Folding Turing is hard but feasible (Abstract)

Cody Geary[1], Pierre-Étienne Meunier[2], Nicolas Schabanel[3], and Shinnosuke Seki[4]

1 **California Institute of Technology, Pasadena, CA, USA.** codyge@gmail.com.
2 **Department of Computer Science, Aalto University, Finland and Aix Marseille Université, CNRS, LIF UMR 7279, 13288, Marseille, France.** http://users.ics.aalto.fi/meunier/
3 **CNRS, Université Paris Diderot, France and IXXI, Université de Lyon, France.** http://www.irif.univ-paris-diderot.fr/users/nschaban/
4 **University of Electro-Communications, Tokyo, Japan.** http://kjk.office.uec.ac.jp/Profiles/69/0006845/prof_e.html

Abstract

We introduce and study the computational power of Oritatami, a theoretical model to explore greedy molecular folding, by which the molecule begins to fold before waiting the end of its production. This model is inspired by our recent experimental work demonstrating the construction of shapes at the nanoscale by folding an RNA molecule during its transcription from an engineered sequence of synthetic DNA. While predicting the most likely conformation is known to be NP-complete in other models, Oritatami sequences fold optimally in linear time. Although our model uses only a small subset of the mechanisms known to be involved in molecular folding, we show that it is capable of efficient universal computation, implying that any extension of this model will have this property as well.

We introduce general design techniques for programming these molecules. Our main result in this direction is an algorithm in time linear in the sequence length, that finds a rule for folding the sequence deterministically into a prescribed set of shapes depending of its environment. This shows the corresponding problem is fixed-parameter tractable although we proved it is NP-complete in the number of possible environments. This algorithm was used effectively to design several key steps of our constructions.

Our present results have been announced at DNA21 (2015) [6] and are currently submitted to ICALP 2016 [7].

1 Introduction

The process by which one-dimensional sequences of nucleotides or amino-acids acquire the complex three-dimensional geometries of biomolecules is a major puzzle of biology today. In particular, the problem of predicting how proteins fold is a major source of interest, as it could potentially allow us to engineer our own proteins.

A few year ago, the *kinetics* of folding, which is the step-by-step dynamics of the reaction, has been demonstrated by biochemists to play a fundamental role in the final shape of molecules [10], and an essential role in the case of RNA [5]. In recent experimental results [9], researchers have been able to *control* this mechanism to engineer their own shapes out of RNA.

One of the most widely used techniques in DNA nanotechnologies, *DNA Origami* [13], requires the molecules to be heated up to high temperature (about 90C) before being slowly cooled down at a precisely controlled rate. In contrast to this, one of the main benefits of RNA Origami [9] is the possibility of controlling folding at temperatures compatible with human life.

Previous theoretical studies on folding focused mostly on the energy optimization mechanisms. For example, in different variants of the *hydrophobic-hydrophilic (HP) model* [4], it has been shown that the problem of predicting the most likely geometry (or *conformation*) of a sequence is NP-complete [14, 12, 1, 2, 3], both in two and three dimensions.

Here, we focus on kinetics, a different and complementary mechanism. We introduce a new model based on the experiments conducted in [9] to explore the perspectives opened by co-transcriptional folding. In particular, in co-transcriptional folding, molecules fold in linear time, which allows us to focus on understanding and developing design paradigms.

Main contributions. We introduce a new model of molecular folding where the molecule gets folded while being produced. More precisely, we consider a sequence of "beads", or abstract basic components which may stand for nucleotides or even sequences of nucleotides (or *domains*). In our model, only the δ latest

Highlights of Algorithms 2016.

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

produced beads of the molecules are allowed to move in order to adopt a more favorable configuration. The folding is driven by the respective attraction between the beads.

We show that our model is able of efficient universal Turing computation. This result heavily relies on the efficient simulation of Turing machines by tag system, from [11]. Building a tag system simulator not only shows the model to be powerful, it also pointed us explicitly to the challenges of molecular engineering. Namely, it led us to develop modular constructions and techniques to produce different shapes from a unique sequence in reaction to its environment. Furthermore, it taught us how one can prepare this environmental changes to trigger calls to specific functions encoded in the sequence. We believe that many of our technics can be used to develop an algorithmic basis for molecular folding engineering.

Moreover, our constructions also motivated the development of an algorithm running in time linear in the sequence length, that finds an attraction rule for folding a single sequence deterministically into a prescribed set of shapes, depending on surrounding beads. As a consequence, even though we will show that the problem of finding a rule is NP-complete, we have been able to implement and use this algorithm to resolve some parts of our designs.

2 Model and Main Results

Oritatami system. Oritatami is about the folding of finite sequences of beads, each from a finite set B of *bead types*, using an attraction rule 💗, on the triangular lattice graph $\mathbb{T} = (\mathbb{Z}^2, \sim)$ where $(x, y) \sim (u, v)$ if and only if $(u, v) \in \{(x \pm 1, y), (x, y \pm 1), (x \pm 1, y \pm 1)\}$. A *conformation* c of a sequence $w \in B^*$ is a self-avoiding path of length ℓ labelled by w in $\mathbb{T}$, i.e. a path whose vertices $c_1, \ldots, c_\ell$ are pairwise distinct and labelled by the letters of w. A *partial conformation* of a sequence w is a conformation of a prefix of w. For any partial conformation c of some sequence w, an *elongation* of c by k beads is a partial conformation of w of length $|c| + k$. We denote by $\mathcal{C}_w$ the set of all partial conformations of w. We denote by $c^{\triangleright k}$ the set of all elongations by k beads of a partial conformation c of a sequence w and by $c^{\triangleleft k}$ the singleton containing the prefix of length $|c| - k$ of c.

An *Oritatami system* $\mathcal{O} = (p, 💗, \delta)$ is composed of (1) a (possibly infinite) *primary structure* p, which is a sequence of *beads*, of a type chosen from a finite set B, (2) an *attraction rule*, which is a symmetric relation $💗 \subseteq B^2$ and (3) a parameter δ called the *delay time*. Given an attraction rule 💗 and a conformation c of a sequence w, we say that there is a *bond* between two adjacent positions c_i and c_j of c in $\mathbb{T}$ if $w_i \, 💗 \, w_j$. The *energy* of a conformation c of w, written $\mathrm{E}(c)$, is the negation of the number of bonds within c: formally, $\mathrm{E}(c) = -|\{(i, j) : c_i \sim c_j,\ j > i + 1,\ \text{and } w_i \, 💗 \, w_j\}|$.

Oritatami dynamics. A *dynamics* for a sequence w is a function $\mathcal{D}_w : 2^{\mathcal{C}_w} \to 2^{\mathcal{C}_w}$ such that for all subset S of partial conformations of length ℓ of w, $\mathcal{D}(S)$ is a subset of the elongations by one bead of the partial conformations in S (thus, partial conformations of length $\ell + 1$).

Given an Oritatami system $\mathcal{O} = (p, 💗, \delta)$ and a *seed conformation* σ of a seed sequence s of length ℓ, the set of partial conformations of the primary structure p at time t under dynamics $\mathcal{D}$ is $\mathcal{D}^t_{sp}(\{\sigma\})$,[1] i.e. the set of all elongations by t beads of the seed conformation prolongated by the primary structure according to dynamics $\mathcal{D}$. In this abstract, we focus on the *oblivious dynamics*, which consists in placing the last δ beads in the minimal energy positions, regardless of their previously adopted positions.[2]

$$\mathcal{O}(S) = \bigcup_{\gamma \in S} \left(\operatorname*{arg\,min}_{c \in (\gamma^{\triangleleft(\delta-1)})^{\triangleright\delta}} \mathrm{E}(c) \right)$$

The resulting conformations are nondeterministic. And, the nondeterministic position of the i-th bead of p is final at time $i + \delta$.

An Oritatami system $\mathcal{O} = (p, 💗, \delta)$ is *deterministic* for dynamics $\mathcal{D}$ and seed σ of sequence s if for all $i \geqslant 1$, the position of the i-th bead of p is deterministic at time $i - 1 + \delta$, i.e. if for all $i \geqslant 1$, $|\{c_{|\sigma|+i} : c \in \mathcal{D}^{i-1+\delta}_{sp}(\{\sigma\})\}| = 1$. We say that $\mathcal{O}$ *stops* at time t with seed σ and dynamics $\mathcal{D}$ if $D^t_{sp}(\{\sigma\}) = \varnothing$ and $\mathcal{D}^z_{sp}(\{\sigma\}) \neq \varnothing$ for $z < t$. Typically, the folding process stops because of geometric obstruction (no more elongation are possible because the conformation gets trapped in a closed area).

[1] Given two words $a, b \in B^*$, we denote by ab their concatenation.
[2] We denote by $\operatorname{arg\,min}_{x \in X} f(x)$ the set of the minima: $\operatorname{arg\,min}_{x \in X} f(x) = \{y \in X : f(y) = \min_{x \in X} f(x)\}$.

Turing universality. Our first main result shows that there is a Turing-universal Oritatami system, able to simulate the execution of any Turing machine with only a polynomial slowdown. (Proof sketch next)

▶ **Theorem 1.** *There is an oblivious deterministic Oritatami system* $\mathcal{U} = (p, \heartsuit, 3)$ *and a* log*-space reduction from any Turing machine* $\mathcal{M}$ *and any input* x *to a seed configuration* $\sigma_{\mathcal{M},x}$*, such that starting from seed conformation* $\sigma_{\mathcal{M},x}$*,* $\mathcal{U}$ *stops if and only if* $\mathcal{M}$ *accepts* x*. Moreover, if* $\mathcal{M}$ *halts after* T *steps on input* x*,* $\mathcal{U}$ *halts after folding* $O(T^2 \log T)$ *beads.*

In particular, the total number of bead types as well as the period of p in $\mathcal{U}$ are bounded by a universal constant.

Rule design. Our second main result concerns the design of a rule for achieving a set of given foldings depending on the environment. (Proof ommitted)

Input: A set of beads $B \supseteq \{1, \ldots, n\}$, a delay time δ, k seed conformations $\sigma_1, \ldots, \sigma_k$ of sequences $s_1, \ldots, s_k \in B^*$ (with possibly different lengths) and k target conformations $c_1, \ldots, c_k$ of the $n - \delta$ first beads of the sequence $p = \langle 1, \ldots, n \rangle$.

Output: A rule $\heartsuit \subseteq B^2$ such that for all $i = 1..k$, the Oritatami system $\mathcal{O} = (p, \heartsuit, \delta)$ folds the $n - \delta$ first beads of p deterministically into $\sigma_i c_i$ from seed conformation σ_i under the oblivious dynamics $\mathcal{O}$, i.e. such that $\mathcal{D}_{s_i p}^{n-\delta}(\{\sigma_i\}) = \{\sigma_i c_i\}$ for all $i = 1..k$.

▶ **Theorem 2.** *The Rule design problem is NP-complete for all* $\delta \geqslant 1$ *and* $n \geqslant 1$*. However, it is FPT as it can be solved in time* $O(C^{\delta \cdot k} n)$ *for some* $C > 0$*, linear in the length* n *of the primary sequence.*

3 A Turing-universal Oritatami system

In this section, we demonstrate the existence of a single periodic primary structure that can simulate any Turing computation. Precisely, our construction simulates a particular type of tag systems which are known to simulate in $O(T^2 \ln T)$ steps any Turing machine running in T steps [11]. Our simulation uses the *oblivious dynamics* with delay time 3. Due to space constraints, we will not provide the full proofs of the correctness of the folding. We refer the reader to the *videos available* at [8] for a full demonstration of the resulting Oritatami system folding its modules live upon itself. The full description of the modules and rule is given in [7]. The full description of G is given as an example in appendix.

Skipping Cyclic Tag systems A *skipping cyclic tag system* consists of a set of n productions $p_0, \ldots, p_{n-1} \in \{0,1\}^*$ and an initial word $w^0 \in \{0,1\}^*$. At each time step, the tag system cycles through the productions and decides to append the current production or not depending on the letter read. We denote by w^t the word at time t. Precisely, at time $t = 0$, the pointer q^0 is set to 0. At all time t,

- If w^t is the empty word ϵ, then the tag system halts and outputs q^t.
- Otherwise, if the first letter w_1^t of w^t is 0, then set $q^{t+1} := (q^t + 1) \bmod n$ and $w^{t+1} := w_2^t \ldots w_{|w^t|}^t$, the suffix of w^t without its first letter.
- And if $w_1^t = 1$, then the tag system appends the next production to w^t and skips to the following production, i.e.: $w^{t+1} := w_2^t \ldots w_{|w^t|}^t . p_{q'}$ where $q' = (q^t + 1) \bmod n$ and $q^{t+1} := (q^t + 2) \bmod n$.

For instance, the skipping tag system $(\epsilon, 100, 1, 0)$ has the following execution $(\langle w^t, p_{q^t} \rangle)_t$ from input word $w^0 = 010$: $\langle 010, \epsilon \rangle, \langle 10, 100 \rangle, \langle 01, 0 \rangle, \langle 1, \epsilon \rangle, \langle 100, 1 \rangle, \langle 000, \epsilon \rangle, \langle 00, 100 \rangle, \langle 0, 1 \rangle, \langle \epsilon, 1 \rangle$ and outputs thus 1. The following of the section will describe how to simulate any skipping cyclic tag system.

Principle of the design. Figure 2 presents the global design for our simulation on the example of the skipping tag system $(\epsilon, 100, 1, 0)$ with the same input word 010 as above. The simulation proceeds in forward-backward swipes of the encoding of the current word. Each forward (left-to-right) swipe trims all the initial 0s (encoded as little bumps from above) from the beginning of the word until a 1 (encoded as flat from above) is met, then rushes to the end of the word to append the corresponding production. The following backward (right-to-left) swipe rewinds to the position in the word just after its first 1 while copying its letters down bellow for the reading of the next swipe. The construction continues until running out of letters in which case the folding gets trapped into a finite space and halts.

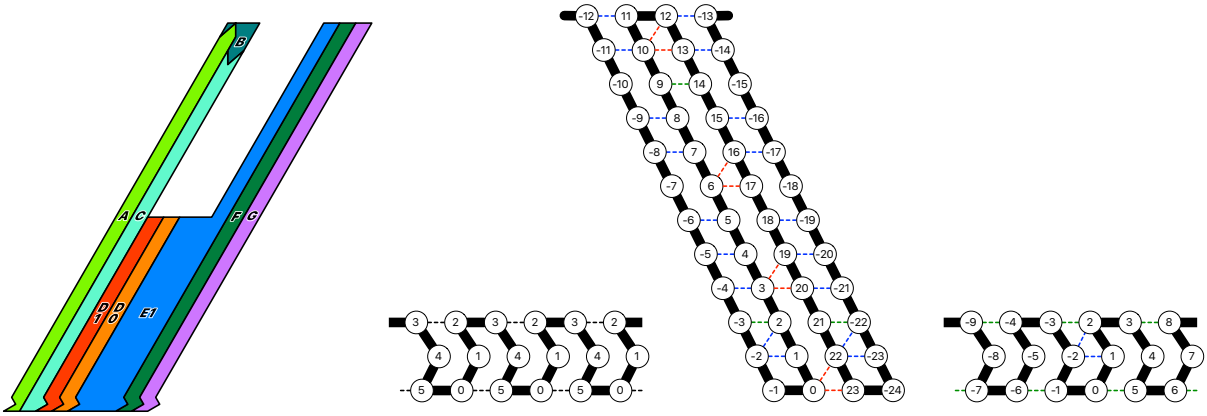

Figure 1 To the left: The production module (folded upright) corresponding to a production 10 in a tag system where all productions have length at most 3 (hence, Padding submodule E_1 takes parameter 1).
To the right: Simple Glider (left), Switchback (middle) and compatible Glider (right).

Production encoding. Each production of the tag system is encoded in the molecule as a module, all of equal length. Each *production module* is composed of the exact same elements, only the letters encoded inside each module changes from one production to another (see Fig. 1). Precisely, if $L = \max_i |p_i|$ denotes the maximum length of a production, the production module for p_i is the sequence of submodules $\langle \mathsf{A}, \mathsf{B}, \mathsf{C}, (\mathsf{D}_a)_{a=(p_i)_j : j=1..|p_i|}, \mathsf{E}_k, \mathsf{F}, \mathsf{G} \rangle$ with $k = L - |p_i|$.

Module A: Init is a simple module building a simple scaffold for the following modules; it always folds in the same way.

Module B: Empty word probe is a very short module that is sensitive to the presence of an non-empty word above it; if the word is empty, then it folds to the left, blocking the molecule into a finite space, halting thus the co-transcriptional folding and simulating the halt of the tag system. Otherwise, it folds to the right and the folding continues.

Module C: End of word probe is sensitive to the end of the word; if the end of word is reached, it folds in a way that initiates the appending of the letters of the production module; otherwise, it initiates the compact folding of the production module.

Modules D_0 and D_1: Letters encode the letters of the production; it can fold into two main forms: compact, where the letter are hidden from the reading head in Module G; or expanded, when the letters are appended at the end of the word.

Module E_k: Padding & Carriage return has two purposes: first, ensure that all production modules have the same length by padding with $k = L - |p_i|$ spaces each production p_i so that they all have the same length; second, reverse the direction of the folding to accomplish a "carriage return of the molecule" once the current production letters (in expanded form) have been appended to the word, marking the end of the forward swipe.

Module F: Term as for Module A, is used to built a scaffold along which the next module folds.

Module G: Read, Copy & Line Feed is the real "brain" of the molecule; in the forward swipe, it first reacts to the letters of the word by folding so as to skip the initial 0s until it finds a 1 which has the effect of mirroring the following production modules; when the production modules are mirrored, G folds in a way that copies the letters read above down bellow; then, at the end of the backward swipe, when it reaches the beginning of the first letter of the current word, the G spontaneously folds to extend further down bellow starting a new line for the next forward swipe to begin.

Designing the modules. The remaining of the section consists in explaining how to design the modules A, ..., G so that the resulting Oritatami system folds as shown in Fig. 2.

Basic scaffolding: Modules A and F. Our construction uses rigid scaffoldings named *gliders*, see Fig. 1 and [7]. Gliders are rigid (they support themselves) and require only few bonds (one every 3 positions on average). It is easy to check that glider folds as expected and requires only 6 different beads, corresponding to a period of the glider pattern. A and F use gliders to build a rigid scaffold along which the following modules will fold.

Adopting either a compact or expanded form: Modules D_0, D_1, E_k, and G. Our design requires to be able to store the letters of the production into a compact form inside the production module and to be able to expand them into a glider when appending the letter at the end of the word. The compact form is called *switchback*. Remarking that the sharp turns of the switchback are similar to the gliders, we have obtained a bonding scheme compatible to both switchback *and* glider as shown on Fig. 1. The magic resides in the fact the form is controlled by the placement of the first three beads: if they adopt a glider form, the rest of the molecule will fold into a glider; if they adopt the switchback form, then the rest of molecule as well. This allows us to have the modules D_0, D_1, E_k, and G to contract or expand at will by forcing the placement of the first three with strong bonding to the environment! Note that each of the switchback strands can be extended as much as wanted by repeating the same 12 beads, this allows to construct switchback compatible with glider with arbitrary multiple of 12 height.

Detecting ends: B, C, and G. End detection is obtained by realizing various level of attachment of a given module: by default it will fold in a certain way, but presented with some specific environment, it will bind strongly with it and change its shape. We refer to [7] and the folding of B for details.

Implementing various functions: G. G is a very sophisticated structure that needs to implement many different functions: reading, copying, and line feeding. It is also responsible for the major changes in the geometry of the folding by reversing the production modules. "Calling" the different functions is achieved by shifting the module along its environment. Precisely, on the one hand, in the upright conformation of a production module, the area bellow the production module is cleared and G will fold its first 8 beads bellow, shift its relative position to the preceeding module F. The effect is striking: G will fold as a glider and enter in its "reading" mode. On the other hand, in the mirrored and rotated conformations, the area above the production module is occupied and G naturally folds along F adopting its switchback shape activating its "copying" mode.

References

1 J. Atkins and W. E. Hart. On the intractability of protein folding with a finite alphabet of amino acids. *Algorithmica*, 25(2–3):279–294, 1999.

2 Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

3 Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding. *Journal of computational biology*, 5(3):423–465, 1998.

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

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

6 Cody Geary, Pierre-Étienne Meunier, Nicolas Schabanel, and Shinnosuke Seki. Efficient universal computation by molecular co-transcriptional folding (short announcement). In *DNA21*, page 39, 2015.

7 Cody Geary, Pierre-Étienne Meunier, Nicolas Schabanel, and Shinnosuke Seki. Folding turing is hard but feasible. `arXiv:1508.00510 [cs.CG]`, Nov. 2015.

8 Cody Geary, Pierre-Étienne Meunier, Nicolas Schabanel, and Shinnosuke Seki. `http://www.dailymotion.com/playlist/x4c560_nicolasschabanel_oritatami`. Folding Turing: videos of the folding of the Oritatami system simulating the Skipping Cyclic Tag System $(100, 1, 0, \epsilon)$ on input word 10, nov. 2015.

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

10 Boyle J, Robillard G, and Kim S. Sequential folding of transfer RNA. a nuclear magnetic resonance study of successively longer tRNA fragments with a common 5' end. *J Mol Biol*, 139:601–625, 1980.

11 Turlough Neary and Damien Woods. P-completeness of cellular automaton rule 110. In *ICALP*, volume LNCS 4051, pages 132–143, 2006.

12 M. Paterson and T. Przytycka. On the complexity of string folding. In F. Meyer and B. Monien, editors, *ICALP 1996*, volume 1099 of *LNCS*, pages 658–669. Springer Berlin Heidelberg, 1996.

13 Paul W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, March 2006.

14 R. Unger and J. Moult. Finding the lowest free energy conformation of a protein is an NP-hard problem: proof and implications. *Bulletin of Mathematical Biology*, 55(6):1183–1198, 1993.

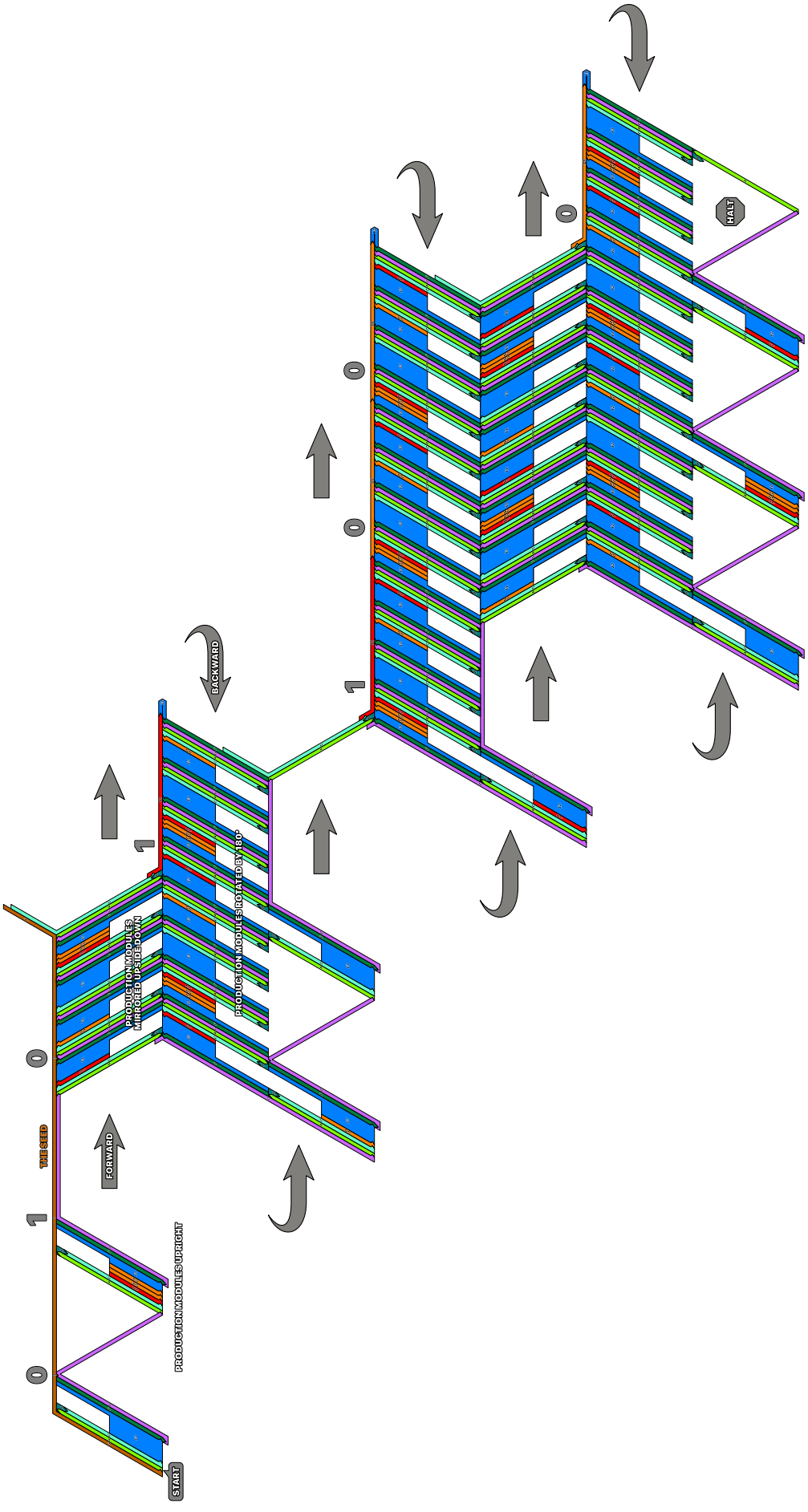

Figure 2 The design of the Oritatami simulation of the execution of skipping cyclic tag system $(\epsilon, 100, 1, 0)$ on input word 010. The little dents $\rangle\rangle$ in each module indicates the locations of the beginning and the end of its folding. Production modules are folded in three main ways: *Upright*, at the beginning of the forward swipe, to read the first letters of the word and trim the 0s until a 1 is found; then *Mirrored* upside-down, in the second phase of the forward swipe, when the first 1 has been found and the remaining letters are copied down bellow; and *Rotated* by 180°, in the backward phase, to copy the letters of the word down bellow from right to left. Note that all these symmetries preserve the neighborhood of the beads and thus do not perturbe the folding of each module. The only thing that impacts the folding is the change of environment: in the Upright folding, the environment is cleared at the basis of the folding (at the bottom); whereas in the Mirrored and Rotated folding, the environment is populated at the basis of the folding (on the top).