Laboratoire de l'Informatique du Parallélisme
Unité mixte CNRS - ENS Lyon - INRIA - UCB Lyon n°5668

Internship report

Oritatami model of molecular cotranscriptional folding: possible and impossible

Daria Pchelina, supervised by Nicolas Schabanel

September 9, 2019

1 Introduction

Molecular folding, also called self-assembly, is a process where sequences of nucleotides or amino-acids are assembled into shapes [5, 22, 23]. This subject got a lot of attention over the last decades. Indeed, being able to assemble arbitrary molecular structures has many applications, from medicine [21] and nanotechnologies [20] to computer science [18]. Many experiments have shown the huge potential of molecular folding. Molecules can assemble into predefined 2D [19] and 3D-shapes [12] and, moreover, they can perform computations [18].

To study such a complex phenomenon, one needs mathematical models. Wet lab experiments require many resources and efforts and do not explain the origin of observed events. Therefore, detailed theoretical analysis is always necessary to fully understand the process. There exist various models of DNA self-assembly: the abstract Tile Assembly Model (TAM), the kinetic TAM [23], the probabilistic TAM [7], the hierarchical self-assembly [4], the nubot model [6]...

DNA self-assembly is a powerful instrument, but it has a number of constraints. First of all, DNA sequences are very stable and do not easily interact with other molecules. Moreover, their folding is possible only after a precise process of heating and cooling which can be attained only in a particular environment.

By contrast, RNA is much less stable and can be assembled with less efforts. This also means that their folding behaviour is harder to simulate and control. For instance, in different variants of the hydrophobic-hydrophilic model [10], even predicting the final shape of a sequence is NP-complete [3].

Transcription is a process where an RNA polymerase (orange in Figure 1) synthesizes a copy (blue) of a gene (gray). The copy, called transcript, folds upon itself immediately as it is produced by the RNA polymerase. This is called cotranscriptional folding [13, 15]. Figure 1 illustrates cotranscriptional folding of a transcript into a rectangular RNA tile designed in [15]. Intuitively, this process is a local energy optimization or, in other words, a greedy algorithm. Understanding this phenomenon is necessary to study RNA self-assembly.

Oritatami system (OS[1]) is a simplified mathematical model of cotranscriptional folding introduced in [14] and based on the experiments from [15]. In this model, nucleotides are abstracted as beads, the molecular sequences are considered as words on a finite alphabet of bead types and the hydrogen bonds between nucleotides are represented by a symmetric relation on the alphabet. The dynamics of the system consists in the following: at each moment of time, a part of the sequence is stabilized on the plane, while the next δ beads are exploring freely the environment in order to find a position optimizing the energy, which will stabilize the next bead. The formal definition of the model is given in Section 2.

[1] The possible confusion with Operating Systems is not that meaningless since Oritatami Systems are Turing complete [16], so we can use them as an abstraction for computations.

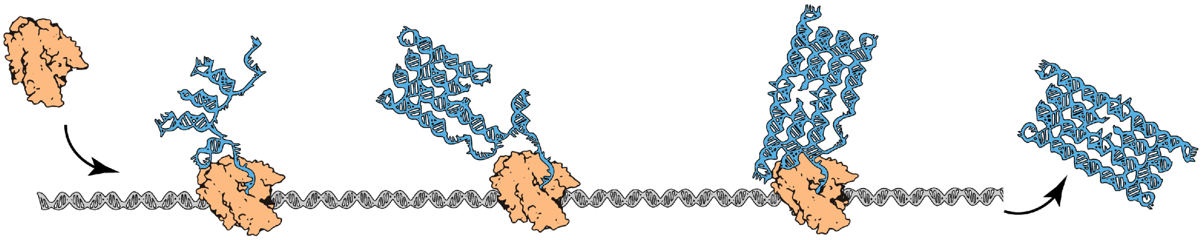

Figure 1: An illustration of cotranscriptional folding from [15].

During my internship, I investigated several aspects of Oritatami systems. First, we have considered this model from the geometric point of view: which shapes can be folded and which can not? The keynote of my research is the borderline between the shapes which are possible and impossible to fold. I started by fixing an existing reasoning, got more general results useful for proving unfoldability of certain shapes (Section 3.1) and applied them in two cases (Sections 3.2 and 3.3). I presented this work during "Journées SDA2 2019" meeting (`https://sda2-2019.sciencesconf.org/279506`). There are results showing that all shapes scaled in a certain way can be self-assembled by an OS [8]. I worked on the last case in a construction folding shapes at a smaller scale, which is explained in Section 3.4.

The Oritatami model was already proven to have full computational power: it can simulate an arbitrary tag system [16] and there is a simplified way to simulate a counter [14]. During my internship, I participated in the exploration of computational properties of this model (see Section 4.2). I took part in the development of a construction to simulate cellular automata by an OS with Nicolas Schabanel, Shinnosuke Seki, and Yuki Ubukata (find more details about it in Section 4.1). This is a candidate to be the most elegant and simple way to show that the Oritatami model is Turing complete, in contrast to the very complicated existing result [16]. Moreover, we hope to use it as a base to implement Oritatami computations in the wet lab. Now, as we know that an OS can perform computations, how complex can be a shape produced by it? Following this question asked during "Journées SDA2" by Guillaume Theyssier, we discovered that any recursively enumerable set can be represented by the terminal shape of an OS.

Currently, I am involved in the writing of three papers. The first paper, where I fixed a proof and obtained some new results (Sections 3.1 - 3.3), is going to be the next version of previously published [8]. The second one is about the simulation of cellular automata (Section 4.1). The last paper in progress describes how any recursively enumerable set can be produced by an OS (Section 4.2).

Doing only theory up to now, I considered this internship as a possibility to go beyond the world of purely theoretical research. The Oritatami model is based on wet lab experiments and, even working on theoretical questions, we need to understand the underlying real processes in order to move forward. The wet lab experiments which I attended inspired me to contribute to work in this direction and eventually perform experiments myself.

2 Definitions

In this section, we will introduce the Oritatami model. Let us begin by giving the formal mathematical definition of an assembled part of a transcript in a given moment of time. This is what we call a configuration (a state) of our dynamical system.

Let us first introduce the triangular lattice $\mathbb{T}$. By definition, $\mathbb{T} = (\mathbb{Z}^2, \sim)$ where $(x, y) \sim (u, v)$ if and only if $(u, v) \in \bigcup_{\epsilon=\pm 1} \{(x+\epsilon, y), (x, y+\epsilon), (x+\epsilon, y+\epsilon)\}$; $x \sim y$ means that x and y are adjacent.

Given a set of *bead types* B which is a finite alphabet, a sequence of beads $\omega \in B^*$ is called a *transcript*. A *configuration* c is a self-avoiding path on the triangular lattice of length $|\omega|$ labeled by beads of the transcript ω. The i-th position in c is denoted by c_i. A configuration of a prefix of ω is called a *partial configuration*. An example of a set of bead types, a transcript and a partial configuration is given in Figure 2.

The hydrogen bonds between nucleotides are represented by an *attraction rule* $\heartsuit \subseteq B^2$ which is a symmetric equivalence relation on bead types. Any pair of nonconsecutive adjacent beads attracted to each other forms a *bond*. Given an attraction rule $\heartsuit$ and a configuration c, we denote the number of bonds created by beads in c by $h(c) := |\{(i, j) : c_i \sim c_j,\ j > i + 1,\ \omega_i \heartsuit \omega_j\}|$. The number of bonds created by the i-th bead of c is denoted by

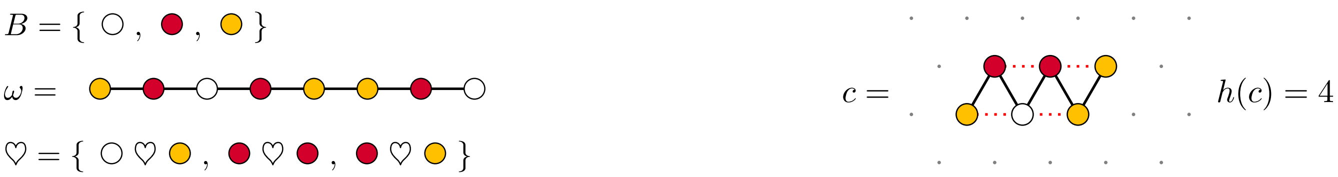

Figure 2: An example of a set B of bead types, a transcript ω, an attraction rule $\heartsuit$, and a partial configuration c whose bonds denoted by red dotted lines.

$h_i(c)$. Find an example of an attraction rule and bonds of a configuration in Figure 2.

Let us define the dynamics of an OS, in other words, how its configuration evolves with time. Let us begin with some intuition about this process. Given a set of beads, a transcript, and an attraction rule, folding starts with an initial configuration called a *seed* which determines the positions of several first beads of the transcript. The forthcoming beads of the transcript are stabilized in their positions one by one: at each step, a fixed number of not yet stabilized beads move freely "looking for" the configurations maximizing the number of bonds. The next bead is stabilized according to these favorable configurations.

Given $\omega \in B^*$ and its partial configuration c, a partial configuration c' of size $|c| + \delta$ is called a δ-*elongation* of c if its first $|c|$ beads are positioned as in c. We denote by $c^{\triangleright\delta}$ the set of all δ-elongations of c. If the transcript is finite and $\delta > |\omega| - |c|$ then δ-elongations of c do not exist since the end of the transcript is too close. In this case, we consider the longest existing elongations: $c^{\triangleright\delta} := c^{\triangleright|\omega|-|c|}$. Given $c' \in c^{\triangleright\delta}$, its last δ beads are called *nascent*. Configuration c' is called a *favorable elongation* if it maximizes the number of bonds among all δ-elongations of c, more formally, if $c' \in \arg\max\limits_{\gamma \in c^{\triangleright\delta}} h(\gamma)$. The set of all elongations and the favorable elongations of a partial configuration are illustrated in Figure 3

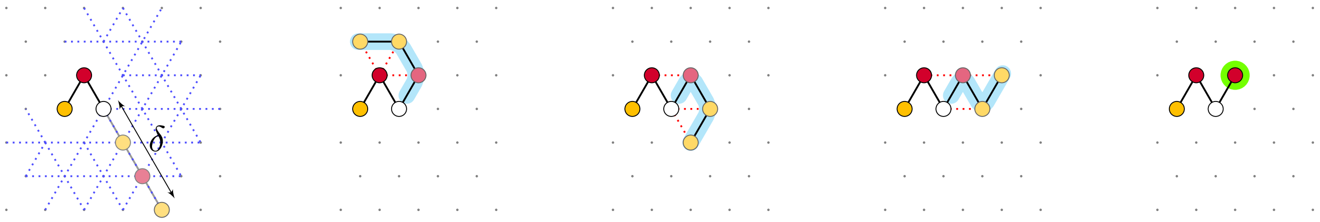

Figure 3: Dynamics of an OS with $\delta = 3$: a partial configuration with the set of its elongations marked by blue dotted lines (on the left), the three favorable elongations whose nascent beads are highlighted in blue (●), and its successor with the newly stabilized bead circled in green (○).

Let us denote by $\mathcal{N}(c)$ the number of bonds produced by the nascent beads of a favorable elongation of c: $\mathcal{N}(c) := \max\limits_{\gamma \in c^{\triangleright\delta}} h(\gamma) - h(c)$.

Let $F(c)$ denote the set of favorable elongations of c, so $F(c) := \arg\max\limits_{\gamma \in c^{\triangleright\delta}} h(\gamma)$.

An *Oritatami system* (*OS*) O is a tuple $(B, \omega, \heartsuit, \delta)$ where B is a set of bead types, ω is a transcript, $\heartsuit$ is an attraction rule and $\delta \in \mathbb{N} \setminus \{0\}$ is a *delay*.

Given an OS O, consider a partial configuration c of length $i < |\omega|$ and one of its 1-elongations c'. We say that c' is the *successor* of c and write $c \vdash c'$ if the first nascent bead of all favorable elongations of c is placed at the same position as the last bead of c'. Figure 3 shows a configuration, its three favorable elongations, and its successor (at the right).

Given an OS O with a seed σ, which is a partial configuration of the transcript, we say that O is *deterministic* if there exists a sequence $(c^i)_{i=0}^{|\omega|-|\sigma|}$ of configurations "folded" by O defined as follows. The initial configuration $c^0 := \sigma$ and for $i > 0$, c^i is the successor of c^{i-1}, i.e., a partial configuration of ω of length $i + |\sigma|$ such that $c^{i-1} \vdash c^i$.

Notice that inside a configuration c^i, the index of a bead is the moment when its final locations is determined. The beads of the seed have thus indices from $1 - |\sigma|$ to 0 and for any $i \geq 0$, $c^i = c^0_{1-|\sigma|}, \ldots, c^i_i$. In this notation, for all i, for any $\tilde{c} \in F(c^i)$, $\tilde{c}_{i+1} = c^{i+1}_{i+1}$. Besides this, given an OS and its configurations, we simplify the notation: $F_i := F(c^i)$ and $\mathcal{N}_i := \mathcal{N}(c^i)$.

Let us define the terminal configuration c^∞ of deterministic Oritatami folding of an OS O. If ω is finite, then let c^∞ be equal to $c^{|\omega|-|\sigma|}$. Otherwise, since for all i, c^i is a prefix of c^{i+1}, c^∞ is defined as the limiting configuration of $(c^n)_{n\in\mathbb{N}}$. We say that *O folds into the configuration c^∞*.

We call a *shape* a subset of points of the triangular grid. Given a configuration c, the *shape of c* is just the set of positions of all beads in c (we forget the bead types and the bonds). Then the *terminal shape* of an OS is the shape of its terminal configuration.

We say that an OS is *nonblocking* if for all i, at least one favorable elongation of c_i can be prolonged (otherwise, the last bead of any elongation is blocked from all sides). This notion was introduced by me and Nicolas Schabanel during my internship to fix an existing proof. In this report, we mostly talk about nonblocking OSs except Section 3.2.3, where we discuss the difference between blocking and nonblocking ones.

3 Geometric properties of Oritatami

Self-assembly was introduced in the first place as a method to fold nanostructures out of molecules. For DNA, there are many theoretic and experimental results showing that it can self-assemble into complex shapes in 2D [19] and in 3D [5, 12]. Here we investigate which structures can be folded by RNA using the Oritatami model.

A purely geometric outcome of self-assembly of any OS is a shape on the triangular lattice. In this section, we investigate how to fold certain shapes and which shapes cannot be assembled at all.

Sections 3.1-3.3, are focused on the limits between the shapes that can be folded and the ones that can not. I got involved in this subject since the beginning of my internship, when I tried to fix an already written reasoning about the shape which could be assembled by a system with a delay δ but not less; you can find it in [8]. I ended up obtaining more general results (Section 3.1) and applying them to self-assembly of a straight line (Section 3.2) and, eventually, to the delay-borderline shape mentioned above (Section 3.3).

In Section 3.4, we consider a large family of shapes which can be assembled by OSs with a small delay and a constant number of bead types. We will see what was done on this subject and how I tried to move it forward: even though no result was obtained yet, it is a work in progress.

3.1 How to see that a shape cannot be assembled

There are shapes that cannot be assembled at all, for instance, non connected ones. Moreover, among connected shapes, infinite shapes with finite cut are also impossible to fold [9]. Eventually, we would like to understand which shapes are foldable and which are not. In order to do this, we shall be able not only to construct systems assembling foldable shapes, but also to prove nonexistence of such systems for unfoldable shapes. In this section, we will develop necessary tools for "unfoldability" proofs.

Consider a deterministic nonblocking OS $O = (B, \omega, \heartsuit, \delta)$. By definition, its folding is determined by configurations $c^0, c^1, \ldots, c^\infty$.

For $j < k < j + \delta$, we say that the step j is *crucial for the k-th bead* if all favorable nascent configurations of c^j place this bead in its correct position in c^∞, while it was not the case at the previous step. More formally, j is crucial for the k-th bead if for all $c' \in F_j$, $c'_k = c^\infty_k$ and there exists $c'' \in F_{j-1}$ such that $c''_k \neq c^\infty_k$. We call a step j *crucial* if there exists k such that $j < k < j+\delta$ and the j-th step is crucial for the k-th bead. Figure 4 illustrates the favorable elongations at step $j - 1$ (on the left) and at step j (on the right) in the case where j is crucial for the k-th bead.

The next claim says that if at step i there is no favorable nascent elongation placing the ϵ-th bead at its correct place, then there is a crucial step between i and $i + \epsilon$.

Claim 1. If for some $\epsilon \leq \delta$, there is $c' \in F_i$, $c'_{i+\epsilon} \neq c^\infty_{i+\epsilon}$ then there exists j, $i < j < i+\epsilon$ such that the j-th step is crucial.

Proof. Since the bead $i+\epsilon$ is stabilized at step $i+\epsilon-1$, for any configuration c'' in $F_{i+\epsilon-1}$, $c''_{i+\epsilon} = c^\infty_{i+\epsilon}$. That means there exists $j \in \{i+1, \ldots, i+\epsilon-1\}$ such that F_j "starts" to consist only of configurations correctly placing the $(i+\epsilon)$-th bead. □

The following lemma gives us upper bounds on the decrease of $\mathcal{N}_i$.

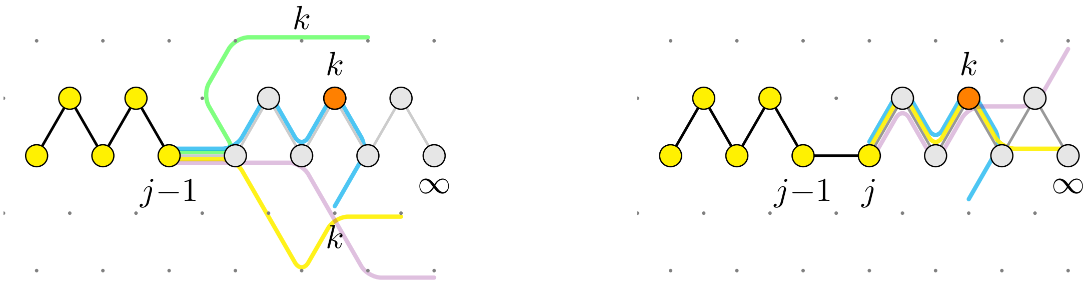

Figure 4: Example of a crucial step j: already placed beads are yellow (●), terminal positions of not yet placed beads are grey (○), nascent beads of favorable elongations are highlighted in different colors, $\delta = 5$.

Lemma 2. For any step i of folding,

(1) $\mathcal{N}_{i-1} \leq \mathcal{N}_i + h_i(c^i)$
(2) if i is crucial, then $\mathcal{N}_{i-1} \leq \mathcal{N}_i + h_i(c^i) - 1$

Proof. Consider any favorable δ-elongation of c^{i-1}, $c' \in F_{i-1}$. In Figure 5, its nascent beads are blue (●). Recall that $c'_i = c^\infty_i$ and thus, any elongation of c' by one bead is a δ-elongation of c^i. Consider such elongation c'', whose nascent beads, $\{c'_{i+1}, \ldots, c'_{i+\delta-1}, c''_{i+\delta}\}$, are circled violet (◉) in Figure 5. Counting the number of nascent bonds created by beads $\{c'_i, c'_{i+1}, \ldots, c'_{i+\delta-1}, c''_{i+\delta}\} = \{c'_i, c''_{i+1}, \ldots, c''_{i+\delta-1}, c''_{i+\delta}\}$, we can consider them as nascent beads of c' together with the bead $c''_{i+\delta}$ or as the bead c'_i together with the nascent beads of c'':

$$\mathcal{N}_{i-1} \leq h(c') - h(c^{i-1}) + h_{i+\delta}(c'') = h_i(c^i) + h(c'') - h(c^{i-1}) \leq h_i(c^i) + \mathcal{N}_i$$

This inequality proves (1).

Furthermore, if the step i is crucial then there exists j, $i < j < i+\delta$ such that for all $\bar{c} \in F_i$, $\bar{c}_j = c^\infty_j$ and there exists $c' \in F_{i-1}$ such that $c'_j \neq c^\infty_j$. We choose such $c' \in F_{i-1}$ and use the reasoning given above to get the following inequality: $\mathcal{N}_{i-1} \leq h_i(c^i) + h(c'') - h(c^{i-1})$.

Since $c''_j = c'_j \neq c^\infty_j$, elongation c'' is not in F_i and thus, $h(c'') - h(c^{i-1}) < \mathcal{N}_i$. Combining the last two inequalities, we get (2):

$$\mathcal{N}_{i-1} < h_i(c^i) + \mathcal{N}_i$$

□

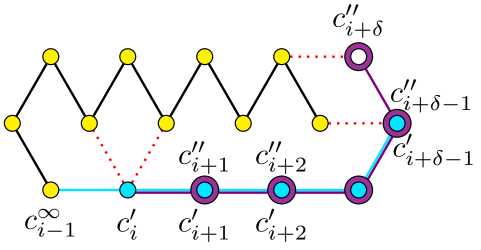

Figure 5: Illustration of Lemma 2 : the nascent beads of c' are blue (●), the nascent beads of c'' are violet (◉).

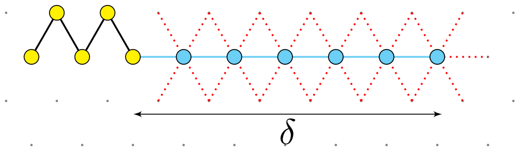

Figure 6: Illustration of Claim 4 : an upper bound on $\mathcal{N}_i$.

Corollary 3. For all i, $\mathcal{N}_i$ is greater or equal to the diference between the number of crucial steps and the number of bonds created during the first i steps:

$$\mathcal{N}_i = \mathcal{N}_0 + \sum_{j=1}^{i} \mathcal{N}_j - \mathcal{N}_{j-1} \geq \sum_{j=1}^{i} \mathbb{1}_{\{\text{j is crucial}\}} - \sum_{j=1}^{i} h_j(c^j)$$

.

The next claim gives us an upper bound on the number of bonds created by a δ-elongation.

Claim 4. For any $i \geq 0$, $\mathcal{N}_i \leq 4\delta + 1$.

Proof. As shown in Figure 6, each of the first $\delta - 1$ beads of a δ-elongation can create at most 4 bonds since two of 6 its neighbors are its predecessor and its successor in the elongation. The last bead can create at most 5 bonds since it has no successor. This gives us $\mathcal{N}_i \leq 4(\delta - 1) + 5 = 4\delta + 1$. □

Corollary 3, together with Claim 4, give us a strong constraint on any OS: the number of crucial steps can never exceed the number of created bonds by more than $4\delta + 1$. Thus, to prove that a shape can not be assembled, it is enough to show that any OS folding this shape would have too many crucial steps and would not create enough bonds. Let us now use these tools to obtain results about foldability of certain shapes.

3.2 Straight line

In this section, given a seed, which is a straight segment, and a delay, we try to extend the line as far as possible. More precisely, we investigate the maximal length of a straight line which can be folded by an OS with the seed σ and delay δ.

3.2.1 A nonblocking OS cannot fold too long straight lines

Let us suppose that a line of length $|\sigma| + N$ is foldable. Consider a nonblocking OS $O_N = (B, \omega, \heartsuit, \delta)$ such that the shape of its N-th configuration c^N is a line of length $|\sigma| + N$. In this part, we will find an upper bound on N. This kind of bound (though, a greater one) was already discussed by Jacob Hendricks and Matthew J. Patitz, but here we present a different proof.

Let us notice that since O_N folds into a line, at each step $i \leq N$, the i-th bead creates no bonds: $h_i(c^i) = 0$. Let $d = \lceil \frac{\delta}{2} \rceil$. The following lemma shows that during folding of a line, there is at least one crucial step at each segment of length d.

Lemma 5. For all $k \in \{0, 1, \dots, \lfloor \frac{N}{d} \rfloor - 1\}$, there exists $i_k \in \{kd + 1, \dots, kd + d\}$ such that i_k is a crucial step.

Proof. It is enough to prove that at any step i of folding, all favorable nascent elongations place the $(i + d + 1)$-th bead at an incorrect position. Indeed, after this, by Claim 1, we obtain the result.

Suppose by contradiction that for some i, $0 \leq i \leq N{-}d{-}1$, for any $c' \in F_i$, $c'_{i+d+1} = c^N_{i+d+1}$. Since c^N is a straight line, it follows that for all k between i and $i{+}d{+}1$, $c'_k = c^N_k$. Let us now consider a δ-elongation $\tilde{c}$ which is obtained by rotating the nascent part of c' around c'_i by $\frac{\pi}{3}$ clockwise. The elongations c' (nascent beads are marked violet: ●) and $\tilde{c}$ (nascent beads are marked blue: ●) are shown in Figures 7 and 8 respectively. This action creates no intersections and does not change the number of bonds since the first $d{+}1$ nascent beads of c' are at the line and the last $\delta{-}d{-}1$ beads are contained in a ball with radius d with center in c'_{i+d+1} (since $\delta{-}d{-}1 = \lfloor \frac{\delta}{2} \rfloor {-}1 < d$). The ball (filled by pink in Figures 7 and 8) contains no neighbors of the configuration c^i (thus, c' has no bonds with c^i), and rotations of this ball around c'_i by $\frac{\pi}{3}$ do not intersect c^i. Thus, $\tilde{c}$ is also a favorable elongation of c^i which leads to contradiction since $\tilde{c}_{i+1} \neq c'_{i+1}$. □

Corollary 6. The number of crucial steps among steps from 1 to N is greater or equal to $\lfloor \frac{N}{d} \rfloor$.

The next lemma gives an upper bound on the length of a foldable straight line.

Lemma 7. Given a seed σ and a delay δ, there is no nonblocking OS assembling a straight line of length greater than $|\sigma| + 2\delta^2 + 5\delta + 2$.

Proof. Since no bonds are created during folding, by Corollary 3 and Corollary 6,

$$\mathcal{N}_N \geq \sum_{i=1}^{N} \mathbb{1}_{\{\text{i is crucial}\}} - h_i(c^i) = \sum_{i=1}^{N} \mathbb{1}_{\{\text{i is crucial}\}} \geq \left\lfloor \frac{N}{d} \right\rfloor$$

.

Then, using Claim 4, we obtain $\left\lfloor \frac{N}{d} \right\rfloor \leq \mathcal{N}_n - \mathcal{N}_0 < \mathcal{N}_n \leq 4\delta + 1$. Therefore, we get an upper bound on the line length: $N \leq \lceil \frac{\delta}{2} \rceil (4\delta + 2) \leq 2\delta^2 + 5\delta + 2$. □

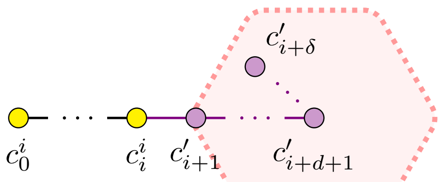

Figure 7: Elongation c' from Lemma 5.

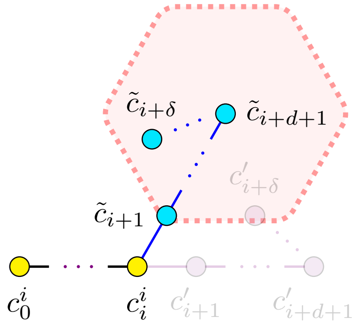

Figure 8: Elongation $\tilde{c}$ from Lemma 5.

3.2.2 A nonblocking OS folding a not-so-long straight line

Now we know that straight lines of length greater than $2\delta^2 + 5\delta + 2$ cannot be folded by nonblocking OSs. Then which lines can be folded? It turns out that for $\delta \geq 9$, there is an OS that assembles deterministically into a line of length $2\lfloor \frac{\delta}{3} \rfloor - 1$ from a seed of length at most 4, and becomes nondeterministic just after. The OS illustrated in Figures 9 and 10 does this in the case where δ is a multiple of 3 ($\delta = 3d$). The cases where $\delta = 3d+1$ and $\delta = 3d+2$ are similar, find them in Figures 11-14.

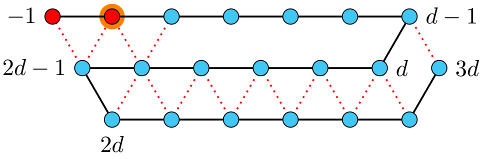

Figure 9: Line folding at step 0 for $\delta = 3d$: the seed is red (●), nascent beads are blue (●)

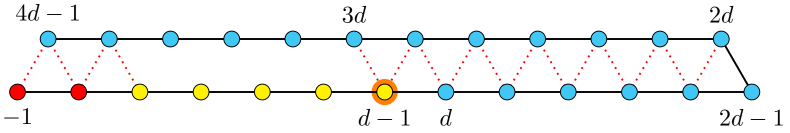

Figure 10: Line folding at step $d-1$ for $\delta = 3d$: the seed is red (●), folded beads are yellow (●), nascent beads are blue (●)

The folding of this OS proceeds as follows. During the first $d-1$ steps, the first $3d$ beads of all favorable elongations are placed as in Figure 9 (or symmetrically). Thus, at step $d-1$, the first $d-1$ beads create a straight segment (yellow beads in Figure 10). Besides that, starting from the step $d-1$, the first $4d-1$ beads of all favorable elongations are placed as in Figure 10 (or symmetrically). Therefore, at step $2d-1$, the first $2d-1$ beads form a straight line, and that is precisely what we need.

As you see, the step $d-1$ is crucial: at this moment, the favorable elongations do not form three layers as in Figure 9 anymore but pass to two layers as in Figure 10 which guarantees that the next d beads also form a straight segment.

However, it is yet unknown if we can fold segments of length strictly greater than δ by a nonblocking OS. Besides that, since the upper bound from the previous section is quadratic, there is a huge gap to fill between foldable and unfoldable straight lines.

3.2.3 A blocking OS that easily folds into an infinite line

Initially, we did not see that some OSs are blocking, so we worked with nonblocking ones without precising it in the definitions. Our first blocking OS was inspired by videos about shrimps and chameleons: this OS looks a bit like a chameleon's tongue or a party horn (see Figure 15). In this section, we will describe this chameleon OS which we discovered with Nicolas Schabanel.

In the previous section we saw that any OS folding a line has a crucial step each d steps. A crucial step means breaking links from ancient favorable elongation to create new, "more interesting", favorable elongations. For a nonblocking OS, ancient elongations can be rejected only when new elongations create strictly more bonds, and in

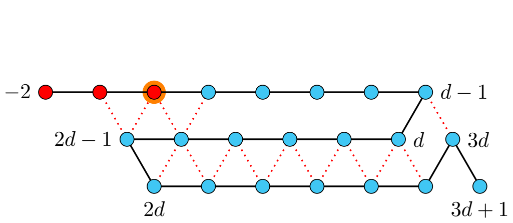

Figure 11: Line folding at step 0 for $\delta = 3d + 2$: the seed is red (●), nascent beads are blue (●)

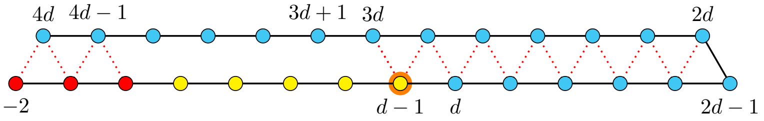

Figure 12: Line folding at step $d-1$ for $\delta = 3d+2$: the seed is red (●), folded beads are yellow (●), nascent beads are blue (●)

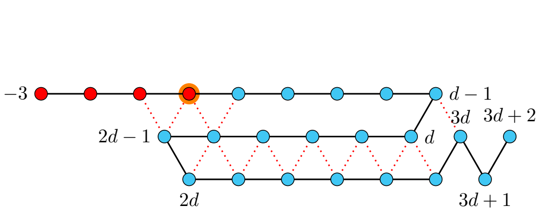

Figure 13: Line folding at step 0 for $\delta = 3d+2$: the seed is red (●), nascent beads are blue (●)

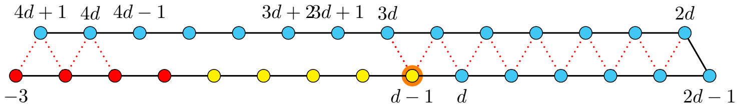

Figure 14: Line folding at step $d-1$ for $\delta = 3d+2$: the seed is red (●), folded beads are yellow (●), nascent beads are blue (●)

this case, the number of bonds $\mathcal{N}_i$ grows and finally reaches the upper bound which makes it impossible to fold too long lines. Blocking OSs, in contrast, can get rid of ancient elongations just by blocking them geometrically: if no favorable elongation can be prolonged, then the new ones will appear without increasing the number of bonds. That is precisely what the chameleon OS does.

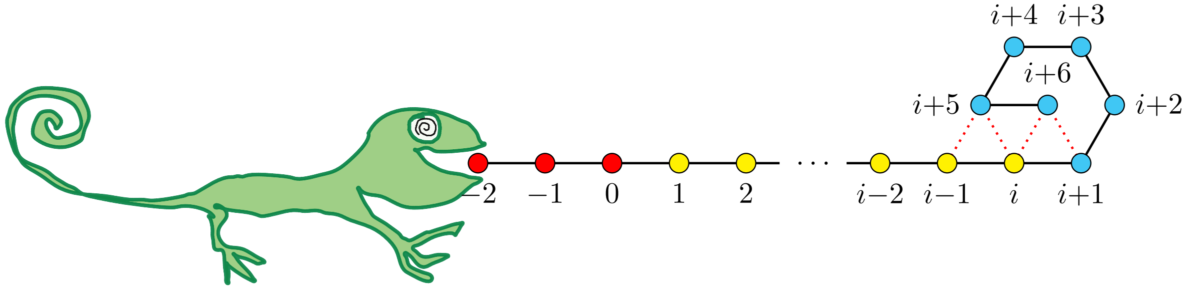

Figure 15: Folding of the chameleon OS: the seed is red (●), folded beads are yellow (●), nascent beads of a favorable elongation are blue (●).

The chameleon OS uses 18 beads $B := \{b_0, \ldots, b_{17}\}$ and the attraction rule $\heartsuit$ is such that for all $0 \le i \le 17$, we have $b_i \heartsuit b_{i+5 \bmod 18}$ and $b_i \heartsuit b_{i+6 \bmod 18}$. The transcript is just a periodic sequence: $\omega = (b_0 b_1 \ldots b_{17})^{\mathbb{N}}$, and the seed $c^0 = b_0 b_1 b_2$. Finally, the delay of the system $\delta = 6$.

It turns out, that at each step of folding, there are two symmetrically equivalent favorable elongations which are blocking: they cannot be prolonged (nascent beads of a favorable elongation are marked blue in Figure 15). Since both of them stabilize the next bead at its terminal position, the folding is correct and results in a straight infinite half-line.

3.3 Shape S_δ^∞

Delay is an important parameter of an OS: intuitively, it is how far the not yet stabilized part of the sequence can "see". This raises a question of how the power of an OS changes with delay. Is it monotonous? Are systems with greater delay more powerful? In this section, we describe a "δ-borderline" shape for which there is an OS with delay δ assembling it, but there is no OS with lesser delay folding into this shape. This example confirms that the power of an OS might increase strictly with delay.

The results from this section are initially based on Section 6 from [8]. I made definitions more precise, added a missing constraint which is that an OS should be nonblocking, changed the design of a shape, and simplified all proofs.

For $\delta > 4$, let S_δ^∞ denote the shape shown in Figure 17, its construction is described below. First, in Section 3.3.1, we prove that this shape is foldable by a nonblocking OS with delay δ. In Section 3.3.2, we show that no nonblocking OS with lesser delay can assemble it.

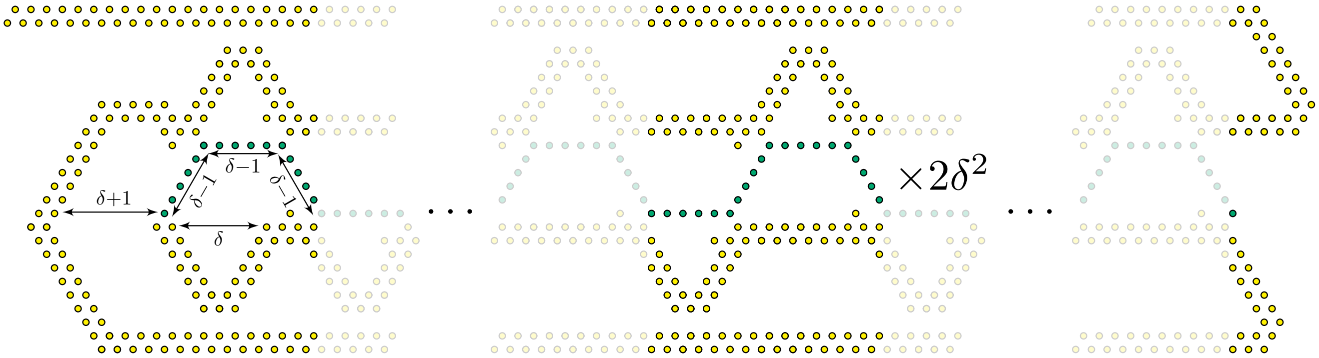

Figure 16: The shape S_δ^0 (green and yellow beads: ● ●) and the snake L^0 (green beads: ●) for $\delta = 6$.

The shape S_δ^0 for $\delta = 6$ is given in Figure 16. Notice that S_δ^0 is a concatenation of a thick "path" of width two and a thin "path" of width one. The latter one, colored in green in Figure 16, is called the *snake* and is denoted by L^0. The central part of L^0 is periodic and consists of a "bend" of size $4(\delta-1)$ beads repeated $2\delta^2$ times. Taking into account the beginning and the end of L^0, its size is equal to $M := |L^0| = 8(\delta-1)\delta^2 + 3\delta - 2$.

We denote by S_δ^∞ the infinite union of translated copies of S_δ^0 shown in Figure 17. More precisely, $S_\delta^\infty = \bigcup\limits_{n\in\mathbb{N}} S_\delta^n$ where $S_\delta^n := S_\delta + pn$ and p is the period vector, connecting the end of a shape with the beginning of the next one. Let L^n denote the snake of S_δ^n. The snake, walking back and forth between two yellow walls, is crucial in this construction: it is the reason why this shape cannot be folded with delay less than δ. The snake is also the trickier part to fold with delay δ.

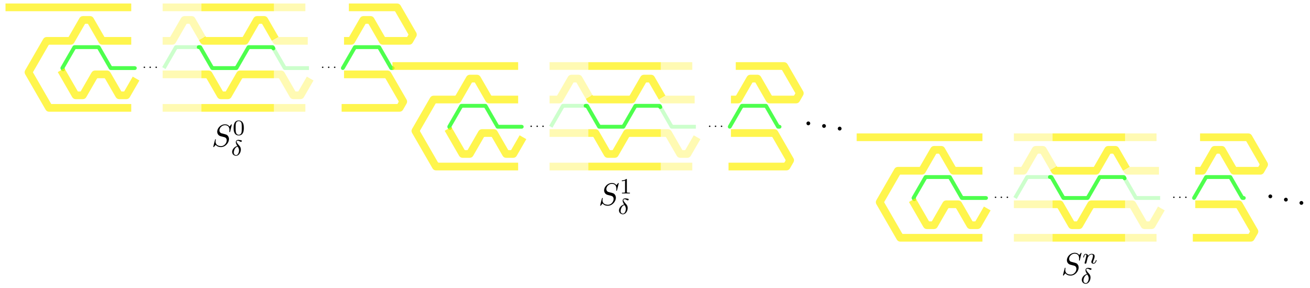

Figure 17: The shape S_δ^∞.

3.3.1 An OS that folds S_δ^∞

In this section, we prove that the shape S_δ^∞ is foldable with delay δ. To show this, we construct a nonblocking OS $O_\delta = (B_\delta, \omega_\delta, \heartsuit, \delta)$ folding into S_δ^∞ with the seed σ which consists of three beads marked red (●) in Figure 18.

The transcript, $\omega_\delta = (\omega_\delta^0)^{\mathbb{N}}$, is a repeated transcript of S_δ^0: the n-th copy of ω_δ^0 folds into the shape S_δ^n. Figure 18 represents the result of folding of ω_δ^0, the shape S_δ^0. For the sake of simplicity, we split ω_δ^0 in several parts called *modules* which are marked with different colors and labeled by capital letters in Figure 18. By definition, $\omega_\delta^0 = AB(CD)^{2\delta^2+1}EA'F(D\tilde{C})^{2\delta^2+1}GH$.

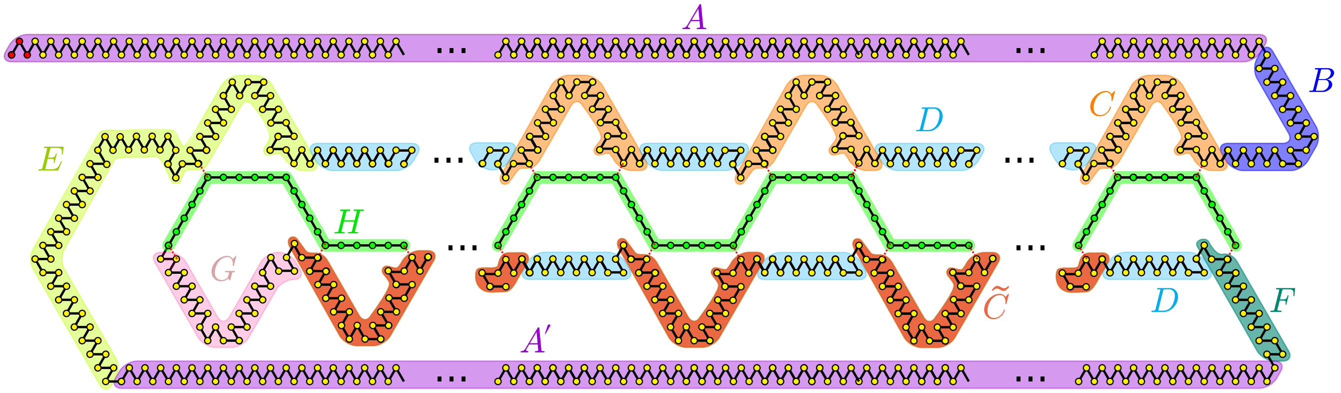

Figure 18: The partition of the shape S_δ^0 for $\delta = 6$.

Before we move forward to the detailed description of the modules, let us notice that the bead types in modules denoted by different letters are different. Besides that, A, A' and H are made of repeated sequences, while all the other modules consist of distinct beads. More precisely, let $|A'| = m' := 8\delta^3 + 12\delta^2 + 4\delta + 8$ and $|A| = m := m' + (2\delta + 4)$. Let k and l satisfy $0 \leq l < 2\delta + 2$ and $m' = k(2\delta + 2) + l$. Then $A' = (a_1 \dots a_{2\delta+2})^k a_1 \dots a_l$ and $A = a_{-1}a_0a_1 \dots a_{2\delta+2}A'$; $H = h_0(h_1h_2^{\delta-3}h_3)^{8\delta^2+3}$.

Let us now totaly describe the modules together with the relation $\heartsuit$: Table 19 gives the sequence of beads for each module and defines the relation $\heartsuit$ on the bead types inside each modul (see the third column of Table 19). To define $\heartsuit$ completely, besides the relations of beads inside each module, we need to give the relations between the beads of distinct modules (see Table 20). Bonds inside modules and between modules are illustrated in Figures 21 - 24. Notice that since A and A' consists of the same beads we merge them in Table 20; we also merge C and $\tilde{C}$ since their beads have the same relations with beads from other modules. Up to now, we described the bonds inside the shape S_i^δ, besides that, we shall take into account its connection with the next shape. There are three bonds between the modules H, F of S_i^δ and the first two beads of the module A of S_{i+1}^δ: they are the only bonds between modules of distinct shapes.

module	sequence	$\heartsuit$ inside the module
A	$a_{-1}a_0(a_1 \dots a_{2\delta+2})^{k+1}a_1 \dots a_l$	$\{a_i \heartsuit a_{i+2} \mid -1 \leq i \leq 2\delta\} \cup \{a_{2\delta+1} \heartsuit a_1, a_{2\delta+2} \heartsuit a_2\}$
B	$b_1 \dots b_{4\delta}$	$\{b_i \heartsuit b_{i+2} \mid 1 \leq i \leq 4\delta - 2,\ i \neq 2\delta+1, 2\delta+4\} \cup \{b_{2\delta} \heartsuit b_{2\delta+3}, b_{2\delta+3} \heartsuit b_{2\delta+6}\}$
C	$c_1 \dots c_{4\delta+5}$	$\{c_i \heartsuit c_{i+2} \mid 1 \leq i \leq 4\delta+2,\ i \neq 4,5,7,2\delta+2,2\delta+5\} \cup \{c_2 \heartsuit c_7,\ c_4 \heartsuit c_7,\ c_6 \heartsuit c_9,\ c_{2\delta+1} \heartsuit c_{2\delta+4},\ c_{2\delta+4} \heartsuit c_{2\delta+7}\}$
$\tilde{C}$	$\tilde{c_1} \dots \tilde{c}_{4\delta+5}$	as for C
D	$d_1 \dots d_{2\delta+1}$	$\{d_i \heartsuit d_{i+2} \mid 1 \leq i \leq 2\delta - 1, i \neq 2\} \cup \{d_1 \heartsuit d_4\}$
E	$e_1 \dots e_{10\delta+10}$	$\{e_i \heartsuit e_{i+2} \mid 1 \leq i \leq 10\delta+8,\ i \neq 4,5,7,2\delta+2,2\delta+5,4\delta+5,6\delta+4,8\delta+4\} \cup \{e_2, e_4 \heartsuit e_7, e_6 \heartsuit e_9,\ e_{2\delta+1} \heartsuit e_{2\delta+4},\ e_{2\delta+4} \heartsuit e_{2\delta+7},\ e_{4\delta+4} \heartsuit e_{4\delta+7},\ e_{6\delta+3} \heartsuit e_{6\delta+6},\ e_{8\delta+8} \heartsuit e_{8\delta+11}\}$
A'	$(a_1 \dots a_{2\delta+2})^k a_1 \dots a_l$	as for A
F	$f_1 \dots f_{2\delta+5}$	$\{f_i \heartsuit f_{i+2} \mid 1 \leq i \leq 2\delta+2,\ i \neq 2\delta\} \cup \{f_{2\delta} \heartsuit f_{2\delta+3}\}$
G	$g_1 \dots g_{4\delta+4}$	$\{g_i \heartsuit g_{i+2} \mid 1 \leq i \leq 4\delta+2,\ i \neq 2,4,2\delta+2\} \cup \{g_1 \heartsuit g_4,\ g_3 \heartsuit g_6,\ g_{2\delta+1} \heartsuit g_{2\delta+5}\}$
H	$h_0(h_1h_2^{\delta-3}h_3)^{8\delta^2+3}$	

Table 19: Transcripts of the modules and the $\heartsuit$ rule inside modules.

As all parameters of O_δ are defined, we will now prove that it is a deteministic nonblocking OS whose terminal configuration is equal to S_δ^∞.

Given a deterministic OS and its configuration c^i, we say that c^i is *straightforward* if it has only one favorable

	A A'	B	C $\tilde{C}$	D	E	F	G	H
A A'		$a_l\heartsuit b_1$, $a_l\heartsuit b_2$				$a_{l-2}\heartsuit f_2$, $a_l\heartsuit f_2$		
B			$b_{4\delta-1}\heartsuit c_1$, $b_{4\delta}\heartsuit c_2$					
C $\tilde{C}$				$c_{4\delta+2}\heartsuit d_2$, $c_{4\delta+4}\heartsuit d_1$, $c_{4\delta+4}\heartsuit d_2$			$c_{4\delta+2}\heartsuit g_2$, $c_{4\delta+4}\heartsuit g_1$, $c_{4\delta+4}\heartsuit g_2$	
D			$d_{2\delta}\heartsuit c_1$, $d_{2\delta+1}\heartsuit c_2$		$d_{2\delta}\heartsuit e_1$,$d_{2\delta+1}\heartsuit e_2$			
E	$e_{10\delta+9}\heartsuit a_1$ $e_{10\delta+9}\heartsuit a_2$							
F	$f_{2\delta+1}\heartsuit a_{-1}$			$f_{2\delta+2}\heartsuit d_1$, $f_{2\delta+4}\heartsuit d_1$, $f_{2\delta+4}\heartsuit d_2$				
G								$g_{4\delta+3}\heartsuit h_0$
H	$h_3\heartsuit a_0$		$h_1\heartsuit c_{4\delta+4}$, $h_1\heartsuit c_4$, $h_3\heartsuit c_{4\delta+3}$, $h_3\heartsuit c_3$		$h_1\heartsuit e_{4\delta+4}$, $h_1\heartsuit e_4$, $h_3\heartsuit e_{4\delta+3}$, $h_3\heartsuit e_3$	$h_3\heartsuit f_{2\delta+3}$		

Table 20: The $\heartsuit$ rules between modules.

elongation equal to $c^{i+\delta}$: $F_i = \{c^{i+\delta}\}$. This means that all beads are already at their treminal positions. It turns out that almost at each step i of folding of O_δ, c^i is straightforward. It happens because every bead can create only the bonds that are present in the terminal configuration, so there is no ambiguity during folding. Let us explain it in details.

Let P denote the set of indices in ω_δ of the beads $\{c_5, c_6, c_{4\delta+5}, d_1, e_5, e_6, e_{4\delta+4}, e_{4\delta+5}, f_{2\delta+5}\}$, they are violet in Figure 21. Let $P^{-\delta} = \{i - \delta |\, i \in P\}$ denote the steps at which the last bead of a δ-elongation is in P.

Claim 8. During folding of all modules except G and H, at each step $i \notin P^{-\delta}$, c^i is straightforward.

Proof. Indeed, at each step i, each bead in the elongation $c^{i+\delta}$, placing all beads in their terminal positions, creates all bonds it can create (we built our system in such a way that any bead of an elongation can be attracted only be the beads with which it creates bonds in the terminal configuration). Moreover, for all these steps, if all the first $i-1$ beads are stabilized, only the terminal position of the i-th bead permits it to create all its bonds. The bonds of the central part of the shape are shown in Figure 21. The bonds and certain beads of the left part of S^0_δ are shown in Figure 23. Figure 24 shows the central repeated part of the shape in details. Finally, the right part of S^0_δ is illustrated by Figure 22. □

If a step i is from $P^{-\delta}$ then the last bead of its δ-elongation is one of the beads filled with violet in Figure 21: the positions of these beads are not directly determined by their predecessors. However all the other beads are stabilized: all favorable elongations agree on the first $\delta - 2$ nascent beads, for the same reason as previously. Thus, since $\delta > 4$, the folding of all the parts specified in Claim 8 is correct. Let us now prove that modules G and H are also correctly folded. All the beads of G and the first bead of H (h_0) are determined by the previous beads, whereas the next $\delta - 3$ beads of H create no bonds at all.

Let us show that H is folded correctly. Since H and its neighborhood are periodic, we only need to show it for the period. A part of folded H is given in Figure 24. Suppose we are at step i and the last stabilized bead was h_3 (it is in the corner of the snake). The only beads of a δ-elongation that can create bonds are at positions $i+1, i+\delta-1, i+\delta$. Their types are h_1, h_3, h_1 respectively. No elongation can create all these 3 bonds, and there is a single elongation creating two bonds with the beads $i+\delta-1$ and $i+\delta$. In this elongation, all beads except the last one are placed in their final positions. During the next $\delta - 2$ steps, nothing changes: no other beads create bonds in the elongation, and all beads until $i+\delta-1$ are at their final places. At step $i+\delta-2$, the last bead of the elongation can create a bond, but it changes nothing. At step $i+\delta-1$, we have just stabilized the corner bead, so we are in the situation where we started and all beads are at their correct terminal positions.

The only remaining part to treat is the connection between module the module H and the module A of the next copy of S^0_δ. Figure 22 illustrates the bonds between last beads of H and first beads of A. Let t be such that the last bead of c^t is the right top corner bead of the snake in the Figure 22 (its type is h_3). At this step, there is only one favorable elongation creating two bonds, they are highlighted in orange; this elongation coincides with the terminal configuration. For all the next steps until the end of the module H, the situation is the same: only the prefix of the terminal configuration creates all possible bonds. Thus, this part is also correctly folded. This concludes our proof that the shape S_δ^∞ is foldable by the nonblocking OS O_δ.

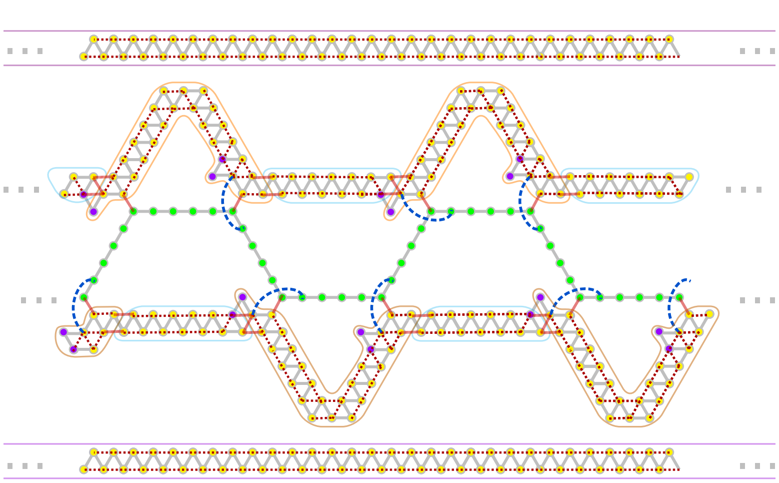

Figure 21: Central part of the terminal configuration of S_δ^0: bonds which are not present in the terminal configuration are blue and dashed, others are red and dotted.

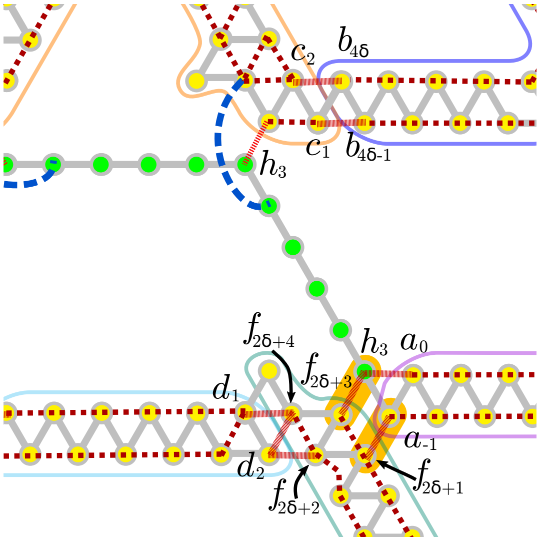

Figure 22: Bonds between the module H of S_δ^n and the module A of S_δ^{n+1}.

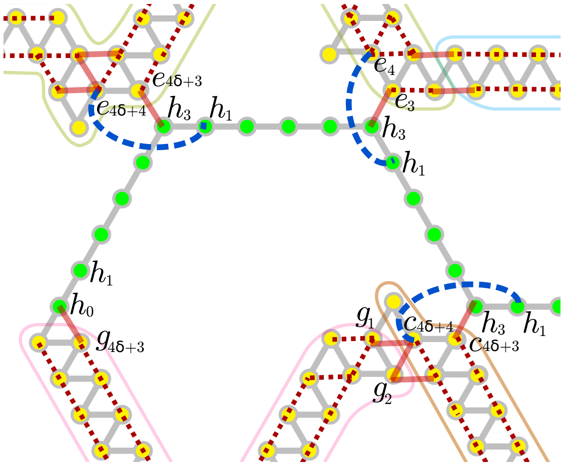

Figure 23: Bonds in the left part of S_δ^0, modules $D, \tilde{C}, G, H$, and E.

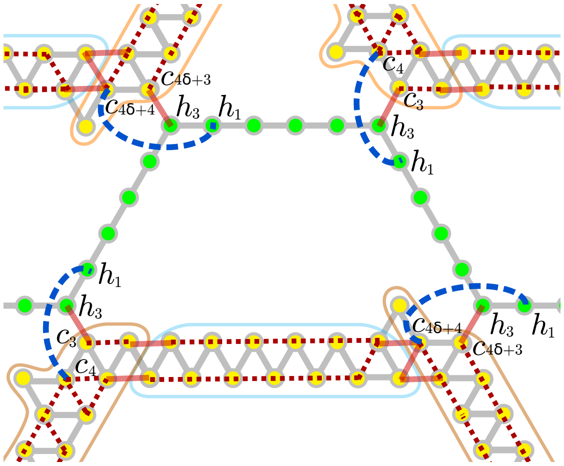

Figure 24: Bonds in the central repeated part of S_δ^0, modules $C, D, \tilde{C}$, and H.

3.3.2 S_δ^∞ cannot be folded with delay less than δ

In this section, we prove that there is no OS with delay less than δ which would fold into S_δ^∞. The idea of the proof is similar to the one about a straight line from Section 3.2.1: we show that an OS folding S_δ^∞ would not create enough bonds to compensate the number of crucial steps. This happens during folding of the snake which has a small number of neighbors and thus, not enough potential bonds.

Reasoning by contradiction, consider $\delta' < \delta$ and a nonblocking OS $O = (B, \omega, \heartsuit, \delta')$ with seed σ which assembles into S_δ^∞. Thus, the set of positions occupied by beads in c^∞ is equal to S_δ^∞.

The following lemma shows that during folding, the snake in one of the copies of S^0_δ is constructed contiguously: the beads marked green in Figure 16 are added one by one, with no yellow beads between them.

Lemma 9. There exists n such that the beads of L^n form a contiguous subsequence of c^∞.

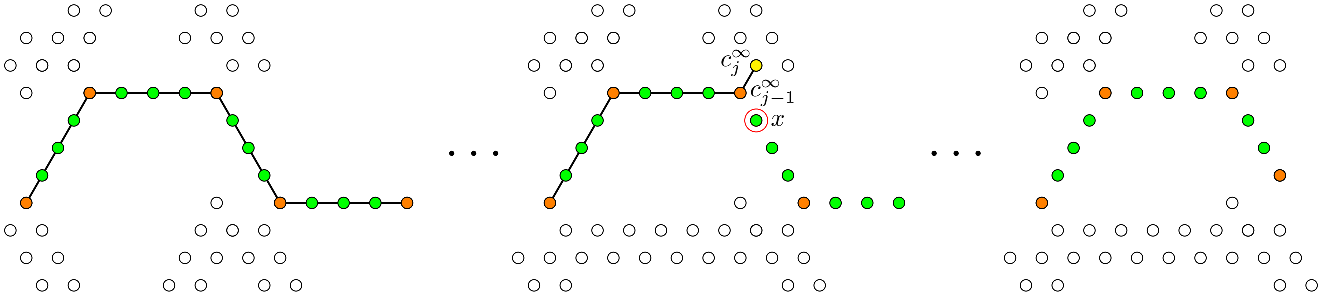

Figure 25: Illustration for Lemma 9 for $\delta = 5$: the snake beads are marked green and orange (●, ●), corners are orange (●), the first bead outside the snake is yellow (●), the bead x is circled in red (○).

Proof. First notice that there exists n such that L^n contains no beads from the seed σ. This holds since $|\sigma|$ is finite and there are infinitely many snakes in S^∞_δ. We will now prove that the snake L^n form a contiguous sequence in the resulting configuration. Suppose by contradiction that there exists a bead not in L^n which appears in c^∞ between a pair of beads from L^n. Consider the first such bead, denote its position by j. By definition, $c^\infty_j \notin L^n$, $c^\infty_{j-1} \in L^n$ and the distance between c^∞_{j-1} and c^∞_j is 1. The only points of L^n with neighbors in $S^\infty_\delta \setminus L^n$ are the corners (marked orange in Figure 25), so c^∞_{j-1} is a corner of L^n, and it is not the last one. As a corner, c^∞_{j-1} has three neighbors from S^∞_δ, one of which, call it x (circled in red in Figure 25), is from L_n and does not appear in c^∞_j.

Since the folding is infinite, each bead from S^∞_δ, except beads from the seed, has a predecessor and a successor in the folding sequence. The bead x has only two neighbors in S^∞_δ and one of them, which is c^∞_{j-1}, is not its predecessor (since it is the predecessor of another bead $c^\infty_j \notin L_n$) nor successor (since it is stabilized before x) which leads to contradiction.

□

Let us fix n from Lemma 9: beads of the snake L^n form a contiguous subsequence in c^∞ and none of them is from the seed. We denote by l the index of the first element of this subsequence, so $L^n = \{c^\infty_l, c^\infty_{l+1}, \ldots, c^\infty_{l+M-1}\}$.

From now on we will only consider S^n_δ, forgetting about the rest of the shape S^∞_δ. The only reason why the shape in our construction is infinite is that in the last proof we need a part which would not contain beads from the seed. We could attain this by other means, for instance, by fixing the size of the seed $|\sigma|$ and considering the shape $\bigcup_{n=0}^{|\sigma|} S^n_\delta$. We use the variant with an infinite number of copies to make the shape independent of the seed size.

L^n consists of sequences of beads of length $\delta-1$ whose first beads, called *corners*, constitute the set $\{c^\infty_{l+k(\delta-1)} \mid 0 \leq k \leq 4\delta^2+3\}$. The corners are marked orange in Figures 25 and 26. We can consider L^n as a concatenation of those $(\delta-1)$-sequences: $L^n = \bigcup_{k=0}^{4\delta^2+2} \{c^\infty_{l+k(\delta-1)}, \ldots, c^\infty_{l+k(\delta-1)+\delta-2}\} \cup \{c^\infty_{l+(4\delta^2+3)(\delta-1)}\}$.

The following lemma shows that if $\mathcal{N}_i > 1$ then there is at least one crucial step among the next $(\delta'-1)$ beads of L^n.

Lemma 10. For $i \in \{l, l+1, \ldots, l+M-\delta'\}$, if $\mathcal{N}_i > 1$ then there is j, $i < j < i+\delta'$ such that the j-th step is crucial.

Proof. It is enough to prove that if $\mathcal{N}_i > 1$ then, for some $\epsilon \leq \delta'$, at least one favorable nascent elongation places the $(i+\epsilon)$-th bead at an incorrect position: there is $c' \in F_i$, $c'_{i+\epsilon} \neq c^\infty_{i+\epsilon}$. Indeed, after this, by Claim 1, we obtain the result.

Suppose by contradiction that for all $\epsilon \leq \delta'$, for all $c' \in F_i$, $c'_{i+\epsilon} = c^{\infty}_{i+\epsilon}$. This means that F_i consists of a single elongation c', equal to $c^{i+\delta'}$. Notice that there is at most one corner bead among the nascent beads of c' since $\delta' < \delta$. Only corner beads can create bonds, so we get that $\mathcal{N}(c') \leq 1 < \mathcal{N}_i$ which leads to contradiction. □

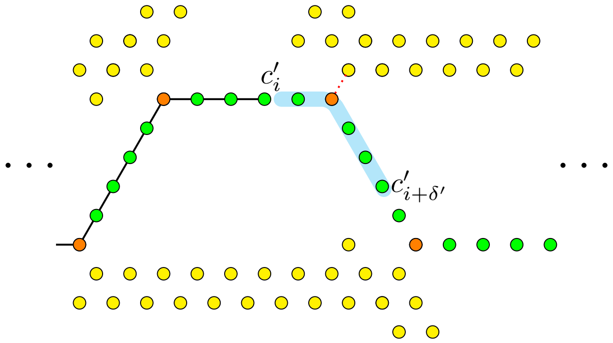

Figure 26: Illustration for Lemma 10 for $\delta = 6$ and $\delta' = 5$: the nascent part of c' is highlighted in blue (●).

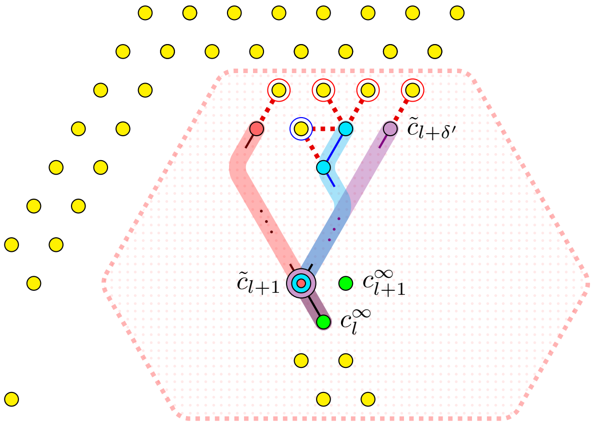

Figure 27: Elongations $\tilde{c}$ from Lemma 11 in the case where the single bond of c' is with some yellow bead at the top.

Lemma 11. At the beginning of the snake L^n, the number of nascent bonds is greater than one: $\mathcal{N}_l > 1$.

Proof. All configurations in F_l stabilize the bead $l+1$ at position c^{∞}_{l+1}. Therefore, $\mathcal{N}_l > 0$: otherwise any elongation would be favorable and the bead $l+1$ would not be stabilized. Suppose by contradiction that $\mathcal{N}_l = 1$ which means that the nascent beads of a favorable elongation of c^l create a single bond. Consider a favorable elongation $c' \in F_l$; the single nascent bond of c' is formed either by a nascent bead and a bead from c^l or by a pair of nascent beads.

In the first case, notice that the single bead from c^l creating a bond with a nascent part of some elongation of c^l is among the 5 beads at the top and a diagonal row of width 2 at the bottom. The area, accessible by a δ'-elongation of c^l whose first nascent bead is in c^{∞}_{l+1}, is covered with pink dots in Figure 27.

Suppose that the single bond is made with one of the five top yellow beads, circled in red (○) and blue (○) in Figure 27. Consider three δ'-elongations $\tilde{c}$ whose nascent beads are marked red (●), blue (●), and violet (●) in Figure 27. In the case where the bond is with the 4 top-most beads (circled in red in Figure 27) it was created by the last bead of the elongation, and one of the elongations $\tilde{c}$ also creates this bond. If the bond is with the remaining top bead (circled blue) then it could be also created by the bead before the last, and the blue elongation $\tilde{c}$ also creates this. Thus, in any case, one of these elongations is favorable. This leads to contradiction since $\tilde{c}_{l+1} \neq c^{\infty}_{l+1}$.

Suppose now that the single bond is with the yellow beads in the bottom. As shown in Figure 28, we denote the right beads of the row by $r_1, \ldots, r_{\delta'}$, starting from the top; the left beads are similarly denoted by $l_1, \ldots, l_{\delta'}$. The single nascent bond is either to the left or to the right of the yellow beads, so it is either between l_i and c'_{l+j} or between r_i and c'_{l+j} for some $1 \leq i, j \leq \delta'$. Notice that in the first case, either $i = j = 1$ or $j \geq i+1$ since the shortest path between c^{∞}_{l} and the neighborhood of l_i is of length $i+1$ for $i > 1$. In the case of r_j, for the same reason, $j \geq i$.

Suppose there is a bond between l_i and c'_{l+j}. Consider the δ'-elongation $\tilde{c}$ whose nascent beads are marked blue (●) in Figure 28. The first nascent beads $(\tilde{c}_l, \tilde{c}_{l+1}, \ldots, \tilde{c}_{l+i})$ just follow the left part of a yellow row. The beads $(\tilde{c}_{l+i}, \ldots, \tilde{c}_{l+\lceil\frac{i+j-1}{2}\rceil})$ form an horizontal line from right to left. The bead $\tilde{c}_{l+\lceil\frac{i+j+1}{2}\rceil}$ goes to the south-west if $j-i$ is even and to the south-east otherwise; the next beads $(\tilde{c}_{l+\lceil\frac{i+j+1}{2}\rceil}, \ldots, \tilde{c}_{l+j})$ form an horizontal line from left to right, below the previous one. The last $\delta' - j$ beads form a south-east line. Notice that by definition of S^{∞}_{δ}, at step l, all cells of the top half of the $(\delta'+1)$-ball around c^{∞}_{l} are empty, so this elongation will not intersect or touch beads from c^l. Whether $j-i$ is even or not, choosing the position of $\tilde{c}_{l+\lceil\frac{i+j+1}{2}\rceil}$, we guarantee that $\tilde{c}_{l+j}$ is the west neighbor of $\tilde{c}_{l+i}$, and they create a bond. Therefore $\tilde{c}$ has at least one nascent bond, so it is a favorable configuration which leads to contradiction since $c^{\infty}_{l+1} \neq \tilde{c}_{l+1}$.

Suppose there is a bond between r_i and c'_{l+j}. Consider the δ'-elongation $\tilde{c}$ whose nascent beads are marked violet (●) in Figure 28. It is constructed in the same way as in the previous case. Here, $\tilde{c}_{l+j}$ is the east neighbor of r_i, this means that $\tilde{c}$ is a favorable elongation which contradicts $c^{\infty}_{l+1} \neq \tilde{c}_{l+1}$.

If the single bond is between two nascent beads, we denote these beads by c'_{l+i} and c'_{l+j}, where $1 \leq i < j \leq \delta'$; this means that $\omega_{l+i} \heartsuit \omega_{l+j}$. Consider the δ'-elongation $\tilde{c}$ shown in Figure 29, whose nascent beads are marked pink (○), it is constructed in the same way as in the previous cases. Here, $\tilde{c}_{l+j}$ is the north-west neighbor of $\tilde{c}_{l+i}$, so they create a bond. As in two previous cases, this contradicts $c^{\infty}_{l+1} \neq \tilde{c}_{l+1}$ and, therefore, we have proved that $\mathcal{N}_l > 1$. □

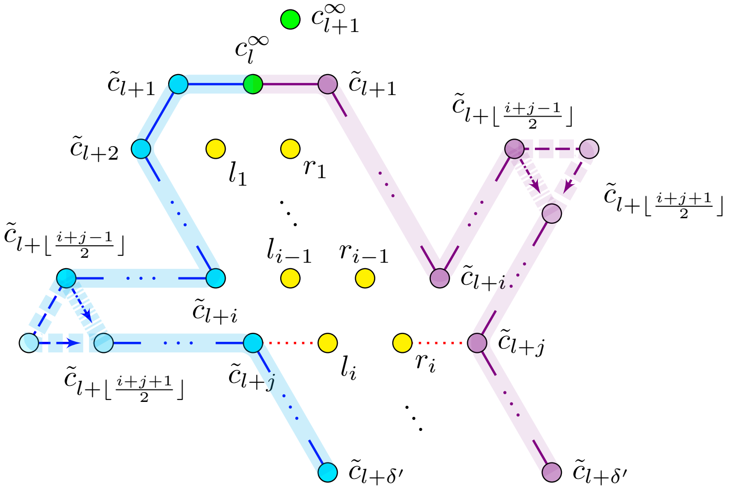

Figure 28: Elongation $\tilde{c}$ from Lemma 11: marked violet in the case where the bond is to the right of the yellow line, blue otherwise.

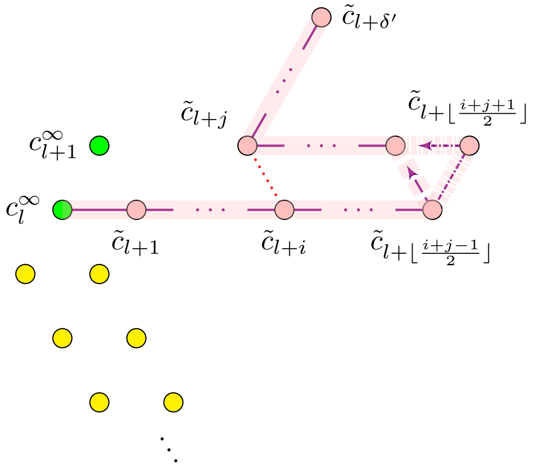

Figure 29: Elongation $\tilde{c}$ from Lemma 11 in the case where the single bond of c' is between c'_{l+i} and c'_{l+j}.

Let us partition the snake L^n into sequences of length $\delta'-1$. The next lemma states that there is a crucial step at each $(\delta'-1)$-segment.

Lemma 12. For all $k \in \{1, \ldots, \lfloor\frac{M}{\delta'-1}\rfloor - 1\}$, there exists a crucial step $i_k \in \{l+k(\delta'-1)+1, \ldots, l+(k+1)(\delta'-1)\}$ and $\mathcal{N}_{l+(k+1)(\delta'-1)} > 1$.

Proof. The first part of the base case, when $k = 0$, follows from Lemma 11 : since $\mathcal{N}_l > 1$, by Lemma 10 , there exists $i_0 \in \{l+1, \ldots l+\delta'-1\}$ such that the step i_0 is crucial. Let us now compute $\mathcal{N}_{l+(\delta'-1)}$:

$$\mathcal{N}_{l+(\delta'-1)} - \mathcal{N}_l = \sum_{i=l+1}^{l+(\delta'-1)} \mathcal{N}_i - \mathcal{N}_{i-1} \geq \sum_{i=l+1}^{l+(\delta'-1)} \mathbb{1}_{\{\text{i is crucial}\}} - h_i(c^i) \geq 1$$

since there are no corner beads: for all $i \in \{l+1, \ldots l+\delta'-1\}$, $h_i(c^i) = 0$. Thus, $\mathcal{N}_{l+(\delta'-1)} > 1$ which concludes the proof of the base case.

Suppose the condition holds for all $k \leq n$; let us prove this for $k = n+1$. By induction hypothesis, $\mathcal{N}_{l+(n+1)\delta'} > 1$. Thus, by Claim 1, there exists $i_{n+1} \in \{l + (n+1)(\delta'-1) + 1, \ldots, l + (n+2)(\delta'-1)\}$ such that the step i_{n+1} is crucial.

Let us now prove that $\mathcal{N}_{l+(n+1)\delta'} > 1$. Since $\delta' < \delta$, there is no more than 1 corner bead among the beads indexed by $\{l + n(\delta'-1) + 1, \ldots, l + (n+1)(\delta'-1)\}$. Let us calculate the difference between $\mathcal{N}_{l+(n+1)(\delta'-1)}$ and $\mathcal{N}_{l+n(\delta'-1)}$:

$$\mathcal{N}_{l+(n+1)(\delta'-1)} - \mathcal{N}_{l+n(\delta'-1)} = \sum_{i=l+n(\delta'-1)+1}^{l+(n+1)(\delta'-1)} \mathcal{N}_i - \mathcal{N}_{i-1} \geq 1 - \sum_{i=l+n(\delta'-1)+1}^{l+(n+1)(\delta'-1)} h_i(c_i) \geq 1{-}1 = 0$$

since there is at most one corner bead, hence, at most one index i such that $h_i(c_i) = 1$: for all others it is equal to 0. Thus, $\mathcal{N}_{l+(n+1)(\delta'-1)} \geq \mathcal{N}_{l+n(\delta'-1)} > 1$ by induction hypothesis.

□

Denote the set of the first $\lfloor \frac{5(\delta-1)^3}{\delta'-1} \rfloor$ crucial steps from Lemma 12 by $I = \{i_k \mid 0 \leq k \leq \lfloor \frac{5(\delta-1)^3}{\delta'-1} \rfloor - 1\}$. The set of the first $5(\delta-1)^2$ indices of the corners is denoted by $C = \{l + k(\delta-1) \mid 0 \leq k \leq 5(\delta-1)^2 - 1\}$.

In the following claim, we directly apply Lemma 2 to all the cases.

Claim 13. For $i \in \{l+1, \dots, l+5(\delta-1)^3-1\}$

(1) If $i \in C \setminus I$ then $h_i(c^i) = 1$, so $\mathcal{N}_i - \mathcal{N}_{i-1} \geq -1$.

(2) If $i \in C \cap I$ then $h_i(c^i) = 1$ and step i is crucial, so $\mathcal{N}_i - \mathcal{N}_{i-1} \geq 1 - 1 = 0$.

(3) If $i \notin C \cup I$ then $h_i(c^i) = 0$ so $\mathcal{N}_i - \mathcal{N}_{i-1} \geq 0$.

(4) If $i \in I \setminus C$ then $h_i(c^i) = 0$ and step i is crucial, so $\mathcal{N}_i - \mathcal{N}_{i-1} \geq 1$.

Since $M = 8(\delta-1)\delta^2 + 3\delta - 2$, we have $l + 5(\delta-1)^3 - 1 < l + M - \delta'$, so all results given above are valid for $l \leq i \leq l + 5(\delta-1)^3 - 1$. Let us consider the difference between $\mathcal{N}_{l+5(\delta-1)^3-1}$ and $\mathcal{N}_l$. By Claim 13,

$$\mathcal{N}_{l+5(\delta-1)^3-1} - \mathcal{N}_l = \sum_{i=l+1}^{l+5(\delta-1)^3-1} \mathcal{N}_i - \mathcal{N}_{i-1} \geq \sum_{i \in I \setminus C} 1 + \sum_{i \in C \setminus I} (-1) = |I| - |C| = \left\lfloor \frac{5(\delta-1)^3}{\delta'-1} \right\rfloor - 5(\delta-1)^2$$

Thus, since $\mathcal{N}_l > 1$ and $\delta \geq \delta' + 1$,

$$\mathcal{N}_{l+5(\delta-1)^3-1} > \frac{5(\delta-1)^3}{\delta-2} - 1 - 5(\delta-1)^2 + 1 = \frac{5(\delta-1)^2}{\delta-2} \geq 5\delta - 5 \geq 5(\delta'+1) - 5 \geq 4\delta' + 1$$

which leads to contradiction by Claim 4.

3.4 Scaling and folding

If a shape is foldable by some OS, then it can be covered by a self-avoiding path which is the terminal configuration of this OS. Thus, a shape is foldable only if its induced graph on the triangular lattice has a Hamiltonian path. The latter question is NP-hard [2], thus, it is also NP-hard to decide if there is an OS that folds into a given finite shape [9]. Even though it is hard to know if an arbitrary shape can be self-assembled, it turns out that a "scaled" version of any shape is foldable.

This section is based on the results from Sections 2-5 of [8]. Together with Nicolas Schabanel, I worked on the elimination of the last case in a construction supplementing the results mentioned above.

Intuitively, scaling means partitioning the triangular lattice into identical shapes and considering them as points of a new, upscaled, triangular lattice. Formally, a *scaling scheme* is an homothetic linear map λ from the triangular lattice to itself, together with a *cell mold* μ containing the origin, which is a shape of an upscaled cell. Given a scaling scheme (λ, μ), the image of a point p after scaling is a shape μ whose origin is placed in $\lambda(p)$. In [8], three scaling schemes are considered, we denote them $\mathscr{A}_n, \mathscr{B}_n$, and $\mathscr{C}_n$. We work on another, more compact, scaling scheme $\mathscr{D}_2$ which was not studied yet. Schemes $\mathscr{A}_3, \mathscr{B}_3, \mathscr{C}_3$, and $\mathscr{D}_2$ are illustrated in Figure 30.

There are two important results about scaled shapes in [8]. First, for any connected finite shape S, for any scaling scheme among $\mathscr{A}_2, \mathscr{B}_2$, and $\mathscr{C}_2$, there is an OS with delay $|S|$ which assembles S at a given scale. This result is interesting but somehow unrealistic because of a huge delay. However, in slightly larger scales, we have the same result with a small constant delay: any shape can be folded by an OS with delay 1 at all scales $\mathscr{A}_n, \mathscr{B}_n$, and $\mathscr{C}_n$ for $n \geq 3$.

However, being interested in folding of tiny shapes, we would like to find a way to fold figures at a smaller scale than $\mathscr{A}_3, \mathscr{B}_3$, and $\mathscr{C}_3$. With Nicolas Schabanel, I worked on a construction of an OS with delay 3 folding any shape at scale $\mathscr{D}_2$, much more compact than the latter ones. The main idea of the proof is the same as for the results about larger scales. To proof the foldability, we build a Hamiltonian path which covers the shape and can be obtained as a terminal configuration of some OS with a given delay. We build this path inductively: starting from the path

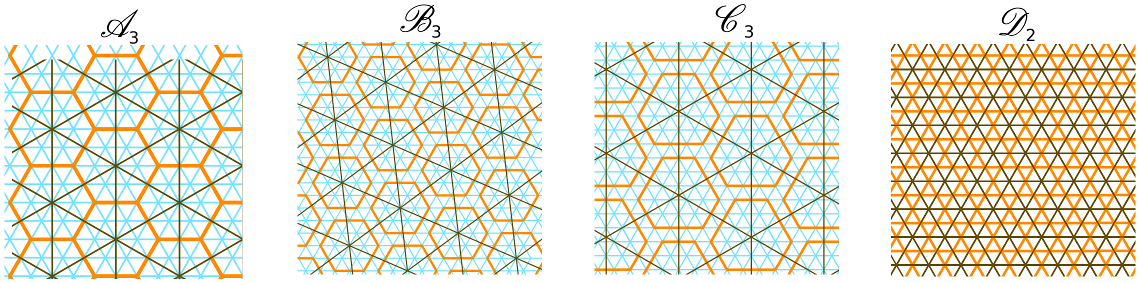

Figure 30: Illustration of the scaling schemes $\mathscr{A}_3, \mathscr{B}_3, \mathscr{C}_3$ and $\mathscr{D}_2$: cell boundaries are orange, images of the triangular grid after scaling is brown.

covering one scaling cell and inserting segments covering other cells, one by one, until the whole shape is covered. All we need is to find how to insert a new path segment filling a cell depending on the neighborhood of this cell, such that the path would stay foldable when this segment is added.

In the construction of an OS folding shapes at scale $\mathscr{D}_2$, there is one case left to treat: we need to design two segments of the Hamiltonian path passing by cells A, B, C, and D in Figure 31 which would satisfy a number of conditions. There are many hidden details in this problem: depending on the already covered adjacent cells and the order in which they were covered, we get different constraints on the path segments. For instance, in Figure 31, only two cells are already occupied around A and B, while in Figure 32, we consider the case where they have 6 covered neighbors.

My approach was to to consider all Hamiltonian paths covering the cells, keep only the paths satisfying the majority of necessary conditions and then, when there is not so many cases to check, analyse the result. I got this data in the end of my internship, so we did not have time to analyse it: it is still a work in progress.

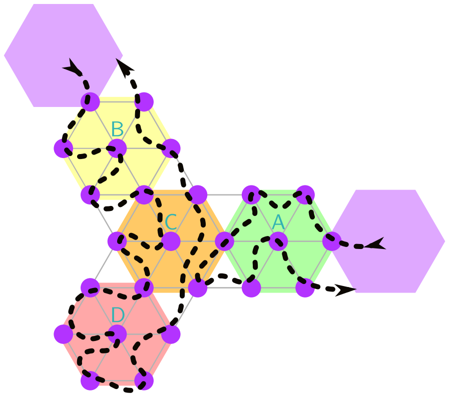

Figure 31: The problematic case. Cells are added in the alphabetic order, both path segments begin and end in the violet cells.

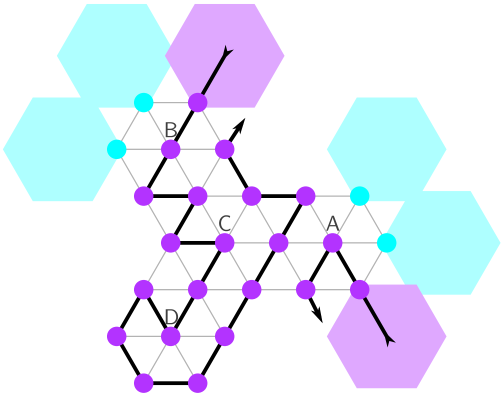

Figure 32: Example of the Hamiltonian paths: already covered cells are blue.

4 Oritatami and computability

Molecular computing [1, 18] gets a lot of attention these days. Indeed, it allows to perform nano-scale parallel computations biologically: isn't it exciting? A lot of progress has been done in the area of DNA computations [11,24]. In this section, we investigate computational properties of Oritatami.

4.1 Simulating cellular automata

It was shown that any tag system (and, therefore, any Turing machine) can be simulated by an OS [16]. However, the design of such an OS is very complex and not intuitive: a simpler model was needed to get closer to the experimental implementation. There is a much simpler and more elegant way to prove the computational universality of Oritatami folding by simulating 1D cellular automata. The simplicity of this system makes it a good candidate for an implementation in the wet lab.

I got involved in the development of this construction with Nicolas Schabanel, Shinnosuke Seki, and Yuki Ubukata. I participated in a simplification of one of the modules. In this section, we will first see an overview of the whole construction and then closely examine the part I worked on. Most of the figures from this section are made by Nicolas Schabanel for his talk at “Journées SDA2”.

Our aim in this work is to simulate any 1D cellular automaton. First of all, using the classical folklore result, we consider a shifted automaton of radius $\frac{1}{2}$ instead of an arbitrary automaton, this simplification is illustrated in Figure 33.

The idea of our construction is to create a space-time diagram of computation of an $\frac{1}{2}$-automaton by moving in a zig-zag way, like it is shown in Figure 34. Each cell of a diagram has a rhombus shape, as in Figure 35. Folding a given cell, our system “reads” two input states, denoted x and y in Figure 35, applies the transition function and “writes” the corresponding two output states x' and y' on the other sides of the cell.

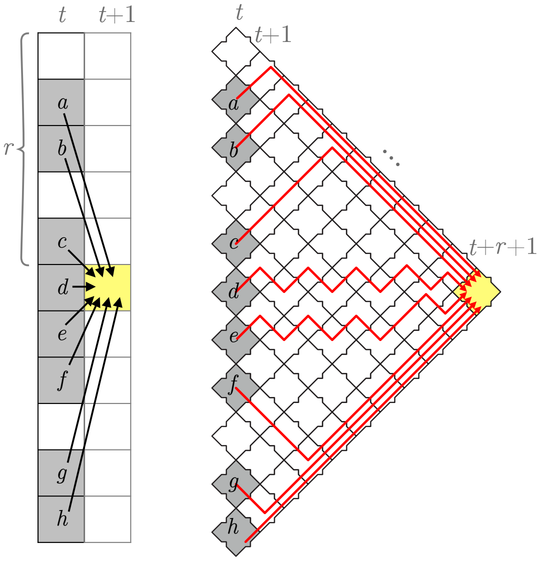

Figure 33: A reduction from a CA of radius r to a shifted $\frac{1}{2}$-automaton.

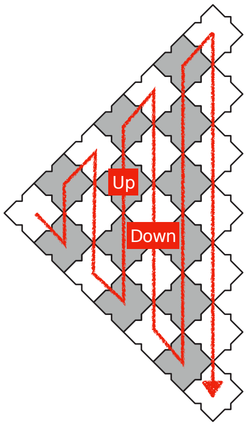

Figure 34: The order of visiting the space-time diagram.

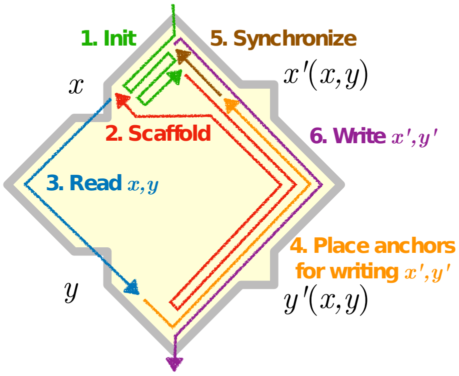

Figure 35: Scheme of the modules in the cell transcript.

The transcripts for all cells of the diagram are equal: reading, applying the transition function and writing differ only in the geometric way in different cells. Thus, all cells are made of the same sequence of beads. In other words, the transcript is periodic and consist of an infinite number of repetitions of the cell transcript. All we need now is to describe the cell transcript.

A cell consists of several modules, their scheme is shown in Figure 35. The first module, marked green, controls the shape and the symmetry of the cell depending on its position in the space-time diagram: this is what “makes the turn” once we go to the next step of computation. The scaffold module, red in the figure, is the skeleton of the cell: it has the same shape in all cells and is used as a landmark for the rest of the transcript. The reading part, which is blue, folds upon the output sides of two adjacent cells, reading two input values. It is designed in such a way that after reading x and y, the transcript has an offset of $2(Qx + y)$ where Q is the number of states. This shift is crucial for the next part of folding, orange in the figure, which applies a transition function to the input placing special marks in order to correctly fold the output values later. Writing starts at a precise position in the cell transcript, so we should remove the offset before it launches. That is why we need the synchronization part, marked brown, which absorbs a misalignment of the transcript using the scaffold as a guideline. After this, comes the writing part of the cell transcript, colored in violet.

I, together with Nicolas Schabanel, worked on the synchronization part. Using a segment of the scaffold as a benchmark, this part should absorb an offset which is bounded by some constant depending on Q. We arrange the beads of the synchronization module in such a way that being well aligned with the scaffold, they form a straight segment. If the transcript is in advance, then some of its parts fold into a zigzag which shortens these parts twice until the transcript is well aligned with the scaffold, e.g., the offset is absorbed. Figure 36 shows the synchronization process more precisely: we see the scaffold module at the bottom, its red beads are the benchmarks used by the synchronization module (at the top). If the transcript is well-aligned, the pink beads of the synchronization module are in front of the yellow beads of the scaffold while the green beads are aligned with the blue beads. Otherwise, once some pink beads are in front of the blue ones, they form a zigzag which absorbs at least one half of the offset each time. Since this pattern repeats log(Maximal offset) times, the shift is guaranteed to be absorbed in the end of the synchronization part.

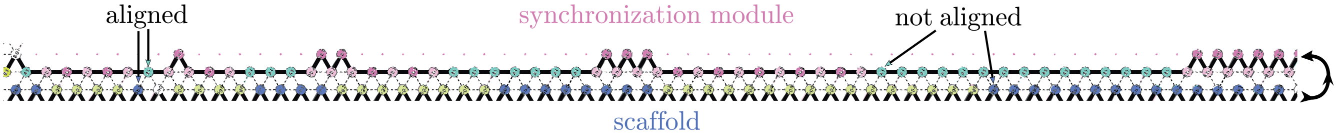

Figure 36: The synchronization module at work: the scaffold is at the bottom, the synchronized part is on the top.

An essential detail in the design of the scaffold and the synchronization parts is the lengths of the pink, green, yellow, and blue segments, they are shown in Figure 37.

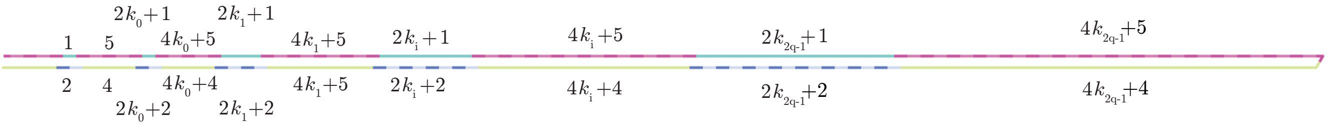

Figure 37: The lengths of segments of beads of different types, $k_0 = 0,\ k_{i+1} = 2^i - 1$.

Figures 38 - 41 show why each blue segment of a scaffold absorbs at least one half of an offset by making the pink beads form a zigzag.

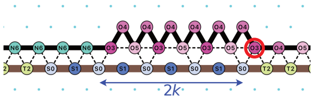

Figure 38: $\Delta = 4k$, then $\Delta' = \frac{\Delta}{2}$.

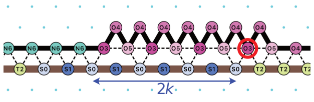

Figure 39: $\Delta = 4k + 1$, then $\Delta' = \lfloor \frac{\Delta}{2} \rfloor$.

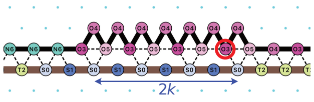

Figure 40: $\Delta = 4k + 2$, then $\Delta' = \frac{\Delta}{2}$.

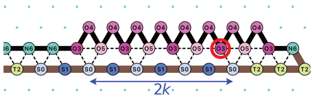

Figure 41: $\Delta = 4k + 3$, then $\Delta' = \lfloor \frac{\Delta}{2} \rfloor$.

4.2 Making complex terminal shapes

During the "Journes SDA2" conference mentioned above, Guillaume Theyssier asked whether there is an OS producing an uncomputable terminal shape. Intuitively, this question is about how far we can go: since the Oritatami model is computationally universal, can it produce an uncomputable outcome? Together with Nicolas Schabanel, Shinnosuke Seki, and Guillaume Theyssier, we found out that there is such an OS. Moreover, any recursively enumerable set can be represented by the terminal shape of the OS that we built. The main idea of this construction is, given a set, to simulate a Turing machine describing this set in parallel on all inputs (we do it using the system from the previous section). The second step is, for each input, to write down in a fixed cell if the Turing machine halts on this input or not. The question asked by Guillaume Theyssier and some ideas about the construction we found were inspired by a similar result for the Tile Assembly Model [17]. However, the Oritatami model is very different: it is much more constrained in space and cannot perform parallel computations, so our proof has many supplementary details. This is a research in progress, and I am highly motivated to keep working on it.

5 My contributions, ongoing work, and further research

My internship covered several subjects: I have obtained complete results in some of them and just moved a little forward in others. Anyway, a lot of work is in progress right now and we have many interesting open questions.

I attempted to complete the construction of an OS folding very compact scaled shapes. My approach consisted in using brute force on Hamiltonian paths with specific properties in order to find the required path segments, this is explained in Section 3.4. I collected the necessary data and now we need to analyze it. Once it is done, we will either complete the construction of an OS, or, if there are no appropriate paths, significantly change this construction. In the future, we shall consider foldability of scaled shapes by blocking OSs: this might allow to reduce the scale.

Working on the limits between foldable and unfoldable shapes, I fixed and simplified an existing design of a delay-borderline shape (see Section 3.3). Doing this, I found a missed constraint leading me to the notion of blocking OSs which were not considered before and are not studied yet: eventually, we need to understand how well this model reflects the experiments and which class of shapes is foldable in this context.

I also obtained general tools for proving unfoldability of various shapes, which are given in Section 3.1, and applied them in a particular case: I got an upper and a lower bound for a length of a straight line foldable by a nonblocking OS, as it is explained in Section 3.2. There is a huge gap between these bounds: one is quadratic in δ and the other one is linear. We would like to make them tighter, and this is an open question. Besides that, we found a blocking OS folding an infinite straight line (see Section 3.2.3) which is the first step in understanding the difference between blocking and nonblocking OSs.

Studying computational properties of Oritatami, I took part in the development of an OS simulating 1D cellular automata, it is explained in Section 4.1. This is a paper in progress. This system is simple enough and we hope to implement some simplifications of its parts in the wet lab. The first step of these experiments would be to build the synchronization part I worked on since its mechanics is rather simple and it is easy to visualize.

Inspired by a question asked by Guillaume Theyssier about uncomputable terminal configurations, we proved that any recursively enumerable set can be represented by a terminal shape of an OS (see Section 4.2). This construction is quite complex and we are in the process of checking all the details in order to write a paper. The fact that such a highly constrained model as Oritatami can produce such complex shapes raises another question: are there even more constrained computational models able to construct a representation of any recursively enumerable set?

References

[1] Leonard M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(11):1021–1024, November 1994.

[2] Esther M. Arkin, Sndor P. Fekete, Kamrul Islam, Henk Meijer, Joseph S.B. Mitchell, Yurai Nez-Rodrguez, Valentin Polishchuk, David Rappaport, and Henry Xiao. Not being (super)thin or solid is hard: A study of grid hamiltonicity. *Computational Geometry*, 42(6):582 – 605, 2009.

[3] Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic (HP) is NP-complete. In *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, RECOMB '98, pages 30–39. ACM, 1998.

[4] Ho-Lin Chen and David Doty. Parallelism and time in hierarchical self-assembly. *SIAM J. Comput.*, 46(2):661–709, 2017.

[5] J. Chen and Nadrian Seeman. Synthesis from DNA of a molecule with the connectivity of a cube. *Nature*, 350(6319):631–633, 1991.

[6] Moya Chen, Doris Xin, and Damien Woods. Parallel computation using active self-assembly. *http://arxiv.org/abs/1405.0527*, 2014.

[7] Matthew Cook, Yunhui Fu, and Robert Schweller. Temperature 1 self-assembly: Deterministic assembly in 3D and probabilistic assembly in 2D. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 11 2009.

[8] Erik D. Demaine, Jacob Hendricks, Meagan Olsen, Matthew J. Patitz, Trent A. Rogers, Nicolas Schabanel, Shinnosuke Seki, and Hadley Thomas. Know when to fold 'em: Self-assembly of shapes by folding in oritatami. `https://arxiv.org/abs/1807.04682`, 2018.

[9] Erik D. Demaine, Jacob Hendricks, Meagan Olsen, Matthew J. Patitz, Trent A. Rogers, Nicolas Schabanel, Shinnosuke Seki, and Hadley Thomas. Know when to fold 'em: Self-assembly of shapes by folding in oritatami. In *DNA Computing and Molecular Programming*, pages 19–36. Springer International Publishing, 2018.

[10] Ken A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6):1501–1509, 1985.

[11] David Doty, Jack H. Lutz, Matthew J. Patitz, Robert T. Schweller, Scott M. Summers, and Damien Woods. The tile assembly model is intrinsically universal. *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 302–310, 2011.

[12] Shawn M. Douglas, Hendrik Dietz, Tim Liedl, Björn Högberg, Franziska Graf, Adam H. Marblestone, Surat Teerapittayanon, Alejandro Vazquez, George M. Church, and William M. Shih. Design and self-assembly of DNA into nanoscale 3D shapes. In *SIGGRAPH Talks*, 2009.

[13] Kirsten L. Frieda and Steven M. Block. Direct observation of cotranscriptional folding in an adenine riboswitch. *Science*, 338(6105):397–400, 2012.

[14] Cody Geary, Pierre-Etienne Meunier, Nicolas Schabanel, and Shinnosuke Seki. Programming biomolecules that fold greedily during transcription. In *MFCS 2016*, volume 58 of *LIPIcs*, pages 43:1–43:14, 2016.

[15] Cody Geary, Paul W. K. Rothemund, and Ebbe S. Andersen. A single-stranded architecture for cotranscriptional folding of RNA nanostructures. *Science*, 345(6198):799–804, 2014.

[16] Cody W. Geary, Pierre-Étienne Meunier, Nicolas Schabanel, and Shinnosuke Seki. Proving the turing universality of oritatami co-transcriptional folding. In *29th International Symposium on Algorithms and Computation, ISAAC 2018, December 16-19, 2018, Jiaoxi, Yilan, Taiwan*, pages 23:1–23:13, 2018.

[17] James I. Lathrop, Jack H. Lutz, Matthew Patitz, and Scott Summers. Computability and complexity in self-assembly. *Theory of Computing Systems*, 48:617–647, 06 2008.

[18] Natasa Jonoska and Nadrian C. Seeman. Computing by molecular self-assembly. *Interface focus*, 2 4:504–11, 2012.

[19] Paul Rothemund. Rothemund, p.w.k.: Folding DNA to create nanoscale shapes and patterns. nature 440, 297-302. *Nature*, 440:297–302, 04 2006.

[20] Karthikeyan Subramani, Ameen Khraisat, and Anne George. Self-assembly of proteins and peptides and their applications in bionanotechnology. *Current Nanoscience*, 4:201 – 207, 05 2008.

[21] Gunjan Verma and Puthusserickal Hassan. Self assembled materials: Design strategies and drug delivery perspectives. *Physical chemistry chemical physics : PCCP*, 15, 08 2013.

[22] GM Whitesides, JP Mathias, and CT Seto. Molecular self-assembly and nanochemistry: a chemical strategy for the synthesis of nanostructures. *Science*, 254(5036):1312–1319, 1991.

[23] Erik Winfree. Algorithmic self-assembly of DNA. *Ph.D. Thesis, California Institute of Technology*, 1998.

[24] Damien Woods, David Doty, Cameron Myhrvold, Joy Hui, Felix Zhou, Peng Yin, and Erik Winfree. Author correction: Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. *Nature*, 08 2019.