i. $\{T_i\}_{i=1}^{k}$ is a sequence of finite sets of tiles. At first step we immerse a seed tile into one of the sets, filter out the assembled shape from this set and place the assembled shape into the next set of tiles where it now acts as a seed tile and assembly continues. Step-wise Assembly Model enables production of more efficient assemblies in terms of tile types at the price of the work of an operator. Here one new complexity measure can be defined as well, the number of steps, or step complexity, which measures the time required for the experiment.

3 Zig-Zag Tile Assembly Model

Informally, a Zig-Zag assembly is an assembly in which the assembly can grow only left to right (or right to left) up to the point at which a first tile is placed into the next row north, and the direction of growth is reversed in the next top row.

Temperature 2 Zig-Zag assembly system: A tile system $\Gamma = \langle T, s, 2\rangle$ is called temperature 2 Zig-Zag system if the tiles in T assures the assembly grows under these two conditions:

1. The assembly sequence, which specifies the order in which tile types are attached to the system is unique, i.e., the system is deterministic.
2. If the $(i-1)$th tile added to the assembly sequence is placed in an even row (counting from the row 0, containing the seed tile) in position (x, y) of the grid, then the ith tile is either placed in position $E(x, y)$ or $N(x, y)$. And if the $(i-1)$th tile added is placed in an odd row, then the next tile to be added (i.e. ith tile in the sequence) is placed either in the position $W(x, y)$ or $N(x, y)$.

Because $\tau = 2$, the key property of Zig-Zag system is interaction of strength at least two between the sides of a tile and size neighbors is required to be able to add the ith tile to the sequence. Right after adding a tile t to the assembly the glues on the sides of t that already have neighbors, are called "*inputs*" and the glues on the sides of t that are exposed are called "*outputs*" of tile t. Cook et al. proved that the deterministic Zig-Zag Tile Assembly Model at temperature 2 is Turing universal [10], and using this they proved the deterministic assembly in 3D and the probabilistic assembly in 2D at temperature 1, are Turing Universal by simulating Zig-Zag assembly System with the idea of "geometric tiles". We will use the same technique to prove our main results.

4 Turing Universality of Step Tile Assembly System

This section and Section 5 contain the main results of this paper. First we prove that every Zig-Zag tile assembly system at temperature 2 can be simulated by a step-wise assembly tile system at temperature 1.

Theorem 1. *For any temperature 2 Zig-Zag tile assembly system $\Gamma = \langle T, s, 2\rangle$ there exists a Step-wise tile assembly system Γ' at temperature 1 that simulates Γ at vertical scale 5, horizontal scale $O(\log(|T|))$, step complexity and tile complexity $O(|T|\log(|T|))$.*

Proof. We follow the idea of the proof in [3]. We represent each tile in Γ with a set of tiles in Γ' called "*macro tiles*" and "fake" the cooperative attachment of tiles in temperature 2.

Let G be the set of all strength 1 glues on north and south side of tiles that appear in the system Γ. Label each type of these glues with numbers 0 to $|G|-1$ in an arbitrary order. For any glue $g \in G$ let $b(g)$ be the binary representation of the label of g. The main idea is to geometrically represent number $b(g)$ on the side of macro tile that is representing the north or south glue of each $t \in T$. We symbolize each bit of $b(g)$, 0 or 1, with bumps. For east and west side glues and for strength 2 glues we do not use geometrical representation. Those are represented with simple strength 1 glues. By alternating between the two tile sets, the system will simulate the input from the south with its bumps, and from left and right the inputs are read with a strength 1 glue. After reading the input, the outputs are unpacked (written) using the same geometrical representation of the bits of the binary coding of output glue on the north (if any) and a glue of strength 1 on east or west side (according to the side of the outputs of the tile in the Zig-Zag system that the system is assembling).

Each macro tile has two bumps of vertical length 2 on beginning and end of its north side as shown in Figure 1. These special bumps will guarantee that macro tiles in the assembly will be aligned. The actions of reading and unpacking glues for a macro tile m that corresponds to a tile $t \in \tau$ with south and west inputs and north and east outputs are discussed in more details in what follows. Examples of tile with other combinations of input and output are illustrated in Figure 1.

Reading the South and West Inputs: Each macro tile starts from the south. The first tile that starts the macro tile has the west input on its west side with one single glue. Then the assembly continues by reading along the north side of the macro tile beneath it. Figure 2 pictures the basic idea behind reading each bit. As it is shown in Figure 2 each bit has width 4 and we recognize whether the bit is 0 or 1 by the location of the bump. Also, each bit is being read in exactly two steps and with exactly two tile sets T_1 an T_2, represented in Figure 2. The final tile sets T_1 and T_2 is a union of all required tiles for different tiles respectively in step 1 and 2.

The tiles shown in Figures 3a and 3c are the key tile types in each tile set (tile set 1 and 2) responsible for reading bit i of the glue on the south side. The

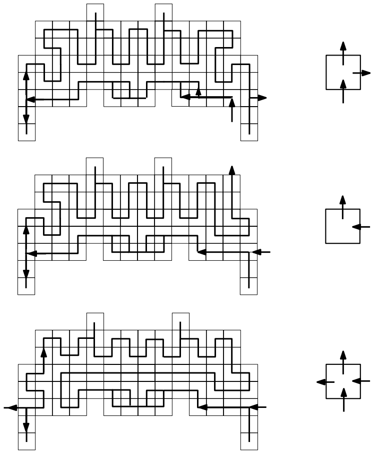

Fig. 1. Macro tiles simulating the tile types from Zig-Zag Tile Assembly System shown on the right side of each macro tile

final assembly of this part of the macro tile which reads the glue on the south of the macro tile for a tile $t \in T$ with glue 110 on the south side, is illustrated in Figure 4.

For reading each bit of the binary representation of the glue on the south side of each macro-tile, say m, there are $O(1)$ unique tiles in tile sets T_1 and $|T_2|$. These tiles assemble together and read the ith bit of the glue on the south side of m when the glue is being read in process of creating m.

After reading $\log(|G|)$ bits, the glue on the south is read completely. The west input is read via the glue on west side of the first step in the assembly sequence of the macro tile. Thus, at this point both inputs are read and the outputs can be determined deterministically (since Zig-Zag TAS is deterministic) and this mapping is done with the tile shown in Figure 3e.

Unpacking the North and East Outputs: When the outputs are determined the assembly crawls back on itself (in this case it crawls back to east). Then it starts "writing" the binary representation of the north glue with illustrating 0 an 1 geometrically. This process is in essence the same as what discussed above but naturally with reverse functionality. Instead of adding the next right digit in this case we delete the right most digit each time at each step. The key tile types necessary for unpacking a glue is illustrated in Figure 5. Note that it unpacks the east glue again with a single unique strength 1 glue on the tile at the very south east position of macro-tile. The complexity of tile types is the same as the reading tiles.

For each bit of the binary representation of the glue on the south or north side of each macro-tile, say m, there are $O(1)$ unique tiles in the tile sets T_1 and T_2. Each macro-tile needs $O(\log(|T|))$ unique tile types. The terminal assembly of Zig-Zag assembly Γ has $O(|T|)$ tiles and all of the $O(\log(|T|)$ tiles assembling

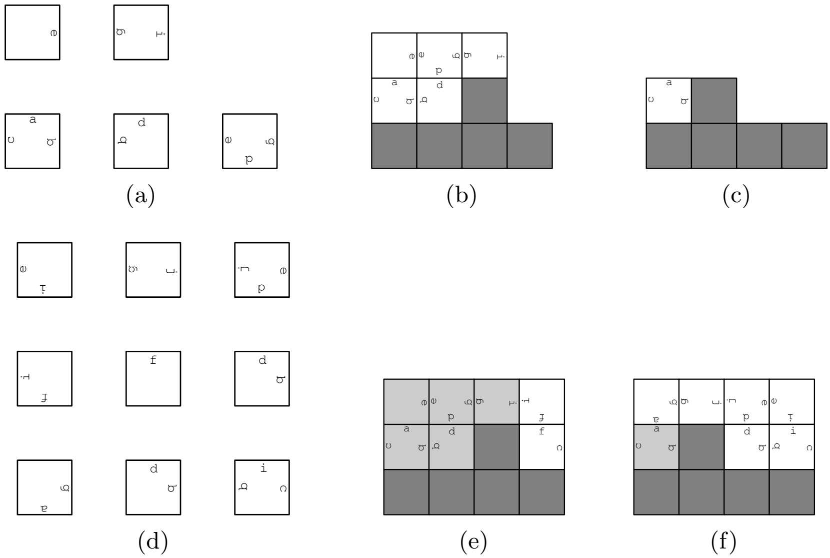

Fig. 2. (a) The first tile set T_1. **(b)** The final assembly for 0 after the first step (after exposing to T_1). The tiles colored light gray belong to the first tile set and the tiles colored with dark gray belong to the macro-tile beneath. The assembly is growing from left to right. **(c)** The final assembly for 1 after the first step (after exposing to T_1). **(d)** The first tile set T_1. **(e)** The final assembly for 0 after the first step (after exposing to T_1). **(f)** The final assembly for 1 after the first step (after exposing to T_1).

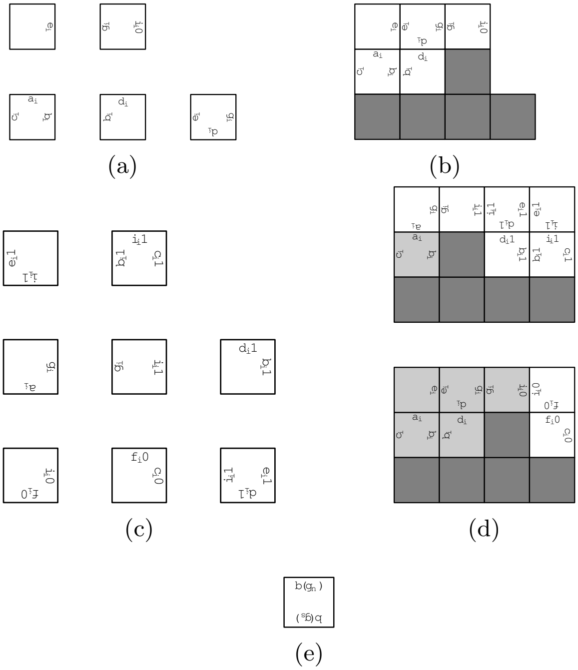

Fig. 3. (a) The subset of T_1 reading the ith bit from left to right. $a_i, \ldots, j_i$ are different representations of the first $i-1$ bits. **(b)** The final assembly reading bit 0 after exposing to the first tile set. The tiles that are colored light gray belong to the first tile set and the tiles colored with dark gray belong to the "reading part" of this macro-tile. **(c)** The subset of T_2 reading the ith bit from left to right. **(d)** The final assembly tile set reading bits 0 and 1 after exposing to second tile set. **(e)** The tile mapping the input of a macro tile that is read to the corresponding output. At this point, the assembly starts crawling back on the assembly and start unpacking.

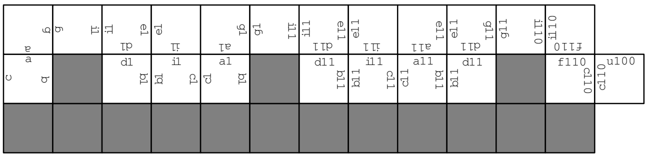

Fig. 4. An example of reading the glue 110 on south side of a macro tile bit by bit from left to right

macro-tile m has to store the original tile that m belongs to in Γ. Thus, we use $O(|T|\log(|T|))$ unique tile types. Also, each macro-tile is constructed in $O(\log(|G|))$ steps. Hence, the tile and step complexity of the final assembly is $O(|T|(\log|T|))$. The assembly system described here clearly simulates Γ.

For every Turing machine M and $w \in \Sigma^*$ there exists a Zig-Zag Tile Assembly Model Γ that simulates M on input w [3]. By Theorem 1 there is a step-Tile Assembly System that simulates Γ. Therefore, we have the following result:

Theorem 2. *The Step-wise Tile Assembly Model is Turing Universal.*

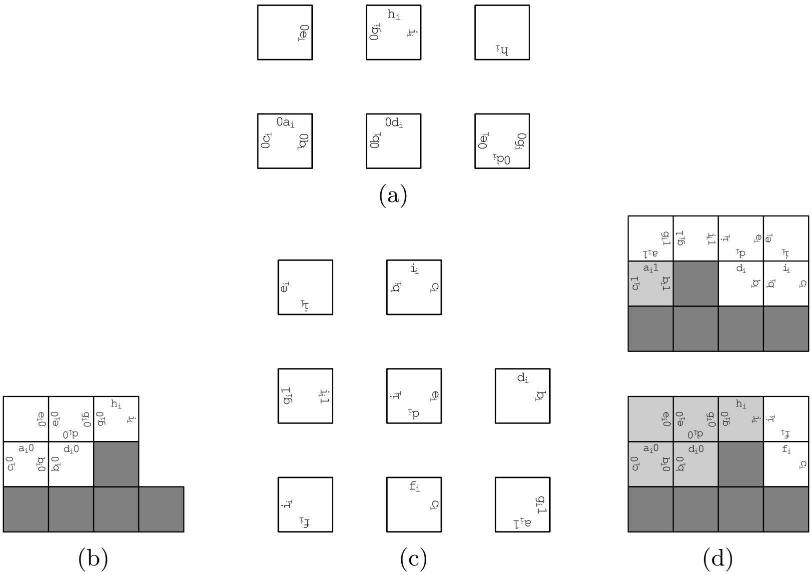

Fig. 5. (a) The subset of T_1 unpacking the ith bit from left to right. $a_i, \ldots, j_i$ are different representations of the first $i-1$ bits. **(b)** The final assembly unpacking bit 0 after exposing to the first tile set. The tiles that are colored light gray belong to the first tile set and the tiles colored with dark gray belong to the macro-tile beneath. **(c)** The subset of T_2 unpacking the ith bit from left to right. **(d)** The final assembly tile set unpacking bits 0 and 1 after exposing to second tile set.

5 Turing Universality of Stage-Assembly System

By performing the exact procedure which is described in Section 4, the Turing Universality of Stage Assembly Model can be proved. But for Stage-wise Assembly Model we can utilize the construction with the "bins" and reduce the number of stages. For this, we use a similar technique as in the construction in Theorem 1. We take an instance of Zig-Zag assembly system, say Γ and simulate it with an instance of stage-assembly system which is constructed with macro tiles as described in the previous section. But here with the use of different bins we can decrease the stage complexity to $O(\log(|T|))$.

Theorem 3. *For any temperature 2 Zig-Zag tile assembly system* $\Gamma = \langle T, s, 2\rangle$ *there exists a Stage Tile Assembly system* Γ' *at temperature 1 that simulates* Γ *at vertical scale 5, horizontal scale* $O(\log(|T|))$*, stage complexity* $O(\log(|T|))$ *and tile complexity* $O(|T|\log(|T|))$*.*

Proof. The construction of Γ' is almost the same as proof of Theorem 1. Again let G be the set of north/south glues. For each $t \in T$ we create a unique macro-tile. The tile sets of Γ' are T_1' and T_2', which are the same as T_1 and T_2 in Γ which is described in the proof of Theorem 1, respectively. The difference is that the tiles in the south side of each macro-tile do not read a binary number. Instead they unpack (write) the corresponding binary number, which is the binary number of the glue of the south side of t. The tile types which assist unpacking this glue are the same as the tiles types that are used to read this glue for the macro tile corresponding to t in the proof of Theorem 1. For each tile $t \in T$ we use a unique bin. The seed tile here is the tile which maps the input to the output of the macro tile which is shown in Figure 3c. So the assembly sequence for the assembly that unpacks the south glue is the reverse of the assembly sequence which reads the south glue in Section 4. After each macro-tile assembled in each bin we will pour the desired assembled macro tiles in one bin with choosing the macro-tile in Γ' corresponding to the seed tile in Γ be the seed configuration in Γ'.

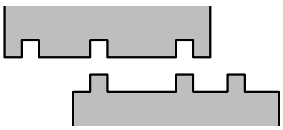

Fig. 6. The possible slips for two macro-tiles one with south side depicting (110) the other with north side (101). The two segments of (10) can be matched together without the vertical teeth of length two on the south of macro tiles.

In each bin it takes $O(\log(|G|))$ stages before the assembly of the macro tile is final, this yields to stage complexity $O(\log |T|)$. Since the tile types are the same in Section 4, the tile complexity is $O(|T| \log(|T|))$.

Now, since Γ is a deterministic system the macro-tiles automatically assemble by matching inputs and outputs uniquely in one step. The vertical bump of length 2 on south side of each macro-tile guarantees that the beginning and the end of each macro-tile matches exactly with the beginning and the end of other macro-tiles in north and south side of the macro-tile. Without these bumps horizontal slips of macro-tiles along each other is possible as shown in Figure 6.

6 Conclusion

In this paper, we proved Turing Universality at temperature 1 for two tile assembly models: Step-wise Tile Assembly Model and Stage Tile Assembly Model. The question whether the Tile Assembly Model at temperature 1 is Turing Universal remains open.

References

1. Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight Jr., T.F., Nagpal, R., Rauch, E., Sussman, G.J., Weiss, R.: Amorphous computing. Commun. ACM 43(5), 74–82 (2000)
2. Adleman, L., Cheng, Q., Goel, A., Huang, M.-D., Kempe, D., de Espanés, P.M., Rothemund, P.W.K.: Combinatorial optimization problems in self-assembly. In: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, pp. 23–32 (2002)
3. Cook, M., Fu, Y., Schweller, R.: Temperature 1 self-assembly: Deterministic assembly in 3D and probabilistic assembly in 2D. In: Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (2011)
4. Demaine, E.D., Demaine, M.L., Fekete, S.P., Ishaque, M., Rafalin, E., Schweller, R.T., Souvaine, D.L.: Staged Self-assembly: Nanomanufacture of Arbitrary Shapes with $O(1)$ Glues. In: Garzon, M.H., Yan, H. (eds.) DNA 13. LNCS, vol. 4848, pp. 1–14. Springer, Heidelberg (2008)
5. Doty, D., Patitz, M.J., Summers, S.M.: Limitations of self-assembly at temperature 1. Theor. Comput. Sci. 412(1-2) (January 2011)
6. Gómez-López, M., Preece, J.A., Fraser Stoddart, J.: The art and science of self-assembling molecular machines. Nanotechnology 7(3), 183 (1996)
7. Lagoudakis, M.G., Labean, T.H.: 2D DNA self-assembly for satisfiability. In: Proceedings of the 5th DIMACS Workshop on DNA Based Computers (1999)
8. Mao, C., Labean, T.H., Reif, J.H., Seeman, N.C.: Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. Nature 407, 493–496 (2000)
9. Maňuch, J., Stacho, L., Stoll, C.: Step-wise tile assembly with a constant number of tile types. In: Natural Computing (to appear)
10. Patitz, M.J., Schweller, R.T., Summers, S.M.: Exact Shapes and Turing Universality at Temperature 1 with a Single Negative Glue. In: Cardelli, L., Shih, W. (eds.) DNA 17 2011. LNCS, vol. 6937, pp. 175–189. Springer, Heidelberg (2011)
11. Reif, J.H.: Local parallel biomolecular computing. In: DNA Based Computers III. DIMACS, vol. 48 (1999)
12. Seeman, N.C.: Nucleic-acid junctions and lattices. Journal of Theoretical Biology 1, 237–247 (1982)
13. Wang, H.: Proving theorems by pattern recognition II. The Bell System Technical Journal XL 1, 1–41 (1961)
14. Winfree, E.: Algorithmic Self-assembly of DNA. PhD thesis, California Institute of Technology (1998)
15. Winfree, E., Yang, X., Seeman, N.C.: Universal computation via self-assembly of dna: Some theory and experiments. In: DNA Based Computers II. DIMACS, vol. 44, pp. 191–213 (1996)