Proving the Turing Universality of Oritatami Co-Transcriptional Folding

Cody Geary, Pierre-Étienne Meunier, Nicolas Schabanel, Shinnosuke Seki

Proving the Turing Universality of oritatami Co-Transcriptional Folding

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 **Tapdance, Inria Paris, France.** pe@pijul.org
3 **CNRS, U. Lyon (LIP), France and IXXI, U. Lyon, France.** perso.ens-lyon.fr/nicolas.schabanel/
4 **oritatami Lab, University of Electro-Communications, Tokyo, Japan.** www.sseki.lab.uec.ac.jp/

Abstract

We study the oritatami model for molecular co-transcriptional folding. In oritatami systems, the transcript (the "molecule") folds as it is synthesized (transcribed), according to a local energy optimisation process, which is similar to how actual biomolecules such as RNA fold into complex shapes and functions as they are transcribed. We prove that there is an oritatami system embedding universal computation in the folding process itself.

Our result relies on the development of a generic toolbox, which is easily reusable for future work to design complex functions in oritatami systems. We develop "low-level" tools that allow to easily spread apart the encoding of different "functions" in the transcript, even if they are required to be applied at the same geometrical location in the folding. We build upon these low-level tools, a programming framework with increasing levels of abstraction, from encoding of instructions into the transcript to logical analysis. This framework is similar to the hardware-to-algorithm levels of abstractions in standard algorithm theory. These various levels of abstractions allow to separate the proof of correctness of the global behavior of our system, from the proof of correctness of its implementation. Thanks to this framework, we were able to computerise the proof of correctness of its implementation and produce certificates, in the form of a relatively small number of proof trees, compact and easily readable/checkable by human, while encapsulating huge case enumerations. We believe this particular type of certificates can be generalised to other discrete dynamical systems, where proofs involve large case enumerations as well.

Digital Object Identifier 10.4230/LIPIcs...1

1 Introduction

Oritatami model was introduced in [5] to try to understand the kinetics of co-transcriptional folding. This process has been shown to play an important role in the final shape of biomolecules [1], especially in the case of RNA [4]. The rationale of this choice is that the wetlab version of Oritatami already exists, and has been successfully used to engineer shapes with RNA in the wetlab [6].

In oritatami, we consider a finite set of *bead types*, and a periodic sequence of *beads*, each of a specific bead type. Beads are attracted to each other according to a fixed symmetric relation, and in any folding (a folding is also called a *configuration*), whenever two beads attracted to each other are found at adjacent positions, a *bond* is formed between them.

At each step, the latest few beads in the sequence are allowed to explore all possible positions, and we keep only those positions that minimise the energy, or otherwise put, those positions that maximise the number of bonds in the folding. "Beads" are a metaphor for domains, i.e. subsequences, in RNA and DNA.

Previous work on oritatami includes the implementation of a binary counter [5], the Heighway dragon fractal [11], folding of shapes at small scale [3], and NP-hardness of the rule minimization [14, 8] and of the equivalence of non-deterministic oritatami systems [9].

Main result. In this paper, we construct a "universal" set of 542 bead types, along with a single universal attraction rule for these bead types, with which we can simulate any tag system, and therefore any Turing machine $\mathcal{M}$, within a polynomial factor of the running time $\mathcal{M}$. The reduction proceeds as follows:

$$\text{Turing machine} \xrightarrow{[15,\,12]} \text{Cyclic tag system} \xrightarrow{\text{Prop. 2}} \text{Skipping cyclic tag system} \xrightarrow{\text{Thm. 6}} \text{Oritatami system}$$

Our result relies on the development of a generic toolbox, which is easily reusable for future work to design complex functions in oritatami systems.

Proving our designs. The main challenge we faced in this paper was the size of our constructions: indeed, while we developed higher-level geometric constructs to program useful shapes, there is a large number of possible interactions between all different parts of the sequence. Getting solid proofs on large objects is a common problem in discrete dynamical systems, for instance on cellular automata [7, 2] or tile assembly systems [10]. In this paper, we introduce a general framework to deal with that complexity, and prove our constructions rigorously. This method proceeds by decomposing the sequence into different *modules*, and the space into different areas: *blocks*, where exactly one step of the simulation is performed, which are composed of *bricks*, where exactly one module grows. We can then reason on the modules separately, and only deal with interactions at the border between all possible modules that can have a common border.

2 Definitions and Main results

2.1 Oritatami Systems

Let B be a finite set of *bead types*. A *configuration* c of a bead type sequence $p \in B^* \cup B^{\mathbb{N}}$ is a directed self-avoiding path in the triangular lattice $\mathbb{T}$,[1] where for all integer i, vertex c_i of c is labelled by p_i. c_i is the *position* in $\mathbb{T}$ of the $(i+1)$th bead, of type p_i, in configuration c. A *partial configuration* of a sequence p is a configuration of a prefix of p.

For any partial configuration c of some sequence p, an *elongation* of c by k beads (or k-*elongation*) is a partial configuration of p of length $|c| + k$ extending by k positions the self-avoiding path c. We denote by $\mathcal{C}_p$ the set of all partial configurations of p (the index p will be omitted when the context is clear). We denote by $c^{\triangleright k}$ the set of all k-elongations of a partial configuration c of sequence p.

Oritatami systems. An *oritatami system* $\mathcal{O} = (p, \heartsuit, \delta)$ is composed of (1) a (possibly infinite) bead type sequence p, called the *transcript*, (2) an *attraction rule*, which is a symmetric relation $\heartsuit \subseteq B^2$, (3) a parameter δ called the *delay*. $\mathcal{O}$ is said *periodic* if p is infinite and periodic. Periodicity ensures that the "program" p embedded in the oritatami system is

[1] The triangular lattice is defined as $\mathbb{T} = (\mathbb{Z}^2, \sim)$, where $(x, y) \sim (u, v)$ if and only if $(u, v) \in \cup_{\epsilon=\pm 1}\{(x+\epsilon, y), (x, y+\epsilon), (x+\epsilon, y+\epsilon)\}$. Every position (x, y) in $\mathbb{T}$ is mapped in the euclidean plane to $x \cdot \vec{E} + y \cdot \vec{SW}$ using the vector basis $\vec{E} = (1, 0)$ and $\vec{SW} = \text{RotateClockwise}\left(\vec{E}, 120°\right) = (-\frac{1}{2}, -\frac{\sqrt{3}}{2})$.

finite (does not hardcode any specific behavior) and at the same time allows arbitrary long computation.

We say that two bead types a and b *attract* each other when $a \heartsuit b$. Furthermore, given a (partial) configuration c of a bead type sequence q, we say that there is a *bond* between two adjacent positions c_i and c_j of c in $\mathbb{T}$ if $q_i \heartsuit q_j$ and $|i-j|>1$. The *number of bonds* of configuration c of q is denoted by $H(c) = |\{(i,j) : c_i \sim c_j,\ j > i+1, \text{ and } q_i \heartsuit q_j\}|$.

Oritatami dynamics. The folding of an oritatami system is controlled by the delay δ. Informally, the configuration grows from a *seed configuration* (the input), one bead at a time. This new bead adopts the position(s) that maximise the potential number of bonds the configuration can make when elongated by δ beads in total. This dynamics is *oblivious* as it keeps no memory of the previously preferred positions; it differs thus slightly from the hasty dynamics studied in [5]; it might also be considered as closer to experimental conditions such as in [6].

Formally, given an oritatami system $\mathcal{O} = (p, \heartsuit, \delta)$ and a *seed configuration* σ of a *seed bead type sequence* s, we denote by $\mathcal{C}_{\sigma,p}$ the set of all partial configurations of the sequence $s \cdot p$ elongating the seed configuration σ. The considered *dynamics* $\mathscr{D} : 2^{\mathcal{C}_{\sigma,p}} \to 2^{\mathcal{C}_{\sigma,p}}$ maps every subset S of partial configurations of length ℓ elongating σ of the sequence $s \cdot p$ to the subset $\mathscr{D}(S)$ of partial configurations of length $\ell + 1$ of $s \cdot p$ as follows:

$$\mathscr{D}(S) = \bigcup_{c \in S} \operatorname*{arg\,max}_{\gamma \in c^{\triangleright 1}} \left(\max_{\eta \in \gamma^{\triangleright(\delta-1)}} H(\eta) \right)$$

The possible configurations at time t of the oritatami system $\mathcal{O}$ are the elongations of the seed configuration σ by t beads in the set $\mathscr{D}^t(\{\sigma\})$.

We say that the oritatami system is *deterministic* if at all time t, $\mathscr{D}^t(\{\sigma\})$ is either a singleton or the empty set. In this case, we denote by c^t the configuration at time t, such that: $c^0 = \sigma$ and $\mathscr{D}^t(\{\sigma\}) = \{c^t\}$ for all $t > 0$; we say that the partial configuration c^t *folds (co-transcriptionally) into* the partial configuration c^{t+1} deterministically. In this case, at time t, the $(t+1)$-th bead of p is placed in c^{t+1} at the position that maximises the number of bonds that can be made in a δ-elongation of c^t.

We say that the oritatami system *halts* at time t if t is the first time for which $\mathscr{D}^t(\{\sigma\}) = \varnothing$. The folding process may only stop because of a geometric obstruction (no more elongation is possible because the configuration is trapped in a closed area).

Please refer to Fig. 1(d) and 1(e) for examples of the dynamical folding of a transcript. Observe that a given transcript may fold (deterministically) into different paths because of its interactions with its local environment (see section 2.3 for more details).

2.2 Main result

Our main result consists in proving the following theorem that demonstrates that oritatami systems are able to complete arbitrary Turing computation. It shows in particular that deciding whether a given oritatami system folds into a finite size shape for a given seed is undecidable.

▶ **Theorem 1** (Main result). *There is a fixed set B of 542 bead types with a fixed attraction rule $\heartsuit$ on B, together with two encodings:*

- *π that maps in polynomial time, any single tape Turing machine $\mathcal{M}$ to a bead type sequence $\pi_{\mathcal{M}} \in B^*$;*

- *(s, σ) that maps in polynomial-time, any single-tape Turing machine $\mathcal{M}$ and any input x of $\mathcal{M}$ to a seed configuration $\sigma_{\mathcal{M}}(x)$ of a bead type sequence $s_{\mathcal{M}}(x)$ of length $O_{\mathcal{M}}(|x|)$, linear in the size of the input x (and polynomial in $|\mathcal{M}|$);*

such that: For any single tape Turing machine $\mathcal{M}$ and every input x of $\mathcal{M}$, the deterministic and periodic oritatami system $\mathcal{O}_{\mathcal{M}} = ((\pi_{\mathcal{M}})^\infty, \heartsuit, 3)$ whose transcript has period $\pi_{\mathcal{M}}$ and whose delay is $\delta = 3$, halts its folding from the seed configuration $\sigma_{\mathcal{M}}(x)$ if and only if $\mathcal{M}$ halts on input x. Furthermore, for all t and all input x of $\mathcal{M}$, if $\mathcal{M}$ halts on x after t steps, then the folding of $\mathcal{O}_{\mathcal{M}}$ from seed configuration $\sigma_{\mathcal{M}}(x)$ halts after folding $O_{\mathcal{M}}(t^4 \log^2 t)$ beads.

There is one Turing-universal periodic transcript. Note that if we apply this theorem to an intrinsically universal single tape Turing machine $\mathcal{U}$ (see [13]), then we obtain one single *absolutely fixed* transcript $\pi_{\mathcal{U}}$ such that the deterministic and periodic oritatami system $\mathcal{O}_{\mathcal{U}} = ((\pi_{\mathcal{U}})^\infty, \heartsuit, 3)$ with 542 bead types can simulate efficiently the halting of any Turing machine $\mathcal{M}$ on any input x using a suitable seed configuration obtained via the encoding of $\mathcal{M}$ and x in $\mathcal{U}$. It follows that this absolutely fixed oritatami system consisting of one single periodic transcript is able of arbitrary Turing computation.

From now on, we only consider deterministic periodic oritatami systems with delay $\delta = 3$.

2.3 Basic design tool: Glider/Switchback

As a warm-up, let us introduce a special type of bead sequence (see Fig. 1) that, depending on the initial context of its folding, either folds as a *glider* (a long and thin self-supported shape heading in a fixed direction) or as *switchbacks* (a narrow and high shape allowing compact storage). This only requires a small number of distinct beads types (12 per switchbacks, that can be repeated every 4 switchbacks). This is achieved by designing a rule with minimum interactions ensuring minimum interferences between both folding patterns. Compatibility between the glider and the turns in switchbacks is ensured by aligning the switchback turns with the turns of the glider, exploiting thus the similarity of their finger-like shape there.

This glider/switchback sequence will be used to store (as switchbacks) and expose (as glider) specific information encoded in the transcript when needed.

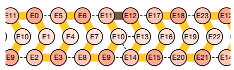
2.4 Skipping Cyclic Tag Systems and Turing-Universality

Our proof of the Turing-universality of oritatami systems consists in simulating a special kind of cyclic tag systems (CTS), called skipping cyclic tag system. Cook introduced CTS in [2] and proved that they combined the tremendous advantage of simulating efficiently any Turing machines, while not requiring a random access lookup table, which makes simulation a lot easier.

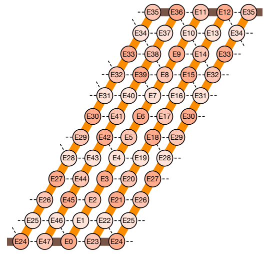
A skipping cyclic tag system (SCTS) consists of a cyclic list of n words $\alpha = \langle \alpha^0, \ldots, \alpha^{n-1} \rangle \in \{0, 1\}^*$, called *appendants*, and an initial *dataword* $u^0 \in \{0, 1\}^*$. Intuitively, α encodes the program and u^0 encodes the input. Its configuration at time t consists of a *marker* m^t, recording the index of the current appendant at time t, and a dataword u^t. Initially, $m^0 = 0$ and the dataword is u^0. At each time step t, the SCTS acts deterministically on configuration (m^t, u^t) in one of three ways:

(Halt step) If u^t is the empty word ϵ, then the SCTS halts;[2]

[2] Note that SCTS halting condition requires the dataword to be empty as opposed to [2, 15] where the

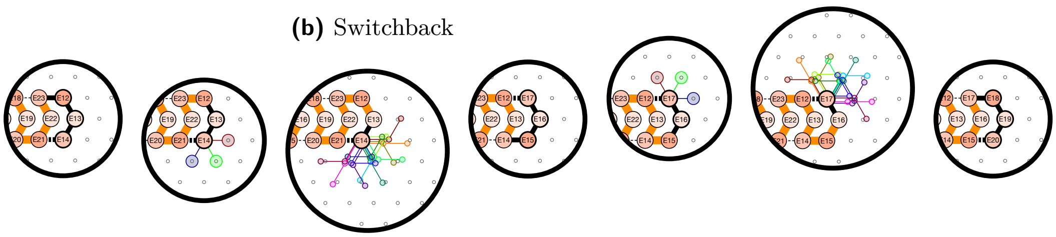


(a) Glider

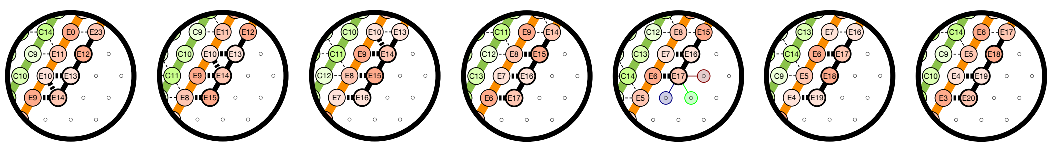


(b) Switchback

(c) Glider/Switchback turn folding compatibility.



(d) from left to right: the folding of the subsequence as a glider.



(e) from left to right: the folding of the subsequence as a switchback.

Figure 1 Glider/switchback subsequence. The folding path of the transcript is represented as the thick colorful line and the ♥-bonds between beads are represented as dashed lines. The bond-maximizing path for the $\delta = 3$ lastly produced beads is represented by a thick black line, possibly terminated by several colorful paths if several paths realize the maximum of number of bonds.

(Nop step) If the first letter u_0^t of u^t is 0**,** then u_0^t is deleted and the marker moves to the next appendant cyclically: i.e., $m^{t+1} = (m^t + 1) \bmod n$ and $u^{t+1} = u_1^t \cdots u_{|u^t|-1}^t$;

(Skip-append step) If $u_0^t = \mathtt{1}$, then u_0^t is deleted, the next appendant $\alpha^{(m^t+1 \bmod n)}$ is appended onto the right end of u^t, and the marker moves to the second next appendant: i.e., $u^{t+1} = u_1^t \cdots u_{|u^t|-1}^t \cdot \alpha^{(m^t+1 \bmod n)}$ and $m^{t+1} = (m^t + 2) \bmod n$.

For example, consider the SCTS $\mathcal{E} = (\langle \mathtt{110}, \epsilon, \mathtt{11}, \mathtt{0}\rangle; u^0 = \mathtt{010})$. Its execution $({}^{[m^t]}u^t)_t$ is:

$$ {}^{[0]}\mathtt{010} \to {}^{[1]}\mathtt{10} \xrightarrow[\text{[2:11]}]{\text{Append}} {}^{[3]}\mathtt{011} \to {}^{[0]}\mathtt{11} \xrightarrow[\text{[1:}\epsilon\text{]}]{\text{Append}} {}^{[2]}\mathtt{1} \xrightarrow[\text{[3:0]}]{\text{Append}} {}^{[0]}\mathtt{0} \to {}^{[1]}\,\mathtt{Halt} $$

Turing universality. A sequence of articles and thesis by Cook [2], and Neary and Woods [15, 12], allows to show that SCTS are able to simulate any Turing machine efficiently in the following sense: (proof deferred to appendix on page 17)

▶ **Proposition 2** ([15, 12])**.** *Let $\mathcal{M}$ be a deterministic Turing machine using a single tape. There is a polynomial algorithm that computes a skipping cyclic tag system $\mathcal{S}_\mathcal{M}$, together with a linear-time encoding $u_\mathcal{M}(x)$ of the input x of $\mathcal{M}$ as an input dataword for $\mathcal{S}_\mathcal{M}$, such that for all input x: $\mathcal{S}_\mathcal{M}$ halts on input dataword $u_\mathcal{M}(x)$ if and only if $\mathcal{M}$ halts on input x. Furthermore, for all t, if $\mathcal{M}$ halts after t steps, then $\mathcal{S}$ halts after $O_\mathcal{M}(t^2 \log t)$ steps. Moreover, the number of appendants of $\mathcal{S}$ is a multiple of 4.*

computation of a cyclic tag system is said to end also if it repeats a configuration.

In order to prove Theorem 1, we are thus left with proving that there is an oritatami system that simulates in quadratic time any SCTS system (see Theorem 6 in appendix for a precise statement).

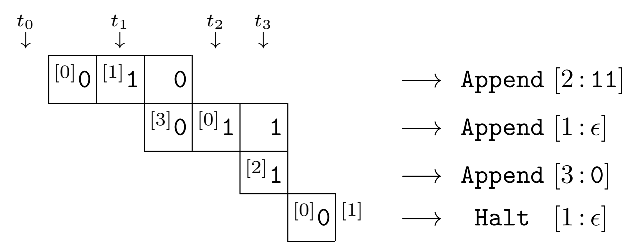
3 The block simulation of SCTS: Proving the correctness of local folding is enough

Given a SCTS $\mathcal{S}$, we design an oritatami system $\mathcal{O}_{\mathcal{S}}$ that folds into a version, at a larger scale, of the *annotated trimmed space-time diagram* of $\mathcal{S}$ (or *trimmed diagram* for short) defined as follows:

Trimmed diagram of SCTS. Any SCTS proceeds as follows: it trims all the leading 0s in the data word and then appends the currently marked appendant when it reads the first 1 (if any; otherwise it halts). It is thus natural to group all these steps (trim leading 0s and process the leading 1) as one single macro step. This motivates the following representation. Given a SCTS $(\alpha^0, \dots, \alpha^{n-1}; u^0)$, we denote by $0 \leqslant t_1 < t_2 < \cdots$ all the times t such that the dataword u^t starts with letter 1 and set $t_0 = -1$ by convention. Let us now group all deletion steps occurring during steps $t_i + 1$ to $t_{i+1} - 1$ by simply indicating in exponent the marker m^t before each letter read. In the case of our STCS $\mathcal{E}$, we have $t_0 = -1, t_1 = 1, t_2 = 3, t_3 = 4$ and its execution is now represented as:

$$^{[0]}0^{[1]}10 \xrightarrow[\text{[2:11]}]{\text{Append}} {}^{[3]}0^{[0]}11 \xrightarrow[\text{[1:}\epsilon\text{]}]{\text{Append}} {}^{[2]}1 \xrightarrow[\text{[3:0]}]{\text{Append}} {}^{[0]}0^{[1]}\ \texttt{Halt}$$

Now, let's align the resulting datawords in a 2D diagram according to their common parts:



This defines the *annotated trimmed space-time diagram* for the SCTS $\mathcal{E}$. Lemma 4 in appendix provides the formal definition for an arbitrary SCTS.

The transcript. The proof of Theorem 6 relies on constructing a transcript (and a fixed rule) that will reproduce faithfully the trimmed diagram of the simulated STCS. Figure 2 illustrates the folded configuration of the transcript corresponding to SCTS $\mathcal{E}$. Macroscopically, the transcript folds into a zig-zag sequence of *blocks*, each performing a specific operation.

The current dataword is encoded at the bottom of each row of blocks: 0s are encoded by a spike, and 1s are encoded by a flat surface.

the seed configuration encodes the initial dataword and opens the first zig row at which the folding of the transcript starts. Letters 0 and 1 are encoded by a *spike* (see Fig. 3(a)) and a *flat surface* (see Fig. 3(b)) respectively.

in each zig row (left to right), the transcript folds into a series of Read0▶ blocks (trimming the leading 0s from the dataword encoded above), then into a Read1▶ block, if the dataword contains a 1, or into a Halt block terminating the folding, otherwise; this is the *zig-up phase*. Then, the transcript starts the *zig-down phase* which consists in folding into Copy▶ block copying the letters encoded above to the bottom of the row; once the end of the dataword is reached, the transcript folds into an Append◀Return block which encodes, at the bottom of the row, the currently marked appendant, and finally, opens the next zag row.

in each zag row (right to left), the transcript folds into Copy◀ blocks copying the dataword encoded above to the bottom of the row. For the leftmost letter, the transcript folds into the special Copy▶LineFeed block which also opens the next zig row.

The transcript is a periodic sequence whose period is the concatenation of n bead type sequences **Appendant** α^0, ..., **Appendant** α^{n-1} called segments, each encoding one appendant.

Encoding of the marker. Read▶ and Append◀Return blocks consist of the folding of *exactly one* segment, whereas Copy▶, Copy◀ and Copy▶LineFeed consist of the folding of *exactly* n segments. It follows that the appendant encoded in the *leading* segment folded inside each block corresponds to the *currently marked* appendant in the simulated SCTS. As a consequence, the appendant contained in the folded Append◀Return block is indeed the appendant to be appended to the dataword.

The segment sequence. Each segment **Appendant** α^i encodes the appendant α^i as a sequence of $6 + |\alpha^i|$ modules: one of each module **A**, **B**, and **C**, then $|\alpha^i|$ of module **D**, then one of each module **E**, **F** and **G**. Each module is a bead type sequence that plays a particular role in the design:

Module A folds into the initial scaffold upon which the next modules rely.

Module B detects if the dataword is empty: if so, it folds to the left and the folding gets trapped in a closed space and halts; otherwise, it folds to the right and the folding continues.

Module C detects the end of the dataword and triggers the appending of the marked appendant accordingly.

Module D encodes each letter of the appendant.

Module E ensures by padding that all appendant sequences have the same length when folded (even if the appendant have different length). It serves two other purposes: Module **B** senses its presence to detect if the dataword is empty; and its folding initiates the opening the zag row once the marked appendant has been appended to the dataword.

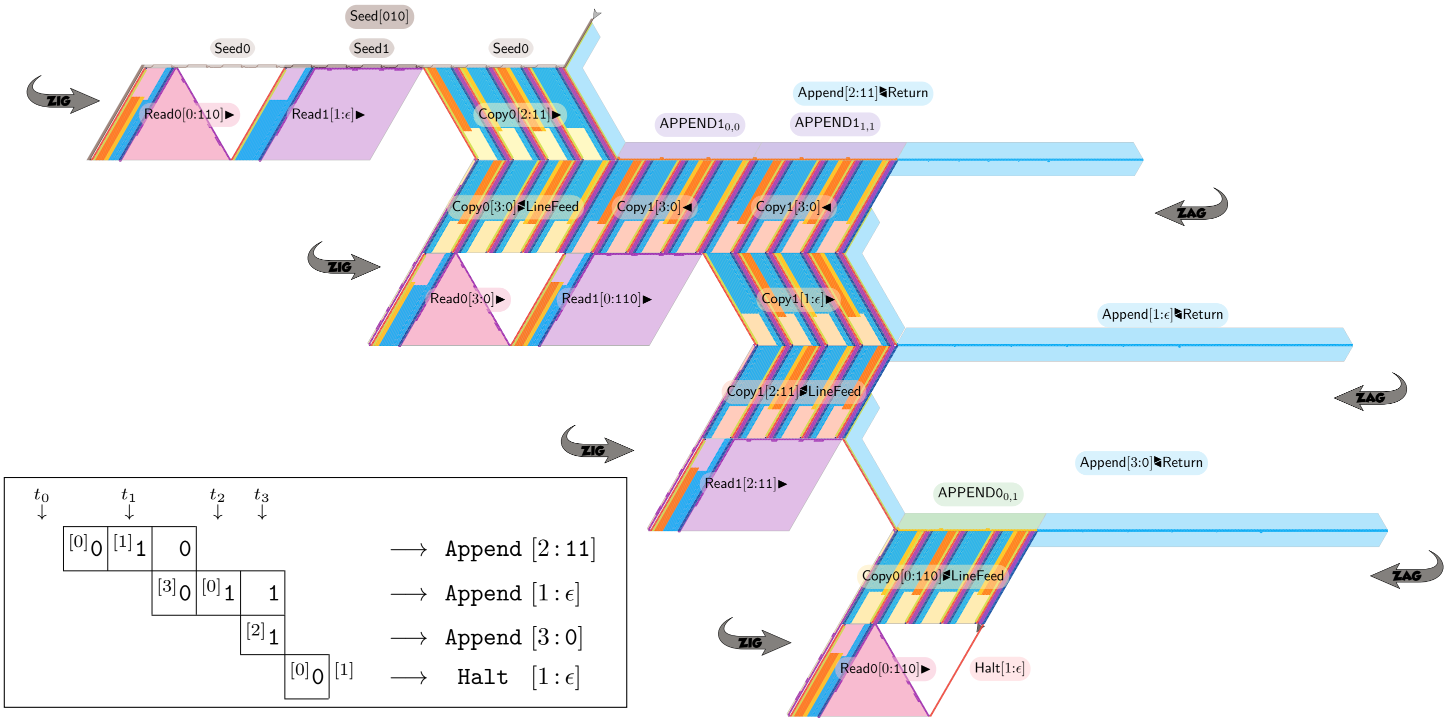
Module F is the scaffold upon which Module **G** folds. It is specially designed to induce two very distinct shapes on **G** depending on the initial shift of **G**. Furthermore, when Module **F** is exposed, Module **C** folds along **F** which triggers the appending of the marked appendant encoded by the modules **D** following **C**.

Module G is the "logical unit" of the transcript. It implements three distinct functions which are triggered by its interactions with its environment: Reading the leading letter of the dataword, Copying a letter of the dataword, and Opening the next zig row at the leftmost end of a zag row.

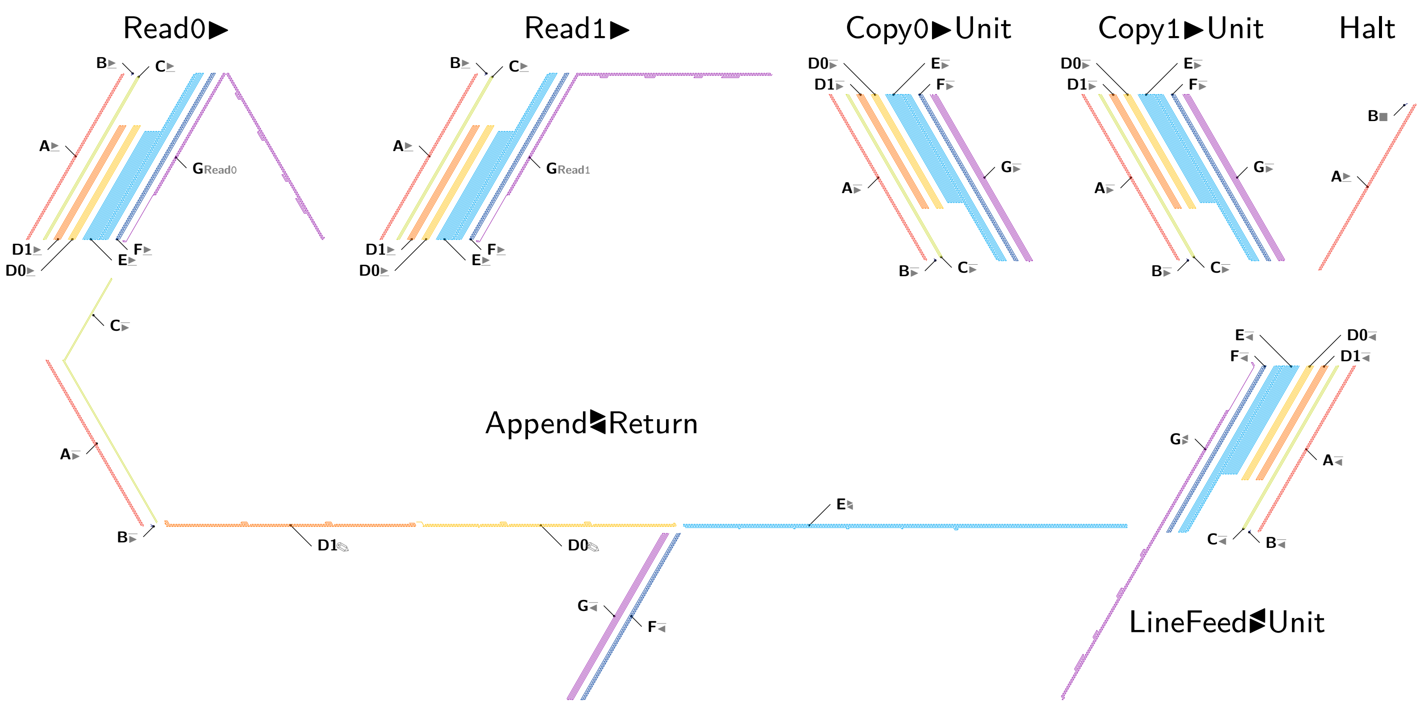
We call *bricks* the folding of each of these modules. The blocks into which the transcript folds, depend on the bricks in which its modules fold, as illustrated in Fig. 2(b). Please refer to sections C to F in appendix for the description of blocks in terms of bricks and of how they articulate with each other to produce the desired macroscopic folding pattern.

The full description of each of these sequences is given in Section F in appendix.

Let $\mathcal{S} = (\alpha^0, \alpha^1, \ldots, \alpha^{n-1}; u^0)$ be a skipping cyclic tag system, and, as before, let for all integer $i \geqslant 0$, t_i be the i^{th} step where u^t starts with 1 (starting from 0, i.e. t_0 is the



(a) Folding of the oritatami system simulating the STCS $\mathcal{E}$.



(b) Exploded view of the bricks and modules inside the blocks involved in the simulation above.

Figure 2 Folding of the transcript simulating the STCS $\mathcal{E}$, and some block internal structures.

first step where u^{t_0} starts with 1). The following lemma shows that the transcript described above folds indeed into blocks that simulates the trimmed diagram of $\mathcal{S}$. Proposition 2 and Theorem 6 are direct corollaries of this lemma.

▶ **Lemma 3** (Key lemma). *There is a bead type set B and a rule ♥ such that: for every SCTS $\mathcal{S}$, there are $\pi_{\mathcal{S}}$ and $(\sigma_{\mathcal{S}}, s_{\mathcal{S}})$ defined as in Theorem 1 such that, for every initial dataword u^0, the (possibly infinite) final folded path of the oritatami system $\mathcal{O}_{\mathcal{S}} = ((\pi_{\mathcal{S}})^\infty, ♥, \delta = 3)$ from the seed configuration $(\sigma(u^0), s(u^0))$ is exactly structured as the following sequence of blocks organized in zig and zag rows as follows: (recall Fig. 2(a))*

- *First, the block* Seed(u^0) *ending at coordinates* $(-1, 0)$.
- *Then, for $i \geqslant 0$, the i-th row consists of a zig row located between $y = 2(i-1)h + 1$ and $y = 2ih$, and a zag row located between $y = 2ih + 1$ and $y = 2(i+1)h$, composed as follows:*
  - **(Compute)** ***if*** $u^{1+t_i} = 0^r 1 \cdot s$ ***and if*** $s \neq \epsilon$ ***or*** $\alpha^{1+i+t_{i+1}} \neq \epsilon$: *then* $r = t_{i+1} - t_i - 1$ *and:*
    - *the i-th zig-row consists from left to right of the following sequence of blocks whose origins are located at the following coordinates:*

$\swarrow y$	$2ih$				$(2i-1)h+1$		
$\to x$	$ih + (1+t_i)W$	$\cdots$	$ih + (t_{i+1}-1)W$	$ih + t_{i+1}W$	$ih + (1+t_{i+1})W - 1$	$\cdots$	$ih + (\lvert s\rvert + t_{i+1})W - 1$	$ih + (1 + \lvert s\rvert + t_{i+1})W - 1$
Blocks	Read0▶	$\cdots$	Read0▶	Read1▶	Copy(s_0)▶	$\cdots$	Copy$(s_{\lvert s\rvert-1})$▶	Append$[\alpha^{1+i+t_{i+1}}]$◣Return
Marker	$i+1+t_i$	$\cdots$	$i+r+t_i$	$i+t_{i+1}$	$i+1+t_{i+1}$	$\cdots$	$i+1+t_{i+1}$	$i+1+t_{i+1}$

This row ends at position $((i+1)h + (1 + \lvert s\rvert + \lvert\alpha^{i+1+t_{i+1}}\rvert + t_{i+1})W - 7, 2ih + 2)$.

- *the i-th zag-row consists from right to left of the following sequence of blocks whose origins are located at the following coordinates:*

$\swarrow y$	$2ih+1$			
$\to x$	$(i+1)h + (2+t_{i+1})W - 8$	$(i+1)h + (3+t_{i+1})W - 8$	$\cdots$	$(i+1)h + (1 + \lvert v\rvert + t_{i+1})W - 8$
Blocks	Copy(v_0)◢LineFeed	Copy(v_1)◀	$\cdots$	Copy$(v_{\lvert v\rvert-1})$◀
Marker	$i+2+t_{i+1}$	$i+2+t_{i+1}$	$\cdots$	$i+2+t_{i+1}$

where $v = u^{1+t_{i+1}} = s \cdot p_{i+1+t_{i+1}} \neq \epsilon$ *(as s and $\alpha^{i+1+t_{i+1}}$ are not both ϵ). This row ends at position* $((i+1)h + (1+t_{i+1})W - 1, 2(i+1)h)$.

  - **(Halt 1)** ***if*** $u^{1+t_i} = 0^r 1$ ***and*** $\alpha^{1+i+t_{i+1}} = \epsilon$: *then* $r = t_{i+1} - t_i - 1$ *and the last rows of the configuration consists from left to right of the following sequence of blocks located at the following coordinates:*

$\swarrow y$	$2ih$				$(2i-1)h+1$	$2(i+1)h$
$\to x$	$ih + (1+t_i)W$	$\cdots$	$ih + (t_{i+1}-1)W$	$ih + t_{i+1}W$	$ih + (1+t_{i+1})W - 1$	$(i+1)h + (1+t_{i+1})W$
Blocks	Read0▶	$\cdots$	Read0▶	Read1▶	CarriageReturn◣LineFeed◢	Halt
Marker	$i+1+t_i$	$\cdots$	$i+t_{i+1}-1$	$i+t_{i+1}$	$i+1+t_{i+1}$	$i+2+t_{i+1}$

  - ***finally,*** **(Halt 2)** ***if*** $u^{1+t_i} = 0^r$ ***for some*** $r \geqslant 0$: *then the i-th zig-row is last row of the configuration and consists of the following sequence of blocks located at the following coordinates:*

$\swarrow y$	$2ih$			
$\to x$	$ih + (1+t_i)W$	$\cdots$	$ih + (r+t_i)W$	$ih + (1+r+t_i)W$
Blocks	Read0▶	$\cdots$	Read0▶	Halt
Marker	$i+1+t_i$	$\cdots$	$i+r+t_i$	$i+r+1+t_i$

The following sections are dedicated to the proof of Key Lemma 3.

4 Advanced Design Tool box

In this section, we present several key tools to program Oritatami design and which we believe to be generic as they allowed us to get a lot of freedom in our design.

4.1 Implementing the logic

As in [5], the internal state of our "molecular computing machinery" consists essentially of two parameters: 1) the *position inside the transcript* of the part currently folding; and 2) the *entry point* of transcript inside the environment. Indeed, depending on the entry point or the position inside the transcript, different beads will be in contact with the environment and thus different *functions* will be applied as a result of their interactions. This happens during the zig phase: in the first (zig-up) part, the transcript starts folding at the bottom, forcing the modules G to fold into G▶Read bricks; whereas during the second (zig-down) part, the transcript starts folding at the top, forcing the modules G to fold into G▶Copy bricks instead. Similarly, the *memory* of the system consists of the beads already placed on the surrounding of the area currently visited (the *environment*). This happens in every row of the folding: depending on the letter encoded at the bottom of the row above, the modules G fold into G▶Read0 or G▶Read1 bricks (zig-up phase), G▶Copy0 or G▶Copy1 bricks (zig-down phase), and G◀Copy0 or G◀Copy1 bricks (zag phase).

At different places, we need the transcript to read information from the environment and trigger the appropriate folding. This is obtained through different mechanisms.

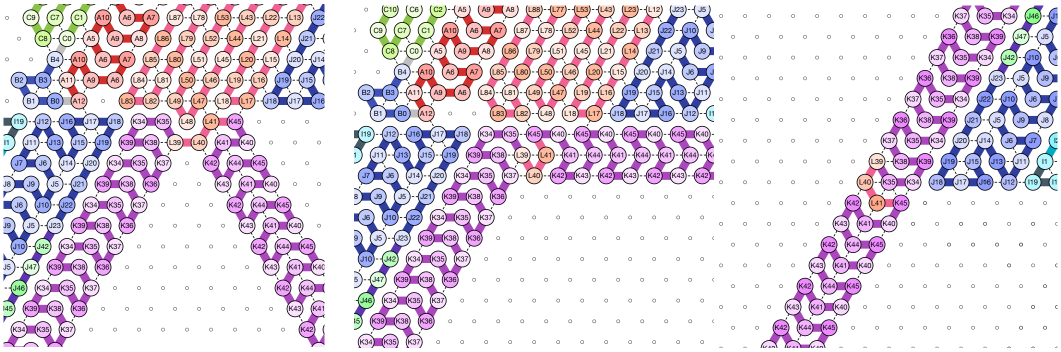
Default folding. By default, during the zig-up phase, B is attracted to the left by F and folds to the right only in presence of E above. This allows to continue the folding only if the tape word is not empty or to halt it otherwise (see Figure 27 in appendix).

Geometry obstruction. An typical example is illustrated by G. During the zig-up phase where the absence of environment below the block Read▶ allows G to fold downward at the beginning (see Figure 41) which shift the transcript by 7 beads along F resulting in G to adopt the glider-shape (more details on this mechanism in the next section). Whereas during the zig-down phase, G cannot make this loop because it is occupied by a previously placed G. This results in a perfect alignment of G with F whose strong attraction forces G to adopt the switchback shape (see Figure 43).

Geometry of the environment. Figure 3 shows how the shape of the environment is used to change the direction of G in glider-shape. This results in modifying the entry point in the environment and allows the Oritatami system to trim the leading 0s in the tape word by going back to the same entry point (Fig. 3(a)), switch from zip-up to zig-down phase when reading a 1 by opening the next block from the top (Fig. 3(b)), and from zag to zig-up phase when it has rewind to the beginning of the tape word, by getting down to the bottom of the next zig row (Fig. 3(c)).

4.2 Easing the design: getting the freedom you need

Several key tools allowed to ease considerably our design, and even in some cases to make it feasible. These tools are generic enough to be considered as *programming paradigms*. One main difficulty we faced is that the different functions one wants to implement tend to concentrate at the same "hot-spots" in the transcript. A typical example is the midpoint of G where one wants to implement all the functions: Read, Copy and Line Feed. The following powerful tools allow to overcome these difficulties:



(a) G bounces southeastward in presence of a spike encoding a 0 and folds into G ▶ Read0.

(b) G bounces eastward on a flat surface encoding a 1, and folds into G ▶ Read1.

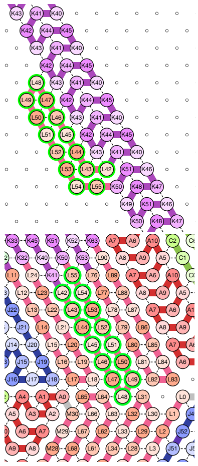
(c) G goes straight southwestward in absence of obstacle, and folds into G ↯ LineFeed.

Figure 3 The interactions of module G in "glider"-mode with different environments result in heading to different entry points to the next area of the folding.

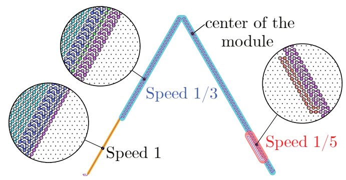
Socks work by letting a glider/switchback module fold into a switchback turn conformation for some time when it would otherwise fold into a glider. Examples are given in Figure 4. They are easy to implement: indeed, the socks naturally adopt the same shape as the corresponding switchback turn and require thus *no extra interfering bonds*. They allow a lot of freedom in the design, for several reasons:

- First, they simplify the design of important switchback part by *lifting the need for implementing the glider configuration* for that part, as shown in Figure 4(a).
- Second, a glider naturally progresses at speed 1/3. Adding a sock allows us to *slow its progression down* to speed 1/5 for some time (see Fig. 4(b)) and therefore to realign them. We used that feature repeatedly to "shift" some modules: starting the folding at an initial speed-1 (i.e. straight line) and then compensating for that speed later on by introducing socks (see Fig. 4(b)). This is a key point in our design, as it allowed us to *spread apart* the Read and Copy functions into different subsequences of module G, and therefore to get less constraints on our rule design. In the specific case of module G, the Copy-function occurs at the center of the module, while the Read-function is implemented earlier in module! (see section F.10 for full details)
- Finally, socks allow to prevent unwanted interactions between beads by *concealing* potentially harmful beads in unreachable area as in Figure 4(c).

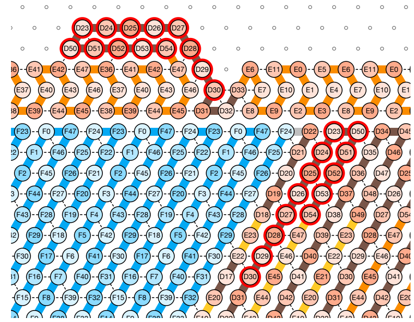
Exponential bead type coloring is a key tool to allow module G to fold into different shapes, glider or switchback, along module F, when folding in the Read▶ configuration. The problem it solves is that in order for G to fold into switchbacks, we need strong interactions between G and its neighboring module F (see Fig. 41), whereas in order for G to fold as glider, we want to avoid those interactions (see Fig. 43). This is made possible because gliders progress at speed 1/3 while switchbacks progress at speed 1. Using a power-of-3 coloring, we manage to easily achieve these contradicting goals altogether (the construction is analysed in Lemma 11 in Section G.1).



(a) Easing the design of switchback/glider by letting the switchback (in green) folds in its natural shape at its extremities even in "glider"-mode.



(b) Module **G**: Realigning a pattern by slowing its folding down at the end to compensate speeding it at the beginning.



(c) Module **D**: Preventing unwanted interactions between the beads outlined in red and the ones below, by concealing the red ones on top of the glider.

Figure 4 Different uses of socks: (a) Easing bond design; (b) Delaying; (c) Preventing unwanted interactions.

5 Correctness of local folding: Proof tree certificates

The goal of this section is to conclude the proof of our design by proving Key Lemma 3. The proof works by induction, assuming that the preceding beads of the transcript fold at the locations claimed by the lemma. We proceed in 3 steps:

- We first enumerate all the possible environments for every part of the transcript. As, we carefully aligned our design, most of the beads only see a small number of different environments.
- For the few cases (three in total) where the number of environments is unbounded, we give an explicit proof of correctness of their folding (Lemmas 9, 10, and 11 in section G.1). This is where the concealing feature of socks and the exponential bead type coloring play a crucial role.
- For all the other cases, we designed human-checkable computer-generated certificates, called *proof trees*. It consists in listing in a compact but readable manner all the possible paths for the transcript in every possible environment. In order to match human readability, paths with identical bonding patterns are grouped into one single ball. Balls containing the paths maximizing the number of bonds are highlighted in bold and organized in a tree. This reduces the number of cases to less than 5 balls in most of the levels of the tree, achieving human-checkability of the computed certificate (see Fig. 5 in appendix). Proof trees are available at `https://www.irif.fr/~nschaban/oritatami/`

  This ensures the highest level of certification of the correctness of our design.

References

1 J. Boyle, G. Robillard, and S. Kim. 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.

2 M. Cook. Universality in elementary cellular automata. *Complex Systems*, 15:1–40, 2004.

3 E.D. Demaine, J. Hendricks, M. Olsen, M.J. Patitz, T. Rogers N. Schabanel, S. Seki, and H. Thomas. Know when to fold 'em: Self-assembly of shapes by folding in oritatami. In *DNA*, 2018. To be published.

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

5 C. Geary, P.-É. Meunier, N. Schabanel, and S. Seki. Programming biomolecules that fold greedily during transcription. In *MFCS 2016*, volume LIPIcs 58, pages 43:1–43:14, 2016.

6 C. Geary, P. W. K. Rothemund, and E. S. Andersen. A single-stranded architecture for cotranscriptional folding of RNA nanostructures. *Science*, 345:799–804, 2014.

7 P. Gács. Reliable cellular automata with self-organization. In *FOCS*, pages 90–99, 1997.

8 Y-S. Han and H. Kim. Ruleset optimization on isomorphic oritatami systems. In *Proc. DNA23*, LNCS 10467, pages 33–45. Springer, 2017.

9 Y.-S. Han, H. Kim, M. Ota, and S. Seki. Nondeterministic seedless oritatami systems and hardness of testing their equivalence. In *DNA*, volume LNCS 9818, pages 19–34, 2016.

10 L. Kari, S. Kopecki, P.-É. Meunier, M. J. Patitz, and S. Seki. Binary pattern tile set synthesis is np-hard. In *ICALP*, pages 1022–1034, 2015.

11 Yusei Masuda, Shinnosuke Seki, and Yuki Ubukata. Towards the algorithmic molecular self-assembly of fractals by cotranscriptional folding. In *CIAA 2018*, LNCS, 2018. in press.

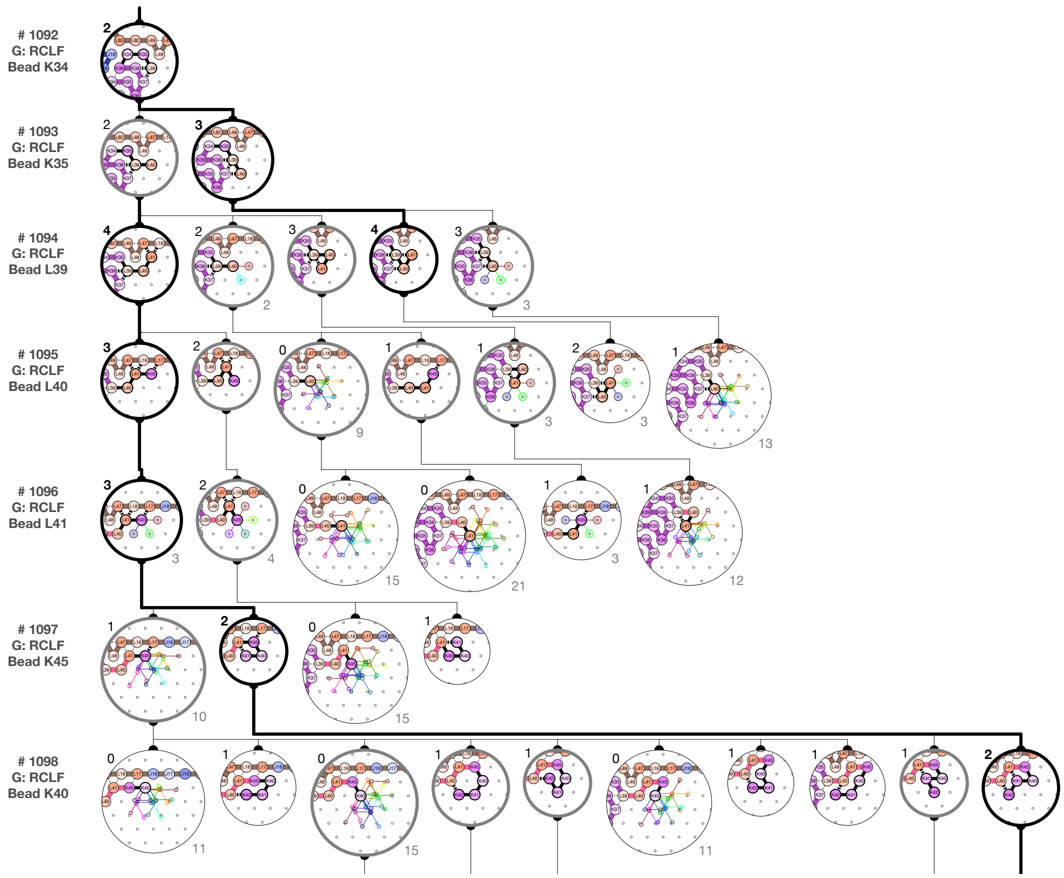
12 Turlough Neary. *Small universal Turing machines.* PhD thesis, NUI, Maynooth, 2008.

13 Nicolas Ollinger. The quest for small universal cellular automata. In *ICALP*, volume LNCS 2380, pages 318–329, 2002.

14 Makoto Ota and Shinnosuke Seki. Ruleset design problems for oritatami systems. *Theor. Comput. Sci.*, 671:26–35, 2017.

15 Damien Woods and Turlough Neary. On the time complexity of 2-tag systems and small universal Turing machines. In *FOCS*, pages 439–448, 2006.

Please find next the omitted part of article due to space constraints.



(a) Proof tree for the glider turn in **G ▸ Read0**.

(b) Proof tree for the glider turn in **G ▸ Read1**.

Figure 5 Two examples of proof trees for the same subsequence in two different environements. The number at the upper-left corner of every ball stands for the number of bonds for the path inside the ball. The number at the lower right corner of each ball stands for the number of paths grouped in the ball, allowing to check that no path was omitted. Balls highlighted in black bold contain the bonds-maximizing paths. Balls highlighted in grey bold contain the paths that places the bead at the same location as the bonds-maximizing paths, and which must thus be considered in the next level as well.

A Skipping Cyclic Tag Systems

Notations. We index the letters of every word $u = u_0 \ldots u_{|u|-1}$ from 0 to $|u| - 1$. Given two words u and v, we denote by $u \cdot v$ their concatenation: $u \cdot v = u_0 \ldots u_{|u|-1} v_0 \ldots v_{|v|-1}$. We denote by u^∞ the oneway infinite periodic word $u \cdot u \cdots$. For all $i \leqslant j$, we denote by $u_{i..j}$ the (possibly empty) factor $u_{\max(0,i)} \ldots u_{\min(j,|u|-1)}$. The empty word is denoted by ϵ. The indices in the notation $O_L()$ where L is a list of variables (for instance $L = A, B$) indicates that the constant in the $O()$ only depends on the variable in L (for instance A and B) and on no other values.

A.1 Trimmed diagram

The following lemma gives the formal description of the trimmed diagram of a SCTS $(\alpha; u^0)$ with marker m^t. Recall that t_i is the i-th time t such that the dataword u^t starts with letter 1 ($t_0 = -1$ by convention).

▶ **Lemma 4.** *The annotated word on the row i (indexed from $i = 0$) of the trimmed diagram is: (the markers in exponent are computed modulo n)*

- ***if $u^{1+t_i} = 0^r 1 \cdot s$ for some $r \geqslant 0$ and $s \in \{0, 1\}^*$:*** *then, $r = t_{i+1} - t_i - 1$ and the annotated word on row i is ${}^{[i+1+t_i]}0 \ldots {}^{[i-1+t_{i+1}]}0^{[i+t_{i+1}]}1 \cdot s$ whose first letter is placed in column $t_i + 1$ (assuming the leftmost column is indexed by 0);*
- ***if $u^{1+t_i} = 0^r$ for some $r > 0$:*** *then, row i is the last row of the diagram and its annotated word is ${}^{[i+1+t_i]}0 \ldots {}^{[i+t_i+r]}0^{[i+t_i+r+1]}$ and starts at column $t_i + 1$.*

Proof sketch. Simply observe that $m^{t_i} = i + t_i \bmod n$, as indeed exactly t_i letters have been read and exactly i appending steps have occurred before reading the i-th 1. ◀

A.2 Turing-universality of Skipping Cyclic Tag Systems

This proof makes use of the time-efficient reduction from Turing machines to cyclic tag systems (CTS) designed in [12, Theorem 4.3.2 p. 65], improving on [2, 15].

A cyclic tag system $\mathcal{C} = (\alpha^0, \ldots, \alpha^{n-1}; v^0)$ consists of a list of n appendants $\alpha^0, \ldots, \alpha^{n-1} \in \{0, 1\}^*$ and an initial dataword $v^0 \in \{0, 1\}^*$. Its configuration at time t consists of a *marker* $m^t = t \bmod n$, recording the index of the current appendant at time t, and a dataword v^t. Initially, $m^0 = 0$ and the dataword is v^0. At each time step t, the CTS acts deterministically on configuration (m^t, v^t) in one of three ways:

(Halt step) If v^t is the empty word ϵ, then the CTS halts;

(Nop step) If the first letter v_0^t of v^t is 0, then v_0^t is deleted and the marker moves to the next appendant cyclically: i.e., $m^{t+1} = (m^t + 1) \bmod n$ and $v^{t+1} = v_1^t \cdots v_{|v^t|-1}^t$;

(Append step) If $v_0^t = 1$, then v_0^t is deleted, the currently marked appendant $\alpha^{(m^t \bmod n)}$ is appended onto the right end of v^t: i.e., $v^{t+1} = v_1^t \cdots v_{|v^t|-1}^t \cdot \alpha^{(m^t \bmod n)}$ and $m^{t+1} = (m^t + 1) \bmod n$ (no skipping).

According to the definition in [12, 2, 15], the computation of a CTS is said to end if either the dataword is ϵ or if it repeats a configuration. This relaxed definition of termination was introduced for the purpose of reducing any Turing machine to cellular automaton rule 110, whose computation never halts. Precisely, [12, Theorem 4.3.2, p. 65] states the following: let $\mathcal{M}$ be any deterministic Turing machine using a single tape; there is a cyclic tag system $\mathcal{C}_\mathcal{M}$ with appendants $\alpha^0, \ldots, \alpha^{n-1}$ and a linear-time encoding $v_\mathcal{M}$ of the input x of $\mathcal{M}$, such that for all input x: (1) $\mathcal{C}_\mathcal{M}$ halts from initial dataword $v_\mathcal{M}(x)$ if and only if $\mathcal{M}$ halts from input x; and (2) for all t, if $\mathcal{M}$ halts after t steps on x, then $\mathcal{T}_\mathcal{M}$ halts after $O_\mathcal{M}(t^2 \log t)$ steps on $v_\mathcal{M}(x)$. For our purpose, we need the computation of the CTS to *stop with an empty dataword* (and not to enter a cycle) if the simulated Turing machine stops, precisely:

▶ **Lemma 5** (Corollary of Theorem 4.3.2 in [12]). *For every Turing machine $\mathcal{M}$, there is CTS $\hat{\mathcal{C}}_\mathcal{M}$ and a linear time encoding $\hat{v}_\mathcal{M}(x) = \chi(v_\mathcal{M}(x))$ that encodes any input x of $\mathcal{M}$ into an initial dataword $\hat{v}^0 = \hat{v}_\mathcal{M}(x)$ such that: (1) $\hat{\mathcal{C}}_\mathcal{M}$ halts from $\hat{v}^0$ with an empty dataword iff $\mathcal{M}$ halts from input x; and (2) if $\mathcal{M}$ halts after t steps from x, then $\hat{\mathcal{C}}_\mathcal{M}$ after $O(t^2 \log t)$ steps from $\hat{v}^0$.*

Proof. We proceed as follows by defining a CTS $\hat{\mathcal{C}}_\mathcal{M}$ "with two processing modes": the first mode emulates $\mathcal{C}_\mathcal{M}$, the second mode just erases the data word; switching from one mode to the other just requires inserting a single letter 0 in the dataword. Consider χ the homomorphism on $\{0,1\}^*$ such that $\chi(0) = 00$ and $\chi(1) = 01$, i.e. χ inserts a 0 before every letter of a word. Then, consider the CTS $\hat{\mathcal{C}}_\mathcal{M}$ with $2n$ appendants: $\hat{\alpha}^{2i+1} = \chi(\hat{\alpha}^i)$ and $\hat{\alpha}^{2i}_\mathcal{M} = \epsilon$, for $0 \leqslant i < n$. An immediate induction shows that $\hat{\mathcal{C}}_\mathcal{M}$ simulates $\mathcal{C}_\mathcal{M}$ exactly twice slower, indeed: if v^t and $\hat{v}^t$ denote the datawords of $\mathcal{C}_\mathcal{M}$ and $\hat{\mathcal{C}}_\mathcal{M}$ starting from the initial datawords v^0 and $\hat{v}^0 = \chi(v^0)$ respectively, then for all time t, $\hat{v}^{2t} = \chi(v^t)$. Now, if we shift the dataword of $\hat{\mathcal{C}}_\mathcal{M}$ by one letter, the appendants that will be appended next, are all the empty word ϵ, and the dataword will be completely erased, yielding to the desired terminaison. Without loss of generality, we assume that $\mathcal{M}$ has an unique final state q_F. According to the design of $\mathcal{C}_\mathcal{M}$ in the proof Theorem 4.3.2 p. 65 in [12], for every step t of $\mathcal{M}$ where the configuration is $\cdots B \cdots B\sigma_1 \ldots \sigma_j[q,d]\sigma_{j+1}\sigma_{j+2}\cdots\sigma_{s-2}B\cdots B\cdots$ (i.e., where the head is over position σ_{j+1}, the current state is q and the next head movement is d, B denotes the blank symbol), the dataword of $\mathcal{C}_\mathcal{M}$ is, at the step $O(t^2 \log t)$ corresponding to first stage of the processing of this configuration by $\mathcal{C}_\mathcal{M}$:

$$\langle 1,q,d\rangle\langle\sigma_{j+1}\rangle\cdots\langle\sigma_{s-2}\rangle\langle\sigma_B\rangle\mu^{s'}\langle\sigma_1\rangle\cdots\langle\sigma_j\rangle$$

and the marker is 0. Each $\langle\cdot\rangle$ and μ stands for a binary encoding containing a single 1. Furthermore, there is at most one pattern $\langle 1,q,d\rangle$ in the dataword of $\mathcal{C}_\mathcal{M}$ at all time. Let i be the index of the only 1 in $\langle 1,q_F,d\rangle$. We then change the appendant $\hat{\alpha}^{2i+1}$ to 0. It follows that, the first time the pattern $\chi(\langle 1,q_F,d\rangle)$ appears, i.e. the first time the simulated Turing machine $\mathcal{M}$ enters the final state, the CTS $\hat{\mathcal{C}}_\mathcal{M}$ switches to the even-indexed empty-appendants-mode, then erases the whole dataword, and halts with an empty dataword, as desired. Furthermore, if $\mathcal{M}$ halts after t steps, $\hat{\mathcal{C}}_\mathcal{M}$ halts after $O(t^2 \log t)$ steps. ◀

We now show how to simulate the CTS $\hat{\mathcal{C}}_\mathcal{M}$ with a SCTS.

Proof of Proposition 2. The original cyclic tag system by Cook [2] differs from the skipping cyclic tag system only in that in the original, the list rotates by 1 no matter which letter the current word begins with. Consider the CTS $\hat{\mathcal{C}}_\mathcal{M}$, given by the lemma above, with $2n$ appendants $\hat{\alpha}^0, \ldots, \hat{\alpha}^{2n-1}$, together with its linear-time input encoding $\hat{v}_\mathcal{M}$. Let χ' be the homomorphism over $\{0,1\}^*$ defined as $\chi'(0) = 00$ and $\chi'(1) = 1$. Let $\mathcal{S}_\mathcal{M}$ be the SCTS with $4n$ appendants: $\beta^{2i} = \epsilon$ and $\beta^{2i+1} = \chi'(\hat{\alpha}^i)$ for $0 \leqslant i < 2n$. An immediate recurrence shows that $\mathcal{S}_\mathcal{M}$ simulates $\hat{\mathcal{C}}_\mathcal{M}$, precisely: if v^t and u^t denote respectively the datawords of $\hat{\mathcal{C}}_\mathcal{M}$ and $\mathcal{S}_\mathcal{M}$ with initial datawords $v^0 \in \{0,1\}^*$ and $u^0 = \chi'(v^0)$ then, for all time t, $u^{t+r_t} = v^t$ where r_t is the number of 0s read by $\hat{\mathcal{C}}_\mathcal{M}$ up to time t (note that $r_t \leqslant t$). Let $u_\mathcal{M}(x) = \chi'(\hat{v}_\mathcal{M}(x))$ denote the linear time encoding of the input x of $\mathcal{M}$ as the initial dataword of $\mathcal{S}_\mathcal{M}$. It follows that (1) $\mathcal{S}_\mathcal{M}$ halts from input dataword $u_\mathcal{M}(x)$ iff $\mathcal{M}$ halts from input x; and (2) if $\mathcal{M}$ halts from input x after t steps, then $\mathcal{S}_\mathcal{M}$ halts from $u_\mathcal{M}(x)$ with an empty dataword after $O(t^2 \log t)$ steps. Note that moreover, the number of appendants of $\mathcal{S}_\mathcal{M}$ is a multiple of 4. ◀

B Proof of main Theorem 1 as a consequence of key Theorem 6

The remaining of this article is dedicated to prove the following theorem which implies Theorem 1 by the proposition above (see appendix on the current page).

▶ **Theorem 6** (Key theorem). *There is a fixed set B of 542 bead types and a fixed attraction rule ♥ on B together with two polynomial-time encodings:*

- *π that maps any SCTS $\mathcal{S}$ with $n \geqslant 8$ appendants $\alpha = \langle \alpha^0, \dots, \alpha^{n-1} \rangle$ where n is a multiple of 4, to a bead-type sequence $\pi_\mathcal{S} \in B^*$ of exact length:*
$$|\pi_\mathcal{S}| = 18Kn(Kn + 12n - 8) + 3n(192n - 171) + 30 \sum_{i=0}^{n-1} |\alpha^i| = O(|\alpha|^4)$$
where $K = L + 12 - (L \bmod 2) \leqslant L + 12$ with $L = \max_i |\alpha^i|$ being the length of the longest appendant, and $|\alpha| = \sum_{i=0}^{n-1} |\alpha^i|$. Note that $\pi_\mathcal{S}$ only depends on the appendants of $\mathcal{S}$.
- *$(s_\mathcal{S}, \sigma_\mathcal{S})$ that maps any input dataword u of $\mathcal{S}$ to a seed configuration $\sigma_\mathcal{S}(u)$ of a bead type sequence $s_\mathcal{S}(u)$ of length $O_\mathcal{S}(|u|)$, precisely:*
$$|\sigma_\mathcal{S}(u)| = 2|u|(3K + 16) + |u|_0 + 9K(n-1) + 36n - 21 = O_\mathcal{S}(|u|)$$
where $|u|_0 = \#\{i : u_i = 0\}$ is the number of 0s in u,

such that: For any SCTS $\mathcal{S}$ with $n \geqslant 8$ appendants, where n is a multiple of 4, and every input dataword u of $\mathcal{S}$, the deterministic and periodic oritatami system $\mathcal{O}_\mathcal{S} = ((\pi_\mathcal{S})^\infty, ♥, 3)$ with bead type sequence $(\pi_\mathcal{S})^\infty$ and delay $\delta = 3$, halts when folding from seed configuration $\sigma_\mathcal{S}(u)$ if and only if $\mathcal{S}$ halts on input dataword u. Furthermore, for all t, if $\mathcal{S}$ halts on u after t steps, then the folding $\mathcal{O}_\mathcal{S}$ from seed configuration $\sigma_\mathcal{S}(s)$ halts after folding $O_\mathcal{S}(t^2)$ beads.

Note that requiring that $n \geqslant 8$ and n being a multiple of 4 does not restrict this result. Indeed, repeating the appendants sequence k times in a SCTS, yields a strictly identical SCTS with k times the number of appendants. These requirements are however necessary to ensure the proper folding alignment in the design of our oritatami system.

Proof of Theorem 1. Consider a universal Turing machine $\mathcal{M}$ and the skipping cyclic tag system $\mathcal{S}_\mathcal{M}$ provided by Proposition 2 together with its linear-time input encoder $u_\mathcal{M}$. Consider the set of 542 bead types B, the rule ♥, the oritatami system $\mathcal{O}_\mathcal{M} = ((\pi_\mathcal{M})^\infty, ♥, 3)$ whose primary structure has period $\pi_\mathcal{M} = \pi_{\mathcal{S}_\mathcal{M}}$, and the linear-time seed encodings $(s_{\mathcal{S}_\mathcal{M}}, \sigma_{\mathcal{S}_\mathcal{M}})$ provided by Theorem 6 when applied to $\mathcal{S}_\mathcal{M}$. Let us define for short $s_\mathcal{M}(x) = s_{\mathcal{S}_\mathcal{M}}(u_\mathcal{M}(x))$ and $\sigma_\mathcal{M}(x) = \sigma_{\mathcal{S}_\mathcal{M}}(u_\mathcal{M}(x))$, the seed bead types sequence and the seed conformation of $\mathcal{O}_\mathcal{M}$ corresponding to the input x of $\mathcal{M}$. Then, by construction:

1. For all input x of $\mathcal{M}$, $\mathcal{M}$ halts on input x, if and only if $\mathcal{S}_\mathcal{M}$ halts on input dataword $u_\mathcal{M}(x)$, if and only if $\mathcal{O}_\mathcal{M}$ halts its folding from seed conformation $\sigma_\mathcal{M}(x) = \sigma_{\mathcal{S}_\mathcal{M}}(u_\mathcal{M}(x))$.
2. For all input x of $\mathcal{M}$ and all time t, if $\mathcal{M}$ halts on x after t steps, then $\mathcal{S}_\mathcal{M}$ halts on input dataword $u_\mathcal{M}(x)$ after $T = O_\mathcal{M}(t^2 \log t)$ steps, and thus $\mathcal{O}_\mathcal{M}$ halts its folding from seed conformation $\sigma_\mathcal{M}(x)$ after $O_{\mathcal{S}_\mathcal{M}}(T^2) = O_\mathcal{M}(t^4 \log^2 t)$ steps.
3. For all input x of $\mathcal{M}$, the length of the seed conformation encoding x in $\mathcal{O}_\mathcal{M}$ is $O_{\mathcal{S}_\mathcal{M}}(|u_\mathcal{M}(x)|) = O_\mathcal{M}(|x|)$, linear in $|x|$.
4. Finally, the oritatami system $\mathcal{O}_\mathcal{M}$ and seed encoding $(s_\mathcal{M}, \sigma_\mathcal{M})$ are obtained in polynomial time from the skipping tag system $\mathcal{S}_\mathcal{M}$, which is also obtained in polynomial time from $\mathcal{M}$. The reduction is thus computed in polynomial time.

◀

C Folding paths of all the bricks of our design

This section presents all the bricks, i.e. all the folding paths of the 7 modules (i.e. subsequences of the transcript) composing each unit of the transcript. The folding of each module into one of these bricks depending on the context, is the key to the correctness of the folding of our transcript design into the shape of the trimmed diagram the simulated STCS as stated in Lemma 3. This section just presents the shape of each brick for each module together with an illustration to scale. Its purpose is to provide a guideline to the description of the blocks in the next section. The full description of each brick will be given in section F.

To ease the reading, the brick name contains a specific symbol indicating to which kind of phase of the folding the brick belongs:

- Zig-up bricks are annotated by ⯈̲
- Zig-down bricks are annotated by ⯈̄
- Appending bricks are annotated by ✎
- Carriage-return bricks (spanning from a zig row to the next zag row) are annoted by ◣
- Zag bricks are annotated by ⯇̄
- Line-feed bricks (spanning from a zag row to the next zig row) are anoted by ◢
- Halting bricks are annotated by ■

The subsequences corresponding to each of the 7 modules are referred by A, B, C, D$(x)_{r,t}$, E$_a$, F, G. Some modules (D and E) have parameters that will be explained later, in the next sections. Their bricks, i.e. their possible folding depending on the context are referred by:

Module A (see section C.1): A⯈̲ (zig-up), A⯈̄ (zig-down) and A⯇̄ (zag).

Module B (see section C.2): B⯈̲ (zig-up), B⯈̄ (zig-down), B⯇̄ (zag) and B■ (halt).

Module C (see section C.3): C⯈̲ (zig-up), C⯈̄ (zig-down), C⯇̄ (zag) and C⯈̲End (zig-up, enf of dataword detected).

Module D$(x)_{r,t}$ (see section C.4): D$(x)_{r,t}$⯈̲ (zig-up), D$(x)_{r,t}$⯈̄ (zig-down), D$(x)_{r,t}$⯇̄ (zag) and ✎D$(x)_{r,t}$ (append).

Module E$_a$ (see section C.5): E$_a$⯈̲ (zig-up), E$_a$⯈̄ (zig-down), E$_a$⯇̄ (zag) and E$_a$◣ (carriage return).

Module F (see section C.6): F⯈̲ (zig-up), F⯈̄ (zig-down) and F⯇̄ (zag).

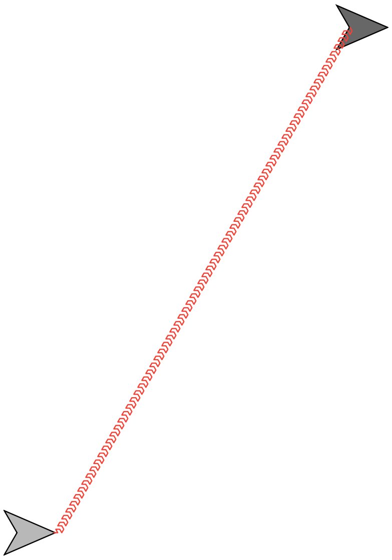
Module G (see section C.7): G⯈̲Read0 (read 0), G⯈̲Read1 (read 1), G⯈̄Copy0 (zig-down), G⯈̄Copy1 (zig-down), G⯇̄Copy0 (zag), G⯇̄Copy1 (zag), and G◢LineFeed (line feed).

In the following figures, a lighter and a darker grey arrow indicates the beginning and the end of the folding path of each brick respectively. The parameter h will be defined later and refers to the height of the blocks composing the folded shape of our design.

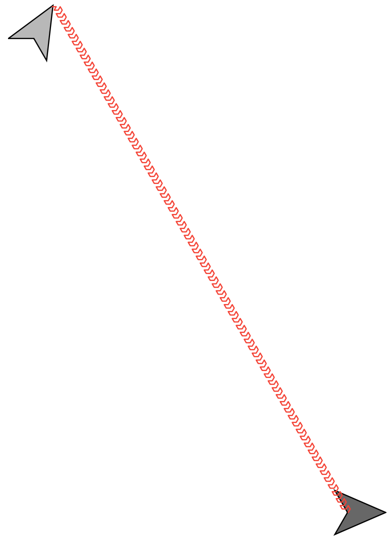
C.1 All bricks for Module A

Module A always folds as a glider of height h and width 3, pointing to NE in zig-up phase, SE in zig-down phase and SW in zag phase. The folding of module A serves as a scaffold for the folding of the next modules in the zig-up and zig-down phases.

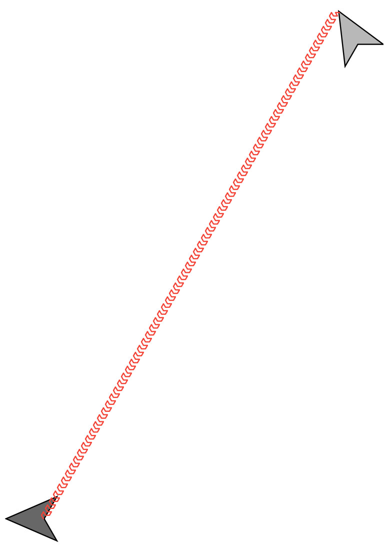
All its possible bricks are displayed in Fig. 6 on the facing page.



(a) The brick A►.



(b) The brick A► = HorizontalMirror(A►).



(c) The brick A◄ = Rotate$_{180^\circ}$(A►).

Figure 6 Folding paths to scale of all the bricks for Module A (see section F.4 for full description).

C.2 All bricks for Module B

Module **B** is 5 beads long. It folds along the preceding brick of Module **A**:

- to the right in zig-down;
- to the left in zag phase;
- to right in zig-up phase if the dataword encoded in the zag-row above is not empty;
- but to the left in the zig-up phase if the dataword encoded above is empty (terminating the folding as it is now trapped in a closed area).

All its possible bricks are displayed in Fig. 7. Note that $\mathbf{B}\,\overline{\blacktriangleright}$ = HorizontalMirror($\mathbf{B}\,\underline{\blacktriangleright}$) and $\mathbf{B}\,\overline{\blacktriangleleft}$ = $\text{Rotate}_{180^\circ}$($\mathbf{B}\,\underline{\blacktriangleright}$).

(a) The brick $\mathbf{B}\,\underline{\blacktriangleright}$. **(b)** The brick $\mathbf{B}\,\overline{\blacktriangleright}$. **(c)** The brick $\mathbf{B}\,\overline{\blacktriangleleft}$. **(d)** The brick $\mathbf{B}\,\blacksquare$.

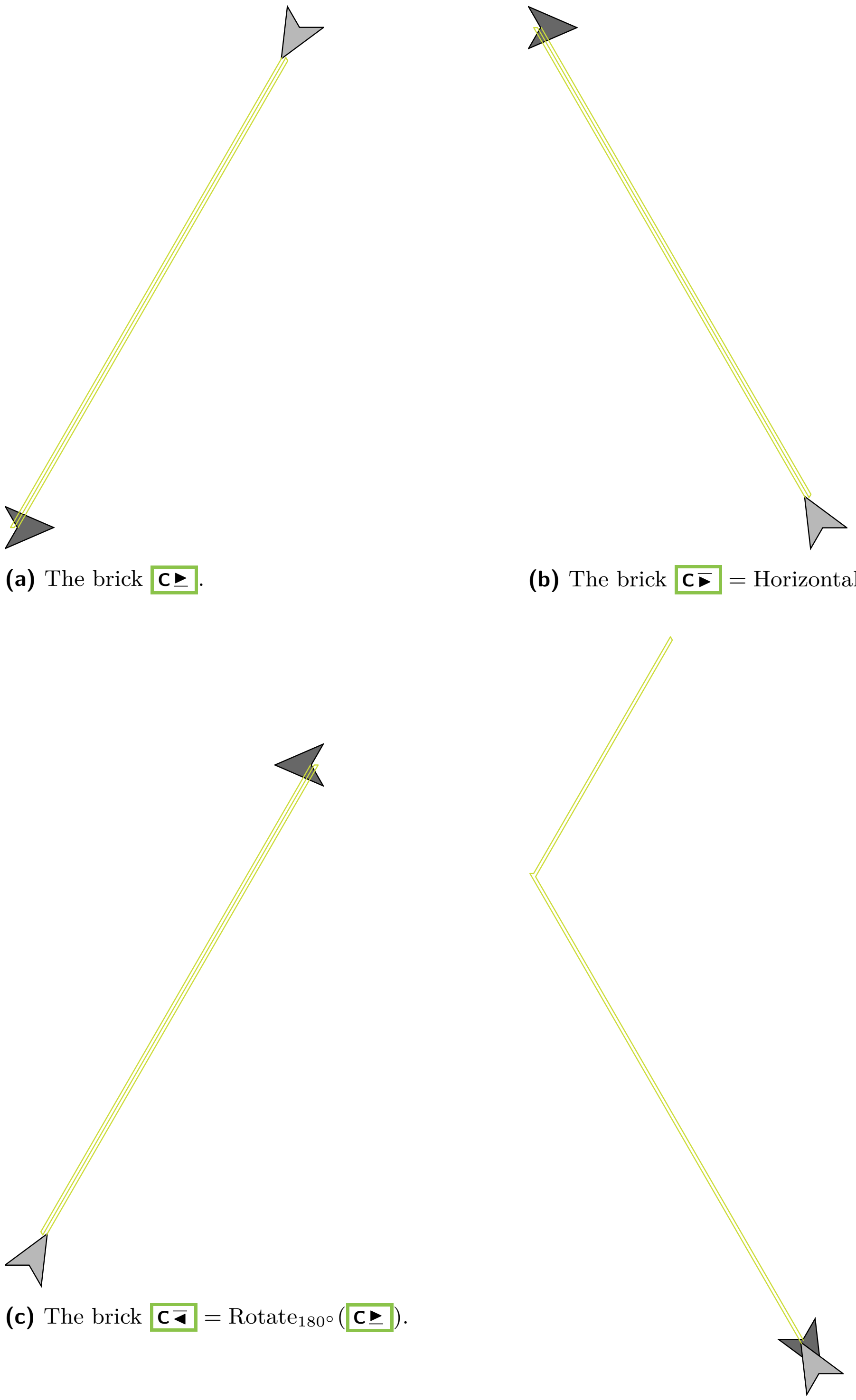
Figure 7 Folding paths to scale of all the bricks for Module **B** (see section F.5 for full description).

C.3 All bricks for Module C

Module **C** folds in switchbacks of height almost h, along the brick of the preceding module **A**:

- in 3 switchbacks in the zag phases;
- in 3 switchbacks in the zig-up or zig-down phases if the current folding did not reach the end of the dataword encoded in the zag-row above yet;
- but in 2 switchbacks (**C▸End**) if this end is reached, creating the initial condition for folding the next modules as the encoding of the letters of the appendant to be appended at this stage of the simulation.

All its possible bricks are displayed in Fig. 8 on the following page.



(a) The brick C▶.

(b) The brick C▶ = HorizontalMirror(C▶).

(c) The brick C◀ = Rotate$_{180°}$(C▶).

(d) The brick C▶End.

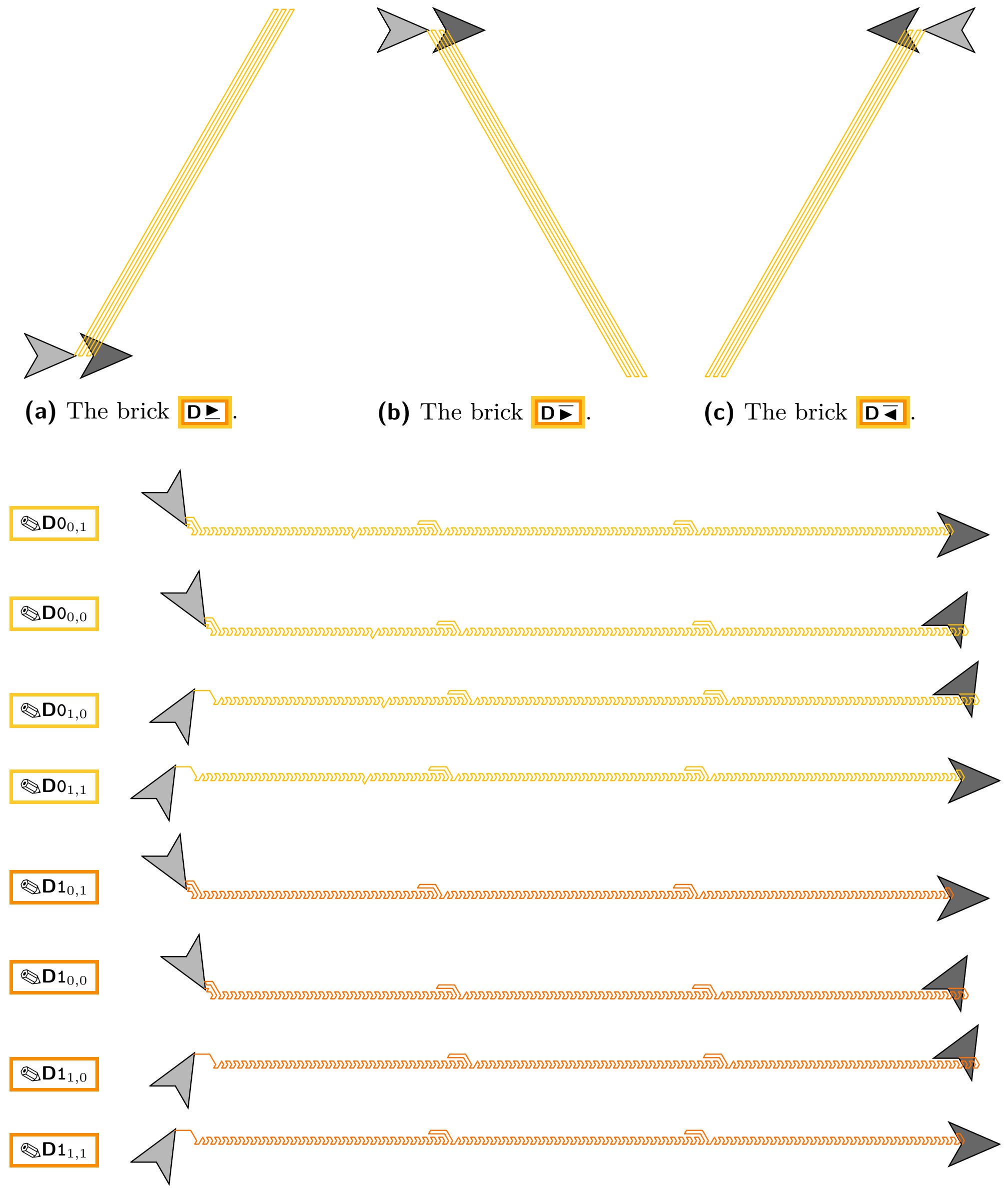
Figure 8 Folding paths to scale of all the bricks for Module C (see section F.6 for full description).

C.4 All bricks for Module D

Module **D**$(x)_{r,t}$ is used to encode each letter of the appendant stored in each block unit. Its parameters x, r, t stands for the letter $x \in \{0, 1\}$ and the position index of that letter in the encoded appendant (r says if it is either at the first, odd or even, and t if it is the last or not). All these variants of module **D** fold slightly differently. Module **D**$(x)_{r,t}$ folds either:

- in 6 switchbacks of height approximatively $h/2$, along the preceding brick of Module **C** in the zig-up, zig-down and zag phases;
- or as a glider in the append phase where it is forced by the preceding brick of module **B** to adopt this shape.

All its possible bricks are displayed in Fig. 9 on the next page.
Note that: **D**▶̄ = HorizontalMirror(**D**▶̲) and **D**◀̄ = Rotate$_{180°}$(**D**▶̲).



(a) The brick D▶. **(b)** The brick D▶. **(c)** The brick D◀.

(d) The bricks D0$_{r,t}$ and D1$_{r,t}$.

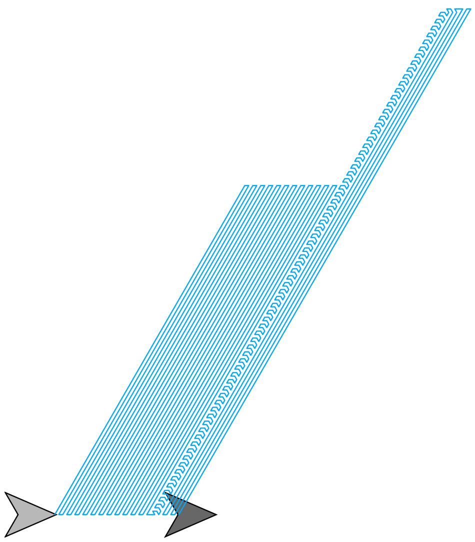
Figure 9 Folding paths to scale of all the bricks for Module D (see section F.7 for full description).

C.5 All bricks for Module E

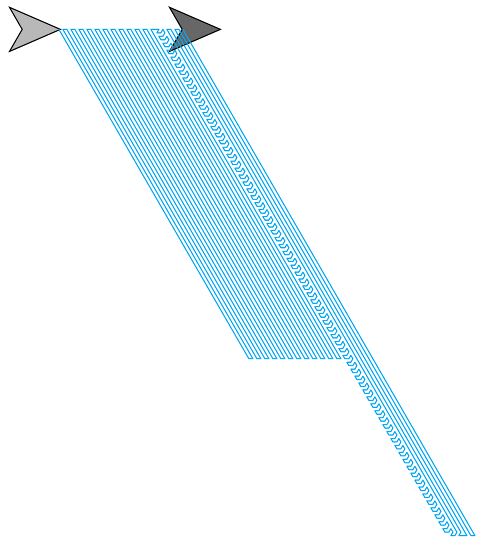
Module $\boxed{E_a}$ is used for padding and carriage return. Its parameter a stands for the number of letter of the appendant it needs to pad so as the block units of all appendants have the same dimensions. The length of module $\boxed{E_a}$ decreases with a accordingly. Module $\boxed{E_a}$ folds either:

- in many short switchbacks of height approximatively $h/2$ followed by a glider and 5 large switchbacks of height h in zig-up, zig-down and zag phases. The many short switchbacks are used to pad the appendants to a fixed length in the block units.
- or as a glider in append phase, achieving the carriage return from zig to zag phase. The glider folding is triggered as for module $\boxed{D}$, either by the preceding brick of module $\boxed{B}$ or the preceding glider append brick of a module $\boxed{D}$.

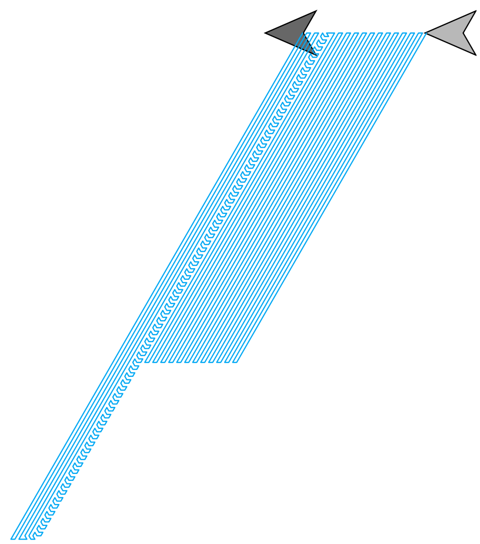
All its possible bricks are displayed in Fig. 10 on the following page.
Note that: $\boxed{E\,\overline{\blacktriangleright}}$ = HorizontalMirror($\boxed{E\,\underline{\blacktriangleright}}$) and $\boxed{E\,\overline{\blacktriangleleft}}$ = Rotate$_{180^\circ}$($\boxed{E\,\underline{\blacktriangleright}}$).



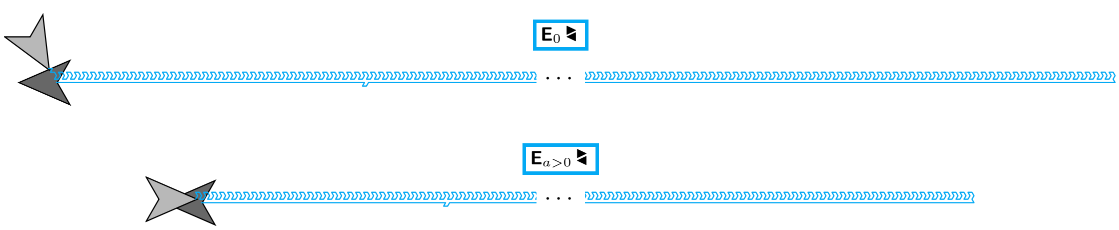
(a) The bricks E_a ▶.



(b) The bricks E_a ▶.



(c) The bricks E_a ◀.



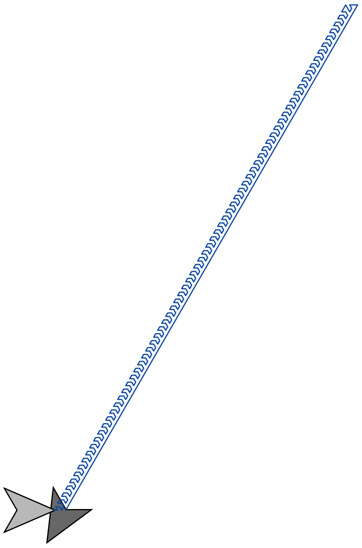
(d) The bricks E_a.

Figure 10 Folding paths to scale of all the bricks for Module E_a (see section F.8 for full description).

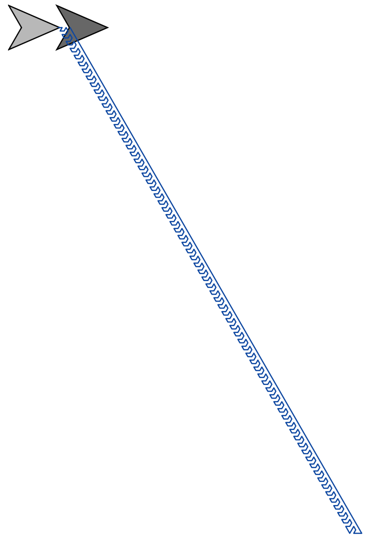
C.6 All bricks for Module F

Module F always folds as a glider of height h and width 4, pointing to NE in zig-up phase, SE in zig-down phase and SW in zag phase. As for module A in the zig phases, the folding of module F serves as a scaffold for the folding of the next modules in the zag phases.

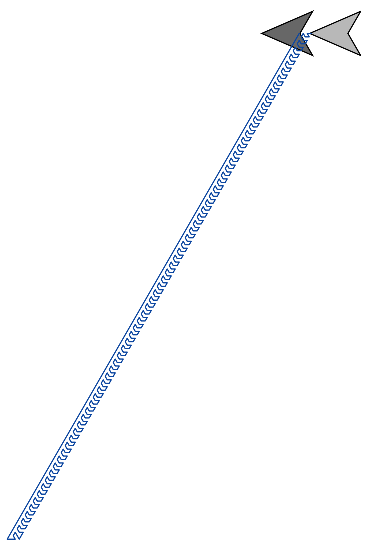
All its possible bricks are displayed in Fig. 11.



(a) The brick F►.



(b) The brick F► = HorizontalMirror(F►).



(c) The brick F◄ = Rotate$_{180°}$(F►).

Figure 11 Folding paths to scale of all the bricks for Module F (see section F.9 for full description).

C.7 All bricks for Module G

Module G is the most sophisiticated module as it can fold, depending of the context, into very different shapes:

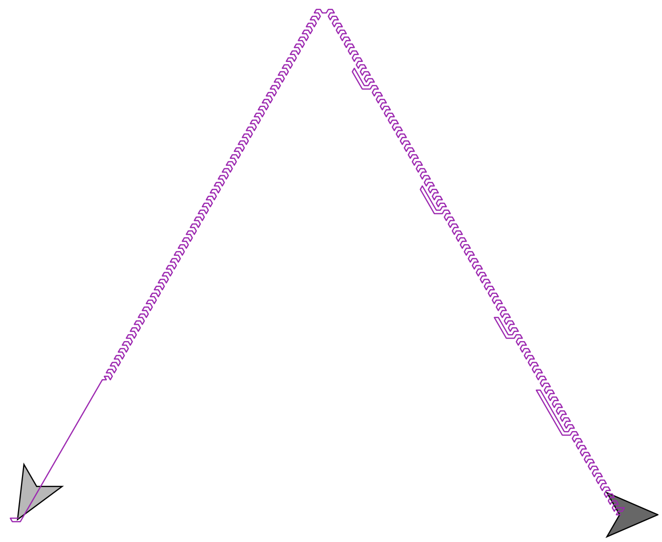
Read bricks (Fig. 12): In zig-up phase, module G folds as a glider of total length approximatively $2h$, heading first to NE and then bouncing to SE (G ▶ Read0) or E (G ▶ Read1) depending on whether it hits the encoding of letter `0` or `1` in the zag row above respectively.

Copy bricks (Fig. 13): In zig-down and zag phases, module G folds into 6 switchbacks of height h, copying the letter, `0` or `1`, encoded at the bottom of the row above, to the top of the row bellow.

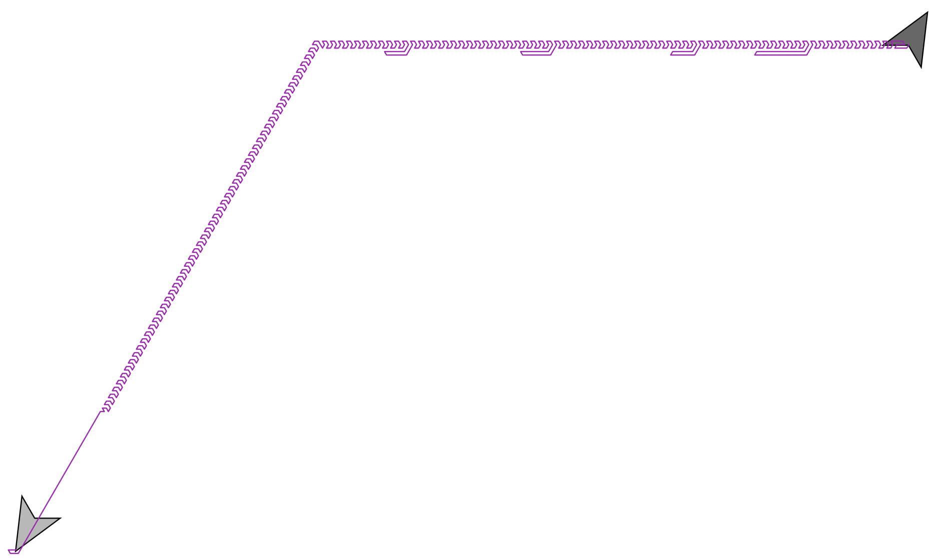
Line feed brick (Fig. 14): At the end of the zag row, module G folds into a glider of length $2h$ heading SW opening the next zig row.

The design of module G requires a lot of care and was made possible thanks to the advanced design tools we developed for that purpose.

All its possible bricks are displayed in Fig. 12, 13 and 14.

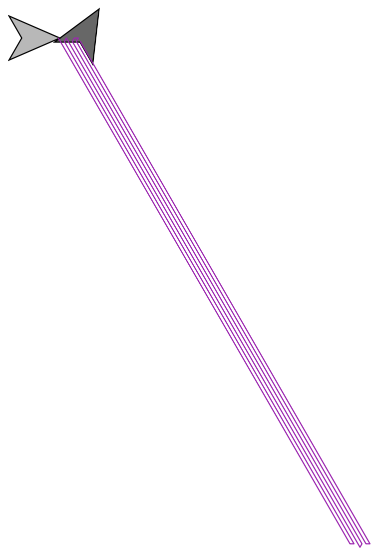


(a) The brick **G ▶ Read0**.

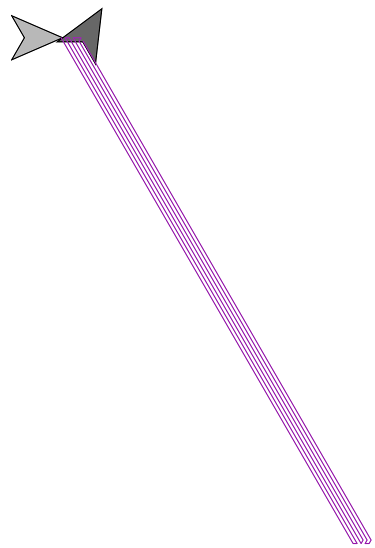


(b) The brick **G ▶ Read1**.

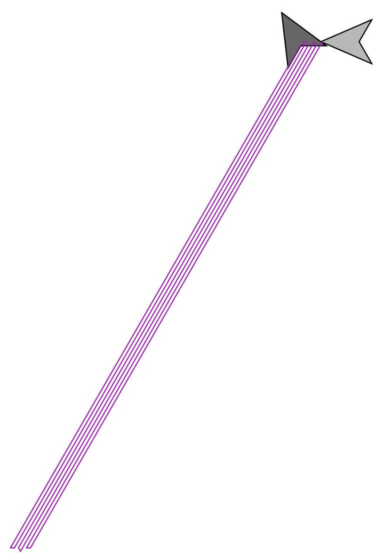
Figure 12 Folding paths to scale of all the Read bricks for Module **G** (see section F.10 for full description).



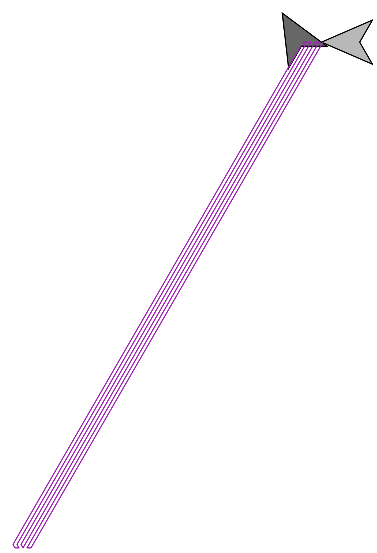
(a) The brick **G ▶ Copy0**.



(b) The brick **G ▶ Copy1**.

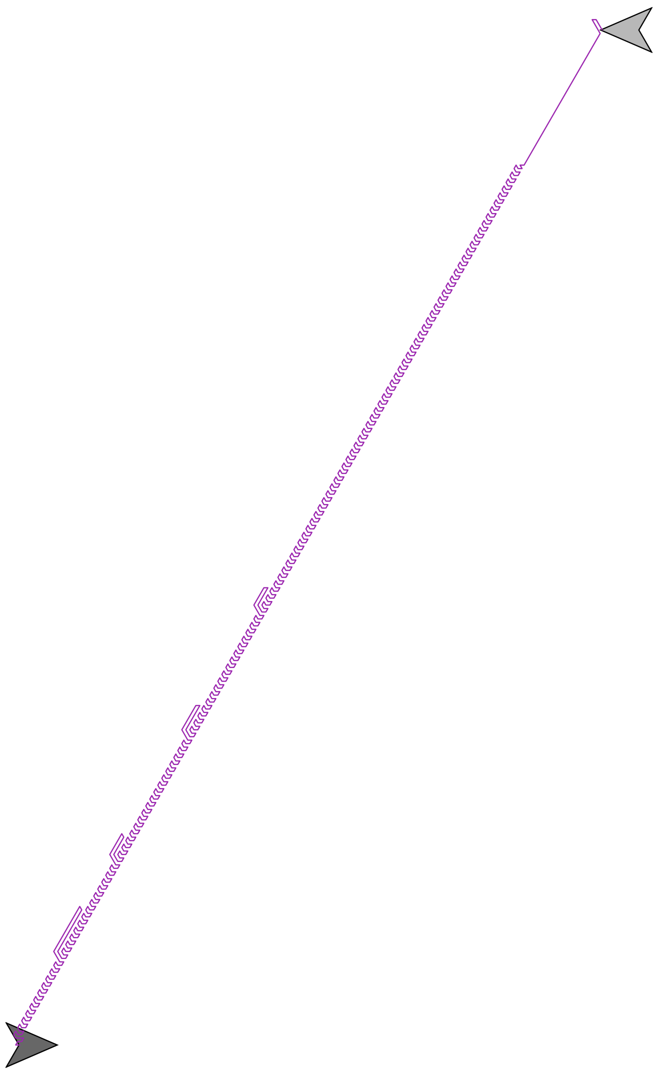


(c) The brick **G ◀ Copy0**.



(d) The brick **G ▶ Copy1**.

Figure 13 Folding paths to scale of all the Copy bricks for Module **G** (see section F.10 for full description).



(a) The brick **G ◀ LineFeed**.

Figure 14 Folding paths to scale of all the Line Feed brick for Module **G** (see section F.10 for full description).

D Inside of the blocks

This section presents the exact content of each block, i.e. the bricks they are composed of.

Notation. We describe conformations using the following notations: given a conformation σ and a bead type b, we denote by $\sigma \underset{\texttt{E}}{\rightarrow} b$, $\sigma \underset{\texttt{SE}}{\searrow} b$, $\sigma \underset{\texttt{SW}}{\swarrow} b$, $\sigma \underset{\texttt{W}}{\leftarrow} b$, $\sigma \overset{\texttt{NW}}{\nwarrow} b$, and $\sigma \overset{\texttt{NE}}{\nearrow} b$ the elongations of the conformation σ by one bead of bead type b located respectively to the east, south-east, south-west, west, north-west, and north-east of the last bead of σ. We refer by b the conformation that consists of a single bead of bead type b located at $(0, 0)$. Fig. 15 illustrates this notation.

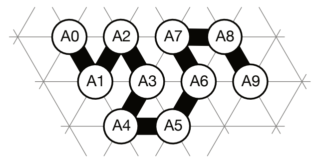


Figure 15 The conformation encoded by **A0** $\underset{\texttt{SE}}{\searrow}$ **A1** $\overset{\texttt{NE}}{\nearrow}$ **A2** $\underset{\texttt{SE}}{\searrow}$ **A3** $\underset{\texttt{SW}}{\swarrow}$ **A4** $\underset{\texttt{E}}{\rightarrow}$ **A5** $\overset{\texttt{NE}}{\nearrow}$ **A6** $\overset{\texttt{NW}}{\nwarrow}$ **A7** $\underset{\texttt{E}}{\rightarrow}$ **A8** $\underset{\texttt{SE}}{\searrow}$ **A9**.

We extend naturally this notation to two conformations: for instance, $\sigma_1 \underset{\texttt{E}}{\rightarrow} \sigma_2$ is the conformation beginning with conformation σ_1 followed by conformation σ_2 whose origin has been translated to the vertex at the east of the last bead of σ_1.

We denote by HorizontalMirror(σ), VerticalMirror(σ) and $\text{Rotate}_{180°}(\sigma)$ the conformations obtained by respectively mirroring horizontally, vertically and rotating by 180° the conformation σ.

Tables 1 and 2 on pages 37 and 38, present an exploded view of the bricks inside each zig- and zag-block respectively.

The folded paths of the bricks composing the blocks in the definitions bellow have been presented in Section C. Their full description will be given in section F.

These blocks are composed of one or n *block units* where n is the number of appendants in the simulated STCS. Each block unit encodes an appendant α^i inside and consists of a sequence of $6 + |\alpha^i|$ bricks: one of each module **A**, module **B**, and module **C**, then $|\alpha^i|$ of module **D**, then one of each module **E**, module **F** and module **G**. Read and Append blocks are composed of one block unit, whereas Copy blocks are composed of n block units.

We annotate each block by $[i]$ where i is the index of the appendant α^i it contains. A block composed of n block units is annotated by the index of its leading unit (i.e., the leftmost block unit of zig-block and the rightmost block unit of a zag-block). These indices are computed modulo n, thus in the following $[i]$ refers to $[i \bmod n]$.

As for the bricks before, specific symbols indicate to which kind of phase the blocks belong:

- Zig-up blocks are annotated by ▶̲
- Zig-down blocks are annotated by ▶̄
- Appending blocks are annotated by ✎
- Carriage-return blocks (spanning from a zig row to the next zag row) are annotated by ◥◀
- Zag blocks are annotated by ◀̄
- Line-feed blocks (spanning from a zag row to the next zig row) are annotated by ◢▶
- Halting blocks are annotated by ■

This section presents the exact composition in terms of bricks of each block. It is more like a reference section. The next section presents with illustration the exact geometry of each block. We encourage the reader to use the figures of the next section to picture each block when reading the block definition. We will refer to the appropriate figure after each block definition.

D.1 The bricks encoding the appendants

Let us first describe how the appendants are encoded by D- and E-bricks inside the blocks.

Each letter of an appendant α^i is encoded by a module D. There are 8 variants $\mathsf{D}(x)_{r,t}$ of modules D depending on:

- the letter $x \in \{0,1\}$
- the rank $r \in \{0,1,2\}$ of the letter in α^i: $r = 0$ for the first letter, $r = 1$ if the index of letter is odd, and $r = 2$ if the index of the letter is positive and even.
- $t \in \{0,1\}$ where $t = 1$ iff it is the last letter of α^i.

Each of these variant folds into a slightly different brick. This is why we must take the index of the letter into account in the definitions bellow.

$$\mathsf{Appendant}(\epsilon)\blacktriangleright = \mathsf{E}_L\blacktriangleright$$

$$\mathsf{Appendant}(x)\blacktriangleright = \mathsf{D}(x)_{0,1}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{E}_{L-1}\blacktriangleright, \qquad \text{for letter } x \in \{0,1\}$$

$$\mathsf{Appendant}(v)\blacktriangleright = \mathsf{D}(v_0)_{0,0}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \bigodot_{i=1}^{|v|-2} \left(\mathsf{D}(v_i)_{2-(i \bmod 2),0}\blacktriangleright \overset{\rightarrow}{\mathtt{E}}\right) \mathsf{D}(v_{|v|-1})_{2-((|v|-1) \bmod 2),1}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{E}_{L-|v|}\blacktriangleright,$$
$$\text{for } v \in \{0,1\}^* \text{ with } |v| \geqslant 2$$

$$\mathsf{Appendant}(v)\overline{\blacktriangleright} = \text{HorizontalMirror}\left(\mathsf{Appendant}(v)\blacktriangleright\right) \qquad \text{for all } v \in \{0,1\}^*$$

$$\mathsf{Appendant}(v)\blacktriangleleft = \text{Rotate}_{180^\circ}\left(\mathsf{Appendant}(v)\blacktriangleright\right) \qquad \text{for all } v \in \{0,1\}^*$$

D.2 The Zig-Blocks

Let us now describe the bricks inside each of the blocks present on a zig row, that is to say: Read▶ and Copy▶ blocks.

$$\mathsf{Read0}[i]\blacktriangleright = \mathsf{A}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{B}\blacktriangleright \overset{\swarrow}{\mathtt{SW}} \mathsf{C}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{Appendant}(\alpha^i)\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{F}\blacktriangleright \overset{\swarrow}{\mathtt{SW}} \mathsf{G}\blacktriangleright\mathsf{Read0} \qquad \text{(See fig. 16(a) p. 40)}$$

$$\mathsf{Read1}[i]\blacktriangleright = \mathsf{A}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{B}\blacktriangleright \overset{\swarrow}{\mathtt{SW}} \mathsf{C}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{Appendant}(\alpha^i)\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{F}\blacktriangleright \overset{\swarrow}{\mathtt{SW}} \mathsf{G}\blacktriangleright\mathsf{Read1} \qquad \text{(See fig. 16(b) p. 40)}$$

$$\mathsf{Copy0}[i]\blacktriangleright\mathsf{Unit} = \mathsf{A}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{B}\blacktriangleright \overset{\mathtt{NW}}{\nwarrow} \mathsf{C}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{Appendant}(\alpha^i)\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{F}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{G}\blacktriangleright\mathsf{Copy0}$$

$$\mathsf{Copy1}[i]\blacktriangleright\mathsf{Unit} = \mathsf{A}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{B}\blacktriangleright \overset{\mathtt{NW}}{\nwarrow} \mathsf{C}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{Appendant}(\alpha^i)\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{F}\blacktriangleright \overset{\rightarrow}{\mathtt{E}} \mathsf{G}\blacktriangleright\mathsf{Copy1}$$

$$\mathsf{Copy0}[i]\blacktriangleright = \mathsf{Copy0}[i]\blacktriangleright\mathsf{Unit} \bigodot_{j=1}^{n-1} \left(\overset{\rightarrow}{\mathtt{E}}\, \mathsf{Copy1}[i+j]\blacktriangleright\mathsf{Unit}\right) \qquad \text{(See fig. 17(a) p. 41)}$$

$$\mathsf{Copy1}[i]\blacktriangleright = \bigodot_{j=0}^{n-1} \left(\mathsf{Copy1}[i+j]\blacktriangleright\mathsf{Unit} \overset{\rightarrow}{\mathtt{E}}\right) \qquad \text{(See fig. 17(b) p. 41)}$$

Note that the Copy▶ blocks are composed of n copy block units with consecutive appendant indices modulo n. They are indexed by the index of their leading copy block unit. Recall that the index $[i+j]$ is computed modulo n and refers thus to $[(i+j) \bmod n]$.

D.3 Append and Carriage return blocks

Let us define the following convenience block:

$$\textsf{AppendAppendant}(\epsilon)\,◥ = \textsf{E}_L\,◥$$

$$\textsf{AppendAppendant}(x)\,◥ = \textsf{D}(x)_{0,1} \overset{\rightarrow}{\text{E}} \textsf{E}_{L-1}\,◥, \qquad \text{for letter } x \in \{0,1\}$$

$$\textsf{AppendAppendant}(v)\,◥ = \textsf{D}(v_0)_{0,0} \overset{\rightarrow}{\text{E}} \bigodot_{i=1}^{|v|-2} \left(\textsf{D}(v_i)_{2-(i \bmod 2),0} \overset{\rightarrow}{\text{E}} \right) \textsf{D}(v_{|v|-1})_{2-((|v|-1) \bmod 2),1} \overset{\rightarrow}{\text{E}} \textsf{E}_{L-1}\,◥,$$

for $v \in \{0,1\}^*$ with $|v| \geqslant 2$

Now,

$$\textsf{Append}[i]◥\textsf{Return} = \textsf{A}\,▶ \overset{\rightarrow}{\text{E}} \textsf{B}\,▶ \overset{\text{NW}}{\nwarrow} \textsf{C}\,▶\,\textsf{End} \overset{\searrow}{\text{SE}} \textsf{AppendAppendant}(\alpha^i)\,◥ \overset{\leftarrow}{\text{W}} \textsf{F}\,◀ \overset{\leftarrow}{\text{W}} \textsf{G}\,◀\,\textsf{Copy1} \qquad \text{(See fig. 21 p. 46)}$$

D.4 Zag row

Naturally, by symmetry, let:

$$\textsf{Copy0}[i]◀\textsf{Unit} = \text{VerticalMirror}\left(\textsf{Copy0}[i]▶\textsf{Unit}\right)$$

$$\textsf{Copy1}[i]◀\textsf{Unit} = \text{VerticalMirror}\left(\textsf{Copy1}[i]▶\textsf{Unit}\right)$$

$$\textsf{LineFeed}[i]◤\textsf{Unit} = \textsf{A}\,◀ \overset{\leftarrow}{\text{W}} \textsf{B}\,◀ \overset{\text{NE}}{\nearrow} \textsf{C}\,◀ \overset{\leftarrow}{\text{W}} \textsf{Appendant}(\alpha^i)\,◀ \overset{\leftarrow}{\text{W}} \textsf{F}\,◀ \overset{\leftarrow}{\text{W}} \textsf{G}\,◤\,\textsf{LineFeed}$$

Then,

$$\textsf{Copy0}[i]◀ = \bigodot_{j=0}^{n-3} \left(\textsf{Copy1}[i+j]◀\textsf{Unit} \overset{\leftarrow}{\text{W}}\right) \textsf{Copy0}[i+n-2]◀\textsf{Unit} \overset{\leftarrow}{\text{W}} \textsf{Copy1}[i+n-1]◀\textsf{Unit} \qquad \text{(See fig. 19(a) p. 43)}$$

$$\textsf{Copy1}[i]◀ = \bigodot_{j=0}^{n-1} \left(\textsf{Copy1}[i+j]◀\textsf{Unit} \overset{\leftarrow}{\text{W}}\right) \qquad \text{(See fig. 19(b) p. 43)}$$

$$\textsf{Copy0}[i]◤\textsf{LineFeed} = \bigodot_{j=0}^{n-3} \left(\textsf{Copy1}[i+j]◀\textsf{Unit} \overset{\leftarrow}{\text{W}}\right) \textsf{Copy0}[i+n-2]◀\textsf{Unit} \overset{\leftarrow}{\text{W}} \textsf{LineFeed}[i+n-1]◤\textsf{Unit} \qquad \text{(See fig. 20(a) p. 44)}$$

$$\textsf{Copy1}[i]◤\textsf{LineFeed} = \bigodot_{j=0}^{n-2} \left(\textsf{Copy1}[i+j]◀\textsf{Unit} \overset{\leftarrow}{\text{W}}\right) \textsf{LineFeed}[i+n-1]◤\textsf{Unit} \qquad \text{(See fig. 20(b) p. 45)}$$

D.5 The special case: appending ϵ to an empty dataword

$$\textsf{CarriageReturn}◥\textsf{LineFeed}◤ = \textsf{A}\,▶ \overset{\rightarrow}{\text{E}} \textsf{B}\,▶ \overset{\text{NW}}{\nwarrow} \textsf{C}\,▶\,\textsf{End} \overset{\searrow}{\text{SE}} \textsf{AppendAppendant}(\epsilon)\,◥ \overset{\leftarrow}{\text{W}} \textsf{F}\,◀ \overset{\leftarrow}{\text{W}} \textsf{G}\,◤\,\textsf{LineFeed}$$

$$\textsf{CarriageReturn}◤\textsf{LineFeed}◀\textsf{Halt}■ = \textsf{CarriageReturn}◥\textsf{LineFeed}◤ \overset{\rightarrow}{\text{E}} \textsf{A}\,▶ \overset{\leftarrow}{\text{W}} \textsf{B}\,■ \qquad \text{(See fig. 23 p. 48)}$$

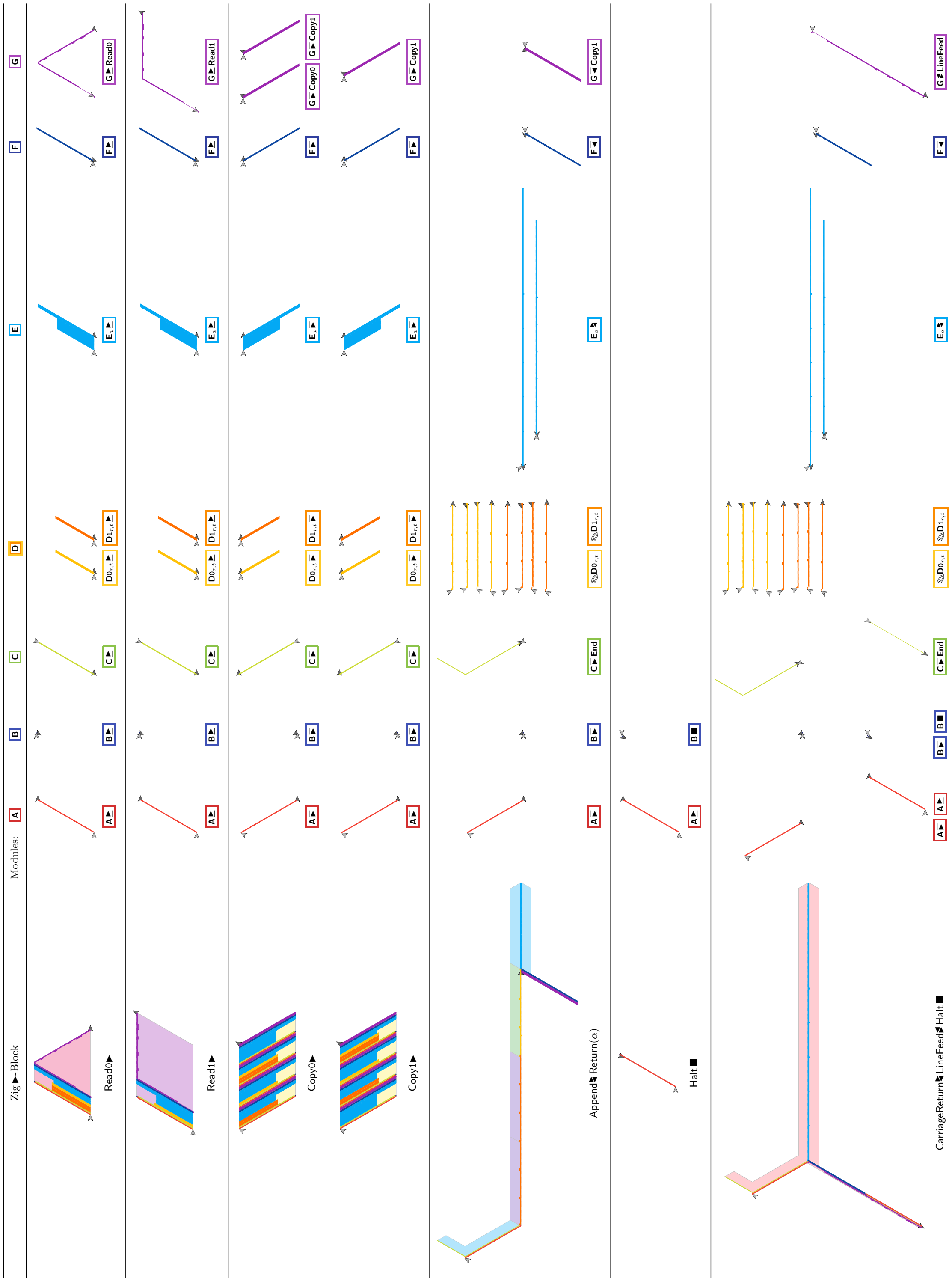


Table 1 The bricks inside the zig ▶-blocks.

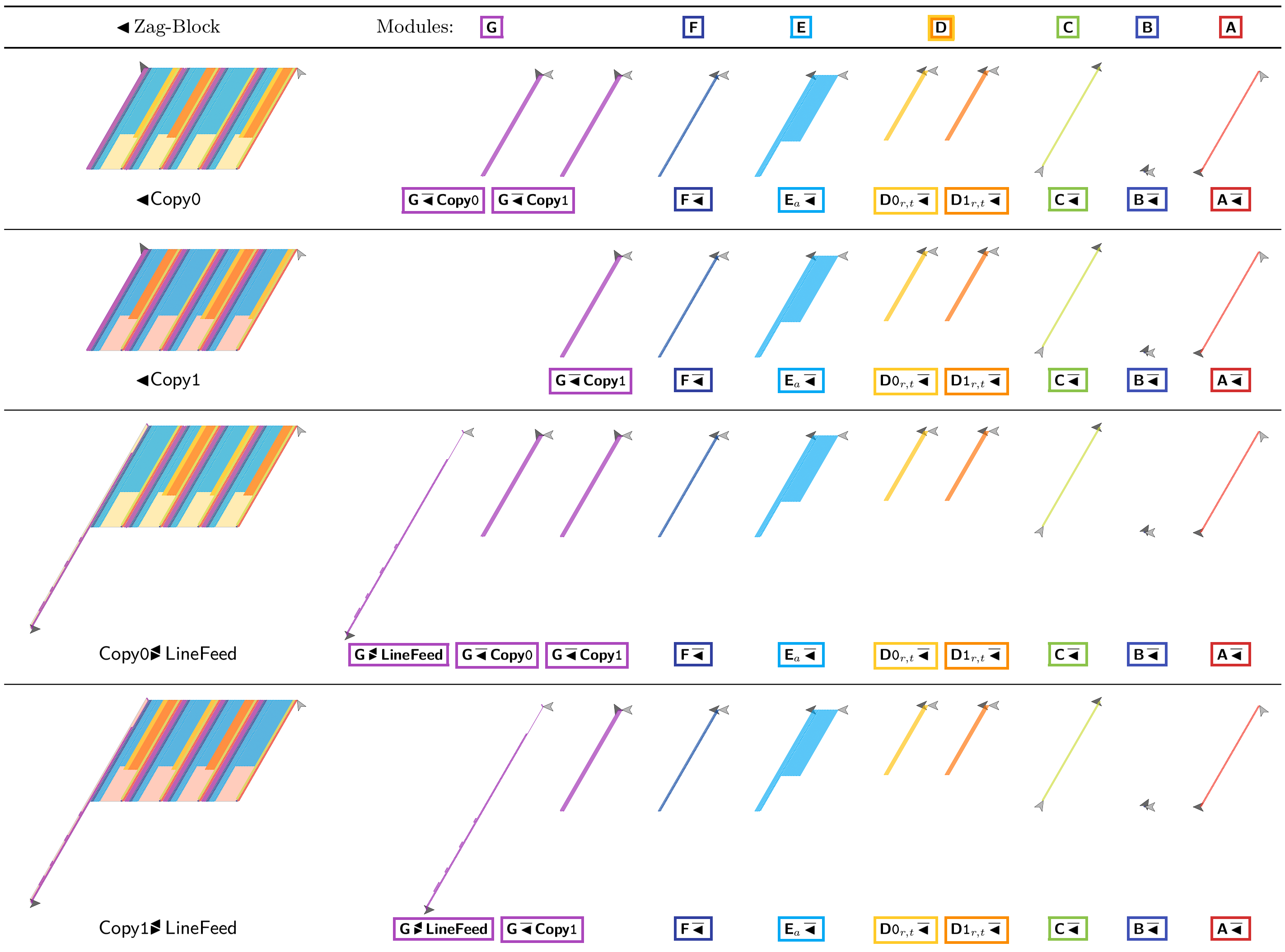


Table 2 The bricks inside the ◀ zag-blocks.

E Geometry of the blocks

We give here the geometrical description of each block. We describe in particular the positions of the important features of each block, such as the positions where a letter is read, copied or written.

Convenience variables. Let $\mathcal{S} = (\alpha; u^0)$ be the SCTS to be simulated. In order to simplify the description, we introduce the following variables which correspond to key geometrical parameters of the different blocks:

$$L = \max_{0 \leqslant i < n} |\alpha_i| \qquad \text{the maximum length of an appendant in } \mathcal{S}$$
$$P = 12 - (L \bmod 2) \qquad \text{the } \textit{padding} \text{ constant, such that } L + P \text{ is even and at least 12}$$
$$w = 6(L+P) + 18 \qquad \text{the width of the appendant module excluding its read/copy/linefeed part}$$
$$W = n \cdot (w+6) \qquad \text{the width of the Read▶, Copy▶, and Copy◀ blocks}$$
$$h = W - (w+3) \qquad \text{the height of the zig and zag rows}$$
$$c(a) = (6\lambda(L - a + P) + 8h - 16)/4 \qquad \text{the width of the brick } \mathsf{E}_a◀$$

As n is a multiple of 4 and as $L+P$ is even, we have:

▶ **Fact 7.** *All convenience variables $L, P, w, W, h, c(a)$ are integers. Furthermore,*

$$w = 6 \bmod 12, \quad W = 0 \bmod 48, \quad h = 3 \bmod 12, \quad c(a) = 2 \bmod 12, \textit{ for all } 0 \leqslant a \leqslant L.$$

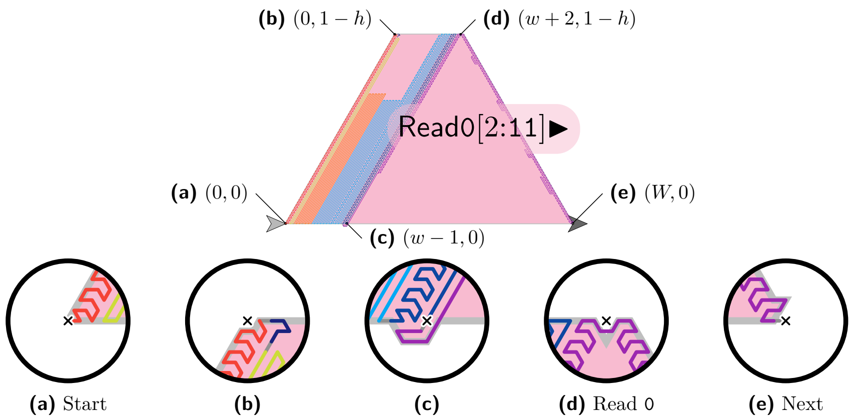
These relations ensure for instance that all gliders finish in the correct position and that all patterns are properly aligned.

- Read▶ blocks are described in figure 16
- Copy▶ blocks are described in figure 17
- Seed blocks are described in figure 18
- Copy◀ blocks are described in figure 19
- Copy▶LineFeed blocks are described in figure 20
- Append◀Return blocks are described in figure 21
- CarriageReturn▶LineFeed◀Halt■ and Halt blocks are described in figures 23 and 24

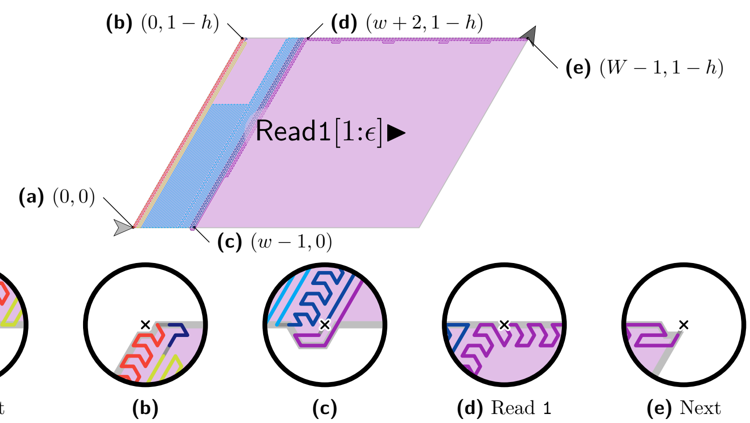
Intuitively, each zig block corresponds to one cell of the trimmed diagram of the simulated SCTS, upscaled to a parallelogram of width W and heigth h. Zag-blocks are just used to copy these cells while returning at the beginning of the current datawork.

Recall that the coordinates are expressed according to the east and south-west axis: every position (x, y) in $\mathbb{T}$ is mapped in the euclidean plane to $x \cdot \vec{E} + y \cdot \vec{SW}$ using the vector basis $\vec{E} = (1, 0)$ and $\vec{SW} = \text{RotateClockwise}\left(\vec{E}, 120°\right) = (-\frac{1}{2}, -\frac{\sqrt{3}}{2})$.

Figure 16 Geometry of the Read▶ blocks. Note that the internal structures (the lines in white) of both blocks Read0▶ and Read1▶ agree until position $(w+2, 1-h)$ where the presence or absence of a spike, encoding a 0, at the bottom of the row above forces the block to adopt the shape Read0▶ or Read1r▶espectively.

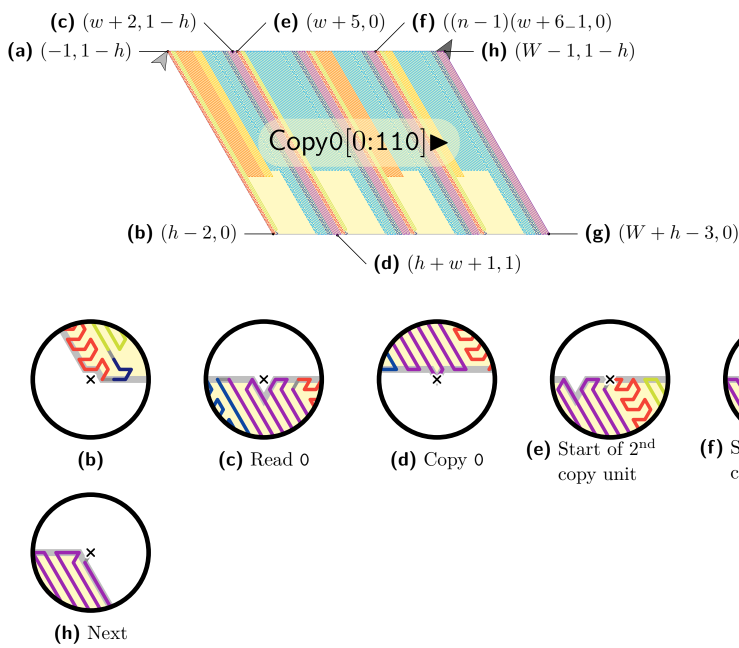


(a) The Read0▶ block has the shape of a trapezium whose bottom basis has length W and top basis has length $w+5$, with height h. It has a dent (an empty position) located at $(w+2, -h+1)$ (w.r.t. to its origin at the bottom left corner), in which plugs the spike of the block from the row above it, encoding the letter 0. The next block will start folding at the bottom right corner, at $(W, 0)$.

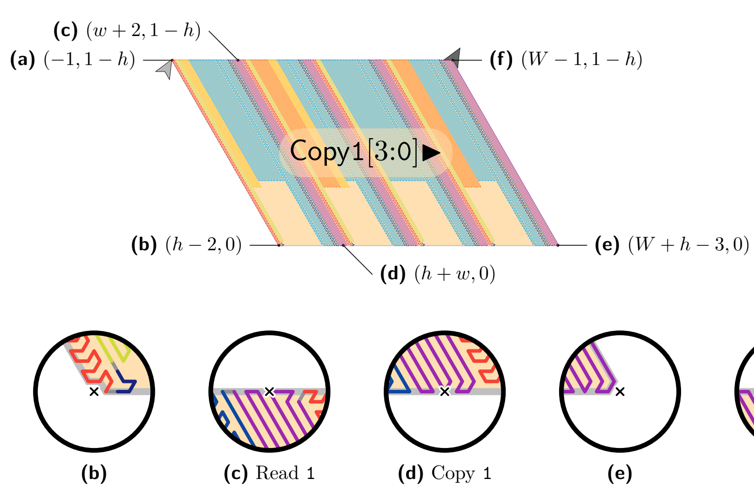


(b) The Read1▶ block has the shape of a parallelogram with horizontal side length W and vertical side length h. The red rectangle area at position $(w+2, 1-h)$ (w.r.t. its origin at the bottom left corner) aligns with the flat bottom block above encoding the letter 1 (as opposed to a spiked-block encoding a 0). The next block will start folding at the top right corner, at $(W-1, 1-h)$.

Figure 17 Geometry of the Copy▶ blocks. The Copy0▶ and Copy1▶ blocks have both the shape of a parallelogram with horizontal side length W and vertical side length h. For both, the next block will start folding at the top right corner, at $(W-1, 1-h)$. Note that the Copy0▶ and Copy1▶ blocks have identical internal structure apart from the line joining the two purple areas at $(w+2, 1-h)$ and $(h+w+1, 1)$. Indeed, when folding, the part of the transcript located in the red area, either: (1) detects a spike on top (encoding a 0) and then folds into a dent on top which induces spike at the bottom (copying the 0 below, the block Copy0▶); or (2) folds flat (encoding a 1) on top which induces a flat folding at the bottom, copying the 1 from the top to the bottom of the Zig-row (the block Copy1▶). Furthermore, this block is made of n Copy▶Unit blocks (from left to right), each of width $w+6$ and height h.



(a) The Copy0▶ block has a dent (an empty position) located at $(w+2, 1-h)$, in which plugs the spike of the block from the row above it, and which induces (when folding) a spike at the bottom at $(h+w+1, 1)$, copying the letter 0 from the top to the bottom of the Zig-row. Note that this block is made of n Copy▶Unit blocks (from left to right): one Copy0▶Unit followed by $n-1$ Copy1▶Unit, each of width $w+6$ and height h.



(b) The Copy1▶ block is flat at $(w+2, 1-h)$, which, aligned with a flat block above (encoding a 1), induces (when folding) a flat bottom at $(h+w, 0)$, copying the letter 1 from the top to the bottom of the Zig-row. Note that this block is made of n Copy1▶Unit blocks (from left to right), each of width $w+6$ and height h.

Figure 18 Geometry of the Seed block. This block encodes the initial word so that the oritatami system simulates properly the corresponding tag system. It consists of placing the different letter at the expected Write positions. Its rightmost part consists in a northeast-bound segment signalling the end of the (initial) word. Its leftmost part ends at the position $(-1, 0)$ where the transcript will start folding the first zig-row.

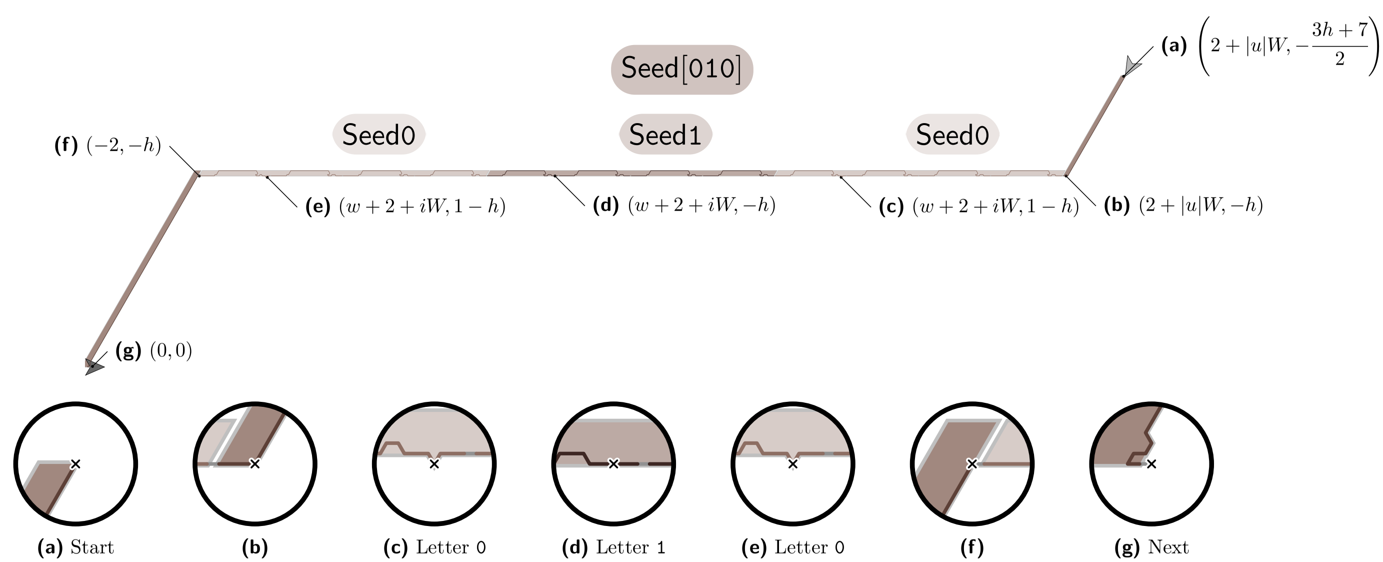
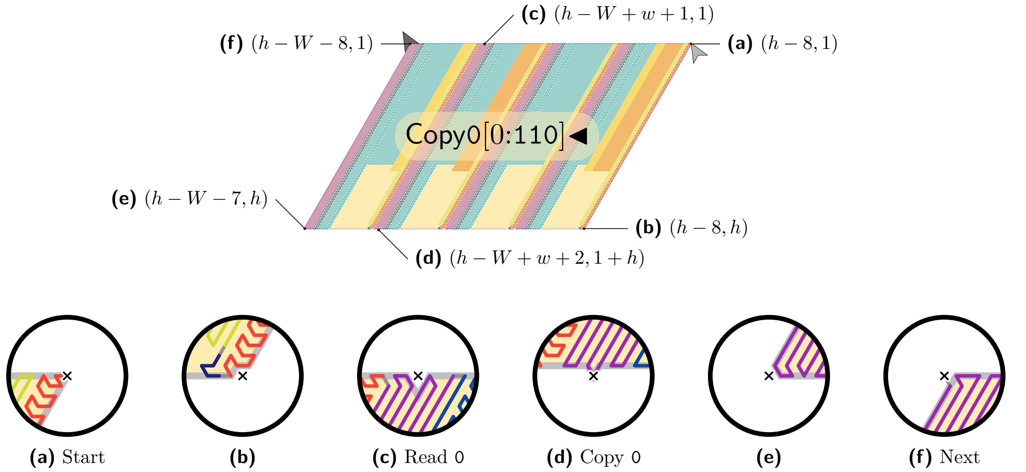
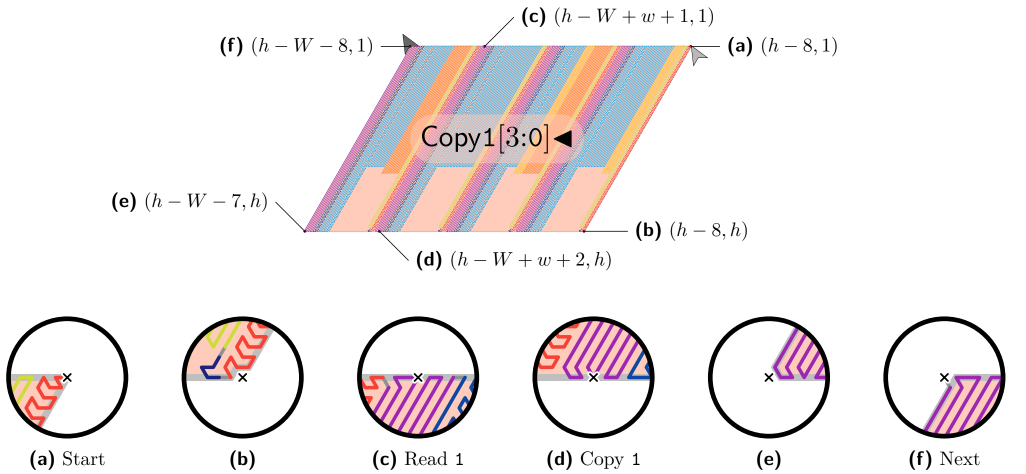


Figure 19 Geometry of the Copy◀ blocks. The Copy0◀ and Copy1◀ blocks are the horizontal mirror images of the Copy0▶ and Copy1▶ blocks (see Figure 17).

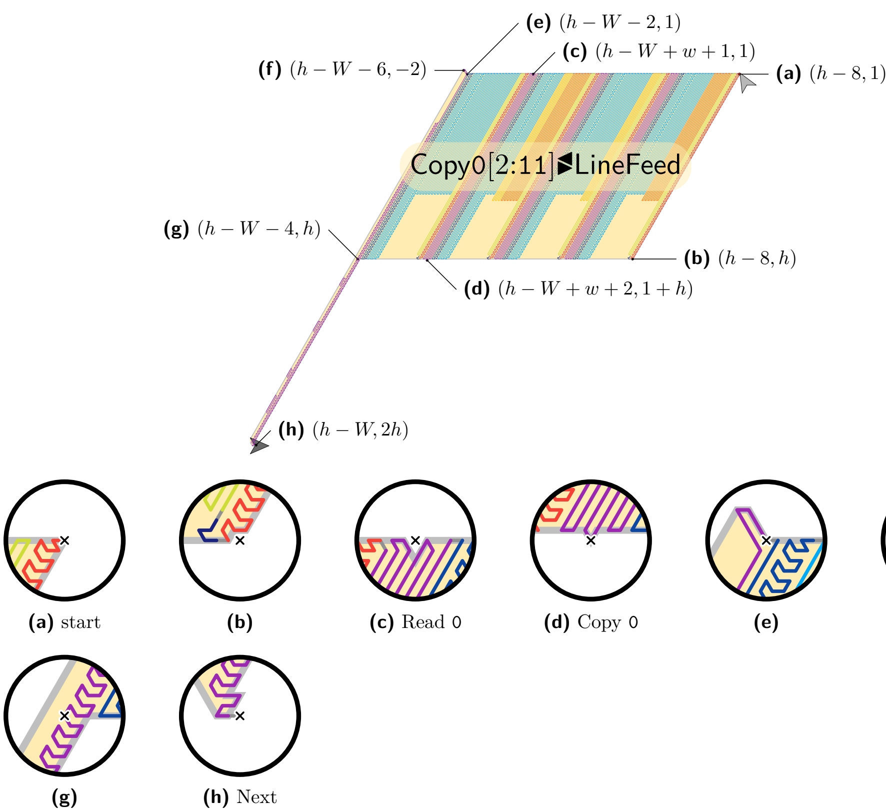


(a) The Copy0◀ block is the horizontal mirror image of the Copy0▶ block (see Figure 17(a)).



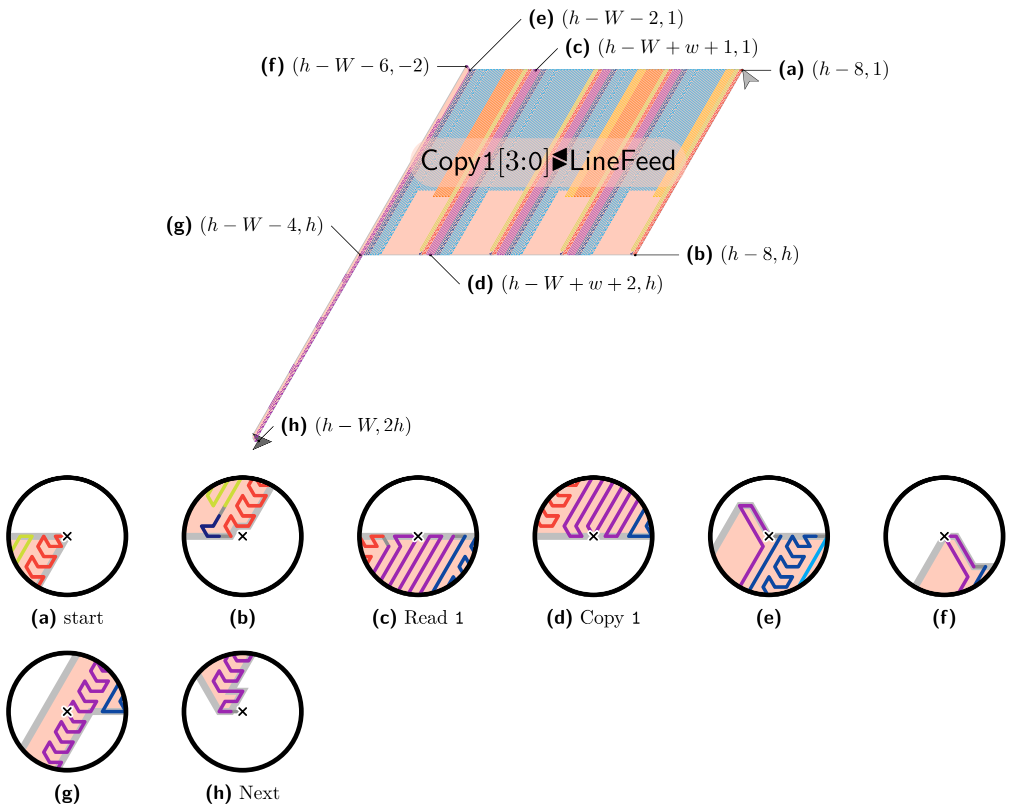
(b) The Copy1◀ block is the horizontal mirror image of the Copy1▶ block (see Figure 17(b))

Figure 20 Geometry of the Copy◀▶LineFeed blocks. These blocks adopt the shape of a $(W-6) \times h$-parallelogram prolongated by an southwestbound "arm" hoping to the beginning of the next zig-row. Folding from right to left, the Copy◀▶LineFeed blocks are identical to the Copy◀ blocks until position $(h-W-2,0)$ where it detects that there are no more blocks (encoding letter) in the row above (the detection of the absence of a block on top is made possible by the horizontal offset of 7 beads between the zig- and zag-rows). Then, instead of completing a parallelogram, the end of the Copy◀▶LineFeed blocks is attracted upwards and then folds into a southwestbound glider pattern to reach the opening position of the next zig-row. The next block will start folding at $(h-W, 2h)$. Furthermore, this block is made (from right to left) of $n-2$ Copy1◀Unit blocks followed by a Copy(x)◀Unit and a LineFeed◀▶Unit, where x is the letter copied.



(a) The Copy0◀▶LineFeed block proceeds as Copy0◀ to copy the spike encoding a 0 from the row above to the row below. It has a dent (an empty position) at $(h-W+w+1,1)$ in which plugs the spike (encoding a 0) of the block above. When folding, this dent induces a spike at the bottom at position $(h-W+w+2,1+h)$. Note that the spike below is at position $(h-W+w+2,1+h)$, which is consistent with the position of the dent in the Read0▶ block that will fold from $(h-W,2h)$ (see Figure 16(a)).

Figure 20 Geometry of the Copy◀LineFeed blocks. (Continued)



(b) The Copy1◀LineFeed block.

Figure 21 Geometry of the Append(u)**◀Return blocks.** The folding into this block is triggered by the absence of a block in row above (indicating the end of the word). It has one northeastbound 2-beads wide arm climbing along the east side of the block in the row above then a southeastbound 4-beads wide arm stopping at the bottom of the current zig-row. Then, the block consists in an 3-beads high $|u|W$-beads long eastbound glider path going along the bottom of the current zig-row and encoding each letter of u: the path contains a spike (below, and a dent on top) for each $u_j = \texttt{0}$ at position $(jW + w + h + 1, 1)$ (1s are encoded by the absence of spike). It then expands upto position $(|u|W + h + c(|u|), 0)$ and go back to its origin and grows a 10-beads wide h-beads high southwestbound arm opening the next zag-row to end at the position $(|u|W + h - 9, 2)$, at the top right corner of the upcoming zag-row. The next block will start at $(|u|W + h - 8, 1)$.

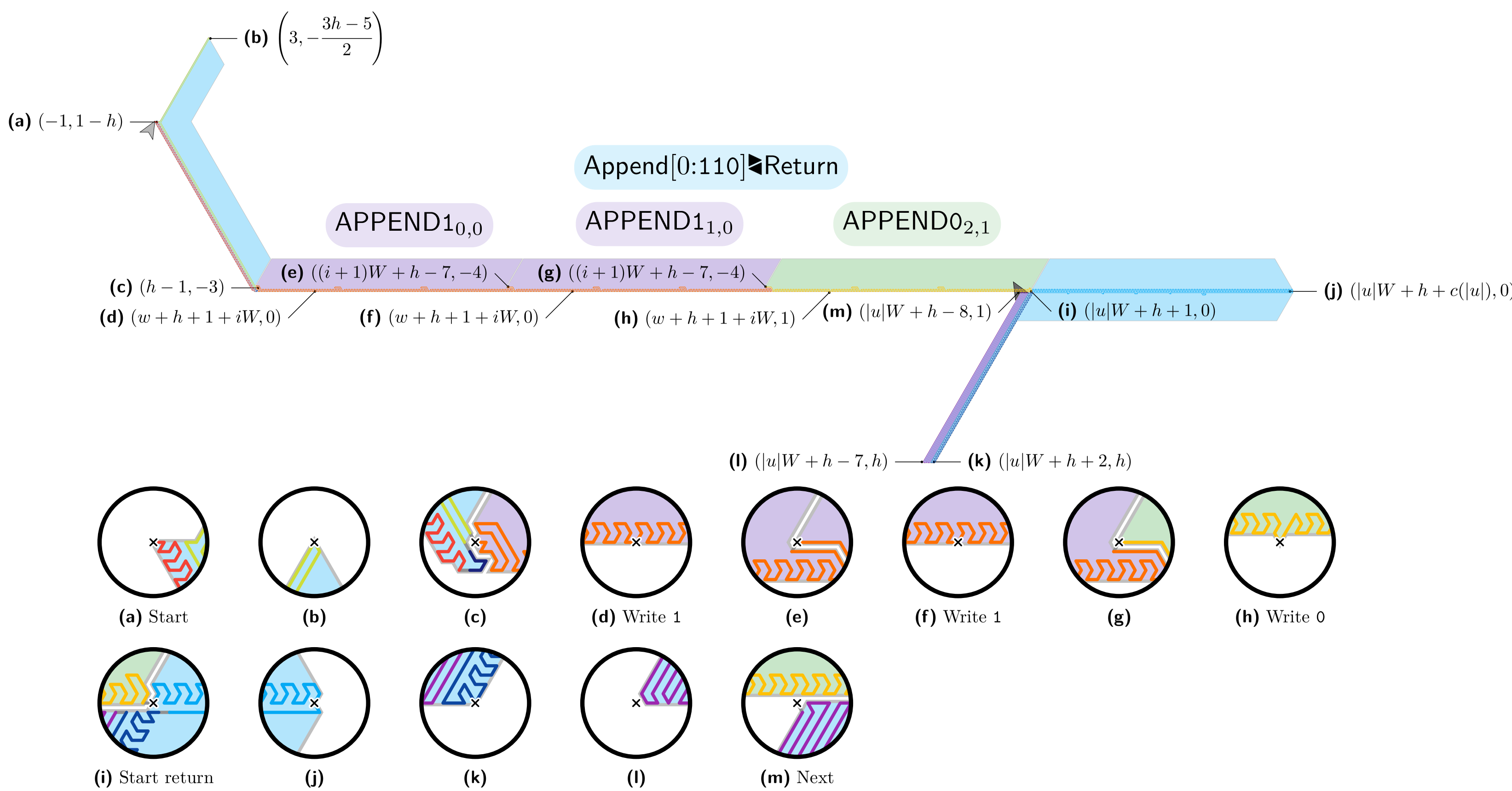


Figure 22 Geometry of the Append(ϵ)◀Return block. This block is the special case of Figure 21 where $u = \epsilon$. It is given for clarity.

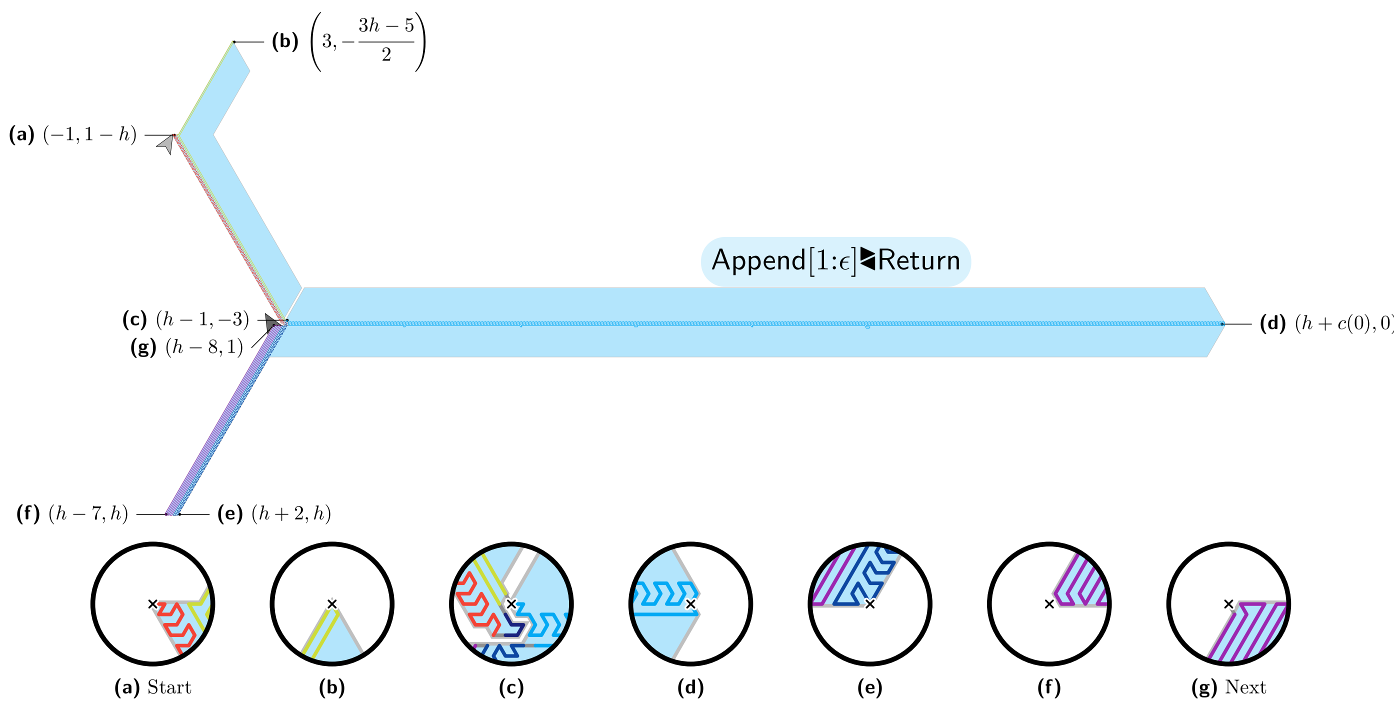


Figure 23 Geometry of the CarriageReturn◀LineFeed◀Halt■ block. This block is identical to the Append(ϵ)◀Return block until it reaches position $(h-1,1)$. Then, when, folding, it detects the absence of a block above which indicates that the current word is empty. It then folds as the lefttmost part of the Copy◀LineFeed blocks (see Figure 20) to open a new zig-row at $(h-1,2h)$. It then goes up to $(h,h-1)$. And as there are no block on the zag-row above, it is attracted inside itself and gets blocked at $(h-1,2h-3)$.

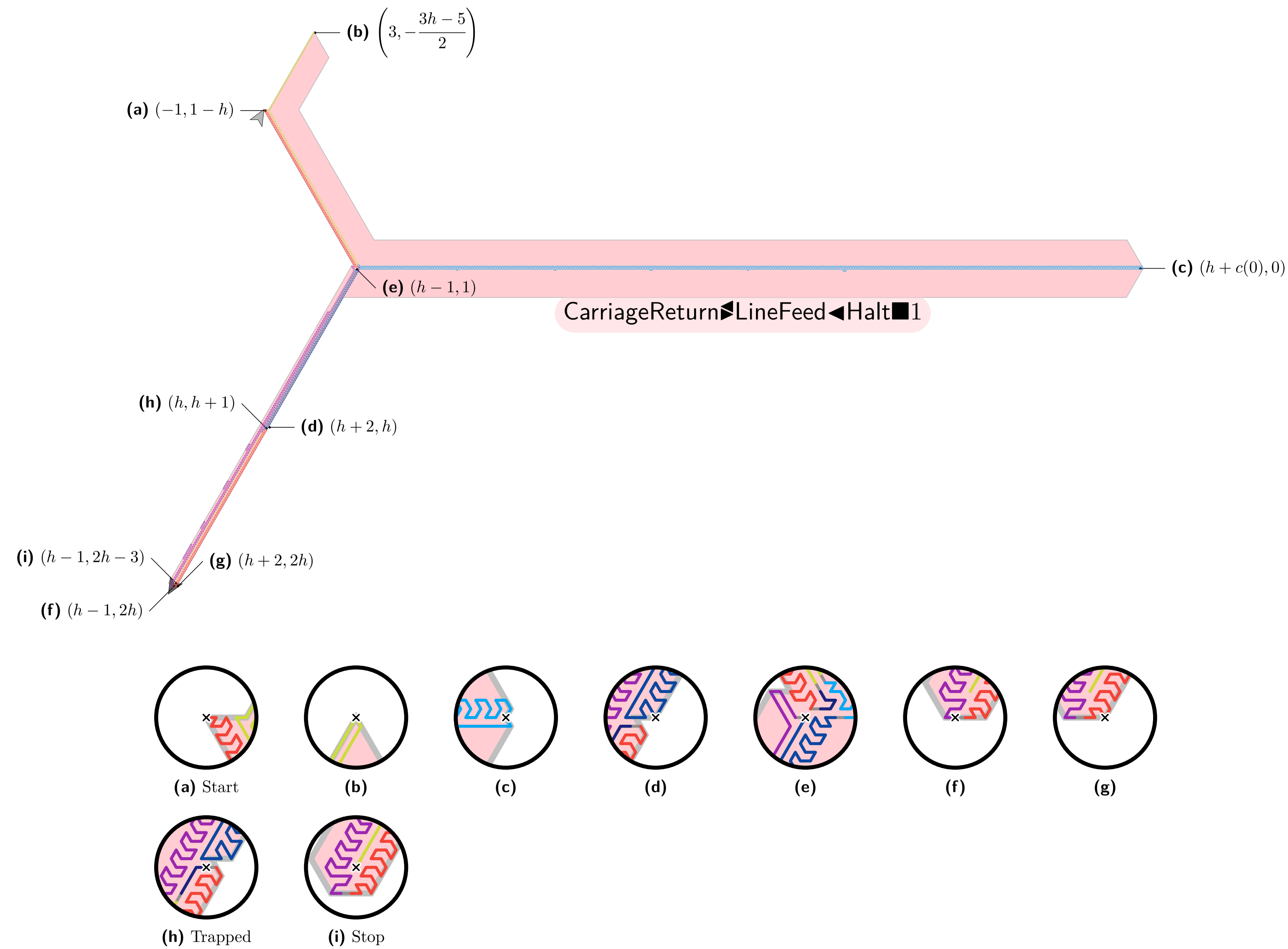
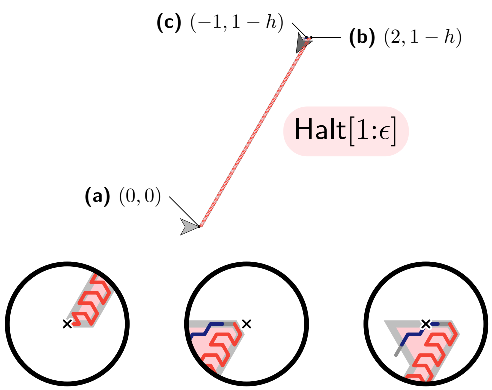


Figure 24 Geometry of the Halt block. This block appears at the end of the computation. It starts as a Read▶ block with a 3-beads wide h-beads high southeastbound glider until it reaches position $(2, 1-h)$. But, as there are no block in the zag-row above, the next beads are attracted to the left and the construction stops there.



(a) Start **(b)** **(c)** Trapped

F Full description of the SCTS oritatami simulator

Consider a SCTS $\mathcal{S}$ with n appendants $\alpha_0, \ldots, \alpha_{n-1} \in \{0,1\}^*$, where n is at least 8 and a multiple of 4, together with an input dataword $u \in \{0,1\}^*$.

We give here the full description of the primary structure $\pi_{\mathcal{S}}$ of the oritatami systems $\mathcal{O}_{\mathcal{S}} = ((\pi_{\mathcal{S}})^\infty, ♥, 3)$ together with its seed conformation $\sigma_{\mathcal{S}}(u)$, which simulates step by step the computation of $\mathcal{S}$ on input dataword u.

Convenience variables. In order to simplify the description, we introduce the following variables which correspond to key geometrical parameters of the different modules: (note that some of the variables were already introduced in Section E)

$L = \max_{0 \leqslant i < n} \lvert\alpha_i\rvert$	the maximum length of an appendant in $\mathcal{S}$
$P = 12 - (L \bmod 2)$	the *padding* constant, such that $L + P$ is even and at least 12
$w = 6(L + P) + 18$	the width of the appendant module excluding its read/copy/linefeed part
$W = n \cdot (w + 6)$	the width of the Read▶, Copy▶, and Copy◀ blocks
$h = W - (w + 3)$	the height of the zig and zag rows
$k = (h - 3)/6$	the number of periods of a glider of length h
$\lambda = W/2$	the height $(\lambda + 5)$ of the letter modules inside the Read▶, Copy▶, and Copy◀ blocks
$\kappa = W/24$	the number of periods of the glider/switchback pattern in the letter and padding modules
$q = (h - 3)/4$	the number of periods of the glider in the backbone of module E
$c(a) = (6\lambda(L - a + P) + 8h - 16)/4$	the width of the brick E_a◣

As n is a multiple of 4 and as $L + P$ is even, we have:

▶ **Fact 8.** *All convenience variables $L, P, w, W, h, k, \lambda, \kappa, q, c(a)$ are integers. Furthermore,*

$$w = 6 \bmod 12, \quad W = 0 \bmod 48, \quad h = 3 \bmod 12, \quad k = 0 \bmod 2,$$
$$\lambda = 0 \bmod 24, \quad \kappa = 0 \bmod 2, \quad q = 0 \bmod 3, \quad c(a) = 2 \bmod 12, \text{ for all } 0 \leqslant a \leqslant L.$$

Notations for describing of the bead type sequences. if u and v are two finite bead type sequences, we write their concatenation as $u \cdot v$. For any two integers $0 \leqslant i \leqslant j < |u|$, we write $u_{i\ldots j}$ for $u_i u_{i+1} \ldots u_j$. The *reverse sequence* of u, written as u^R, is $u_{|u|-1} u_{|u|-2} \ldots u_1 u_0$.

Finally, given a sequence u, we write $u\langle\langle a_1 @ i_1, \ldots, a_k @ i_k\rangle\rangle$ for the sequence w where the bead type indexed by i_j in u has been replaced by a_j for $j = 1, \ldots, k$:

$$w_i = \begin{cases} a_j & \text{if } i = i_j \text{ for some } j \\ u_i & \text{otherwise} \end{cases}$$

By extension, we write $u\langle\langle v @ k..l\rangle\rangle$ for the sequence w where for all $i \in \{k, k+1, \ldots, l\}$, the beads at indices k to l of u have been replaced by the word v (of length $l - k + 1$):

$$w_i = \begin{cases} v_{i-k} & \text{if } k \leqslant i \leqslant l \\ u_i & \text{otherwise} \end{cases}$$

For an infinite sequence of (finite) words $(u_i)_{i \geqslant 1}$, we denote by $\bigodot_{i \geqslant 1} u_i$ the infinite word $u_1 u_2 \cdots u_i \cdots$ obtained by containing all the words $u_1, u_2, \ldots$

Notation for describing conformations. Given an infinite sequence of directions $(d_i) \in \{\overset{\texttt{NE}}{\nearrow}, \overset{\rightarrow}{\texttt{E}}, \underset{\texttt{SE}}{\searrow}, \underset{\texttt{SW}}{\swarrow}, \overset{\leftarrow}{\texttt{W}}, \overset{\texttt{NW}}{\nwarrow}\}^{\mathbb{N}}$, and a finite bead type sequence $b \in B^*$, we denote by $\mathsf{Conformation}(b, d)$ the conformation $b_0 d_0 b_1 d_1 \cdots d_{|b|-2} b_{|b|-1}$ that maps b along the path d. We will use the following convenience functions which will ease the description of the bricks:

- $\mathsf{E\text{-}path}(b) = \mathsf{Conformation}(b, (\overset{\rightarrow}{\texttt{E}})^\infty)$ and similarly, SE-path, SW-path, W-path, NW-path and NE-path that map a bead type sequence along the paths $(\underset{\texttt{SE}}{\searrow})^\infty$, $(\underset{\texttt{SW}}{\swarrow})^\infty$, $(\overset{\leftarrow}{\texttt{W}})^\infty$, $(\overset{\texttt{NW}}{\nwarrow})^\infty$, and $(\overset{\texttt{NE}}{\nearrow})^\infty$ respectively.
- $\mathsf{E\text{-}glider}(b) = \mathsf{Conformation}\left(b, \left(\overset{\texttt{NE}}{\nearrow}, \overset{\texttt{NW}}{\nwarrow}, \overset{\rightarrow}{\texttt{E}}, \underset{\texttt{SE}}{\searrow}, \underset{\texttt{SW}}{\swarrow}, \overset{\rightarrow}{\texttt{E}}\right)^\infty\right)$
- $\mathsf{E\text{-}glider}'(b) = \mathsf{Conformation}\left(b, \left(\underset{\texttt{SE}}{\searrow}, \underset{\texttt{SW}}{\swarrow}, \overset{\rightarrow}{\texttt{E}}, \overset{\texttt{NE}}{\nearrow}, \overset{\texttt{NW}}{\nwarrow}, \overset{\rightarrow}{\texttt{E}}\right)^\infty\right)$
- $\mathsf{SE\text{-}rev\text{-}glider}(b) = \mathsf{Conformation}\left(b, \left(\overset{\texttt{NE}}{\nearrow}, \overset{\rightarrow}{\texttt{E}}, \underset{\texttt{SE}}{\searrow}, \overset{\leftarrow}{\texttt{W}}, \underset{\texttt{SW}}{\swarrow}, \underset{\texttt{SE}}{\searrow}\right)^\infty\right)$
- $\mathsf{NE\text{-}glider}(b) = \mathsf{Conformation}\left(b, \left(\overset{\texttt{NW}}{\nwarrow}, \overset{\leftarrow}{\texttt{W}}, \overset{\texttt{NE}}{\nearrow}, \overset{\rightarrow}{\texttt{E}}, \underset{\texttt{SE}}{\searrow}, \overset{\texttt{NE}}{\nearrow}\right)^\infty\right)$

Recall that the coordinates are expressed according to the east and south-west axis: every position (x, y) in $\mathbb{T}$ is mapped in the euclidean plane to $x \cdot \vec{E} + y \cdot \vec{SW}$ using the vector basis $\vec{E} = (1, 0)$ and $\vec{SW} = \text{RotateClockwise}\left(\vec{E}, 120°\right) = (-\frac{1}{2}, -\frac{\sqrt{3}}{2})$.

F.1 The periodic primary structure $(\pi_{\mathcal{S}})^\infty$

The period of the primary structure consists in the concatenation of one appendant bead type sequence for each appendant:

$$\pi_{\mathcal{S}} = \boxed{\textsf{Appendant } \alpha^0} \cdot \boxed{\textsf{Appendant } \alpha^1} \cdot \cdots \cdot \boxed{\textsf{Appendant } \alpha^{n-1}}$$

F.2 The appendant bead type sequences

Each appendant bead type sequence $\boxed{\textsf{Appendant } \alpha^i}$ has the exact same structure: it is the concatenation of 6 sequences: Modules $\boxed{\textsf{A}}$, $\boxed{\textsf{B}}$, and $\boxed{\textsf{C}}$, followed a sequence $\boxed{\textsf{word}(\alpha_i)}$ encoding the appendant α_i itself, followed by modules $\boxed{\textsf{F}}$ and $\boxed{\textsf{G}}$:

$$\boxed{\textsf{Appendant } \alpha^i} = \boxed{\textsf{A}} \cdot \boxed{\textsf{B}} \cdot \boxed{\textsf{C}} \cdot \boxed{\textsf{word}(\alpha_i)} \cdot \boxed{\textsf{F}} \cdot \boxed{\textsf{G}}$$

For each word $v \in \{0, 1\}^*$ with $|v| \leqslant L$, $\boxed{\textsf{word}(v)}$ encodes each letter of v using one of the 6 variants of module $\boxed{\textsf{D}}$ and terminates with a padding sequence $\boxed{\textsf{E}_{L-|v|}}$ which ensures that the folded size of $\boxed{\textsf{word}(v)}$ is independent of the length of v. The 6 variants of the module $\boxed{\textsf{D}}$ are $\boxed{\textsf{D}(x)_{r,t}}$ where:

- $x \in \{0, 1\}$ is the encoded letter;
- $r \in \{0, 1, 2\}$ is the rank of the letter inside the encoded word v: $r = 0$ if it is the first letter of v; $r = 1$ if its index in v is odd; and $r = 2$ if its index in v is even but not 0;
- $t \in \{0, 1\}$ is 1 if the encoded letter is the last letter of v, and 0 otherwise.

The definition of the sequence $\boxed{\textsf{word}(v)}$ follows as:

$$\boxed{\textsf{word}(\epsilon)} = \boxed{\textsf{E}_L}$$
$$\boxed{\textsf{word}(0)} = \boxed{\textsf{D}0_{0,1}} \cdot \boxed{\textsf{E}_{L-1}}$$
$$\boxed{\textsf{word}(1)} = \boxed{\textsf{D}1_{0,1}} \cdot \boxed{\textsf{E}_{L-1}}$$

and for $|v| \geqslant 2$, $$\boxed{\textsf{word}(v)} = \boxed{\textsf{D}(v_0)_{0,0}} \cdot \left(\bigodot_{i=1}^{|v|-2} \boxed{\textsf{D}(v_i)_{2-(i \bmod 2),0}}\right) \cdot \boxed{\textsf{D}(v_{n-1})_{2-((|v|-1) \bmod 2),1}} \cdot \boxed{\textsf{E}_{L-|v|}}$$

The next section concludes the full description of the primary structure by giving the sequences for modules **A**, **B**, **C**, **D**$(x)_{r,t}$, **E**$_a$, **F**, and **G**.

F.3 Modules bead type sequences and brick conformations

The modules are given using 546 bead types. However, 5 of them, **A6**, **C16**, **J9**, **L86**, and **L89**, are *neutral*, i.e. do not have any interaction with any other bead types (see Section H) and can thus be all substituted by one single *neutral* bead type **N0**. The total number of distinct bead types is thus 542 (541 + 1 neutral bead type **N0**). However, in the description bellow, we prefer to use **A6**, **C16**, **J9**, **L86**, and **L89** (and not **N0**) as it keeps the bead types homogenous and continuously numbered in each module.

F.4 Module A

F.4.1 Bead type sequence of Module A

Length: $3h-2$; 13 bead types used: **A0..12**

$$\boxed{\textsf{A}} = \textbf{A0..4} \cdot (\textbf{A5..10})^{3k-1} \cdot \textbf{A5..7} \cdot \textbf{A6} \cdot \textbf{A9..10} \cdot \textbf{A11..12}$$

F.4.2 The bricks for module A

Module **A** adopts three brick conformations: **A▶**, **A▶**, and **A◀**, where the two last ones are just obtained by mirroring and rotating the first:

$$\boxed{\textsf{A}\blacktriangleright} = \textbf{A0} \overset{\rightarrow}{\scriptstyle E} \textbf{A1} \overset{NW}{\nwarrow} \textbf{A2} \overset{\rightarrow}{\scriptstyle E} \textbf{A3} \underset{SE}{\searrow} \textbf{A4} \overset{NE}{\nearrow} (\textbf{A5} \overset{NW}{\nwarrow} \textbf{A6} \overset{\leftarrow}{\scriptstyle W} \textbf{A7} \overset{NE}{\nearrow} \textbf{A8} \overset{\rightarrow}{\scriptstyle E} \textbf{A9} \underset{SE}{\searrow} \textbf{A10} \overset{NE}{\nearrow})^{3k-1} \textbf{A5} \overset{NW}{\nwarrow}$$
$$\textbf{A6} \overset{\leftarrow}{\scriptstyle W} \textbf{A7} \overset{NE}{\nearrow} \textbf{A6} \overset{\rightarrow}{\scriptstyle E} \textbf{A9} \underset{SE}{\searrow} \textbf{A10} \overset{NE}{\nearrow} \textbf{A11} \overset{NW}{\nwarrow} \textbf{A12}$$

$$\boxed{\textsf{A}\blacktriangleright} = \text{HorizontalMirror}(\boxed{\textsf{A}\blacktriangleright})$$

$$\boxed{\textsf{A}\blacktriangleleft} = \text{Rotate}_{180^\circ}(\boxed{\textsf{A}\blacktriangleright})$$

Figure 25 displays brick **A▶**.

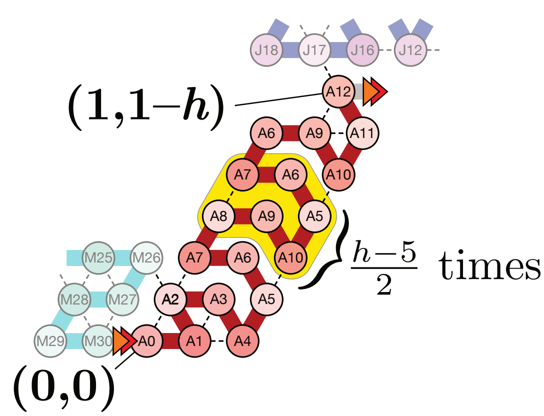


Figure 25 Module **A**: Brick **A▶**.

F.4.3 Subrule for Module A

Module [A] interacts with [A], [B], [C], [D], [F] and [G]. Figure 26 presents the subrule for the interactions between the beads of [A] and the beads of the other modules.

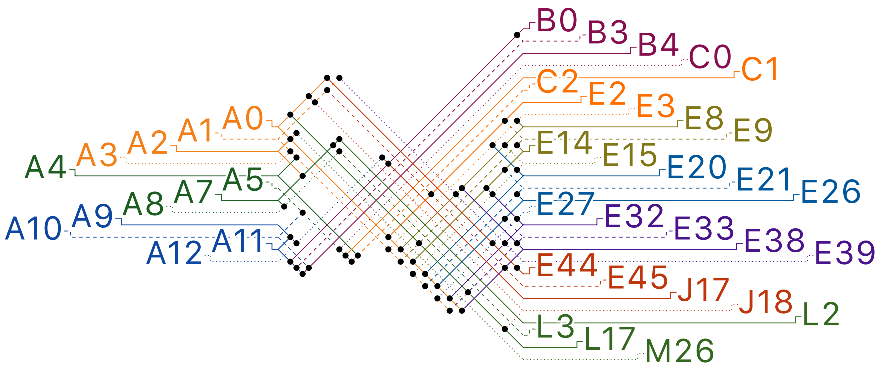


Figure 26 Subrule for Module [A].

F.5 Module B

F.5.1 Bead type sequence for Module B

Length: 5; 5 bead types used: **B0..4**

[B] = **B0..4**

F.5.2 The bricks for module B

Module [B] adopts four brick conformations plus some incomplete ones if the folding stops because the dataword in the simumlated SCTS is empty.

[B▶̲] = **B0** $\underset{E}{\rightarrow}$ **B1** $\underset{SE}{\searrow}$ **B2** $\underset{W}{\leftarrow}$ **B3** $\underset{SW}{\swarrow}$ **B4**

[B▶̄] = HorizontalMirror([B▶̲])

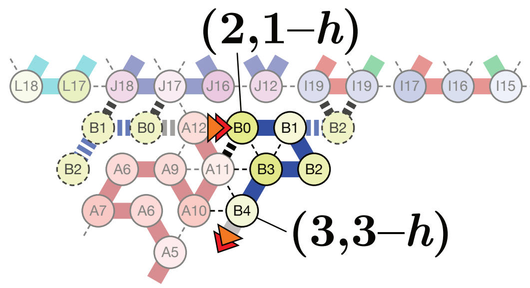
[B◀] = $\text{Rotate}_{180^\circ}$([B▶̲])

[B■] = **B0** $\underset{W}{\leftarrow}$ **B1** $\underset{SW}{\swarrow}$ **B2** $\underset{SW}{\swarrow}$ **B3** $\underset{SW}{\swarrow}$ **B4**

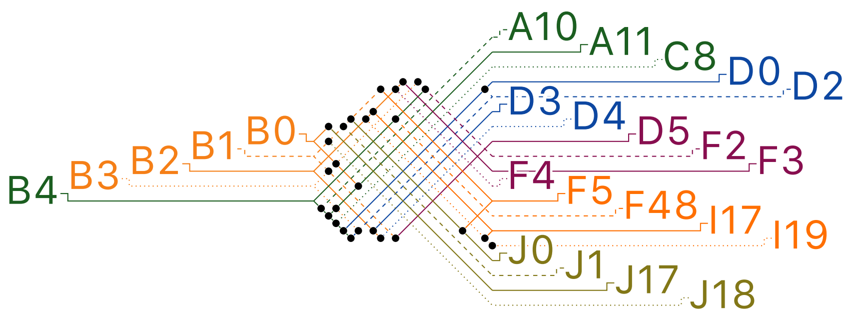
Figure 27 displays bricks [B▶̲] and [B■] (shaded).

F.5.3 Subrule for Module B

Module [B] interacts with [A], [C], [D], [E] and [F]. Figure 28 presents the subrule for the interactions between the beads of [B] and the beads of the other modules.



■ **Figure 27** Module B: Brick B▶ to the right, and brick B■ shaded to the left.



■ **Figure 28** Subrule for Module B.

F.6 Module C

F.6.1 Bead type sequence for Module C

Length: $3h - 10$; 17 bead types used: **C0..16**

$$\boxed{\mathsf{C}} = (\mathbf{C0..2})^{2k} \cdot (\mathbf{C3..5})^{k} \cdot \mathbf{C3} \cdot \mathbf{C7..8} \cdot (\mathbf{C6..8})^{k-1} \cdot (\mathbf{C9..14})^{k-1} \cdot \mathbf{C9..10} \cdot \mathbf{C15..16} \cdot \mathbf{C13}$$

F.6.2 The bricks for module C

Module C adopts four brick conformations:

$$\boxed{\mathsf{C}\underline{\blacktriangleright}} = (\mathbf{C0} \underset{\texttt{SW}}{\swarrow} \mathbf{C1} \underset{\texttt{SW}}{\swarrow} \mathbf{C2} \underset{\texttt{SW}}{\swarrow})^{2k-1} \mathbf{C0} \underset{\texttt{SW}}{\swarrow} \mathbf{C1} \underset{\texttt{SW}}{\swarrow} \mathbf{C2} \underset{\texttt{E}}{\rightarrow}$$
$$(\mathbf{C3} \overset{\texttt{NE}}{\nearrow} \mathbf{C4} \overset{\texttt{NE}}{\nearrow} \mathbf{C5} \overset{\texttt{NE}}{\nearrow})^{k} \mathbf{C3} \overset{\texttt{NE}}{\nearrow} \mathbf{C7} \overset{\texttt{NE}}{\nearrow} (\mathbf{C8} \overset{\texttt{NE}}{\nearrow} \mathbf{C6} \overset{\texttt{NE}}{\nearrow} \mathbf{C7} \overset{\texttt{NE}}{\nearrow})^{k-1} \mathbf{C8} \underset{\texttt{SE}}{\searrow}$$
$$(\mathbf{C9} \underset{\texttt{SW}}{\swarrow} \mathbf{C10} \underset{\texttt{SW}}{\swarrow} \mathbf{C11} \underset{\texttt{SW}}{\swarrow} \mathbf{C12} \underset{\texttt{SW}}{\swarrow} \mathbf{C13} \underset{\texttt{SW}}{\swarrow} \mathbf{C14} \underset{\texttt{SW}}{\swarrow})^{k-1} \mathbf{C9} \underset{\texttt{SW}}{\swarrow} \mathbf{C10} \underset{\texttt{SW}}{\swarrow} \mathbf{C15} \underset{\texttt{SW}}{\swarrow} \mathbf{C16} \underset{\texttt{SW}}{\swarrow} \mathbf{C13}$$

$$\boxed{\mathsf{C}\overline{\blacktriangleright}} = \mathrm{HorizontalMirror}(\boxed{\mathsf{C}\underline{\blacktriangleright}})$$

$$\boxed{\mathsf{C}\overline{\blacktriangleleft}} = \mathrm{Rotate}_{180^\circ}(\boxed{\mathsf{C}\underline{\blacktriangleright}})$$

$$\boxed{\mathsf{C}\overline{\blacktriangleright}\,\mathsf{End}} = (\mathbf{C0} \overset{\texttt{NW}}{\nwarrow} \mathbf{C1} \overset{\texttt{NW}}{\nwarrow} \mathbf{C2} \overset{\texttt{NW}}{\nwarrow})^{2k-1} \mathbf{C0} \overset{\texttt{NW}}{\nwarrow} \mathbf{C1} \overset{\texttt{NW}}{\nwarrow} \mathbf{C2} \underset{\texttt{E}}{\rightarrow}$$
$$(\mathbf{C3} \overset{\texttt{NE}}{\nearrow} \mathbf{C4} \overset{\texttt{NE}}{\nearrow} \mathbf{C5} \overset{\texttt{NE}}{\nearrow})^{k} \mathbf{C3} \underset{\texttt{SE}}{\searrow}$$
$$\mathbf{C7} \underset{\texttt{SW}}{\swarrow} \mathbf{C8} \underset{\texttt{SW}}{\swarrow} (\mathbf{C6} \underset{\texttt{SW}}{\swarrow} \mathbf{C7} \underset{\texttt{SW}}{\swarrow} \mathbf{C8} \underset{\texttt{SW}}{\swarrow})^{k-1} \mathbf{C9} \underset{\texttt{SW}}{\swarrow}$$
$$(\mathbf{C10} \underset{\texttt{SE}}{\searrow} \mathbf{C11} \underset{\texttt{SE}}{\searrow} \mathbf{C12} \underset{\texttt{SE}}{\searrow} \mathbf{C13} \underset{\texttt{SE}}{\searrow} \mathbf{C14} \underset{\texttt{SE}}{\searrow} \mathbf{C9} \underset{\texttt{SE}}{\searrow})^{k-1} \underset{\texttt{SE}}{\searrow} \mathbf{C10} \underset{\texttt{SE}}{\searrow} \mathbf{C15} \underset{\texttt{SE}}{\searrow} \mathbf{C16} \underset{\texttt{SE}}{\searrow} \mathbf{C13}$$

Figures 29 and Figure 30 display bricks $\boxed{\mathsf{C}\underline{\blacktriangleright}}$ and $\boxed{\mathsf{C}\overline{\blacktriangleright}\,\mathsf{End}}$ respectively.

F.6.3 Subrule for Module C

Module C interacts with A, B, D, E and F and might interact with G. Figure 31 presents the subrule for the interactions between the beads of C and the beads of the other modules.

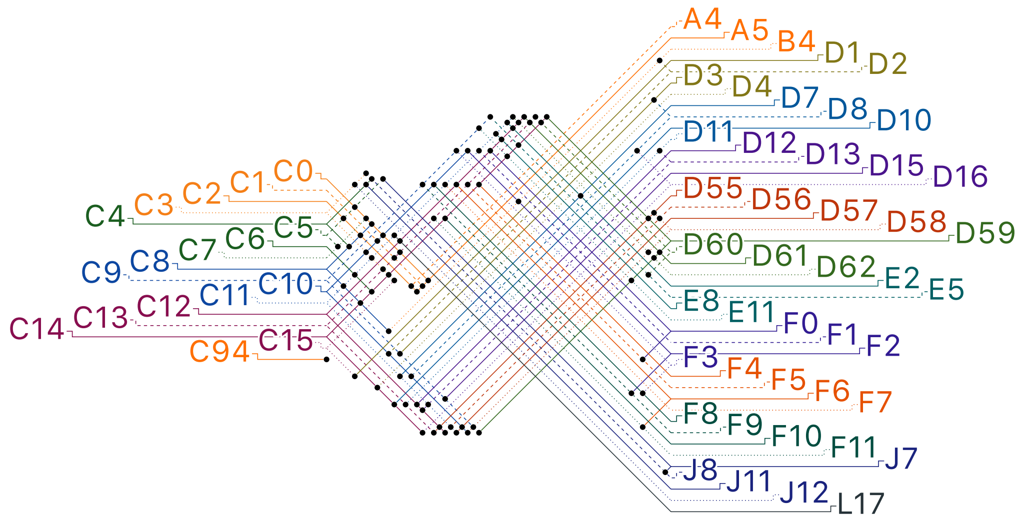


Figure 31 Subrule for Module C.

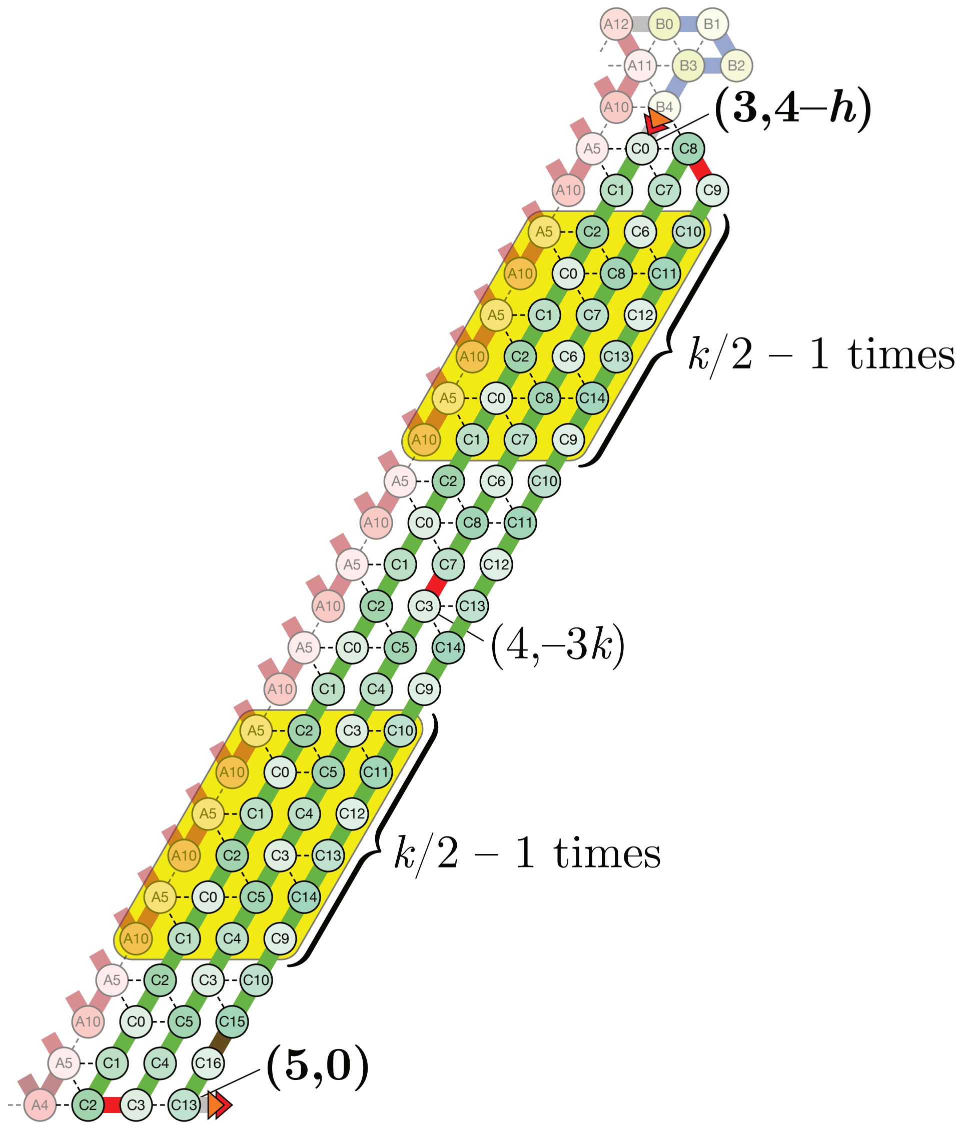


Figure 29 Module C: Brick C►.

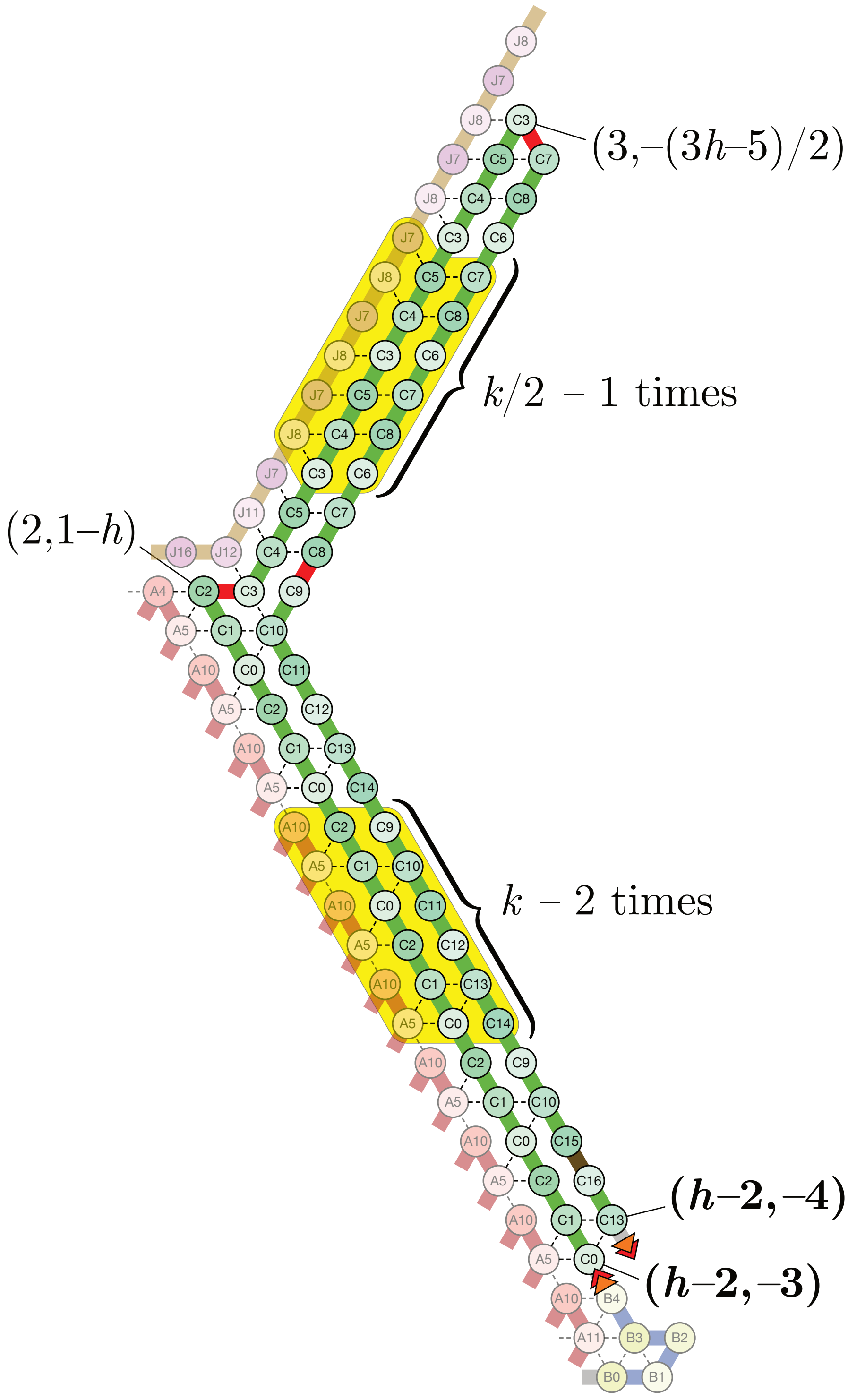


Figure 30 Module C: Brick C►End.

F.7 Modules D

F.7.1 Bead type sequence for Modules D

Length: $3W + 30 = 6(\lambda + 5)$;
uses: 111 proper bead types, **D0..62** and **E0..47**; plus 2 special bead types from **G**, **L17..18**

$$\mathsf{SegD}_0 = \mathbf{D23..33} \cdot \mathbf{E6..11} \cdot (\mathbf{E0..11})^{\kappa-1}$$
$$\mathsf{SegD}_1 = (\mathbf{E12..23})^{\kappa} \cdot \mathbf{D49..45}$$
$$\mathsf{SegD}_2 = \mathbf{D34..44} \cdot \mathbf{E30..35} \cdot (\mathbf{E24..35})^{\kappa-1}$$
$$\mathsf{SegD}_3 = (\mathbf{E36..47})^{\kappa} \cdot \mathbf{D54..50}$$
$$\mathsf{D1}_{2,0} = \mathsf{SegD}_0 \cdot \mathsf{SegD}_1 \cdot \mathsf{SegD}_2 \cdot \mathsf{SegD}_3 \cdot \mathsf{SegD}_0 \cdot \mathsf{SegD}_1$$
$$\mathsf{D1}_{1,0} = \mathsf{SegD}_2 \cdot \mathsf{SegD}_3 \cdot \mathsf{SegD}_0 \cdot \mathsf{SegD}_1 \cdot \mathsf{SegD}_2 \cdot \mathsf{SegD}_3$$
$$\mathsf{D1}_{0,0} = \mathsf{D1}_{2,0}\langle\!\langle \mathbf{D0..16}@0..16 \rangle\!\rangle$$
$$\mathsf{D1}_{r,1} = \mathsf{D1}_{r,0}\langle\!\langle \mathbf{D17}@(3W+22), \mathbf{D18..22}@(3W+25)..(3W+29) \rangle\!\rangle \qquad \text{for } r \in \{0,1,2\}$$
$$\mathsf{D0}_{r,t} = \mathsf{D1}_{r,t}\langle\!\langle \mathbf{L17}@(3w+1), \mathbf{L18}@(3w+2), \mathbf{D55..62}@(3w+6)..(3w+13) \rangle\!\rangle \qquad \text{for } r \in \{0,1,2\} \text{ and } t \in \{0,1\}$$

F.7.2 The bricks for the modules D

F.7.2.1 The zig and zag brick conformations for Module D

$$\mathsf{SegD}_0 \blacktriangleright = \text{NE-path}\big(\mathbf{D23..33} \cdot \mathbf{E6..11} \cdot (\mathbf{E0..11})^{\kappa-1}\big)$$
$$\mathsf{SegD}_1 \blacktriangleright = \text{SW-path}\big((\mathbf{E12..23})^{\kappa} \cdot \mathbf{D49..45}\big)$$
$$\mathsf{SegD}_2 \blacktriangleright = \text{NE-path}\big(\mathbf{D34..44} \cdot \mathbf{E30..35} \cdot (\mathbf{E24..35})^{\kappa-1}\big)$$
$$\mathsf{SegD}_3 \blacktriangleright = \text{SW-path}\big((\mathbf{E36..47})^{\kappa} \cdot \mathbf{D54..50}\big)$$

$$\mathsf{D1}_{2,0} \blacktriangleright = \mathsf{SegD}_0 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_1 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_2 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_3 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_0 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_1 \blacktriangleright$$
$$\mathsf{D1}_{1,0} \blacktriangleright = \mathsf{SegD}_2 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_3 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_0 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_1 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_2 \blacktriangleright \overset{\rightarrow}{\mathrm{E}} \mathsf{SegD}_3 \blacktriangleright$$
$$\mathsf{D1}_{0,0} \blacktriangleright = \mathsf{D1}_{2,0} \blacktriangleright \langle\!\langle (\text{NE-path}(\mathbf{D0..16}) \overset{\mathrm{NE}}{\nearrow})@0..16 \rangle\!\rangle$$
$$\mathsf{D1}_{r,1} \blacktriangleright = \mathsf{D1}_{r,0} \blacktriangleright \langle\!\langle (\mathbf{D17} \underset{\mathrm{SW}}{\swarrow})@(3W+22), \text{SW-path}(\mathbf{D18..22})@(3W+25)..(3W+29) \rangle\!\rangle \qquad \text{for } r \in \{0,1,2\}$$
$$\mathsf{D0}_{r,t} \blacktriangleright = \mathsf{D1}_{r,t} \blacktriangleright \langle\!\langle (\mathbf{L17} \overset{\mathrm{NE}}{\nearrow} \mathbf{L18} \overset{\mathrm{NE}}{\nearrow})@(3w+1..3w+2), (\text{NE-path}(\mathbf{D55..62}) \overset{\mathrm{NE}}{\nearrow})@(3w+6)..(3w+13) \rangle\!\rangle \qquad \text{for } r \in \{0,1,2\} \text{ and } t \in \{0,1\}$$

The zig-down and zag brick conformations are obtained by mirroring and rotating the zig-up brick conformation: for $x \in \{0,1\}$, $r \in \{0,1,2\}$ and $t \in \{0,1\}$, we have

$$\mathsf{D}(x)_{r,t} \blacktriangleright = \text{HorizontalMirror}(\mathsf{D}(x)_{r,t} \blacktriangleright)$$
$$\mathsf{D}(x)_{r,t} \blacktriangleleft = \text{Rotate}_{180^\circ}(\mathsf{D}(x)_{r,t} \blacktriangleright)$$

Figure 32 displays the bricks $\mathsf{D0}_{r,t} \blacktriangleright$.

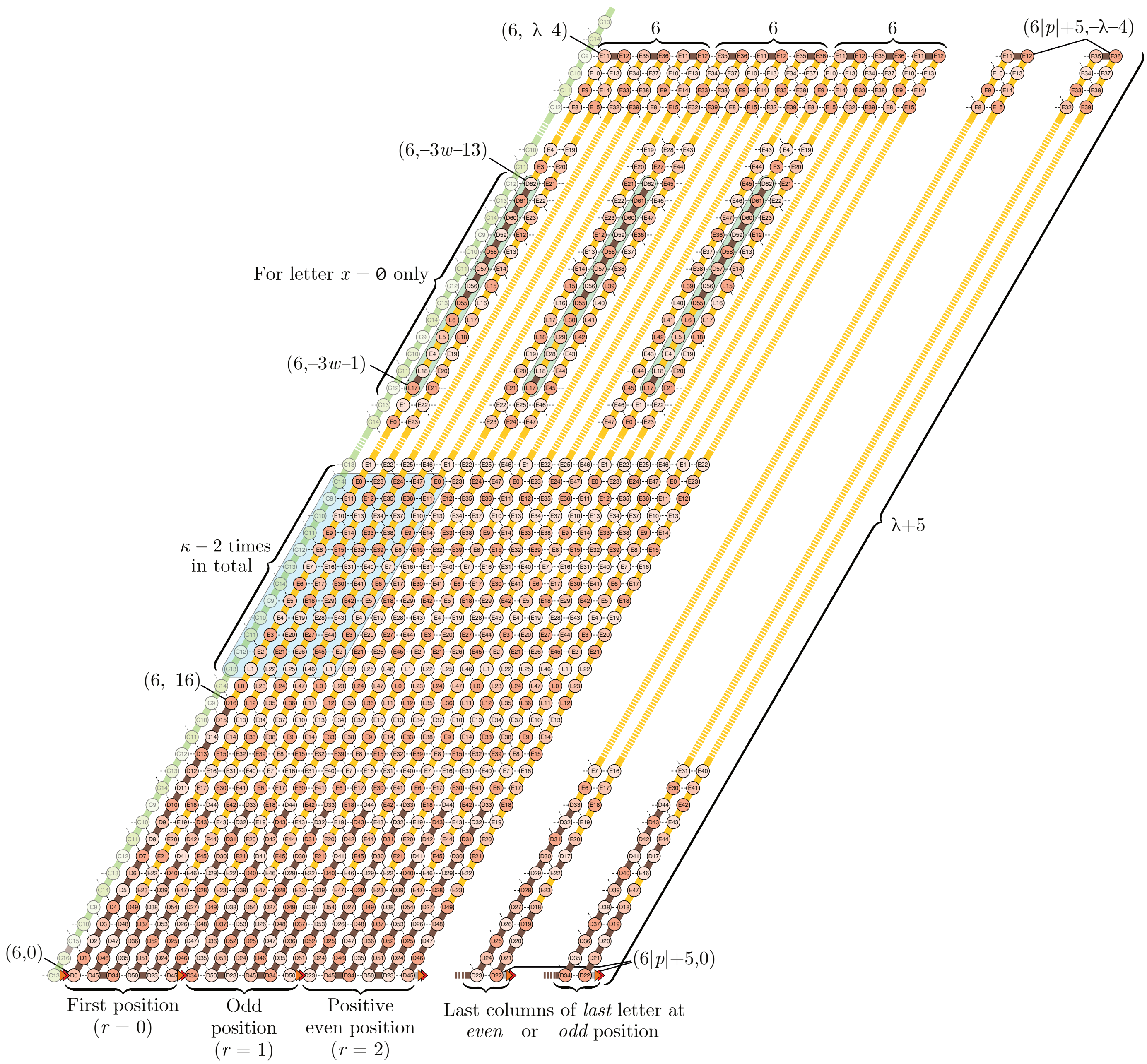


Figure 32 Modules $\mathsf{D0}_{r,t}$: Bricks $\mathsf{D0}_{r,t}$▶.

F.7.2.2 The append brick conformations for Module D

SegD$_0$✎ = E-path(**D23..27**) $\underset{\text{SE}}{\searrow}$ SE-path(**D28..30**) $\underset{\text{SW}}{\swarrow}$ **D31** $\underset{\text{E}}{\to}$ **D32** $\overset{\text{NE}}{\nearrow}$ **D33** $\overset{\text{NE}}{\nearrow}$ E-glider′(**E6..11** · (**E0..11**)$^{\kappa-1}$)

SegD$_1$✎ = E-glider′((**E12..23**)$^{\kappa}$) $\overset{\text{NW}}{\nwarrow}$ W-path(**D49..45**)

SegD$_2$✎ = E-path(**D34..38**) $\underset{\text{SE}}{\searrow}$ SE-path(**D39..41**) $\underset{\text{SW}}{\swarrow}$ **D42** $\underset{\text{E}}{\to}$ **D43** $\overset{\text{NE}}{\nearrow}$ **D44** $\overset{\text{NE}}{\nearrow}$ E-glider′(**E30..35** · (**E24..35**)$^{\kappa-1}$)

SegD$_3$✎ = E-glider′((**E36..47**)$^{\kappa}$) $\overset{\text{NW}}{\nwarrow}$ W-path(**D54..50**)

SegD$_{init}$✎ = **D0** $\underset{\text{E}}{\to}$ **D1** $\underset{\text{SW}}{\swarrow}$ **D2** $\underset{\text{E}}{\to}$ **D3** $\underset{\text{SE}}{\searrow}$ **D4** $\underset{\text{SW}}{\swarrow}$ **D5** $\underset{\text{E}}{\to}$ **D6** $\overset{\text{NE}}{\nearrow}$ **D7** $\overset{\text{NW}}{\nwarrow}$ **D8** $\overset{\text{NW}}{\nwarrow}$ **D9** $\overset{\text{NW}}{\nwarrow}$ **D10** $\underset{\text{W}}{\leftarrow}$ **D11** $\overset{\text{NW}}{\nwarrow}$ **D12** $\underset{\text{E}}{\to}$ **D13** $\underset{\text{E}}{\to}$ **D14** $\underset{\text{SE}}{\searrow}$ **D15** $\underset{\text{SE}}{\searrow}$ **D16**

SegD$_{end}$✎ = **D18** $\underset{\text{E}}{\to}$ **D19** $\underset{\text{SE}}{\searrow}$ **D20** $\underset{\text{SE}}{\searrow}$ **D21** $\underset{\text{SW}}{\swarrow}$ **D22**

SegD$_{spike}$✎ = **D55** $\underset{\text{SW}}{\swarrow}$ **D56** $\underset{\text{SE}}{\searrow}$ **D57** $\overset{\text{NE}}{\nearrow}$ **D58** $\overset{\text{NE}}{\nearrow}$ **D59** $\overset{\text{NE}}{\nearrow}$ **D60** $\underset{\text{SE}}{\searrow}$ **D61** $\underset{\text{SW}}{\swarrow}$ **D62**

✎D1$_{2,0}$ = SegD$_0$✎ $\underset{\text{E}}{\to}$ SegD$_1$✎ $\overset{\text{NE}}{\nearrow}$ SegD$_2$✎ $\underset{\text{E}}{\to}$ SegD$_3$✎ $\overset{\text{NE}}{\nearrow}$ SegD$_0$✎ $\underset{\text{E}}{\to}$ SegD$_1$✎

✎D1$_{1,0}$ = SegD$_2$✎ $\underset{\text{E}}{\to}$ SegD$_3$✎ $\overset{\text{NE}}{\nearrow}$ SegD$_0$✎ $\underset{\text{E}}{\to}$ SegD$_1$✎ $\overset{\text{NE}}{\nearrow}$ SegD$_2$✎ $\underset{\text{E}}{\to}$ SegD$_3$✎

✎D1$_{0,0}$ = ✎D1$_{2,0}$ ⟪(SegD$_{init}$✎ $\underset{\text{SE}}{\searrow}$)@0..16⟫

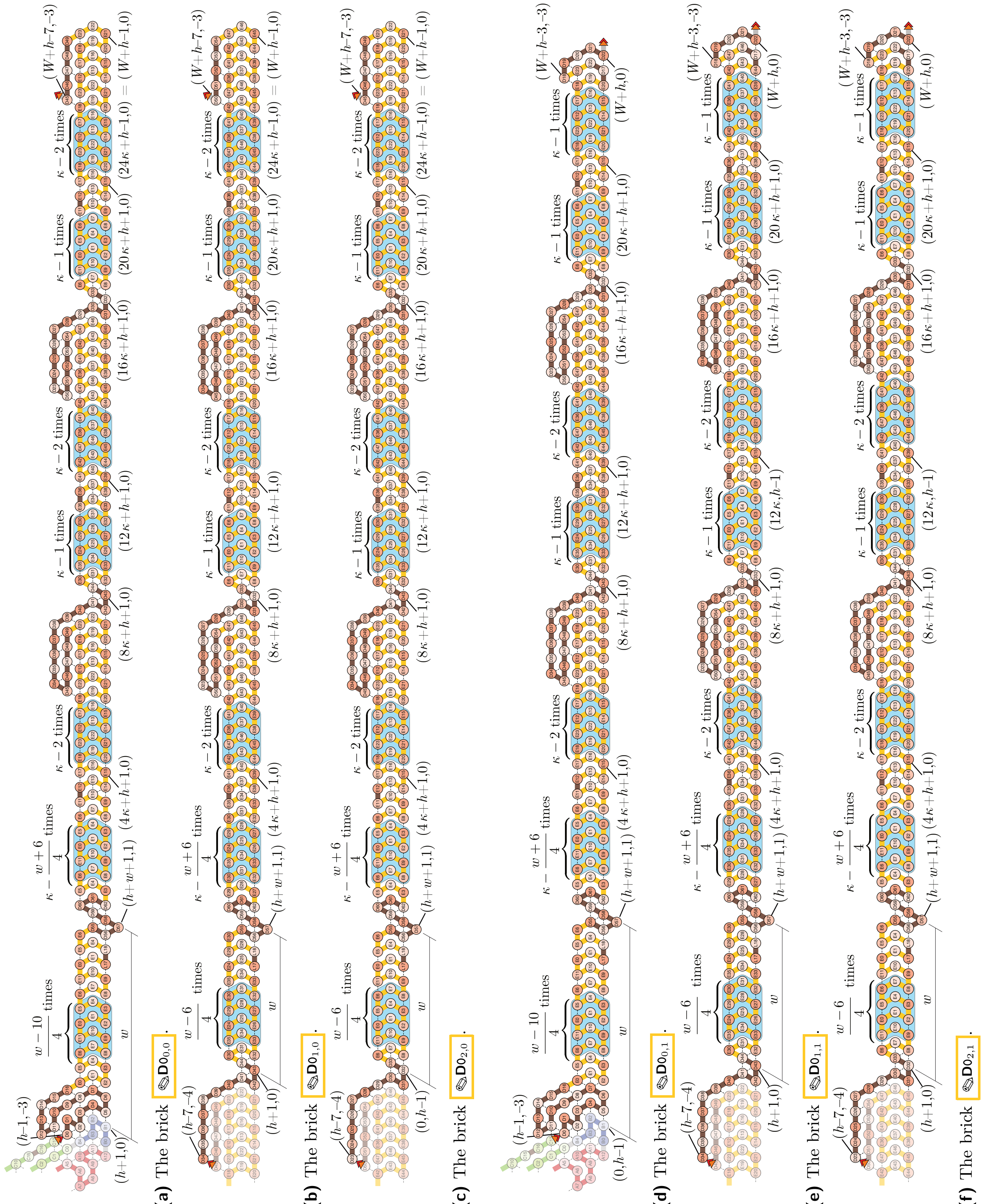
✎D1$_{r,1}$ = ✎D1$_{r,0}$ ⟪(**D17** $\overset{\text{NE}}{\nearrow}$)@$(3W+22)$, ($\overset{\text{NW}}{\nwarrow}$ SegD$_{end}$✎)@$(3W+25)..(3W+29)$⟫ for $r \in \{0,1,2\}$

✎D0$_{r,t}$ = ✎D1$_{r,t}$ ⟪(**L17** $\underset{\text{E}}{\to}$ **L18** $\overset{\text{NE}}{\nearrow}$)@$(3w+1..3w+2)$, (SegD$_{spike}$✎ $\underset{\text{E}}{\to}$)@$(3w+6)..(3w+13)$⟫ for $r \in \{0,1,2\}$ and $t \in \{0,1\}$

Figure 33 displays the bricks ✎D0$_{r,t}$.

F.7.3 Subrule for Module D

Module D interacts with A, B, C, E and G. Figure 34 presents the subrule for the interactions between the beads of D and the beads of the other modules.



(a) The brick ✎$\mathbf{D0}_{0,0}$.

(b) The brick ✎$\mathbf{D0}_{1,0}$.

(c) The brick ✎$\mathbf{D0}_{2,0}$.

(d) The brick ✎$\mathbf{D0}_{0,1}$.

(e) The brick ✎$\mathbf{D0}_{1,1}$.

(f) The brick ✎$\mathbf{D0}_{2,1}$.

Figure 33 Module $\mathbf{D0}_{r,t}$: Bricks $\mathbf{D}$✎.

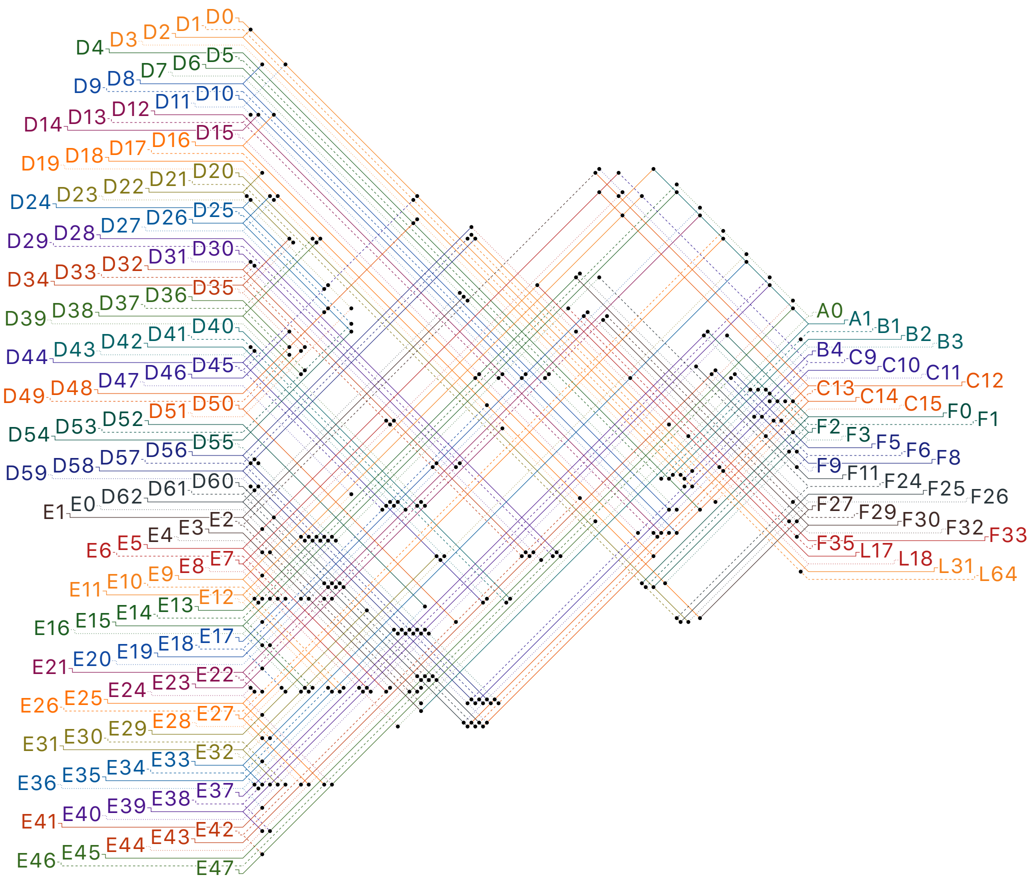


Figure 34 Subrule for Module D.

F.8 Modules E

F.8.1 Bead type sequences for Modules E

Length: $6\lambda(L-a+P)+8h-1$;
146 bead types used: F0..51, G0..48, H0..24, and I0..19.

$$\text{SegEA} = \left((\text{F0..11})^{\kappa} \cdot (\text{F12..23})^{\kappa} \cdot (\text{F24..35})^{\kappa} \cdot (\text{F36..47})^{\kappa}\right)^{\infty}$$

$$\text{SegEB} = \left((\text{G0..11})^{\kappa} \cdot (\text{G12..23})^{\kappa} \cdot (\text{G24..35})^{\kappa} \cdot (\text{G36..47})^{\kappa}\right)^{\infty}$$

$$\text{HeadE}_a = \text{SegEA}_{b..b+3c(a)-2} \cdot \text{F51} \cdot \text{SegEB}_{b+3c(a)..b+K-2}$$

where $b = 0$ if a is even, and $b = 2\lambda$ if a is odd

$$\text{SegEC} = \text{H0..4} \cdot (\text{H5..16})^{q-1} \cdot \text{H5..10} \cdot \text{H17..24}$$

$$\text{SegED}_0 = \text{I15} \cdot \text{I1..5} \cdot (\text{I0..5})^{k-1} \cdot \text{I0..1} \cdot \text{I18}$$

$$\text{SegED}_1 = \text{I19} \cdot \text{I7..8} \cdot (\text{I6..8})^{2k-1} \cdot \text{I6..7} \cdot \text{I15..16}$$

$$\text{SegED}_2 = \text{I17} \cdot \text{I10..11} \cdot (\text{I9..11})^{2k-1} \cdot \text{I9..10} \cdot \text{I19}$$

$$\text{SegED}_3 = \text{I18} \cdot \text{I13..14} \cdot (\text{I12..14})^{2k-1} \cdot \text{I12..13} \cdot \text{I19}$$

$$\text{SegED}_4 = \text{I19} \cdot \text{I1..2} \cdot (\text{I0..2})^{2k}$$

$$\text{TailE} = \text{SegEC} \cdot \text{SegED}_0 \cdot \text{SegED}_1 \cdot \text{SegED}_2 \cdot \text{SegED}_3 \cdot \text{SegED}_4$$

$$\text{E}_0 = \text{HeadE}_0 \langle\!\langle \text{F48..49}@0..1, \text{F50}@11 \rangle\!\rangle \cdot \text{G48} \cdot \text{TailE}$$

$$\text{E}_{a>0} = \text{HeadE}_a \cdot \text{G48} \cdot \text{TailE}$$

F.8.2 The bricks for the modules E

F.8.2.1 Zig-up, zig-down and zag bricks for Module E

$$\boxed{\textsf{SegEA}\blacktriangleright} = \Big(\text{NE-path}\big((\underline{\textbf{F0..11}})^{\kappa}\big) \underset{\texttt{E}}{\vec{\ }} \text{SW-path}\big((\underline{\textbf{F12..23}})^{\kappa}\big) \underset{\texttt{E}}{\vec{\ }} \text{NE-path}\big((\underline{\textbf{F24..35}})^{\kappa}\big) \underset{\texttt{E}}{\vec{\ }} \text{SW-path}\big((\underline{\textbf{F36..47}})^{\kappa}\big)\Big)^{\infty}$$

$$\boxed{\textsf{SegEB}\blacktriangleright} = \Big(\text{NE-path}\big(\underline{\textbf{G0..11}})^{\kappa}\big) \underset{\texttt{E}}{\vec{\ }} \text{SW-path}\big((\underline{\textbf{G12..23}})^{\kappa}\big) \underset{\texttt{E}}{\vec{\ }} \text{NE-path}\big((\underline{\textbf{G24..35}})^{\kappa}\big) \underset{\texttt{E}}{\vec{\ }} \text{SW-path}\big((\underline{\textbf{G36..47}})^{\kappa}\big)\Big)^{\infty}$$

$$\boxed{\textsf{HeadE}_a\blacktriangleright} = \boxed{\textsf{SegEA}\blacktriangleright}_{b..b+3c(a)-2}\ d\ \underline{\textbf{F51}}\ d\ \boxed{\textsf{SegEB}\blacktriangleright}_{b+3c(a)..b+K-2}$$

where $b = 0$ if a is even, and $b = 2\lambda$ if a is odd; and $d = \overset{\texttt{NE}}{\nearrow}$ if $\lfloor \frac{3c-2}{\lambda} \rfloor$ is even, else $d = \underset{\texttt{SW}}{\swarrow}$

$$\boxed{\textsf{SegEC}\blacktriangleright} = \underline{\textbf{H0}} \underset{\texttt{E}}{\vec{\ }} \underline{\textbf{H1}} \overset{\texttt{NW}}{\nwarrow} \underline{\textbf{H2}} \underset{\texttt{E}}{\vec{\ }} \underline{\textbf{H3}} \underset{\texttt{SE}}{\searrow} \underline{\textbf{H4}} \overset{\texttt{NE}}{\nearrow} \text{NE-glider}\big((\underline{\textbf{H5..16}})^{q-1} \cdot \underline{\textbf{H5..10}} \cdot \underline{\textbf{H17..22}}\big) \overset{\texttt{NE}}{\nearrow} \underline{\textbf{H23}} \overset{\texttt{NW}}{\nwarrow} \underline{\textbf{H24}}$$

$$\boxed{\textsf{SegED}_0\blacktriangleright} = \underline{\textbf{I15}} \underset{\texttt{SE}}{\searrow} \text{SW-path}\big(\underline{\textbf{I1..5}} \cdot (\underline{\textbf{I0..5}})^{k-1} \cdot \underline{\textbf{I0..1}} \cdot \underline{\textbf{I18}}\big)$$

$$\boxed{\textsf{SegED}_1\blacktriangleright} = \text{NE-path}\big(\underline{\textbf{I19}} \cdot \underline{\textbf{I7..8}} \cdot (\underline{\textbf{I6..8}})^{2k-1} \cdot \underline{\textbf{I6..7}}\big) \overset{\texttt{NW}}{\nwarrow} \underline{\textbf{I15}} \underset{\texttt{E}}{\vec{\ }} \underline{\textbf{I16}}$$

$$\boxed{\textsf{SegED}_2\blacktriangleright} = \text{SW-path}\big(\underline{\textbf{I17}} \cdot \underline{\textbf{I10..11}} \cdot (\underline{\textbf{I9..11}})^{2k-1} \cdot \underline{\textbf{I9..10}} \cdot \underline{\textbf{I19}}\big)$$

$$\boxed{\textsf{SegED}_3\blacktriangleright} = \text{NE-path}\big(\underline{\textbf{I18}} \cdot \underline{\textbf{I13..14}} \cdot (\underline{\textbf{I12..14}})^{2k-1} \cdot \underline{\textbf{I12..13}} \cdot \underline{\textbf{I19}}\big)$$

$$\boxed{\textsf{SegED}_4\blacktriangleright} = \text{SW-path}\big(\underline{\textbf{I19}} \cdot \underline{\textbf{I1..2}} \cdot (\underline{\textbf{I0..2}})^{2k}\big)$$

$$\boxed{\textsf{TailE}\blacktriangleright} = \boxed{\textsf{SegEC}\blacktriangleright} \underset{\texttt{E}}{\vec{\ }} \boxed{\textsf{SegED}_0\blacktriangleright} \underset{\texttt{E}}{\vec{\ }} \boxed{\textsf{SegED}_1\blacktriangleright} \underset{\texttt{E}}{\vec{\ }} \boxed{\textsf{SegED}_2\blacktriangleright} \underset{\texttt{E}}{\vec{\ }} \boxed{\textsf{SegED}_3\blacktriangleright} \underset{\texttt{E}}{\vec{\ }} \boxed{\textsf{SegED}_4\blacktriangleright}$$

$$\boxed{\textsf{E}_0\blacktriangleright} = \boxed{\textsf{HeadE}_0\blacktriangleright} \langle\!\langle (\underline{\textbf{F48}} \overset{\texttt{NE}}{\nearrow} \underline{\textbf{F49}} \overset{\texttt{NE}}{\nearrow})@0..1, (\underline{\textbf{F50}} \overset{\texttt{NE}}{\nearrow})@11 \rangle\!\rangle \underset{\texttt{SW}}{\swarrow} \underline{\textbf{G48}} \underset{\texttt{E}}{\vec{\ }} \boxed{\textsf{TailE}\blacktriangleright}$$

$$\boxed{\textsf{E}_{a>0}\blacktriangleright} = \boxed{\textsf{HeadE}_a\blacktriangleright} \underset{\texttt{SW}}{\swarrow} \underline{\textbf{G48}} \underset{\texttt{E}}{\vec{\ }} \boxed{\textsf{TailE}\blacktriangleright}$$

$$\boxed{\textsf{E}_{a\geqslant 0}\blacktriangleright} = \text{HorizontalMirror}\big(\boxed{\textsf{E}_a\blacktriangleright}\big)$$

$$\boxed{\textsf{E}_{a\geqslant 0}\blacktriangleleft} = \text{Rotate}_{180^\circ}\big(\boxed{\textsf{E}_a\blacktriangleright}\big)$$

Figure 35 displays the bricks $\boxed{\textsf{E}_a\blacktriangleright}$.

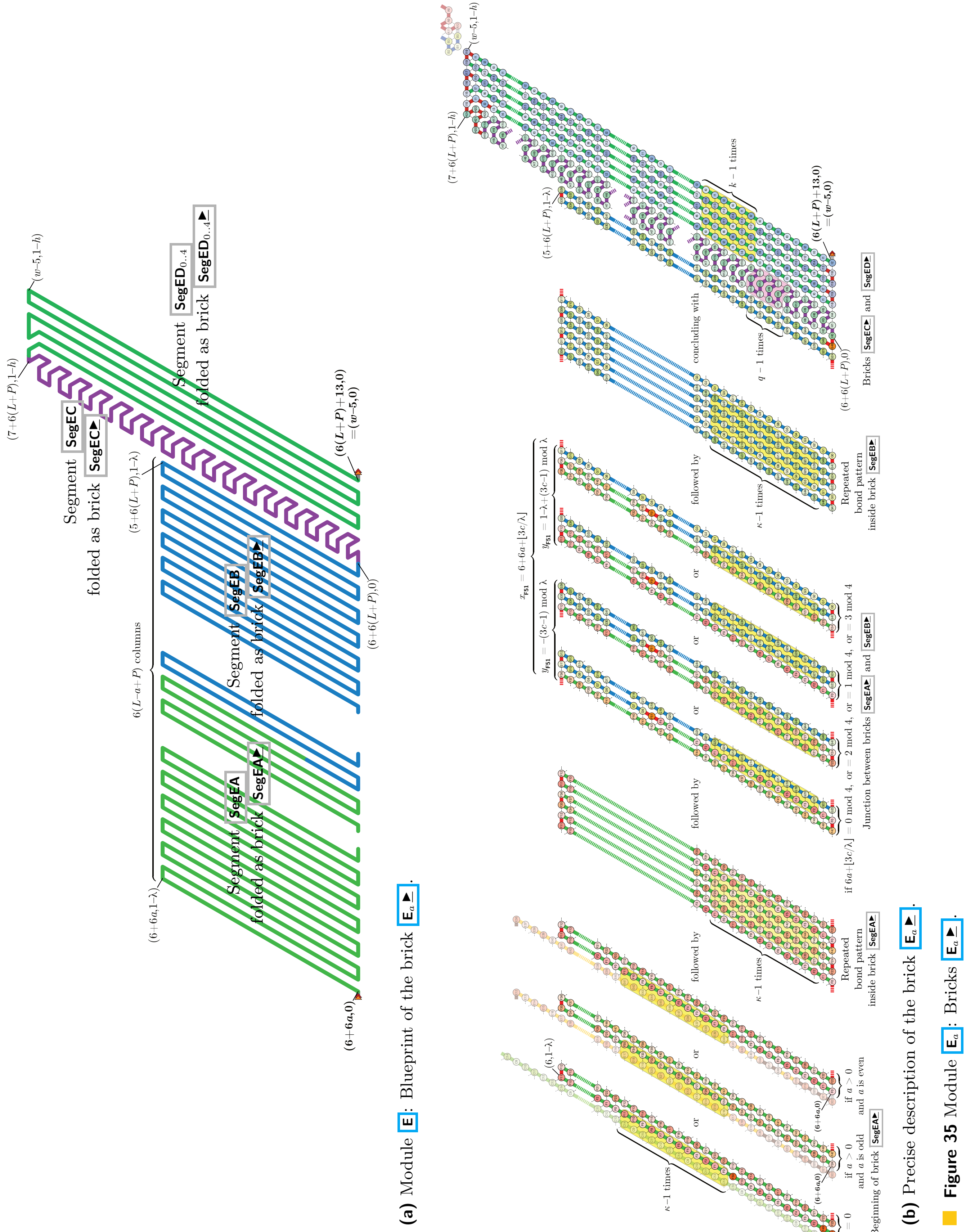


Figure 35 Module E_a: Bricks E_a▶.

F.8.2.2 Carriage-return bricks for Module E

SegEA◣ = E-glider(SegEA)

SegEB◣ = W-path(SegEB)

HeadE$_a$◣ = SegEA◣$_{b..b+3c(a)-2}$ $\overset{\swarrow}{\text{SW}}$ **F51** $\overset{\searrow}{\text{SE}}$ SegEB◣$_{b+3c(a)..b+K-2}$ where $b = 0$ if a is even, and $b = 2\lambda$ if a is odd;

SegEC◣ = W-path(**H0..4** · (**H5..16**)$^{q-1}$ · **H5..10** · **H17..19**) $\overset{\searrow}{\text{SE}}$ **H20** $\overset{\rightarrow}{\text{E}}$ **H21** $\overset{\rightarrow}{\text{E}}$ **H22** $\overset{\swarrow}{\text{SW}}$ **H23** $\overset{\leftarrow}{\text{W}}$ **H24**

SegED$_0$◣ = **I15** $\overset{\text{NW}}{\nwarrow}$ **I1** $\overset{\text{NW}}{\nwarrow}$ W-path(**I2..5** · (**I0..5**)$^{k-1}$ · **I0..1**) $\overset{\searrow}{\text{SE}}$ **I18**

SegED$_1$◣ = **I19** $\overset{\text{NW}}{\nwarrow}$ W-path(**I7..8** · (**I6..8**)$^{2k-1}$ · **I6..7**) $\overset{\searrow}{\text{SE}}$ **I15** $\overset{\leftarrow}{\text{W}}$ **I16**

SegED$_2$◣ = **I17** $\overset{\text{NE}}{\nearrow}$ W-path(**I10..11** · (**I9..11**)$^{2k-1}$ · **I9..10**) $\overset{\searrow}{\text{SE}}$ **I19**

SegED$_3$◣ = **I18** $\overset{\text{NW}}{\nwarrow}$ W-path(**I13..14** · (**I12..14**)$^{2k-1}$ · **I12..13**) $\overset{\swarrow}{\text{SW}}$ **I19**

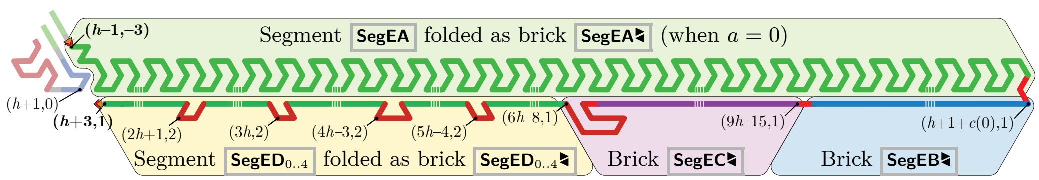
SegED$_4$◣ = **I19** $\overset{\text{NE}}{\nearrow}$ W-path(**I1..2** · (**I0..2**)^{2k})

TailE◣ = SegEC◣ $\overset{\leftarrow}{\text{W}}$ SegED$_0$◣ $\overset{\leftarrow}{\text{W}}$ SegED$_1$◣ $\overset{\leftarrow}{\text{W}}$ SegED$_2$◣ $\overset{\leftarrow}{\text{W}}$ SegED$_3$◣ $\overset{\leftarrow}{\text{W}}$ SegED$_4$◣

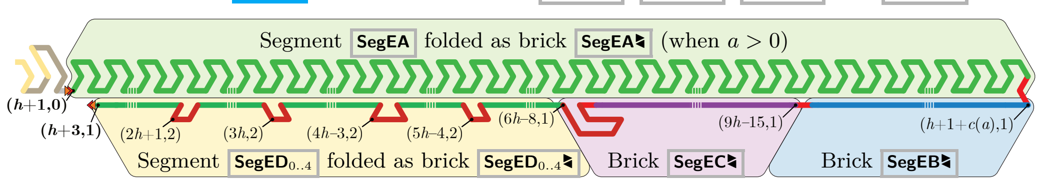
E$_0$◣ = HeadE$_0$◣⟪(**F48** $\overset{\rightarrow}{\text{E}}$ **F49** $\overset{\swarrow}{\text{SW}}$)@0..1, (**F50** $\overset{\rightarrow}{\text{E}}$)@11⟫ $\overset{\leftarrow}{\text{W}}$ **G48** $\overset{\leftarrow}{\text{W}}$ TailE◣

E$_{a>0}$◣ = HeadE$_a$◣ $\overset{\leftarrow}{\text{W}}$ **G48** $\overset{\leftarrow}{\text{W}}$ TailE◣

Figures 36 and 37 display the blueprints and the description of bricks E$_a$◣ respectively.



(a) Blueprint of brick E$_{a=0}$◣, made of four bricks SegEA◣, SegEB◣, SegEC◣ and SegED◣.



(b) Blueprint of brick E$_{a>0}$◣, made of four bricks SegEA◣, SegEB◣, SegEC◣ and SegED◣.

Figure 36 Outline of the four different parts of module E$_a$, when folded at the end of the appended appendant. See Figure 37 for the detailed beads of each part.

F.8.3 Subrule for Module E

Module E interacts with B, C, D and F. Figure 38 presents the subrule for the interactions between the beads of E and the beads of the other modules.

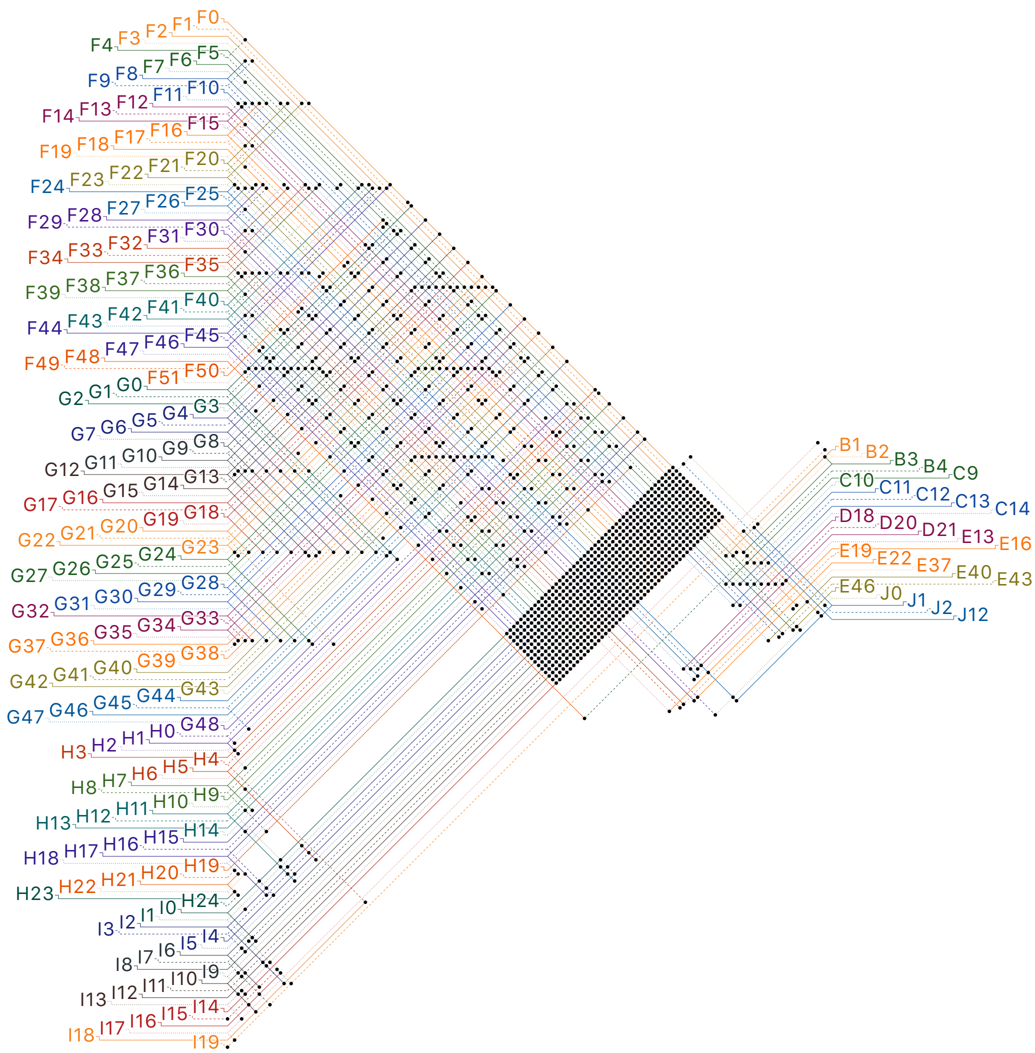
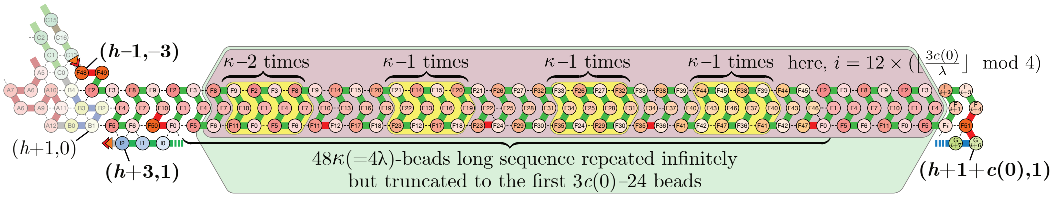
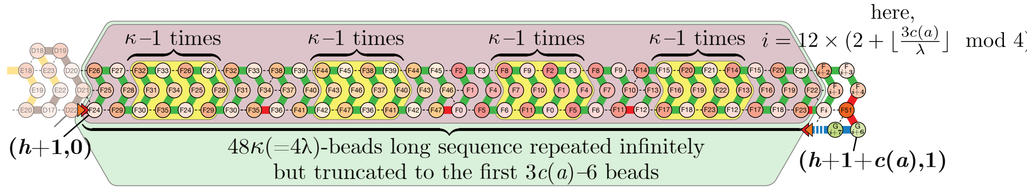


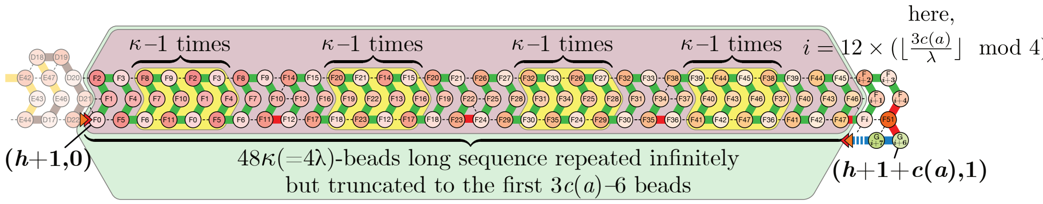
Figure 38 Subrule for Module **E**.



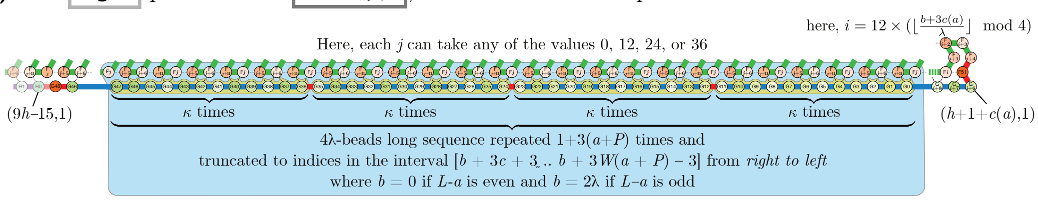
(a) The SegEA-part of brick $\text{HeadE}_{a=0}$, when $a = 0$.



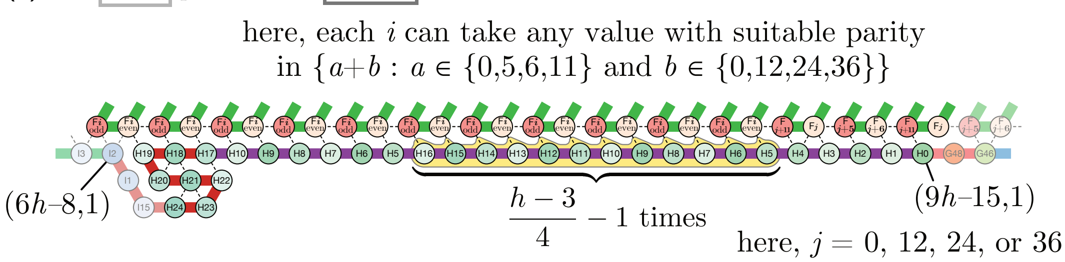
(b) The SegEA-part of brick HeadE_{a}, when a is odd.



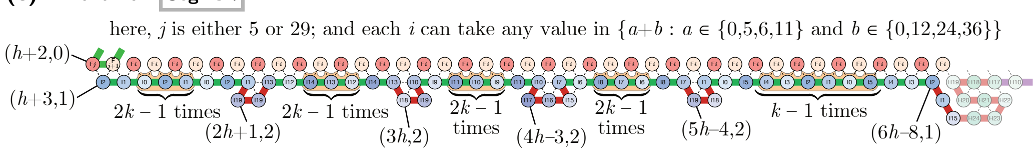
(c) The SegEA-part of brick $\text{HeadE}_{a>0}$, when a is even and positive.



(d) The SegEB-part of brick HeadE_{a}.



(e) The brick SegEC.



(f) The brick $\text{SegED}_{0..4}$.

Figure 37 The bricks E_a.

F.9 Module F

F.9.1 Bead type sequence for Module F

Length: $4h$; 53 bead types used: J0..52.

$$\mathsf{HeadF} = \mathsf{J0..4} \cdot (\mathsf{J5..10})^{3k-1} \cdot \mathsf{J5..7} \cdot \mathsf{J11..23}$$

$$\mathsf{TailF} = \mathsf{J48} \cdot (\mathsf{J51..48})^{9} \cdot \mathsf{J51} \cdot \mathsf{J52} \cdot \mathsf{J49..48}$$

$$\mathsf{SegExp}(2i) = \mathsf{J24..29} \cdot (\mathsf{J30..35})^{3^{2i-1}-1} \qquad \text{for } i \geqslant 1$$

$$\mathsf{SegExp}(2i+1) = \mathsf{J36..41} \cdot (\mathsf{J42..47})^{3^{2i}-1} \qquad \text{for } i \geqslant 1$$

$$\mathsf{SegExpF} = \bigodot_{i \geqslant 2} \mathsf{SegExp}(i)$$

$$\mathsf{F} = \mathsf{HeadF} \cdot \left(\mathsf{J39..41} \cdot \mathsf{SegExpF}_{0..(h-51)}\right)^{R} \cdot \mathsf{TailF}$$

F.9.2 The bricks for the modules F

Module F adopts three brick conformations: zig-up F▶, zig-down F▶, and zag F◀. the two last are obtained by mirroring and rotating the first.

$$\mathsf{HeadF}\blacktriangleright = \mathsf{J0} \overset{\rightarrow}{\scriptstyle E} \mathsf{J1} \overset{\text{NW}}{\nwarrow} \mathsf{J2} \overset{\rightarrow}{\scriptstyle E} \mathsf{J3} \underset{\text{SE}}{\searrow} \mathsf{J4} \overset{\text{NE}}{\nearrow} \text{NE-glider}\left((\mathsf{J5..10})^{3k-1}\right) \overset{\text{NE}}{\nearrow} \mathsf{J5} \overset{\text{NW}}{\nwarrow} \mathsf{J6} \overset{\leftarrow}{\scriptstyle W} \mathsf{J7} \overset{\text{NE}}{\nearrow}$$
$$\mathsf{J11} \overset{\text{NE}}{\nearrow} \mathsf{J12} \underset{\text{SE}}{\searrow} \mathsf{J13} \underset{\text{SE}}{\searrow} \mathsf{J14} \overset{\text{NE}}{\nearrow} \mathsf{J15} \overset{\text{NW}}{\nwarrow} \mathsf{J16} \overset{\rightarrow}{\scriptstyle E} \mathsf{J17} \overset{\rightarrow}{\scriptstyle E} \text{SW-path}(\mathsf{J18..23})$$

$$\mathsf{TailF}\blacktriangleright = \text{SW-path}(\mathsf{J48} \cdot (\mathsf{J51..48})^{9} \cdot \mathsf{J51} \cdot \mathsf{J52} \cdot \mathsf{J49..48})$$

Recall that $\mathsf{SegExpF} = \bigodot_{i \geqslant 2} \mathsf{SegExp}(i)$

$$\mathsf{F}\blacktriangleright = \mathsf{HeadF}\blacktriangleright \underset{\text{SW}}{\swarrow} \text{SW-path}\left(\mathsf{J39..41} \cdot \mathsf{SegExpF}_{0..(h-51)}\right)^{R} \underset{\text{SW}}{\swarrow} \mathsf{TailF}\blacktriangleright$$

$$\mathsf{F}\blacktriangleright = \text{HorizontalMirror}(\mathsf{F}\blacktriangleright)$$

$$\mathsf{F}\blacktriangleleft = \text{Rotate}_{180^\circ}(\mathsf{F}\blacktriangleright)$$

Figure 39 displays the brick F▶.

F.9.3 Subrule for Module F

Module F interacts with A, B, C, E and G. Figure 40 presents the subrule for the interactions between the beads of F and the beads of the other modules.

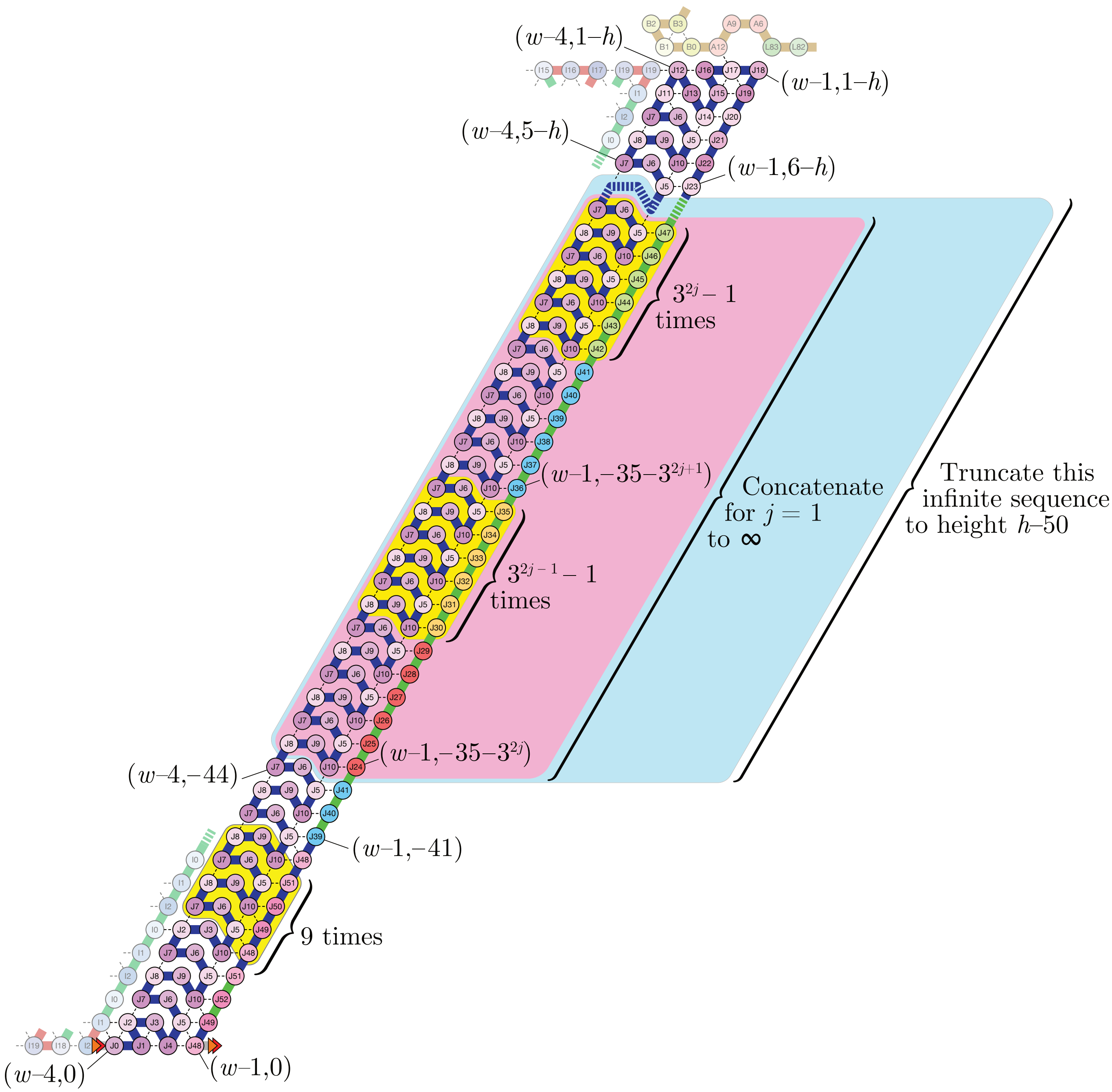


Figure 39 Module F: Brick F▶.

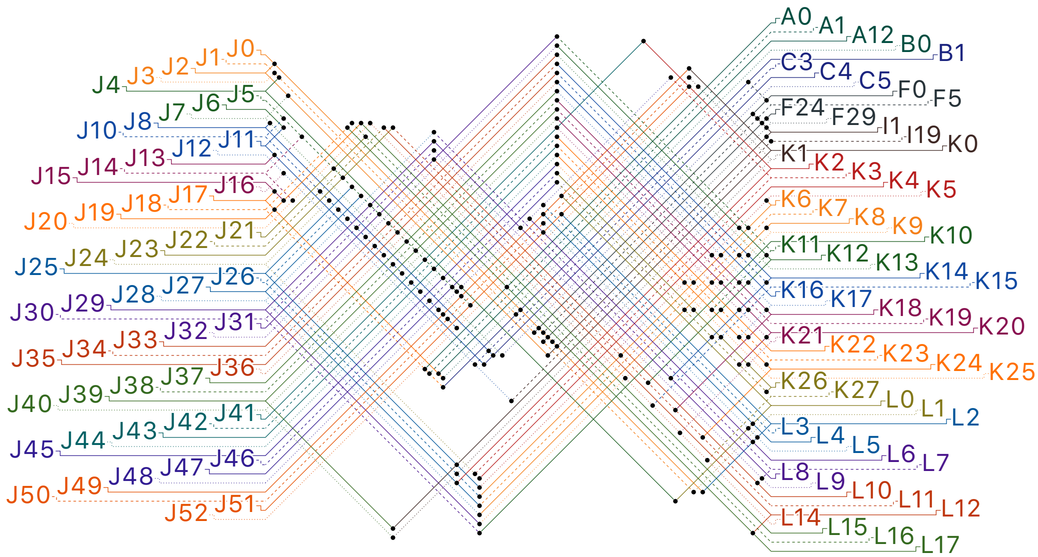


Figure 40 Subrule for Module F.

F.10 Module G

F.10.1 Bead type sequence for Module G

Length: $6h-1$; 201 bead types used: K0..69, L0..99, and M0..30.

SegExp'$(2i)$ = K4..9 · (K10..15)$^{3^{2i-1}-1}$ for $i \geqslant 1$

SegExp'$(2i+1)$ = K16..21 · (K22..27)$^{3^{2i}-1}$ for $i \geqslant 1$

SegExpG = $\bigodot_{i \geqslant 2}$ SegExp'(i)

SegG1 = L0..6 · K3 · (K0..3)^9 · K0..2 · L7..10 · SegExpG$_{8..h-51}$

SegG2 = K32 · K33 · (K28..33)$^{k-3}$ · K28..32

SegG3 = K35..39 · (K34..39)$^{k-14}$ · K34 · K35 · L39..41 · K45 · (K40..45)^{10} · K40 · K41

SegG4 = K50..51 · (K46..51)$^{k-3}$ · K46..48

SegG5 = K55..57 · (K52..57)$^{k-6}$ · K52..53 · L74 · L75 · K56..57 · (K52..57)^2 · K52..53

SegG6 = K63 · (K58..63)$^{k-19}$ · K58..61 · L91..99 · M0..19 · K67..69 · (K64..69)^{10} · M20..30

G = SegG1 · L11..24 · SegG2 · L25..38 · SegG3 · L42..55 · SegG4 · L56..73 · SegG5 · L76..90 · SegG6

F.10.2 The bricks for the modules G

F.10.2.1 Zig-up bricks for Module G

G adopts two different bricks in zig-up phase, depending on the letter encoded in the zag-phase above: G▶Read0 and G▶Read1.

The G▶Read0 brick.

SegG1Start▶Read = L0 $\overset{\leftarrow}{\text{W}}$ L1 $\overset{\leftarrow}{\text{W}}$ L2 $\overset{\searrow}{\text{SE}}$ L3 $\overset{\rightarrow}{\text{E}}$ L4 $\overset{\rightarrow}{\text{E}}$ L5 $\overset{\text{NE}}{\nearrow}$ L6 $\overset{\text{NE}}{\nearrow}$ K3

SegG1GliderNE▶Read = NE-path$\big($(K0..3)^{9} · K0..2$\big)$ $\overset{\rightarrow}{\text{E}}$ L7 $\overset{\text{NW}}{\nwarrow}$ L8 $\overset{\rightarrow}{\text{E}}$ L9 $\overset{\searrow}{\text{SE}}$ L10 $\overset{\text{NE}}{\nearrow}$
NE-glider$\big($SegExpG$_{8..h-51}$ · L11..23$\big)$

SegG2GliderNE▶Read = NE-glider(L24 · K32..33 · (K28..33)$^{k-3}$ · K28..32 · L25..34)

SegG3GliderNE▶Read = NE-glider(L35..38 · K35..39 · (K34..39)$^{k-14}$ · K34 · K35 · L39)

SegG1..3GliderNE▶Read = SegG1GliderNE▶Read $\overset{\text{NE}}{\nearrow}$ SegG2GliderNE▶Read $\overset{\text{NE}}{\nearrow}$ SegG3GliderNE▶Read

SegG3GliderSE▶Read0 = SE-rev-glider(L40..41 · K45 · (K40..45)^{10} · K40..41 · L42)

SegG3SockNW▶Read0 = NW-path(L43..48) $\overset{\swarrow}{\text{SW}}$ SE-path(L49..54) $\overset{\rightarrow}{\text{E}}$ L55

SegG4GliderSE▶Read0 = SE-rev-glider(K50..51 · (K46..51)$^{k-3}$ · K46..48 · L56)

SegG4SockNW▶Read0 = NW-path(L57..64) $\overset{\swarrow}{\text{SW}}$ SE-path(L65..72) $\overset{\rightarrow}{\text{E}}$ L73

SegG5GliderSE▶Read0 = SE-rev-glider$\big($K55..57 · (K52..57)$^{k-6}$ · K52..53 · L74..75 · K56..57 ·
(K52..57)^{2} · K52..53 · L76$\big)$

SegG5SockNW▶Read0 = NW-path(L77..82) $\overset{\leftarrow}{\text{W}}$ SE-path(L83..89) $\overset{\rightarrow}{\text{E}}$ L90

SegG6AGliderSE▶Read0 = SE-rev-glider(K63 · (K58..63)$^{k-19}$ · K58..61 · L91)

SegG6SockNW▶Read0 = NW-path(L92..99 · M0..4) $\overset{\leftarrow}{\text{W}}$ SE-path(M5..18) $\overset{\rightarrow}{\text{E}}$ M19

SegG6BGlider▶Read0 = SE-rev-glider(K67..69 · (K64..69)^{10} · M20..22) $\overset{\searrow}{\text{SE}}$ M23

SegG6End▶Read0 = M24 $\overset{\rightarrow}{\text{E}}$ M25 $\overset{\rightarrow}{\text{E}}$ M26 $\overset{\swarrow}{\text{SW}}$ M27 $\overset{\leftarrow}{\text{W}}$ M28 $\overset{\swarrow}{\text{SW}}$ M29 $\overset{\rightarrow}{\text{E}}$ M30

G▶Read0 = SegG1Start▶Read $\overset{\text{NE}}{\nearrow}$ SegG1..3GliderNE▶Read $\overset{\rightarrow}{\text{E}}$ SegG3GliderSE▶Read0 $\overset{\leftarrow}{\text{W}}$
SegG3SockNW▶Read0 $\overset{\rightarrow}{\text{E}}$ SegG4GliderSE▶Read0 $\overset{\leftarrow}{\text{W}}$ SegG4SockNW▶Read0 $\overset{\rightarrow}{\text{E}}$
SegG5GliderSE▶Read0 $\overset{\leftarrow}{\text{W}}$ SegG5SockNW▶Read0 $\overset{\rightarrow}{\text{E}}$ SegG6AGliderSE▶Read0 $\overset{\leftarrow}{\text{W}}$
SegG6SockNW▶Read0 $\overset{\rightarrow}{\text{E}}$ SegG6BGlider▶Read0 $\overset{\text{NE}}{\nearrow}$ SegG6End▶Read0

Figure 41 displays the brick G▶Read0.

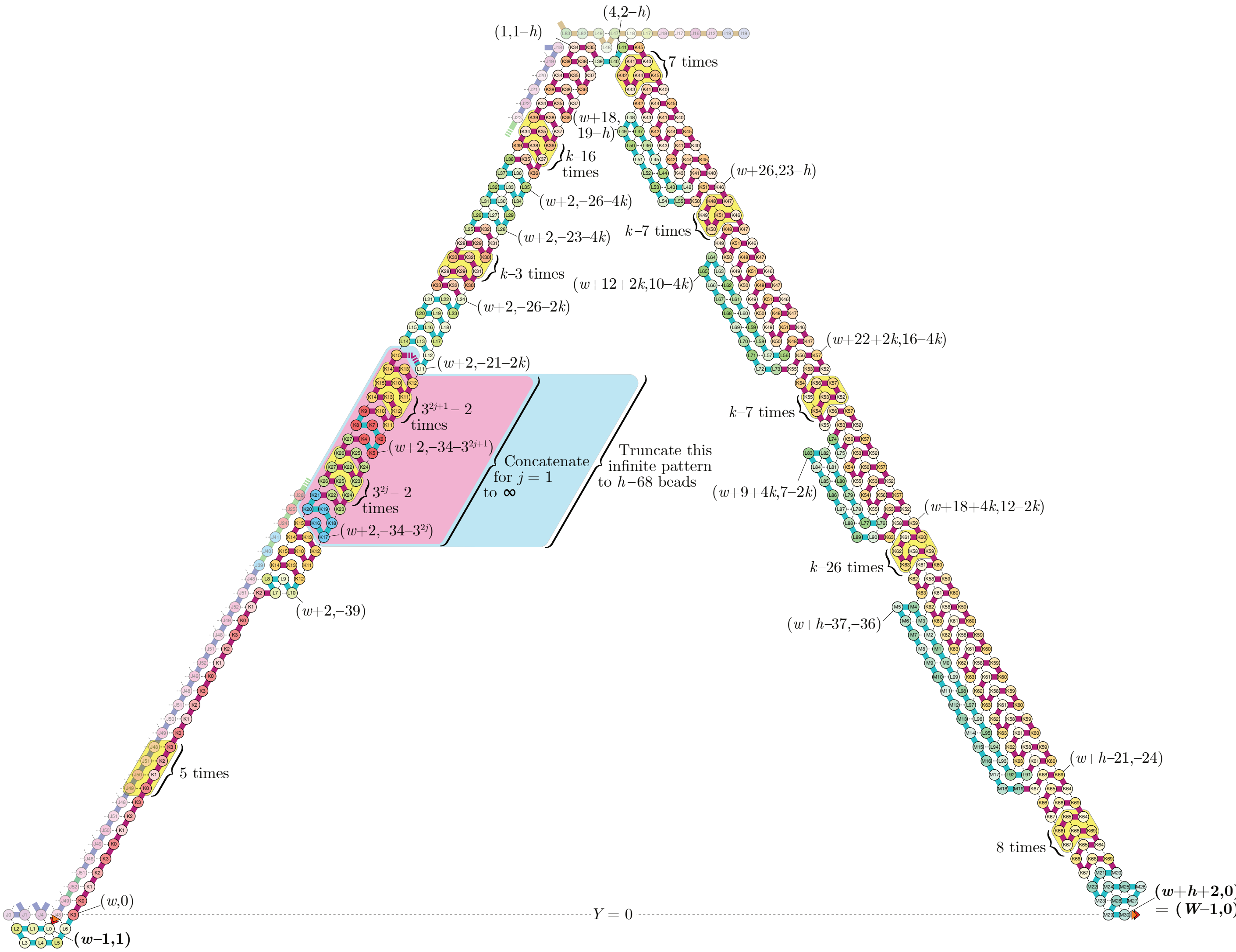


Figure 41 Module G: Brick G ▶ Read0.

The G ▶ Read1 brick.

SegG3GliderE▶Read1 = L41 $\overset{\text{NW}}{\nwarrow}$ K45 $\overset{\to}{\text{E}}$ E-glider′((K40..45)^{10}) $\overset{\to}{\text{E}}$ K40 $\underset{\text{SE}}{\searrow}$ K41 $\underset{\text{SW}}{\swarrow}$ L42

SegG3SockW▶Read1 = W-path(L43..48) $\underset{\text{SE}}{\searrow}$ E-path(L49..54) $\overset{\text{NE}}{\nearrow}$ L55

SegG4GliderE▶Read1 = E-glider(K50..51 · (K46..51)$^{k-3}$ · K46..48 · L56)

SegG4SockW▶Read1 = W-path(L57..64) $\underset{\text{SE}}{\searrow}$ E-path(L65..72) $\overset{\text{NE}}{\nearrow}$ L73

SegG5GliderE▶Read1 = E-glider(K55..57 · (K52..57)$^{k-6}$ · K52..53 · L74..75 · K56..57 · (K52..57)^{2} · K52..53 · L76)

SegG5SockW▶Read1 = W-path(L77..82) $\underset{\text{SW}}{\swarrow}$ E-path(L83..89) $\overset{\text{NE}}{\nearrow}$ L90

SegG6AGliderE▶Read1 = E-glider(K63 · (K58..63)$^{k-19}$ · K58..61 · L91)

SegG6SockW▶Read1 = W-path(L92..99 · M0..4) $\underset{\text{SW}}{\swarrow}$ E-path(M5..18) $\overset{\text{NE}}{\nearrow}$ M19

SegG6BGliderE▶Read1 = E-glider(K67..69 · (K64..69)^{9} · K64..66) $\overset{\to}{\text{E}}$ K67 $\overset{\text{NE}}{\nearrow}$ K68 $\overset{\text{NW}}{\nwarrow}$ K69

SegG3..6GliderE▶Read1 = SegG3GliderE▶Read1 $\underset{\text{SW}}{\swarrow}$ SegG3SockW▶Read1 $\overset{\text{NE}}{\nearrow}$ SegG4GliderE▶Read1 $\underset{\text{SW}}{\swarrow}$ SegG4SockW▶Read1 $\overset{\text{NE}}{\nearrow}$ SegG5GliderE▶Read1 $\underset{\text{SW}}{\swarrow}$ SegG5SockW▶Read1 $\overset{\text{NE}}{\nearrow}$ SegG6AGliderE▶Read1 $\underset{\text{SW}}{\swarrow}$ SegG6SockW▶Read1 $\overset{\text{NE}}{\nearrow}$ SegG6BGliderE▶Read1

SegG6End▶Read1 = E-path(M20..22) $\underset{\text{SE}}{\searrow}$ W-path(M23..25) $\underset{\text{SW}}{\swarrow}$ E-path(M26..29) $\overset{\text{NE}}{\nearrow}$ M30

G ▶ Read1 = SegG1Start▶Read $\overset{\text{NE}}{\nearrow}$ SegG1..3GliderNE▶Read $\underset{\text{SE}}{\searrow}$ L40 $\overset{\text{NE}}{\nearrow}$ SegG3..6GliderE▶Read1 $\overset{\to}{\text{E}}$ SegG6End▶Read1

Figure 42 displays the brick G ▶ Read1.

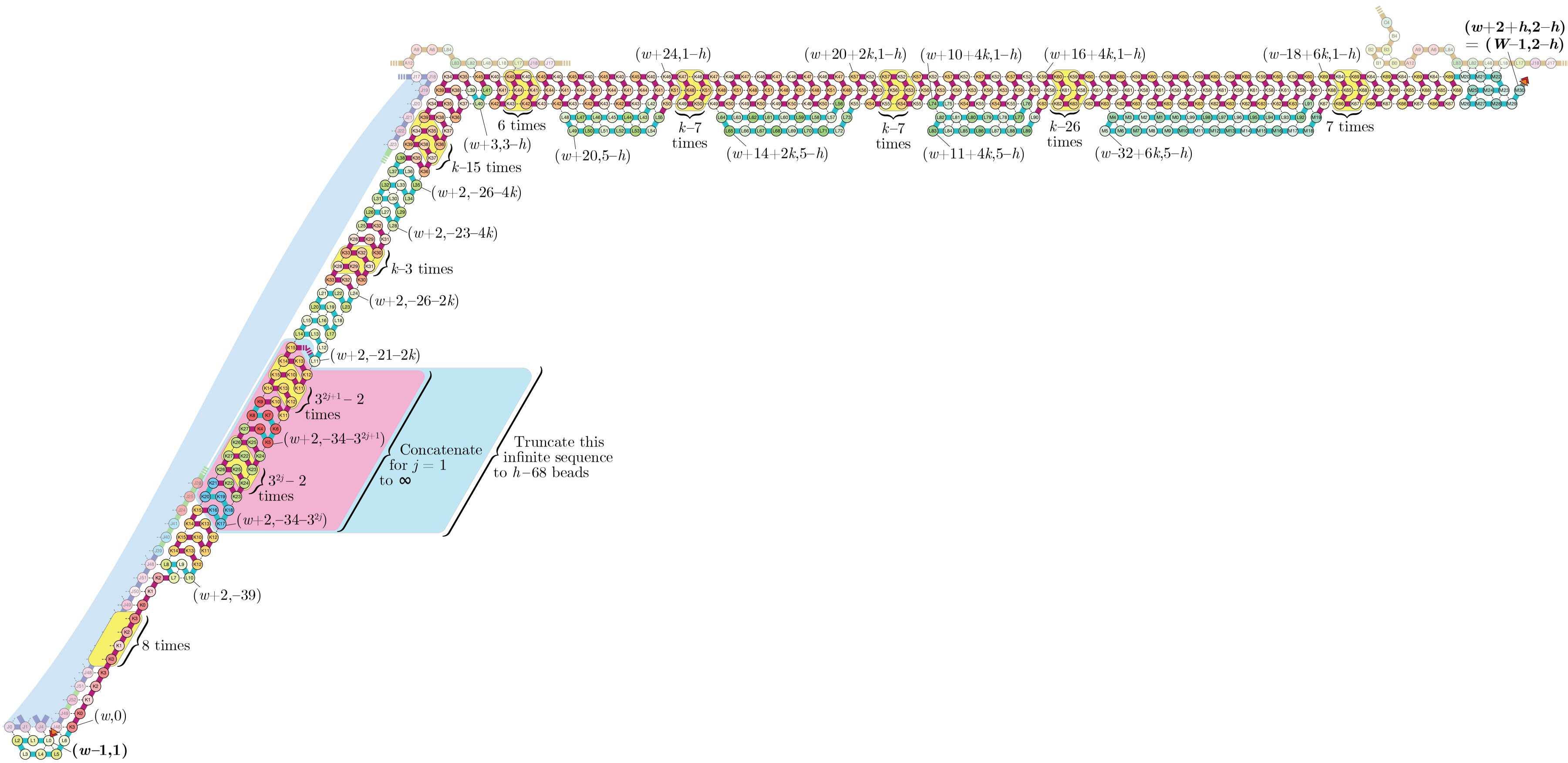


Figure 42 Module G: Brick G▶Read1.

F.10.2.2 Zig-down and Zag bricks for Module G: Copy letter 0 and 1.

[SegG1▶Copy] = SE-path(L0..6 · K3 · (K0..3)^{9} · K0..2 · L7..10 · [SegExpG]$_{8..h-51}$ · L11..17)
[SegG2▶Copy] = NW-path(L18..24 · K32..33 · (K28..33)$^{k-3}$ · K28..32 · L25..30)
[SegG3▶Copy] = SE-path(L32..38 · K35..39 · (K34..39)$^{k-14}$ · K34..35 · L39..41 · K45· (K40..45)^{10} · K40..41 · L42..48)
[SegG4▶Copy] = NW-path(L49..55 · K50..51 · (K46..51)$^{k-3}$ · K46..48 · L56..63)
[SegG5▶Copy] = SE-path(L66..73 · K55..57 · (K52..57)$^{k-6}$ · K52..53 · L74..75 · K56..57· (K52..57)^{2} · K52..53 · L76..81)
[SegG6▶Copy] = NW-path(L84..90 · K63 · (K58..63)$^{k-19}$ · K58..61 · L91..99 · M0..19 · K67..69 · (K64..69)^{10} · M20..30)

[G ▶ Copy0] = [SegG1▶Copy] $\overset{\rightarrow}{\texttt{E}}$ [SegG2▶Copy] $\overset{\texttt{NE}}{\nearrow}$ L31 $\underset{\texttt{SE}}{\searrow}$ [SegG3▶Copy] $\overset{\texttt{NE}}{\nearrow}$ [SegG4▶Copy] $\overset{\texttt{NE}}{\nearrow}$ L64 $\overset{\rightarrow}{\texttt{E}}$ L65 $\underset{\texttt{SW}}{\swarrow}$ [SegG5▶Copy] $\underset{\texttt{SE}}{\searrow}$ L82 $\overset{\rightarrow}{\texttt{E}}$ L83 $\overset{\texttt{NW}}{\nwarrow}$ [SegG6▶Copy]

[G ▶ Copy1] = [SegG1▶Copy] $\overset{\rightarrow}{\texttt{E}}$ [SegG2▶Copy] $\overset{\texttt{NW}}{\nwarrow}$ L31 $\overset{\rightarrow}{\texttt{E}}$ [SegG3▶Copy] $\overset{\texttt{NE}}{\nearrow}$ [SegG4▶Copy] $\overset{\rightarrow}{\texttt{E}}$ L64 $\overset{\rightarrow}{\texttt{E}}$ L65 $\underset{\texttt{SW}}{\swarrow}$ [SegG5▶Copy] $\underset{\texttt{SW}}{\swarrow}$ L82 $\overset{\rightarrow}{\texttt{E}}$ L83 $\overset{\texttt{NE}}{\nearrow}$ [SegG6▶Copy]

[G ◀ Copy0] = VerticalMirror([G ▶ Copy0])
[G ◀ Copy1] = VerticalMirror([G ▶ Copy1])

Figures 43 and 44 display the bricks [G ▶ Copy0] and [G ▶ Copy1].

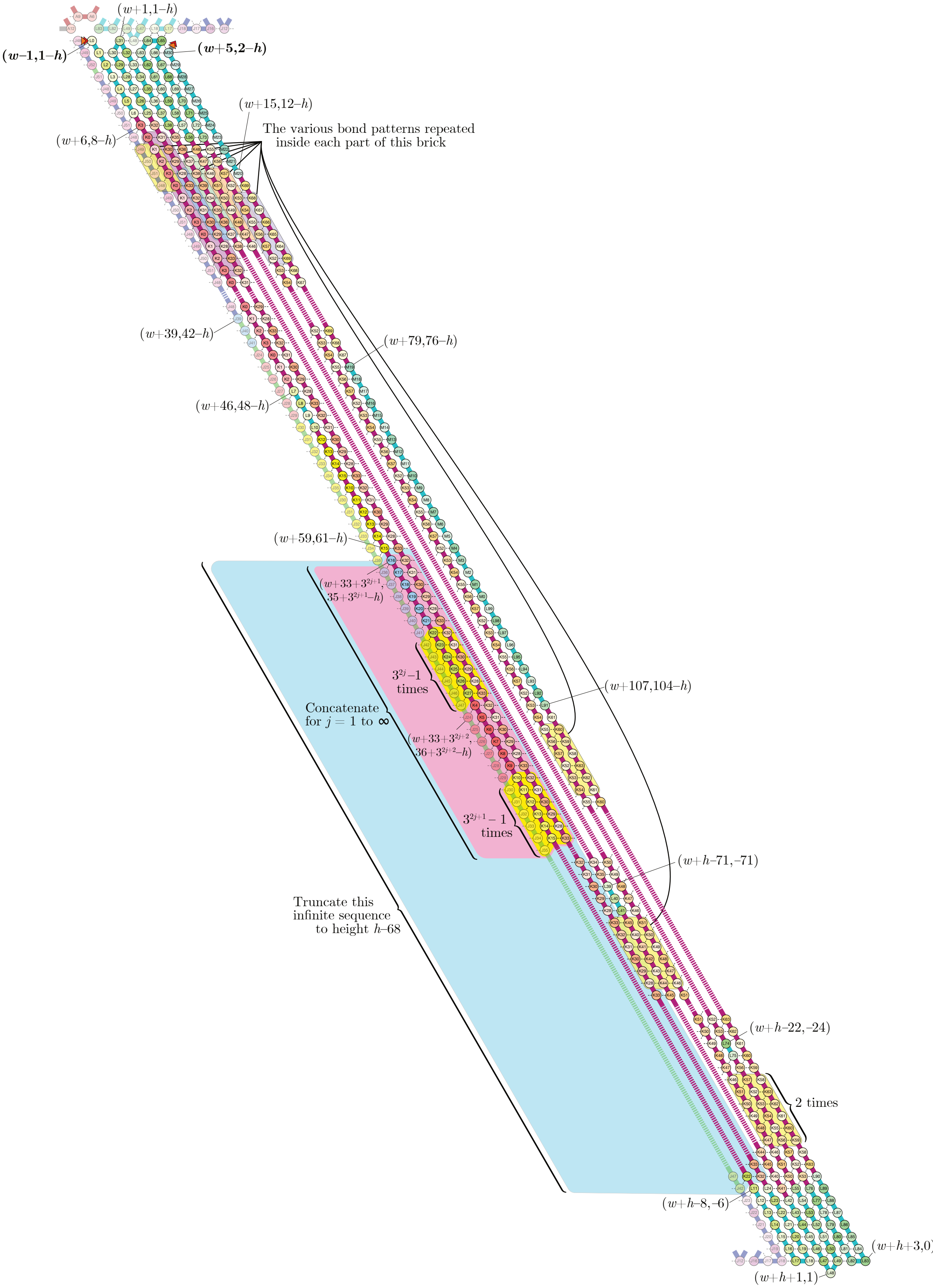


Figure 43 Module **G**: Brick **G ▸ Copy0**.

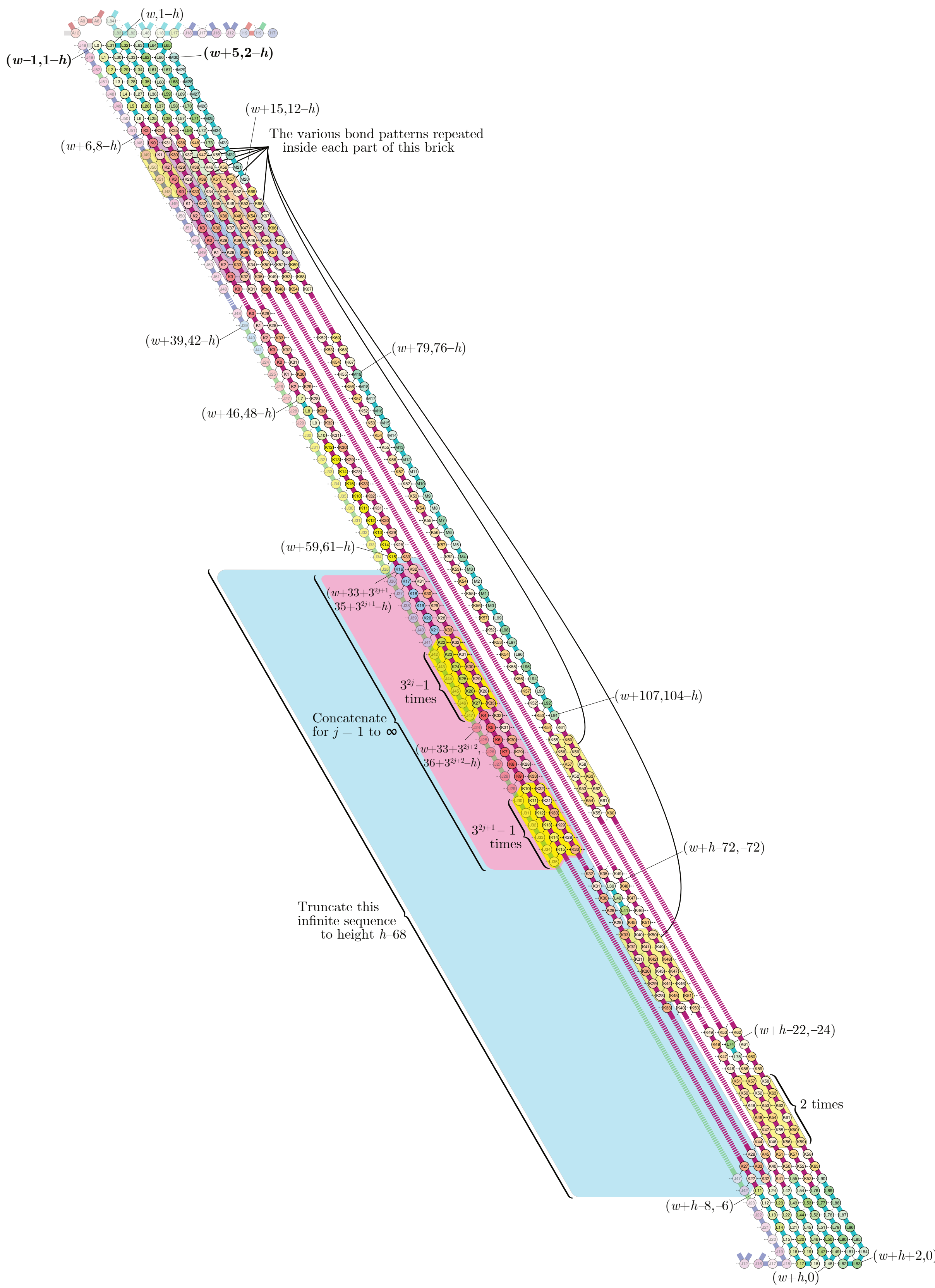


Figure 44 Module G: Brick G ▸ Copy1.

F.10.2.3 Line feed brick for Module G

SegG1A♯LF = NW-path(**L0..3**) ↖W SE-path(**L4..6** · **K3**)

G♯LineFeed = **SegG1A♯LF** ↙SW Rotate$_{180°}$ (**SegG1..3GliderNE►Read**) ↙SW **L40**↘SE RotateClockwise (**SegG3..6GliderE►Read1**, 120°) ↙SW **M20** ↖W **M21** ↖NW **M22** ↙SW **M23** ↘SE **SegG6End►Read0**

Figure 45 displays the brick **G♯LineFeed**.

F.10.3 Subrule for Module G

Module **G** interacts with **A**, **C**, **D**, **F**. Figure 46 presents the subrule for the interactions between the beads of **G** and the beads of the other modules.

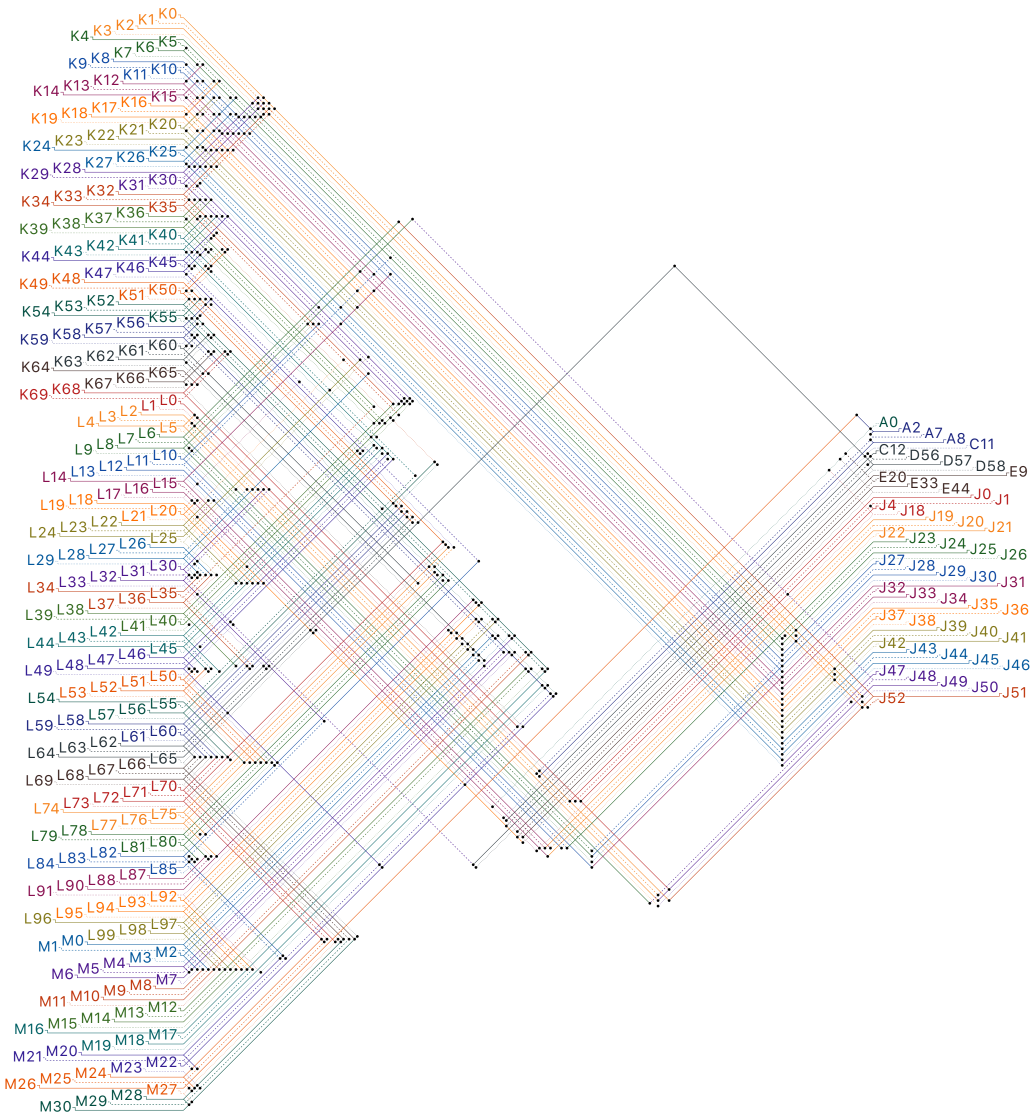


Figure 46 Subrule for Module **G**.

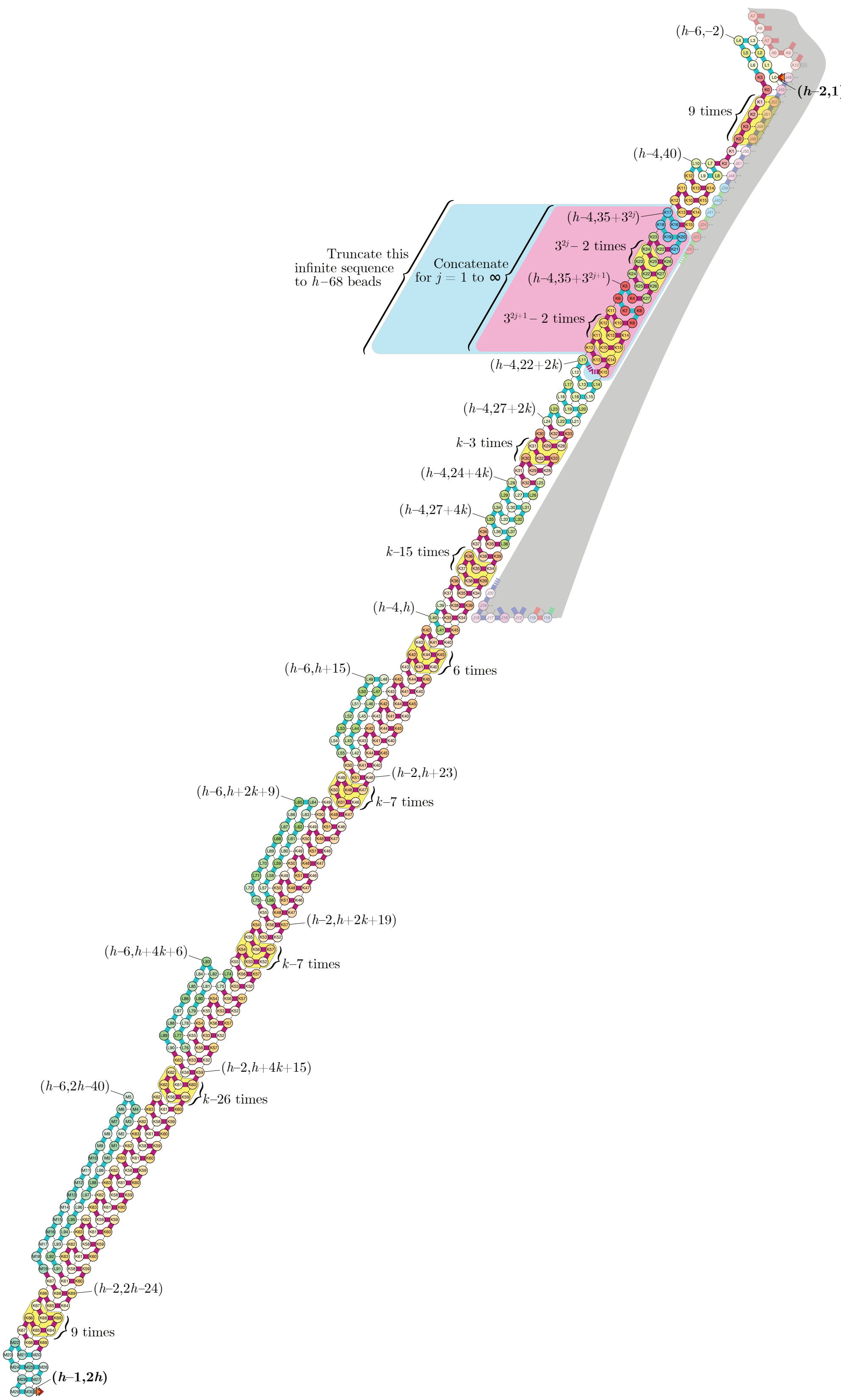


Figure 45 Module G: Brick G LineFeed.

F.11 The seed conformation

The seed $\boxed{\mathsf{Seed}(u)}$ is a conformation encoding the input dataword u of the simulated SCTS. It is made of 4 types of conformations, (see Figure 47 for an illustration):

$$\boxed{\mathsf{SegSeedBegin}} = \left(\mathsf{J8} \underset{\mathtt{SW}}{\swarrow} \mathsf{J7} \underset{\mathtt{SW}}{\swarrow}\right)^{2+\frac{h-3}{4}} \mathsf{J11} \underset{\mathtt{SW}}{\swarrow} \mathsf{J12} \underset{\mathtt{W}}{\leftarrow} \mathsf{J16} \underset{\mathtt{W}}{\leftarrow} \mathsf{J17} \underset{\mathtt{W}}{\leftarrow} \mathsf{J18}$$

$$\boxed{\mathsf{SegSeedSuffix}} = \mathsf{A9} \underset{\mathtt{SW}}{\swarrow} \mathsf{A12} \underset{\mathtt{W}}{\leftarrow} \mathsf{B0} \underset{\mathtt{W}}{\leftarrow} \mathsf{B1} \overset{\mathtt{NW}}{\nwarrow} \mathsf{B2} \underset{\mathtt{E}}{\rightarrow} \mathsf{B3} \overset{\mathtt{NE}}{\nearrow} \mathsf{B4} \overset{\mathtt{NW}}{\nwarrow} \mathsf{C4} \overset{\mathtt{NW}}{\nwarrow} \mathsf{A0} \left(\underset{\mathtt{W}}{\leftarrow} \mathsf{A0}\right)^{w-17} \underset{\mathtt{SW}}{\swarrow} \mathsf{H8} \underset{\mathtt{SW}}{\swarrow} \mathsf{H19} \underset{\mathtt{SW}}{\swarrow} \mathsf{H20} \underset{\mathtt{W}}{\leftarrow} \mathsf{H21} \underset{\mathtt{SW}}{\swarrow} \mathsf{H24} \underset{\mathtt{W}}{\leftarrow} \mathsf{I15} \underset{\mathtt{W}}{\leftarrow} \mathsf{I15} \underset{\mathtt{W}}{\leftarrow} \mathsf{I16} \underset{\mathtt{W}}{\leftarrow} \mathsf{I17} \underset{\mathtt{W}}{\leftarrow} \mathsf{I19} \underset{\mathtt{W}}{\leftarrow} \mathsf{I19} \underset{\mathtt{W}}{\leftarrow} \mathsf{J12} \underset{\mathtt{W}}{\leftarrow} \mathsf{J16} \underset{\mathtt{W}}{\leftarrow} \mathsf{J17} \underset{\mathtt{W}}{\leftarrow} \mathsf{J18}$$

$$\boxed{\mathsf{SegSeed}(0)} = \mathsf{L17} \underset{\mathtt{W}}{\leftarrow} \mathsf{L18} \underset{\mathtt{W}}{\leftarrow} \mathsf{L47} \underset{\mathtt{SW}}{\swarrow} \mathsf{L48} \overset{\mathtt{NW}}{\nwarrow} \mathsf{L49} \underset{\mathtt{W}}{\leftarrow} \mathsf{L82} \underset{\mathtt{W}}{\leftarrow} \mathsf{L83} \overset{\mathtt{NW}}{\nwarrow} \mathsf{A6}$$

$$\boxed{\mathsf{SegSeed}(1)} = \mathsf{L17} \underset{\mathtt{W}}{\leftarrow} \mathsf{L18} \underset{\mathtt{W}}{\leftarrow} \mathsf{L48} \underset{\mathtt{W}}{\leftarrow} \mathsf{L82} \underset{\mathtt{W}}{\leftarrow} \mathsf{L83} \overset{\mathtt{NW}}{\nwarrow} \mathsf{L84} \underset{\mathtt{W}}{\leftarrow} \mathsf{A6}$$

$$\boxed{\mathsf{SegSeedEnd}} = \mathsf{K34} \underset{\mathtt{SW}}{\swarrow} \left(\mathsf{K45} \underset{\mathtt{SW}}{\swarrow} \mathsf{K40} \underset{\mathtt{SW}}{\swarrow}\right)^{11} \left(\mathsf{K46} \underset{\mathtt{SW}}{\swarrow} \mathsf{K47} \underset{\mathtt{SW}}{\swarrow}\right)^{k-2} \left(\mathsf{K57} \underset{\mathtt{SW}}{\swarrow} \mathsf{K52} \underset{\mathtt{SW}}{\swarrow}\right)^{k-2} \left(\mathsf{K59} \underset{\mathtt{SW}}{\swarrow} \mathsf{K60} \underset{\mathtt{SW}}{\swarrow}\right)^{k-18} \left(\mathsf{K69} \underset{\mathtt{SW}}{\swarrow} \mathsf{K64} \underset{\mathtt{SW}}{\swarrow}\right)^{10} \mathsf{K69} \underset{\mathtt{SW}}{\swarrow} \mathsf{M20} \underset{\mathtt{SE}}{\searrow} \mathsf{M26} \underset{\mathtt{SW}}{\swarrow} \mathsf{M27} \underset{\mathtt{W}}{\leftarrow} \mathsf{M28} \underset{\mathtt{SW}}{\swarrow} \mathsf{M29} \underset{\mathtt{E}}{\rightarrow} \mathsf{M30}.$$

Each letter $a \in \{0, 1\}$ is encoded in the seed by the conformation: (note that it heads westwards)

$$\boxed{\mathsf{SegSeedLetter}(a)} = \left(\boxed{\mathsf{SegSeed}(1)} \underset{\mathtt{W}}{\leftarrow} \boxed{\mathsf{SegSeedSuffix}} \underset{\mathtt{W}}{\leftarrow}\right)^{n-1} \boxed{\mathsf{SegSeed}(a)} \underset{\mathtt{W}}{\leftarrow} \boxed{\mathsf{SegSeedSuffix}}$$

Then, the conformation $\boxed{\mathsf{Seed}(u)}$ is: (note that it heads southwestwards, see Fig. 47)

$$\boxed{\mathsf{Seed}(u)} = \boxed{\mathsf{SegSeedBegin}} \underset{\mathtt{W}}{\leftarrow} \left(\bigodot_{i=1}^{|u|} \boxed{\mathsf{SegSeedLetter}(u_{|u|-i})} \underset{\mathtt{W}}{\leftarrow}\right) \boxed{\mathsf{SegSeedEnd}}$$

This completes the definition of the primary structure of the oritatami system $\mathcal{O}_{\mathcal{S}} = ((\pi_{\mathcal{S}})^{\infty}, \heartsuit, 3)$ whose folding from the seed configuration $\boxed{\mathsf{Seed}(u)}$ simulates step-by-step the computation of SCTS $\mathcal{S}$ starting with input dataword u. Section H presents the full description of the attraction rule $\heartsuit$, completing the description of the oritatami system $\mathcal{O}_{\mathcal{S}}$.

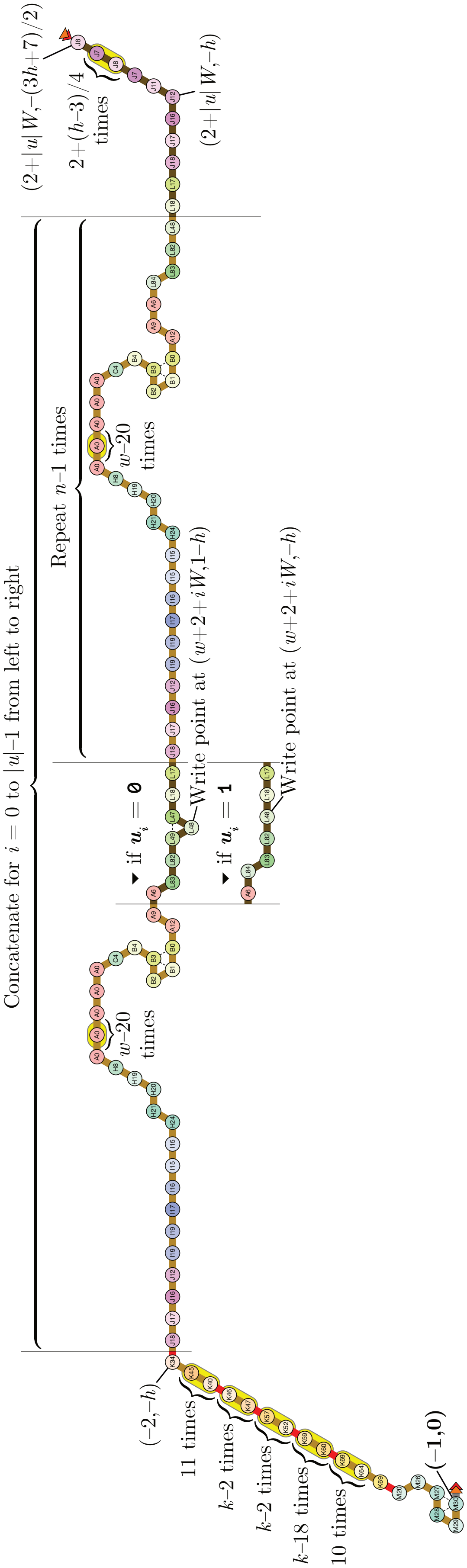


Figure 47 The brick Seed(u).

G Computerized proof of correctness of the STCS oritatami simulation

G.1 Enumerating all possible environments for each module

We will now resume and expand the explanations given in Section 5. Here is how we proceeded to ensure the correctness of our design:

1. Enumerate all the surrounding for each brick of each module
2. Enumerate all possible modules following the module
3. Generate automatically human-readable certificate of the correctness of the folding for each possibility, in the form of *proof trees*.
4. In the few cases where the surrounding may vary, prove that it has no incidence on the folding of the brick. This happens only for three bricks exactly: when the brick **G▶Read** zig-folds along **F▶**, when the top of the brick **G▶Read1** folds, and when the zag-bricks folds under **D✎**.

One can check using Fact 8 that the beads alignments in each brick do not change when n and L vary. This implies that the figures of the bricks are indeed generic. It follows that with the exception of the three cases listed in point 4 above, and handled in Section G.1.1, it is enough to prove the folding of each brick only once. And as most of them are made of repeating patterns, only a finite number of environments have to be considered. That last case will be treated in Section G.1.2 using an automatic procedure which produces human-readable certificates called *proof-trees*.

G.1.1 The three bricks with varying environments

The following lemma show that it is enough to proof one folding of the zag bricks under a **D✎**, all the other are the same since there are no interaction between the **D✎** brick and any zag brick folding immediately below it.

▶ **Lemma 9** (Zag-folding under **D✎**). *The modules zag-folding under the bricks* **D✎** *have no interaction with* **D✎**, *with the only exceptions of:*

- *the beads* **A0** *and* **A1** *of module* **A** *which have bonds with beads* **E**$(2+12i)$ *and* **L17** *for* **A0**, *and* **E**$(9+12i)$ *for* **A1**, *for all* $0 \leqslant i \leqslant 3$.
- *the beads* **L17**, **L18**, **D57**, **D58** *(the bump in module* **D0***) which bond with the beads* **L65**, **L64**, **L31** *so that the corresponding module* **G** *folds into the expected brick* **G◀Copy0**.

Proof. Figure 48 lists all the possible ♥-interactions between the beads accessible from below the **D✎** bricks (to the left) with the beads at the top the modules zag-folding below it that can interact with them (to the right).

The only possible bonds are thus:

with beads D17 and D22: (in green on Figure 48) these are only present at the junction between the bricks **D✎** and **E▶Return**, at the end of the rightmost **D✎** brick. The correctness of the zag-folding of the **F◀** brick below is given next in the proof-trees section.

with beads L17, L18, D56, D57, D58, D62: (in blue on Figure 48) these beads are only present in the spike encoding a 0 in the brick **D✎**, and these interactions are the one expected to ensure the copy of the encoding of 0 by the module **G** that will Zag-fold below.

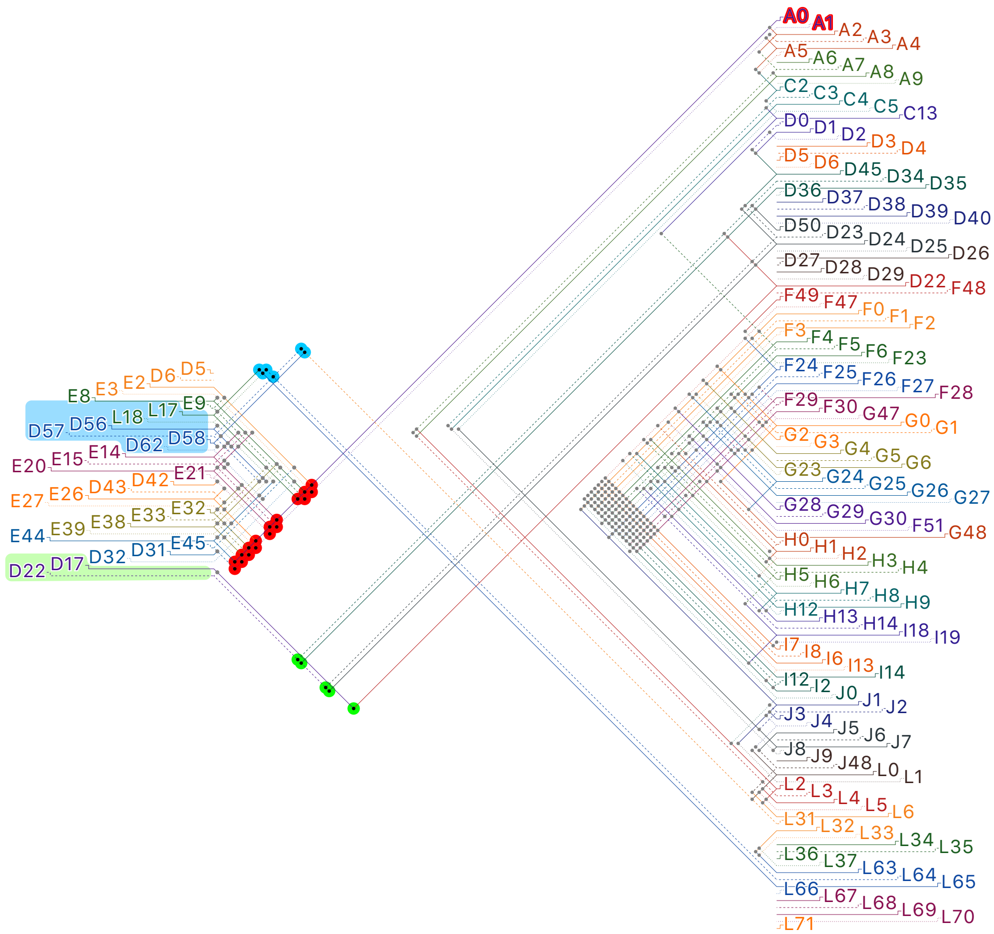


Figure 48 The ♥-rule between the beads accessible from below of brick D✎ and the beads that will get in touch with them from all the modules Zag-folding below.

and finally between beads A0 and A1, and 4 groups of beads: **E2**, **E3**, **E8**, **E9**, then **E14**, **E15**, **E20**, **E21**, then **E26**, **E27**, **E32**, **E33**, and finally **E38**, **E39**, **E44**, **E45** (in red on Figure 48). As the width of a zag-folded production segment is $w + 6 = 0 \bmod 12$, the beads **A0** and **A1** are always aligned with the same beads within each of these groups (see Figure 33), namely **A0** with **E2**, **E14**, **E26** and **E38**, and **A1** with **E9**, **E21**, **E33** and **E45**. Furthermore as the interactions of **A0** and **A1** are the same with each of them, it is enough to prove that the module **A** zag-folds correctly between *one* of these groups only, which is done next in the proof-trees section.

It follows that outside these three cases (each handled by a proof-tree, see later), no interactions are possible and the modules will zag-fold below the **D✎** bricks independently of the exact beads that are present inside. It is thus enough to show that each module zag-folds correctly at any location to ensure that it zag-folds correctly anywhere below the **D✎** brick. ◀

▶ **Lemma 10** (Top of **G▶Read1**). *During the folding of the brick* **G▶Read1**, *no bead in* **G** *interacts with the row above but at its two extremeties, i.e. the 82 top-leftmost beads and the 11 last (***K34***..***L55*** and* **M20***..***M30** *resp. in Figure 42).*

Proof. Figure 49(a) lists the only beads exposed and accessible from below above **G▶Read1**. And Figure 49(b) lists all the possible ♥-interactions between them (to the left) and the beads of the brick **G▶Read1** zig-folding below (to the right).

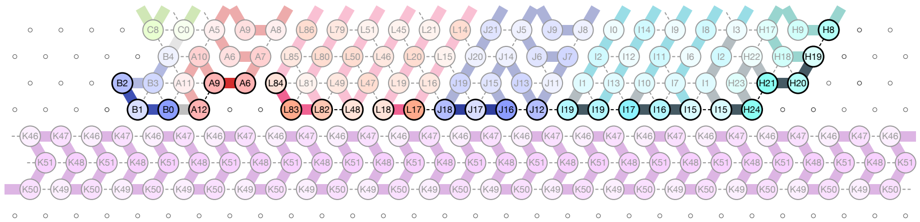
According to the rule in Figure 49(b), besides the interactions at the 82 first beads at the very top-leftmost part of **G▶Read1** (**K34**..**L55** in Figure 42, interactions in green in Figure 49(b)) and the 11 beads at the very end of **G▶Read1** (**M20**..**M30** in Figure 42, interactions in blue in Figure 49(b)), the only possible interaction between **G▶Read1** and the already present beads above it is: **L82**♥**L74**. But **L74** appears only once in **G▶Read1**, at coordinates $(w + 10 + 4k, 1 - h)$ (see Figure **??**), while **L82** appears above **G▶Read1** at coordinates $(w + 1 + i(w + 6), 2 - h)$ for $i = 0..n$. The minimal x-distance between **L82** and **L74** is thus $\min_{i=0..n} |9+4k-i(w+6)|$. But $9+4k-i(w+6) = 9+4(n-1)(w+6)/6-i(w+6) = 9+2(n-1-3i)(2(L+P)+8)$. It follows that the minimum difference in x-coordinate between **L82** and **L74** is:

- $17 + 2(L + P) \geqslant 41$, if $n = 0 \bmod 3$;
- 9, if $n = 1 \bmod 3$; and
- $1 - 2(L + P) \leqslant -23$, if $n = 2 \bmod 3$.

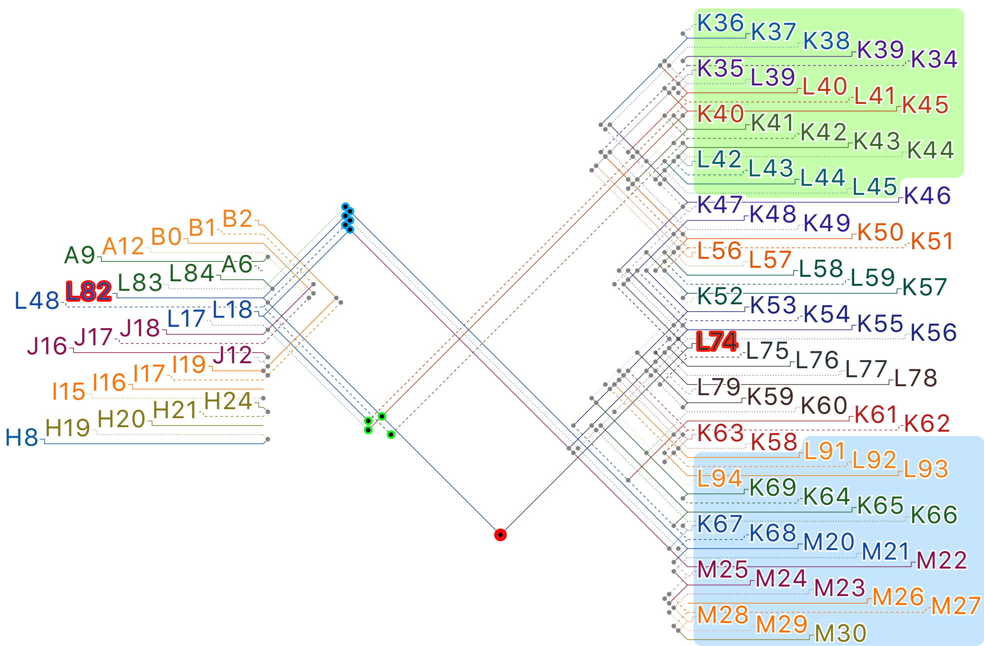
As a consequence, **L74** never gets close enough to interact with **L82** above (see Figure 49(c) for the closest situation). It follows that one only need to take into account the environnement for the folding of the top-leftmost and top-rightmost part of brick **G▶Read1** (which is done next using proof-trees), the glider between them, zig-folds regardless of the beads above in the environment. ◀

▶ **Lemma 11** (**G▶Read1** along **F▶**). *When* **G** *folds into the brick* **G▶Read** *, no bead in* **SegExpG** *can make bonds with the beads in* **F▶** *nearby and thus folds regardless of the beads nearby (as a glider).*

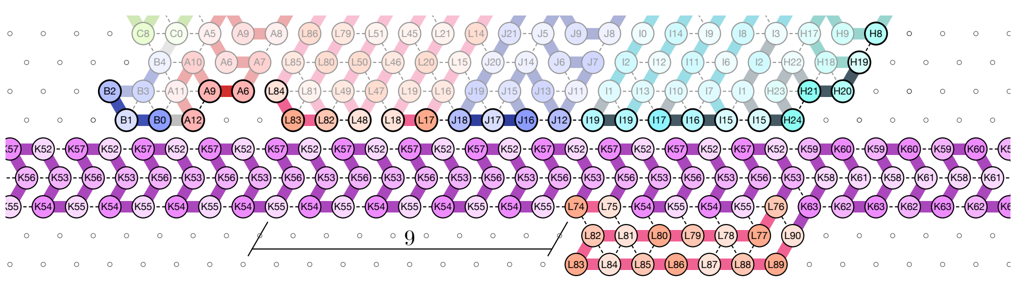
Proof. Figure 50 lists the interactions between the beads in **SegExpG** and the beads in **SegExpF**: these are exactly **K**$(4 + i)$♥**J**$(24 + i)$ for $i = 0..23$; in particular red-shaded beads **K4**..**K9** in **G** (resp. yellow, **K10**..**K15**; blue, **K16**..**K21**; and green, **K22**..**K27**) can only bond with beads of the same shade **J24**..**J29** in **F** (resp. **J30**..**J35**; **J36**..**J41**; **J42**..**J47**).



(a) The beads accessible when the brick G▸Read1 zig-folds itself.



(b) The ♥-rule for the beads accessible by the beads in G▸Read1 as it zig-folds.



(c) The closest bead L74 in brick G▸Read1 can get from one bead L82 above (case $n = 1 \bmod 3$).

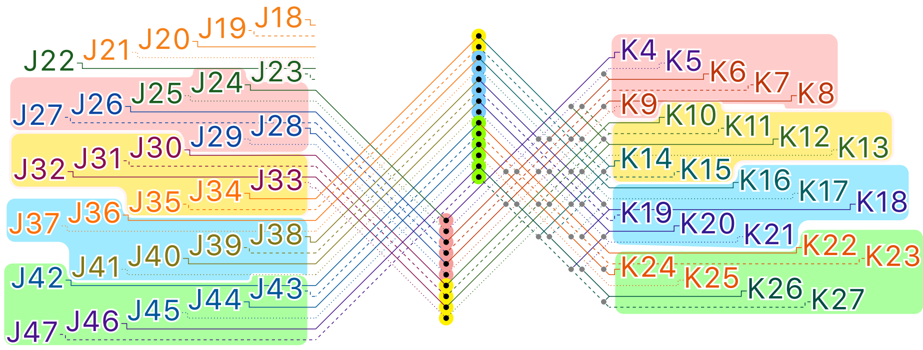


Figure 50 ♥-rule between the two exponential segments in **F** and **G**. Note that each bead makes exactly one bond, with a bead of the same shade, red, blue, yellow or green (see Figure **??** and **??**) and of the same rank within the shade.

As shown on Figure 39 and 41 the y-coordinates explored by these beads are as follows when **G** zig-folds into **G▶Read0** or **G▶Read1**:

Red : the y-coordinates of beads **J24**..**J29** in **F** belong to $\{-40-3^{2j},\dots,-35-3^{2j}\}$ for $j \geqslant 1$, while the corresponding beads **K4**..**K9** in **G** explore y-coordinates in $\{-38-3^{2j'+1},\dots,-34-3^{2j'+1}\}$ for $j' \geqslant 1$.

Yellow : the y-coordinates of beads **J30**..**J35** in **F** belong to $\{-34-3^{2j+1},\dots,-41-3^{2j}\}$ for $j \geqslant 1$, while the corresponding beads **K10**..**K15** in **G** explore y-coordinates in $\{-36-3^{2j'+2},\dots,-36-3^{2j'+1}\}$ for $j' \geqslant 1$.

Blue : the y-coordinates of beads **J36**..**J41** in **F** belong to $\{-40-3^{2j+1},\dots,-35-3^{2j+1}\}$ for $j \geqslant 1$, while the corresponding beads **K16**..**K21** in **G** explore y-coordinates in $\{-38-3^{2j'},\dots,-34-3^{2j'}\}$ for $j' \geqslant 1$.

Green : the y-coordinates of beads **J42**..**J47** in **F** belong to $\{-34-3^{2j+2},\dots,-41-3^{2j+1}\}$ for $j \geqslant 1$, while the corresponding beads **K2**..**K27** in **G** explore y-coordinates in $\{-36-3^{2j'+1},\dots,-36-3^{2j'}\}$ for $j' \geqslant 1$.

Now, as for all $j \geqslant 1$ (with the notation, $a \lessdot b$ iff $a \leqslant b-2$)

$$\begin{aligned} & -35-3^{2j+2} \lessdot -38-3^{2j+1} \lessdot -34-3^{2j+1} \lessdot -40-3^{2j} \\ \text{and} \quad & -36-3^{2j+1} \lessdot -34-3^{2j+1} \lessdot -41-3^{2j} \lessdot -36-3^{2j} \\ \text{and} \quad & -34-3^{2j+2} \lessdot -40-3^{2j+1} \lessdot -35-3^{2j+1} \lessdot -38-3^{2j} \\ \text{and} \quad & -41-3^{2j+1} \lessdot -36-3^{2j+1} \lessdot -36-3^{2j} \lessdot -34-3^{2j} \end{aligned}$$

none of the (same-shade) interacting beads ever get close enough to each other and the beads in the segment **SegExpG** folds without making any bond (into a glider), regardless of the beads next to them in **F▶** when **G** zig-folds into brick **G▶Read**. ◀

G.1.2 Proof-trees: An automated human-readable certificate for the correctness of oritatami system

A *proof-tree* is a compact representation of the enumeration of all the possible paths the molecule explores as it folds. Figure 51 presents the proof-tree for the folding of **G** when

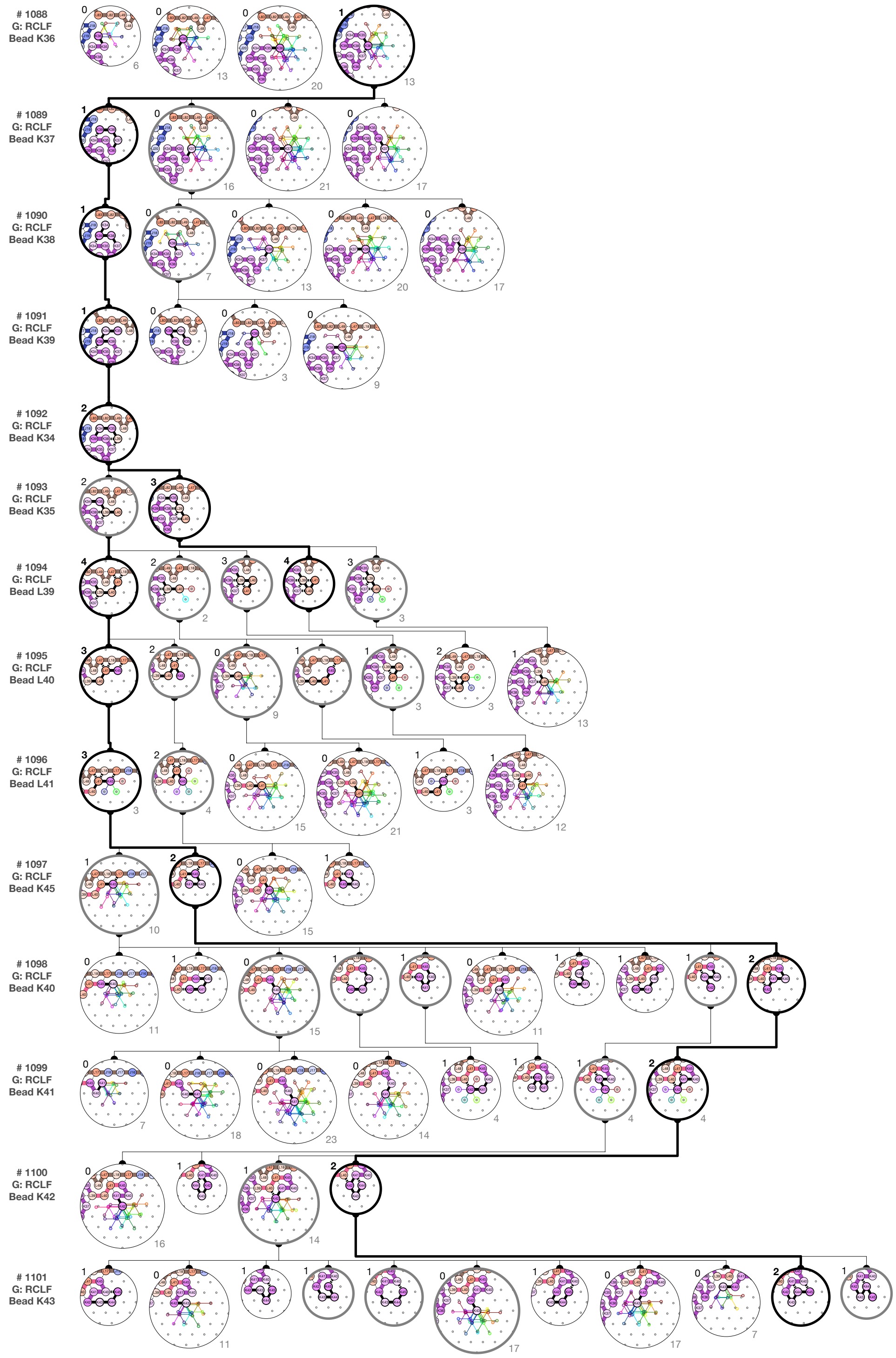


Figure 51 Excerpt from the proof-tree certificate for the folding of G into G ▶ Read0 when bouncing on a spike encoding a 0.

bouncing on a bump encoding a 0 in **G▶Read0**. For the sake of readability, several paths are drawn in the same ball when they share the same beginning up to their last bond with the environment; then, as a sanity check, the grey number at the bottom left of the ball indicates how many paths are drawn in this ball. The black number in the top right corner of each ball indicates how many bonds are made by the paths with the environment. The ball(s) with the maximum number of bonds is(are) highlighted in black and go to the next round, together with the balls that place the first bead at the same position.

These proof-trees are automatically generated as the molecule folds. Each environment (surrouding + the three beads currently folding) is given a number (written #xxxx). When an already studied environment is encountered, the proof-tree is stopped, and the next (already encountered) environment number is written, allowing easy navigation in the proof — note that Figure 51 is an excerpt from a larger proof-tree and does not show its beginning nor its end, this is why the navigation tag cannot be observed in this figure.

The complete proof certificates may be found on the website:

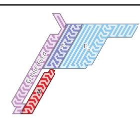
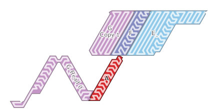
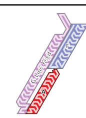
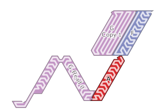
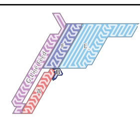
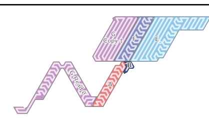
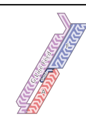
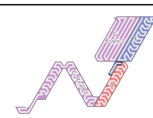
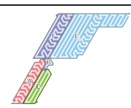
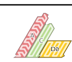
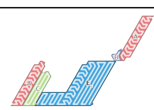
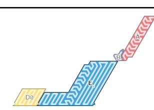
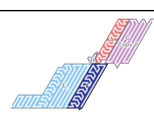
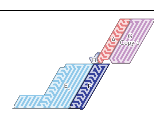
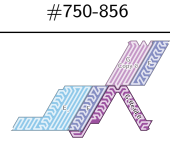
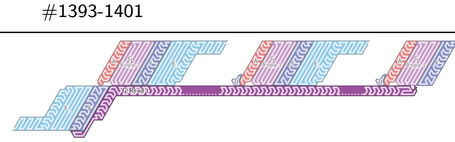
`https://www.irif.fr/∼nschaban/oritatami/prooftrees/`

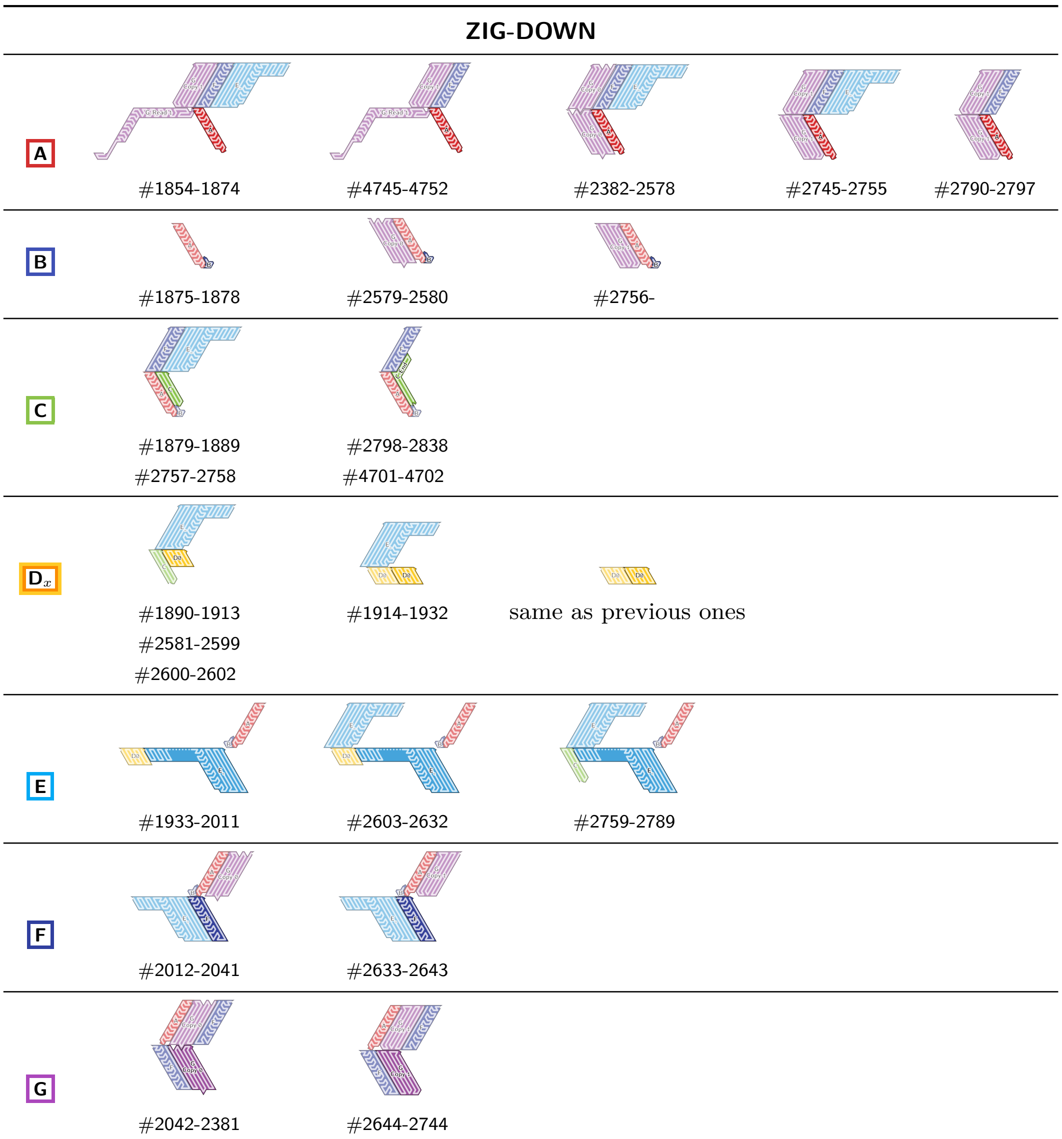
G.2 Computer-generated proof trees for each possible environment

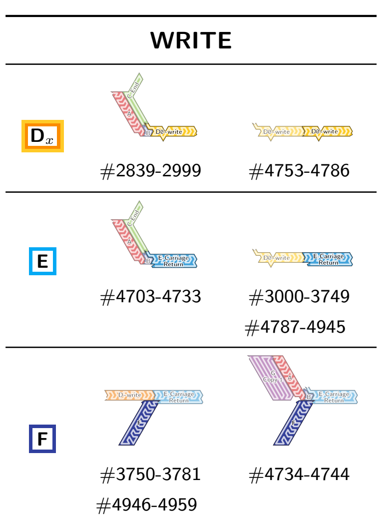
The following tables refer to the proof-trees on the website:

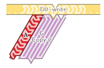
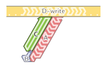
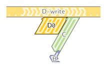
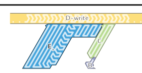
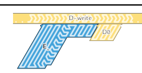
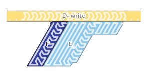
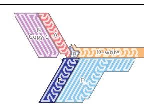
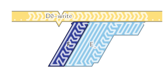
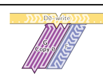
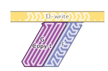
https://www.irif.fr/∼nschaban/oritatami/prooftrees/

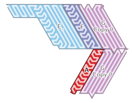
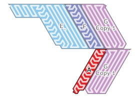
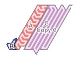
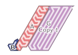
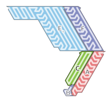
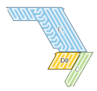
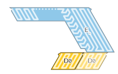
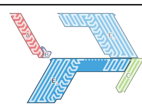
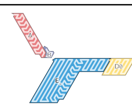
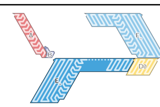
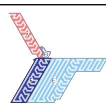
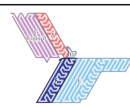
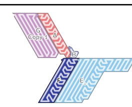
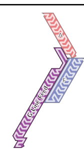
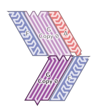
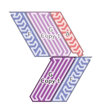
proving the correctness of the folding of our design in every possible surroundings.

ZIG-UP				
A	 #0-98	 #1286-1312	 #-	 #4995-4997
B	 #99-103	 #1313-	 #-	 #4998-5000
C	 #104-159 #1314-1315			
D_x	 #160-339	 #340-384		
E	 #1316-1392	 #385-749		
F	 #750-856	 #1393-1401		
G	 #857-1285	 #1402-1853		


ZIG-DOWN
A
#1854-1874
#4745-4752
#2382-2578
#2745-2755
#2790-2797
B
#1875-1878
#2579-2580
#2756-
C
#1879-1889
#2757-2758
#2798-2838
#4701-4702
D_x
#1890-1913
#2581-2599
#2600-2602
#1914-1932
same as previous ones
E
#1933-2011
#2603-2632
#2759-2789
F
#2012-2041
#2633-2643
G
#2042-2381
#2644-2744


WRITE
D_x
#2839-2999
#4753-4786
E
#4703-4733
#3000-3749
#4787-4945
F
#3750-3781
#4946-4959
#4734-4744

ZAG-WRITE			
A	 #4994-	 #3806-3816 , #4059-4069 , #4215-4225 #4310-4320	
C	 #3817-3827 , #4070-4080 , #4226-4236 #4321-4331 , #4459-4460 , #4475-4476 #4550-4551 , #4605-4606		
D_x	 #3828-3851 #4237-4263 #4332-4355	 #3939-3941 , #4434-4439 , #4525-4527 #4571-4573 , #4587-4589 , #4595-4597 #4618-4620	
E	 #4081-4176 , #4461-4474 #4552-4570 , #4607-4617	 #3852-3924 , #3942-4020 , #4264-4271 #4356-4428 , #4440-4458 , #4504-4519 #4528-4549 , #4574-4581 , #4590-4594 #4598-4604 , #4621-4644	
F	 #3925-3938 , #4021-4034 , #4177-4190 #4272-4285 , #4429-4433 , #4520-4524 #4582-4586	 #4645-4653	 #4960-4967
G	 #4968-4993	 #3782-3805 , #4035-4058 , #4191-4214 #4286-4309	

	ZAG		
A	 see Zig-Down	 see Zig-Down	
B	 see Zig-Down	 see Zig-Down	
C	 see Zig-Down		
D_x	 see Zig-Down	 see Zig-Down	 see Zig-Down
E	 see Zig-Down	 see Zig-Down	 see Zig-Down
F	 see Zig-Down	 see Zig-Down	
G	 #4654-4700	 see Zig-Down	 see Zig-Down

H The complete attraction rule

We first gives the rule in text. Fig. 52 displays it as a matrix.

A

A00 ♥ A02
A00 ♥ E02
A00 ♥ E08
A00 ♥ E14
A00 ♥ E20
A00 ♥ E26
A00 ♥ E32
A00 ♥ E38
A00 ♥ E44
A00 ♥ J18
A00 ♥ L17
A01 ♥ A03
A01 ♥ A04
A01 ♥ E03
A01 ♥ E09
A01 ♥ E15
A01 ♥ E21
A01 ♥ E27
A01 ♥ E33
A01 ♥ E39
A01 ♥ E45
A01 ♥ J18
A02 ♥ A00
A02 ♥ A07
A02 ♥ M26
A03 ♥ A01
A04 ♥ A01
A04 ♥ C02
A05 ♥ A10
A05 ♥ C00
A05 ♥ C01
A05 ♥ C02
A07 ♥ A02
A07 ♥ A08
A07 ♥ L02
A07 ♥ L03
A08 ♥ A07
A08 ♥ L03
A09 ♥ A11
A09 ♥ A12
A10 ♥ A05
A10 ♥ B04
A11 ♥ A09
A11 ♥ B00
A11 ♥ B03
A11 ♥ B04
A12 ♥ A09
A12 ♥ J17

B

B00 ♥ A11
B00 ♥ B03
B00 ♥ J17
B01 ♥ B03
B01 ♥ D05
B01 ♥ F05
B01 ♥ J00
B01 ♥ J01
B01 ♥ J18
B02 ♥ D03
B02 ♥ D04
B02 ♥ F03
B02 ♥ F04
B02 ♥ I17
B02 ♥ I19
B03 ♥ A11
B03 ♥ B00
B03 ♥ B01
B03 ♥ D02
B03 ♥ F02
B04 ♥ A10
B04 ♥ A11
B04 ♥ C08
B04 ♥ D00
B04 ♥ D02
B04 ♥ F02
B04 ♥ F48

C

C00 ♥ A05
C00 ♥ C05
C00 ♥ C07
C00 ♥ C08
C00 ♥ C10
C00 ♥ C13
C01 ♥ A05
C01 ♥ C10
C01 ♥ C13
C02 ♥ A04
C02 ♥ A05
C02 ♥ C05
C02 ♥ C08
C03 ♥ C10
C03 ♥ C11
C03 ♥ C13
C03 ♥ C14
C03 ♥ C15
C03 ♥ J08
C03 ♥ J12
C04 ♥ C08
C04 ♥ J08
C04 ♥ J11
C05 ♥ C00
C05 ♥ C02
C05 ♥ C07
C05 ♥ J07
C06 ♥ C11
C06 ♥ C14
C07 ♥ C00
C07 ♥ C05
C08 ♥ B04
C08 ♥ C00
C08 ♥ C02
C08 ♥ C04
C08 ♥ C11
C08 ♥ C14
C09 ♥ D03
C09 ♥ D15
C09 ♥ D58
C09 ♥ D59
C09 ♥ E05
C09 ♥ E11
C09 ♥ F03
C09 ♥ F09
C10 ♥ C00
C10 ♥ C01
C10 ♥ C03
C10 ♥ D02
C10 ♥ D08
C10 ♥ D57
C10 ♥ D58
C10 ♥ F02
C10 ♥ F08
C11 ♥ C03
C11 ♥ C06
C11 ♥ C08
C11 ♥ D07
C11 ♥ D13
C11 ♥ D56
C11 ♥ D57
C11 ♥ D62
C11 ♥ E02
C11 ♥ E08
C11 ♥ F01
C11 ♥ F07
C11 ♥ L17
C12 ♥ D12
C12 ♥ D13
C12 ♥ D55
C12 ♥ D56
C12 ♥ D61
C12 ♥ D62
C12 ♥ E02
C12 ♥ E08
C12 ♥ F00
C12 ♥ F06
C12 ♥ L17
C13 ♥ C00
C13 ♥ C01
C13 ♥ C03
C13 ♥ D11
C13 ♥ D55
C13 ♥ D60
C13 ♥ D61
C13 ♥ F05
C13 ♥ F11
C14 ♥ C03
C14 ♥ C06
C14 ♥ C08
C14 ♥ D04
C14 ♥ D10
C14 ♥ D16
C14 ♥ D59
C14 ♥ D60
C14 ♥ E05
C14 ♥ E11
C14 ♥ F04
C14 ♥ F10
C15 ♥ C03
C15 ♥ D01

D

D00 ♥ B04
D00 ♥ D02
D00 ♥ D11
D00 ♥ D45
D01 ♥ C15
D01 ♥ D45
D02 ♥ B03
D02 ♥ B04
D02 ♥ C10
D02 ♥ D00
D03 ♥ B02
D03 ♥ C09
D03 ♥ D08
D03 ♥ D48
D04 ♥ B02
D04 ♥ C14
D04 ♥ D48
D05 ♥ B01
D06 ♥ E22
D07 ♥ C11
D07 ♥ E00
D07 ♥ E01
D07 ♥ E02
D07 ♥ E22
D08 ♥ C10
D08 ♥ D03
D08 ♥ D16
D09 ♥ E19
D10 ♥ C14
D10 ♥ D14
D10 ♥ E19
D11 ♥ C13
D11 ♥ D00
D11 ♥ D13
D12 ♥ C12
D12 ♥ E16
D13 ♥ C11
D13 ♥ C12
D13 ♥ D11
D13 ♥ E16
D14 ♥ D10
D15 ♥ C09
D15 ♥ E13
D16 ♥ C14
D16 ♥ D08
D16 ♥ E13
D17 ♥ D22
D18 ♥ D27
D18 ♥ D38
D18 ♥ E18
D18 ♥ E42
D18 ♥ F03
D18 ♥ F27
D19 ♥ D26
D19 ♥ D27
D19 ♥ D37
D19 ♥ D38
D19 ♥ E23
D19 ♥ E47
D20 ♥ F02
D20 ♥ F26
D21 ♥ F00
D21 ♥ F01
D21 ♥ F24
D21 ♥ F25
D22 ♥ D17
D22 ♥ D23
D22 ♥ D24
D22 ♥ D34
D22 ♥ D35
D23 ♥ D22
D23 ♥ D45
D23 ♥ D51
D24 ♥ D22
D24 ♥ D45
D25 ♥ D53
D26 ♥ D19
D26 ♥ D48
D26 ♥ D54
D27 ♥ D18
D27 ♥ D19
D27 ♥ D48
D28 ♥ E46
D28 ♥ E47
D29 ♥ E22
D29 ♥ E46
D30 ♥ D32
D30 ♥ D33
D30 ♥ E22
D31 ♥ E43
D31 ♥ E45
D32 ♥ D30
D32 ♥ E08
D32 ♥ E19
D32 ♥ E42
D32 ♥ E43
D33 ♥ D30
D33 ♥ E07
D33 ♥ E08
D33 ♥ E19
D33 ♥ E42
D34 ♥ D22
D34 ♥ D46
D34 ♥ D50
D35 ♥ D22
D35 ♥ D50
D36 ♥ D48
D37 ♥ D19
D37 ♥ D49
D37 ♥ D53
D38 ♥ D18
D38 ♥ D19
D38 ♥ D53
D39 ♥ E22
D39 ♥ E23
D40 ♥ E22
D40 ♥ E46
D41 ♥ D43
D41 ♥ D44
D41 ♥ E46
D42 ♥ E19
D42 ♥ E21
D43 ♥ D41
D43 ♥ E18
D43 ♥ E19
D43 ♥ E32
D43 ♥ E43
D44 ♥ D41
D44 ♥ E18
D44 ♥ E31
D44 ♥ E32
D44 ♥ E43
D45 ♥ D00
D45 ♥ D01
D45 ♥ D23
D45 ♥ D24
D45 ♥ E18
D46 ♥ D34
D47 ♥ E12
D48 ♥ D03
D48 ♥ D04
D48 ♥ D26
D48 ♥ D27
D48 ♥ D36
D49 ♥ D37
D50 ♥ D34
D50 ♥ D35
D50 ♥ E42
D51 ♥ D23
D52 ♥ E36
D53 ♥ D25
D53 ♥ D37
D53 ♥ D38
D54 ♥ D26
D55 ♥ C12
D55 ♥ C13
D55 ♥ D58
D55 ♥ D59
D55 ♥ E15
D55 ♥ E16
D55 ♥ E39
D55 ♥ E40
D56 ♥ C11
D56 ♥ C12
D56 ♥ D58
D56 ♥ E14
D56 ♥ E15
D56 ♥ E38
D56 ♥ E39
D56 ♥ L18
D57 ♥ C10
D57 ♥ C11
D57 ♥ E13
D57 ♥ E14
D57 ♥ E37
D57 ♥ E38
D57 ♥ L31
D57 ♥ L64
D58 ♥ C09
D58 ♥ C10
D58 ♥ D55
D58 ♥ D56
D58 ♥ D62
D58 ♥ E12
D58 ♥ E13
D58 ♥ E36
D58 ♥ E37
D58 ♥ L31
D59 ♥ C09
D59 ♥ C14
D59 ♥ D55
D59 ♥ D61
D59 ♥ D62
D59 ♥ E11
D59 ♥ E12
D59 ♥ E35
D59 ♥ E36
D60 ♥ C13
D60 ♥ C14
D60 ♥ E05
D60 ♥ E22
D60 ♥ E23
D60 ♥ E29
D60 ♥ E46
D60 ♥ E47
D61 ♥ C12
D61 ♥ C13
D61 ♥ D59
D61 ♥ E21
D61 ♥ E22
D61 ♥ E45
D61 ♥ E46
D62 ♥ C11
D62 ♥ C12
D62 ♥ D58
D62 ♥ D59
D62 ♥ E20
D62 ♥ E21
D62 ♥ E44
D62 ♥ E45
E00 ♥ D07
E00 ♥ E05
E00 ♥ E23
E00 ♥ E46
E01 ♥ D07
E01 ♥ E22
E01 ♥ E46
E02 ♥ A00
E02 ♥ C11
E02 ♥ C12
E02 ♥ D07
E02 ♥ E09
E02 ♥ E45
E03 ♥ A01
E03 ♥ E08
E03 ♥ E20
E04 ♥ E19
E04 ♥ E42
E05 ♥ C09
E05 ♥ C14
E05 ♥ D60
E05 ♥ E00
E05 ♥ E42
E06 ♥ E11
E06 ♥ E17
E07 ♥ D33
E07 ♥ E16
E07 ♥ E39
E07 ♥ E40
E08 ♥ A00
E08 ♥ C11
E08 ♥ C12
E08 ♥ D32
E08 ♥ D33
E08 ♥ E03
E08 ♥ E15
E08 ♥ E38
E08 ♥ E39
E09 ♥ A01
E09 ♥ E02
E09 ♥ E14
E09 ♥ L17
E10 ♥ E13
E10 ♥ E14
E10 ♥ E36
E11 ♥ C09
E11 ♥ C14
E11 ♥ D59
E11 ♥ E06
E11 ♥ E36
E12 ♥ D47
E12 ♥ D58
E12 ♥ D59
E12 ♥ E17
E12 ♥ E34
E12 ♥ E35
E13 ♥ D15
E13 ♥ D16
E13 ♥ D57
E13 ♥ D58
E13 ♥ E10
E13 ♥ F26
E13 ♥ F27
E14 ♥ A00
E14 ♥ D56
E14 ♥ D57
E14 ♥ E09
E14 ♥ E10
E14 ♥ E21
E14 ♥ E32
E15 ♥ A01
E15 ♥ D55
E15 ♥ D56
E15 ♥ E08
E15 ♥ E20
E15 ♥ E31
E15 ♥ E32
E16 ♥ D12
E16 ♥ D13
E16 ♥ D55
E16 ♥ E07
E16 ♥ E31
E16 ♥ F24
E16 ♥ F35
E17 ♥ E06
E17 ♥ E12
E18 ♥ D18
E18 ♥ D43
E18 ♥ D44
E18 ♥ D45
E18 ♥ E23
E18 ♥ E28
E18 ♥ E29
E19 ♥ D09
E19 ♥ D10
E19 ♥ D32
E19 ♥ D33
E19 ♥ D42
E19 ♥ D43
E19 ♥ E04
E19 ♥ F32
E19 ♥ F33
E20 ♥ A00
E20 ♥ D62
E20 ♥ E03
E20 ♥ E15
E20 ♥ L18
E21 ♥ A01
E21 ♥ D42
E21 ♥ D61
E21 ♥ D62
E21 ♥ E14
E21 ♥ E26
E22 ♥ D06
E22 ♥ D07
E22 ♥ D29
E22 ♥ D30
E22 ♥ D39
E22 ♥ D40
E22 ♥ D60
E22 ♥ D61
E22 ♥ E01
E22 ♥ E24
E22 ♥ E25
E22 ♥ F29
E22 ♥ F30
E23 ♥ D19
E23 ♥ D39
E23 ♥ D60
E23 ♥ E00
E23 ♥ E18
E24 ♥ E22
E24 ♥ E29
E24 ♥ E47
E25 ♥ E22
E25 ♥ E46
E26 ♥ A00
E26 ♥ E21
E26 ♥ E33
E27 ♥ A01
E27 ♥ E32
E27 ♥ E44
E28 ♥ E18
E28 ♥ E43
E29 ♥ D60
E29 ♥ E18
E29 ♥ E24
E30 ♥ E35
E30 ♥ E41
E31 ♥ D44
E31 ♥ E15
E31 ♥ E16
E31 ♥ E40
E32 ♥ A00
E32 ♥ D43
E32 ♥ D44
E32 ♥ E14
E32 ♥ E15
E32 ♥ E27
E32 ♥ E39
E33 ♥ A01
E33 ♥ E26
E33 ♥ E38
E33 ♥ L17
E34 ♥ E12
E34 ♥ E37
E34 ♥ E38

E35 ♥ D59
E35 ♥ E12
E35 ♥ E30
E36 ♥ D52
E36 ♥ D58
E36 ♥ D59
E36 ♥ E10
E36 ♥ E11
E36 ♥ E41
E37 ♥ D57
E37 ♥ D58
E37 ♥ E34
E37 ♥ F02
E37 ♥ F03
E38 ♥ A00
E38 ♥ D56
E38 ♥ D57
E38 ♥ E08
E38 ♥ E33
E38 ♥ E34
E38 ♥ E45
E39 ♥ A01
E39 ♥ D55
E39 ♥ D56
E39 ♥ E07
E39 ♥ E08
E39 ♥ E32
E39 ♥ E44
E40 ♥ D55
E40 ♥ E07
E40 ♥ E31
E40 ♥ F00
E40 ♥ F11
E41 ♥ E30
E41 ♥ E36
E42 ♥ D18
E42 ♥ D32
E42 ♥ D33
E42 ♥ D50
E42 ♥ E04
E42 ♥ E05
E42 ♥ E47
E43 ♥ D31
E43 ♥ D32
E43 ♥ D43
E43 ♥ D44
E43 ♥ E28
E43 ♥ F08
E43 ♥ F09
E44 ♥ A00
E44 ♥ D62
E44 ♥ E27
E44 ♥ E39
E44 ♥ L18
E45 ♥ A01
E45 ♥ D31
E45 ♥ D61
E45 ♥ D62
E45 ♥ E02
E45 ♥ E38
E46 ♥ D28
E46 ♥ D29
E46 ♥ D40
E46 ♥ D41
E46 ♥ D60
E46 ♥ D61
E46 ♥ E00
E46 ♥ E01
E46 ♥ E25
E46 ♥ F05
E46 ♥ F06
E47 ♥ D19
E47 ♥ D28
E47 ♥ D60
E47 ♥ E24
E47 ♥ E42

E

F00 ♥ C12
F00 ♥ D21
F00 ♥ E40
F00 ♥ F05
F00 ♥ F23
F00 ♥ F46
F00 ♥ F51
F00 ♥ G00
F00 ♥ G04
F00 ♥ G08
F00 ♥ G12
F00 ♥ G16
F00 ♥ G20
F00 ♥ G23
F00 ♥ G24
F00 ♥ G28
F00 ♥ G32
F00 ♥ G36
F00 ♥ G40
F00 ♥ G44
F00 ♥ H03
F00 ♥ H04
F00 ♥ H07
F00 ♥ H11
F00 ♥ H15
F00 ♥ H17
F00 ♥ I00
F00 ♥ I01
F00 ♥ I02
F00 ♥ I03
F00 ♥ I04
F00 ♥ I05
F00 ♥ I06
F00 ♥ I07
F00 ♥ I08
F00 ♥ I09
F00 ♥ I10
F00 ♥ I11
F00 ♥ I12
F00 ♥ I13
F00 ♥ I14
F00 ♥ J00
F00 ♥ J01
F01 ♥ C11
F01 ♥ D21
F01 ♥ F22
F01 ♥ F46
F01 ♥ G22
F01 ♥ I00
F01 ♥ I01
F01 ♥ I02
F01 ♥ I03
F01 ♥ I04
F01 ♥ I05
F01 ♥ I06
F01 ♥ I07
F01 ♥ I08
F01 ♥ I09
F01 ♥ I10
F01 ♥ I11
F01 ♥ I12
F01 ♥ I13
F01 ♥ I14
F02 ♥ B03
F02 ♥ B04
F02 ♥ C10
F02 ♥ D20
F02 ♥ E37
F02 ♥ F09
F02 ♥ F45
F02 ♥ G21
F02 ♥ I00
F02 ♥ I01
F02 ♥ I02
F02 ♥ I03
F02 ♥ I04
F02 ♥ I05
F02 ♥ I06
F02 ♥ I07
F02 ♥ I08
F02 ♥ I09
F02 ♥ I10
F02 ♥ I11
F02 ♥ I12
F02 ♥ I13
F02 ♥ I14
F03 ♥ B02
F03 ♥ C09
F03 ♥ D18
F03 ♥ E37
F03 ♥ F08
F03 ♥ F20
F03 ♥ F43
F03 ♥ F44
F03 ♥ G20
F03 ♥ I00
F03 ♥ I01
F03 ♥ I02
F03 ♥ I03
F03 ♥ I04
F03 ♥ I05
F03 ♥ I06
F03 ♥ I07
F03 ♥ I08
F03 ♥ I09
F03 ♥ I10
F03 ♥ I11
F03 ♥ I12
F03 ♥ I13
F03 ♥ I14
F04 ♥ B02
F04 ♥ C14
F04 ♥ F19
F04 ♥ F42
F04 ♥ G19
F04 ♥ I00
F04 ♥ I01
F04 ♥ I02
F04 ♥ I03
F04 ♥ I04
F04 ♥ I05
F04 ♥ I06
F04 ♥ I07
F04 ♥ I08
F04 ♥ I09
F04 ♥ I10
F04 ♥ I11
F04 ♥ I12
F04 ♥ I13
F04 ♥ I14
F05 ♥ B01
F05 ♥ C13
F05 ♥ E46
F05 ♥ F00
F05 ♥ F42
F05 ♥ G02
F05 ♥ G06
F05 ♥ G10
F05 ♥ G14
F05 ♥ G18
F05 ♥ G22
F05 ♥ G26
F05 ♥ G30
F05 ♥ G34
F05 ♥ G38
F05 ♥ G42
F05 ♥ G46
F05 ♥ H02
F05 ♥ H06
F05 ♥ H10
F05 ♥ H14
F05 ♥ I00
F05 ♥ I01
F05 ♥ I02
F05 ♥ I03
F05 ♥ I04
F05 ♥ I05
F05 ♥ I06
F05 ♥ I07
F05 ♥ I08
F05 ♥ I09
F05 ♥ I10
F05 ♥ I11
F05 ♥ I12
F05 ♥ I13
F05 ♥ I14
F05 ♥ J00
F06 ♥ C12
F06 ♥ E46
F06 ♥ F11
F06 ♥ F17
F06 ♥ F50
F06 ♥ F51
F06 ♥ G01
F06 ♥ G02
F06 ♥ G06
F06 ♥ G10
F06 ♥ G13
F06 ♥ G14
F06 ♥ G17
F06 ♥ G18
F06 ♥ G22
F06 ♥ G25
F06 ♥ G26
F06 ♥ G30
F06 ♥ G34
F06 ♥ G37
F06 ♥ G38
F06 ♥ G42
F06 ♥ G46
F06 ♥ H01
F06 ♥ H05
F06 ♥ H09
F06 ♥ H13
F06 ♥ H19
F06 ♥ I00
F06 ♥ I01
F06 ♥ I02
F06 ♥ I03
F06 ♥ I04
F06 ♥ I05
F06 ♥ I06
F06 ♥ I07
F06 ♥ I08
F06 ♥ I09
F06 ♥ I10
F06 ♥ I11
F06 ♥ I12
F06 ♥ I13
F06 ♥ I14
F07 ♥ C11
F07 ♥ F16
F07 ♥ F39
F07 ♥ F51
F07 ♥ G16
F07 ♥ I00
F07 ♥ I01
F07 ♥ I02
F07 ♥ I03
F07 ♥ I04
F07 ♥ I05
F07 ♥ I06
F07 ♥ I07
F07 ♥ I08
F07 ♥ I09
F07 ♥ I10
F07 ♥ I11
F07 ♥ I12
F07 ♥ I13
F07 ♥ I14
F08 ♥ C10
F08 ♥ E43
F08 ♥ F03
F08 ♥ F15
F08 ♥ F39
F08 ♥ G15
F08 ♥ I00
F08 ♥ I01
F08 ♥ I02
F08 ♥ I03
F08 ♥ I04
F08 ♥ I05
F08 ♥ I06
F08 ♥ I07
F08 ♥ I08
F08 ♥ I09
F08 ♥ I10
F08 ♥ I11
F08 ♥ I12
F08 ♥ I13
F08 ♥ I14
F09 ♥ C09
F09 ♥ E43
F09 ♥ F02
F09 ♥ F14
F09 ♥ G14
F09 ♥ I00
F09 ♥ I01
F09 ♥ I02
F09 ♥ I03
F09 ♥ I04
F09 ♥ I05
F09 ♥ I06
F09 ♥ I07
F09 ♥ I08
F09 ♥ I09
F09 ♥ I10
F09 ♥ I11
F09 ♥ I12
F09 ♥ I13
F09 ♥ I14
F10 ♥ C14
F10 ♥ F13
F10 ♥ F14
F10 ♥ F36
F10 ♥ G13
F10 ♥ I00
F10 ♥ I01
F10 ♥ I02
F10 ♥ I03
F10 ♥ I04
F10 ♥ I05
F10 ♥ I06
F10 ♥ I07
F10 ♥ I08
F10 ♥ I09
F10 ♥ I10
F10 ♥ I11
F10 ♥ I12
F10 ♥ I13
F10 ♥ I14
F11 ♥ C13
F11 ♥ E40
F11 ♥ F06
F11 ♥ F36
F11 ♥ G00
F11 ♥ G04
F11 ♥ G08
F11 ♥ G12
F11 ♥ G16
F11 ♥ G20
F11 ♥ G24
F11 ♥ G28
F11 ♥ G32
F11 ♥ G36
F11 ♥ G40
F11 ♥ G44
F11 ♥ H00
F11 ♥ H04
F11 ♥ H08
F11 ♥ H12
F11 ♥ H16
F11 ♥ H18
F11 ♥ I00
F11 ♥ I01
F11 ♥ I02
F11 ♥ I03
F11 ♥ I04
F11 ♥ I05
F11 ♥ I06
F11 ♥ I07
F11 ♥ I08
F11 ♥ I09
F11 ♥ I10
F11 ♥ I11
F11 ♥ I12
F11 ♥ I13
F11 ♥ I14
F12 ♥ F17
F12 ♥ F34
F12 ♥ F35
F12 ♥ F51
F12 ♥ G00
F12 ♥ G04
F12 ♥ G08
F12 ♥ G12
F12 ♥ G16
F12 ♥ G20
F12 ♥ G24
F12 ♥ G28
F12 ♥ G32
F12 ♥ G34
F12 ♥ G35
F12 ♥ G36
F12 ♥ G40
F12 ♥ G44
F12 ♥ H03
F12 ♥ H04
F12 ♥ H07
F12 ♥ H11
F12 ♥ H15
F12 ♥ H17
F12 ♥ I00
F12 ♥ I01
F12 ♥ I02
F12 ♥ I03
F12 ♥ I04
F12 ♥ I05
F12 ♥ I06
F12 ♥ I07
F12 ♥ I08
F12 ♥ I09
F12 ♥ I10
F12 ♥ I11
F12 ♥ I12
F12 ♥ I13
F12 ♥ I14
F13 ♥ F10
F13 ♥ G33
F13 ♥ I00
F13 ♥ I01
F13 ♥ I02
F13 ♥ I03
F13 ♥ I04
F13 ♥ I05
F13 ♥ I06
F13 ♥ I07
F13 ♥ I08
F13 ♥ I09
F13 ♥ I10
F13 ♥ I11
F13 ♥ I12
F13 ♥ I13
F13 ♥ I14
F14 ♥ F09
F14 ♥ F10
F14 ♥ F21
F14 ♥ G32
F14 ♥ I00
F14 ♥ I01
F14 ♥ I02
F14 ♥ I03
F14 ♥ I04
F14 ♥ I05
F14 ♥ I06
F14 ♥ I07
F14 ♥ I08
F14 ♥ I09
F14 ♥ I10
F14 ♥ I11
F14 ♥ I12
F14 ♥ I13
F14 ♥ I14
F15 ♥ F08
F15 ♥ F20
F15 ♥ F31
F15 ♥ F32
F15 ♥ G31
F15 ♥ I00
F15 ♥ I01
F15 ♥ I02
F15 ♥ I03
F15 ♥ I04
F15 ♥ I05
F15 ♥ I06
F15 ♥ I07
F15 ♥ I08
F15 ♥ I09
F15 ♥ I10
F15 ♥ I11
F15 ♥ I12
F15 ♥ I13
F15 ♥ I14
F16 ♥ F07
F16 ♥ G30
F16 ♥ I00
F16 ♥ I01
F16 ♥ I02
F16 ♥ I03
F16 ♥ I04
F16 ♥ I05
F16 ♥ I06
F16 ♥ I07
F16 ♥ I08
F16 ♥ I09
F16 ♥ I10
F16 ♥ I11
F16 ♥ I12
F16 ♥ I13
F16 ♥ I14
F17 ♥ F06
F17 ♥ F12
F17 ♥ F51
F17 ♥ G02
F17 ♥ G06
F17 ♥ G10
F17 ♥ G14
F17 ♥ G18
F17 ♥ G22
F17 ♥ G26
F17 ♥ G29
F17 ♥ G30
F17 ♥ G34
F17 ♥ G38
F17 ♥ G42
F17 ♥ G46
F17 ♥ H02
F17 ♥ H06
F17 ♥ H10
F17 ♥ H14
F17 ♥ I00
F17 ♥ I01
F17 ♥ I02
F17 ♥ I03
F17 ♥ I04
F17 ♥ I05
F17 ♥ I06
F17 ♥ I07
F17 ♥ I08
F17 ♥ I09
F17 ♥ I10
F17 ♥ I11
F17 ♥ I12
F17 ♥ I13
F17 ♥ I14
F18 ♥ F23
F18 ♥ F28
F18 ♥ F29
F18 ♥ F51
F18 ♥ G01
F18 ♥ G02
F18 ♥ G06
F18 ♥ G10
F18 ♥ G13
F18 ♥ G14
F18 ♥ G18
F18 ♥ G22
F18 ♥ G25
F18 ♥ G26
F18 ♥ G28
F18 ♥ G30
F18 ♥ G34
F18 ♥ G37
F18 ♥ G38
F18 ♥ G42
F18 ♥ G46
F18 ♥ H01
F18 ♥ H05
F18 ♥ H09
F18 ♥ H13
F18 ♥ H19
F18 ♥ I00
F18 ♥ I01
F18 ♥ I02
F18 ♥ I03
F18 ♥ I04
F18 ♥ I05
F18 ♥ I06
F18 ♥ I07
F18 ♥ I08
F18 ♥ I09
F18 ♥ I10
F18 ♥ I11
F18 ♥ I12
F18 ♥ I13
F18 ♥ I14
F19 ♥ F04
F19 ♥ F27
F19 ♥ G27
F19 ♥ I00
F19 ♥ I01
F19 ♥ I02
F19 ♥ I03
F19 ♥ I04
F19 ♥ I05
F19 ♥ I06
F19 ♥ I07
F19 ♥ I08
F19 ♥ I09
F19 ♥ I10
F19 ♥ I11
F19 ♥ I12
F19 ♥ I13
F19 ♥ I14
F20 ♥ F03
F20 ♥ F15
F20 ♥ F27
F20 ♥ G26
F20 ♥ I00
F20 ♥ I01
F20 ♥ I02
F20 ♥ I03
F20 ♥ I04
F20 ♥ I05
F20 ♥ I06
F20 ♥ I07
F20 ♥ I08
F20 ♥ I09
F20 ♥ I10
F20 ♥ I11

F20 ♥ I12	F23 ♥ I08	F26 ♥ F21	F29 ♥ G38	F31 ♥ I02	F34 ♥ I14	F36 ♥ I09	F40 ♥ I06	F42 ♥ I00
F20 ♥ I13	F23 ♥ I09	F26 ♥ F33	F29 ♥ G42	F31 ♥ I03	F35 ♥ E16	F36 ♥ I10	F40 ♥ I07	F42 ♥ I01
F20 ♥ I14	F23 ♥ I10	F26 ♥ G45	F29 ♥ G46	F31 ♥ I04	F35 ♥ F12	F36 ♥ I11	F40 ♥ I08	F42 ♥ I02
F21 ♥ F14	F23 ♥ I11	F26 ♥ I00	F29 ♥ H02	F31 ♥ I05	F35 ♥ F30	F36 ♥ I12	F40 ♥ I09	F42 ♥ I03
F21 ♥ F26	F23 ♥ I12	F26 ♥ I01	F29 ♥ H06	F31 ♥ I06	F35 ♥ G00	F36 ♥ I13	F40 ♥ I10	F42 ♥ I04
F21 ♥ G25	F23 ♥ I13	F26 ♥ I02	F29 ♥ H10	F31 ♥ I07	F35 ♥ G04	F36 ♥ I14	F40 ♥ I11	F42 ♥ I05
F21 ♥ I00	F23 ♥ I14	F26 ♥ I03	F29 ♥ H14	F31 ♥ I08	F35 ♥ G08	F37 ♥ F34	F40 ♥ I12	F42 ♥ I06
F21 ♥ I01	F24 ♥ D21	F26 ♥ I04	F29 ♥ I00	F31 ♥ I09	F35 ♥ G12	F37 ♥ G09	F40 ♥ I13	F42 ♥ I07
F21 ♥ I02	F24 ♥ E16	F26 ♥ I05	F29 ♥ I01	F31 ♥ I10	F35 ♥ G16	F37 ♥ I00	F40 ♥ I14	F42 ♥ I08
F21 ♥ I03	F24 ♥ F22	F26 ♥ I06	F29 ♥ I02	F31 ♥ I11	F35 ♥ G20	F37 ♥ I01	F41 ♥ F30	F42 ♥ I09
F21 ♥ I04	F24 ♥ F29	F26 ♥ I07	F29 ♥ I03	F31 ♥ I12	F35 ♥ G24	F37 ♥ I02	F41 ♥ F36	F42 ♥ I10
F21 ♥ I05	F24 ♥ F47	F26 ♥ I08	F29 ♥ I04	F31 ♥ I13	F35 ♥ G28	F37 ♥ I03	F41 ♥ F51	F42 ♥ I11
F21 ♥ I06	F24 ♥ F51	F26 ♥ I09	F29 ♥ I05	F31 ♥ I14	F35 ♥ G32	F37 ♥ I04	F41 ♥ G02	F42 ♥ I12
F21 ♥ I07	F24 ♥ G00	F26 ♥ I10	F29 ♥ I06	F32 ♥ E19	F35 ♥ G36	F37 ♥ I05	F41 ♥ G05	F42 ♥ I13
F21 ♥ I08	F24 ♥ G04	F26 ♥ I11	F29 ♥ I07	F32 ♥ F15	F35 ♥ G40	F37 ♥ I06	F41 ♥ G06	F42 ♥ I14
F21 ♥ I09	F24 ♥ G08	F26 ♥ I12	F29 ♥ I08	F32 ♥ F27	F35 ♥ G44	F37 ♥ I07	F41 ♥ G10	F43 ♥ F03
F21 ♥ I10	F24 ♥ G12	F26 ♥ I13	F29 ♥ I09	F32 ♥ F39	F35 ♥ H00	F37 ♥ I08	F41 ♥ G14	F43 ♥ F28
F21 ♥ I11	F24 ♥ G16	F26 ♥ I14	F29 ♥ I10	F32 ♥ G39	F35 ♥ H04	F37 ♥ I09	F41 ♥ G18	F43 ♥ G03
F21 ♥ I12	F24 ♥ G20	F27 ♥ D18	F29 ♥ I11	F32 ♥ I00	F35 ♥ H08	F37 ♥ I10	F41 ♥ G22	F43 ♥ I00
F21 ♥ I13	F24 ♥ G24	F27 ♥ E13	F29 ♥ I12	F32 ♥ I01	F35 ♥ H12	F37 ♥ I11	F41 ♥ G26	F43 ♥ I01
F21 ♥ I14	F24 ♥ G28	F27 ♥ F19	F29 ♥ I13	F32 ♥ I02	F35 ♥ H16	F37 ♥ I12	F41 ♥ G30	F43 ♥ I02
F22 ♥ F01	F24 ♥ G32	F27 ♥ F20	F29 ♥ I14	F32 ♥ I03	F35 ♥ H18	F37 ♥ I13	F41 ♥ G34	F43 ♥ I03
F22 ♥ F24	F24 ♥ G36	F27 ♥ F32	F29 ♥ J00	F32 ♥ I04	F35 ♥ I00	F37 ♥ I14	F41 ♥ G38	F43 ♥ I04
F22 ♥ F25	F24 ♥ G40	F27 ♥ F44	F30 ♥ E22	F32 ♥ I05	F35 ♥ I01	F38 ♥ F33	F41 ♥ G42	F43 ♥ I05
F22 ♥ F49	F24 ♥ G44	F27 ♥ G44	F30 ♥ F35	F32 ♥ I06	F35 ♥ I02	F38 ♥ F34	F41 ♥ G46	F43 ♥ I06
F22 ♥ G24	F24 ♥ G47	F27 ♥ I00	F30 ♥ F41	F32 ♥ I07	F35 ♥ I03	F38 ♥ F45	F41 ♥ H02	F43 ♥ I07
F22 ♥ I00	F24 ♥ H03	F27 ♥ I01	F30 ♥ F51	F32 ♥ I08	F35 ♥ I04	F38 ♥ G08	F41 ♥ H06	F43 ♥ I08
F22 ♥ I01	F24 ♥ H04	F27 ♥ I02	F30 ♥ G01	F32 ♥ I09	F35 ♥ I05	F38 ♥ I00	F41 ♥ H10	F43 ♥ I09
F22 ♥ I02	F24 ♥ H07	F27 ♥ I03	F30 ♥ G02	F32 ♥ I10	F35 ♥ I06	F38 ♥ I01	F41 ♥ H14	F43 ♥ I10
F22 ♥ I03	F24 ♥ H11	F27 ♥ I04	F30 ♥ G06	F32 ♥ I11	F35 ♥ I07	F38 ♥ I02	F41 ♥ I00	F43 ♥ I11
F22 ♥ I04	F24 ♥ H15	F27 ♥ I05	F30 ♥ G10	F32 ♥ I12	F35 ♥ I08	F38 ♥ I03	F41 ♥ I01	F43 ♥ I12
F22 ♥ I05	F24 ♥ H17	F27 ♥ I06	F30 ♥ G13	F32 ♥ I13	F35 ♥ I09	F38 ♥ I04	F41 ♥ I02	F43 ♥ I13
F22 ♥ I06	F24 ♥ I00	F27 ♥ I07	F30 ♥ G14	F32 ♥ I14	F35 ♥ I10	F38 ♥ I05	F41 ♥ I03	F43 ♥ I14
F22 ♥ I07	F24 ♥ I01	F27 ♥ I08	F30 ♥ G18	F33 ♥ E19	F35 ♥ I11	F38 ♥ I06	F41 ♥ I04	F44 ♥ F03
F22 ♥ I08	F24 ♥ I02	F27 ♥ I09	F30 ♥ G22	F33 ♥ F26	F35 ♥ I12	F38 ♥ I07	F41 ♥ I05	F44 ♥ F27
F22 ♥ I09	F24 ♥ I03	F27 ♥ I10	F30 ♥ G25	F33 ♥ F38	F35 ♥ I13	F38 ♥ I08	F41 ♥ I06	F44 ♥ F39
F22 ♥ I10	F24 ♥ I04	F27 ♥ I11	F30 ♥ G26	F33 ♥ G38	F35 ♥ I14	F38 ♥ I09	F41 ♥ I07	F44 ♥ G02
F22 ♥ I11	F24 ♥ I05	F27 ♥ I12	F30 ♥ G30	F33 ♥ I00	F36 ♥ F10	F38 ♥ I10	F41 ♥ I08	F44 ♥ I00
F22 ♥ I12	F24 ♥ I06	F27 ♥ I13	F30 ♥ G34	F33 ♥ I01	F36 ♥ F11	F38 ♥ I11	F41 ♥ I09	F44 ♥ I01
F22 ♥ I13	F24 ♥ I07	F27 ♥ I14	F30 ♥ G37	F33 ♥ I02	F36 ♥ F41	F38 ♥ I12	F41 ♥ I10	F44 ♥ I02
F22 ♥ I14	F24 ♥ I08	F28 ♥ F18	F30 ♥ G38	F33 ♥ I03	F36 ♥ F51	F38 ♥ I13	F41 ♥ I11	F44 ♥ I03
F23 ♥ F00	F24 ♥ I09	F28 ♥ F43	F30 ♥ G41	F33 ♥ I04	F36 ♥ G00	F38 ♥ I14	F41 ♥ I12	F44 ♥ I04
F23 ♥ F18	F24 ♥ I10	F28 ♥ G43	F30 ♥ G42	F33 ♥ I05	F36 ♥ G04	F39 ♥ F07	F41 ♥ I13	F44 ♥ I05
F23 ♥ F49	F24 ♥ I11	F28 ♥ I00	F30 ♥ G46	F33 ♥ I06	F36 ♥ G08	F39 ♥ F08	F41 ♥ I14	F44 ♥ I06
F23 ♥ G00	F24 ♥ I12	F28 ♥ I01	F30 ♥ H01	F33 ♥ I07	F36 ♥ G10	F39 ♥ F32	F42 ♥ F04	F44 ♥ I07
F23 ♥ G04	F24 ♥ I13	F28 ♥ I02	F30 ♥ H05	F33 ♥ I08	F36 ♥ G11	F39 ♥ F44	F42 ♥ F05	F44 ♥ I08
F23 ♥ G08	F24 ♥ I14	F28 ♥ I03	F30 ♥ H09	F33 ♥ I09	F36 ♥ G12	F39 ♥ G07	F42 ♥ F47	F44 ♥ I09
F23 ♥ G12	F24 ♥ J00	F28 ♥ I04	F30 ♥ H13	F33 ♥ I10	F36 ♥ G16	F39 ♥ I00	F42 ♥ F51	F44 ♥ I10
F23 ♥ G16	F24 ♥ J01	F28 ♥ I05	F30 ♥ H19	F33 ♥ I11	F36 ♥ G20	F39 ♥ I01	F42 ♥ G01	F44 ♥ I11
F23 ♥ G20	F25 ♥ D21	F28 ♥ I06	F30 ♥ I00	F33 ♥ I12	F36 ♥ G24	F39 ♥ I02	F42 ♥ G02	F44 ♥ I12
F23 ♥ G24	F25 ♥ F22	F28 ♥ I07	F30 ♥ I01	F33 ♥ I13	F36 ♥ G28	F39 ♥ I03	F42 ♥ G04	F44 ♥ I13
F23 ♥ G28	F25 ♥ F46	F28 ♥ I08	F30 ♥ I02	F33 ♥ I14	F36 ♥ G32	F39 ♥ I04	F42 ♥ G06	F44 ♥ I14
F23 ♥ G32	F25 ♥ G46	F28 ♥ I09	F30 ♥ I03	F34 ♥ F12	F36 ♥ G36	F39 ♥ I05	F42 ♥ G10	F45 ♥ F02
F23 ♥ G36	F25 ♥ I00	F28 ♥ I10	F30 ♥ I04	F34 ♥ F37	F36 ♥ G40	F39 ♥ I06	F42 ♥ G13	F45 ♥ F38
F23 ♥ G40	F25 ♥ I01	F28 ♥ I11	F30 ♥ I05	F34 ♥ F38	F36 ♥ G44	F39 ♥ I07	F42 ♥ G14	F45 ♥ G01
F23 ♥ G44	F25 ♥ I02	F28 ♥ I12	F30 ♥ I06	F34 ♥ G37	F36 ♥ H03	F39 ♥ I08	F42 ♥ G18	F45 ♥ I00
F23 ♥ H00	F25 ♥ I03	F28 ♥ I13	F30 ♥ I07	F34 ♥ I00	F36 ♥ H04	F39 ♥ I09	F42 ♥ G22	F45 ♥ I01
F23 ♥ H04	F25 ♥ I04	F28 ♥ I14	F30 ♥ I08	F34 ♥ I01	F36 ♥ H07	F39 ♥ I10	F42 ♥ G25	F45 ♥ I02
F23 ♥ H08	F25 ♥ I05	F29 ♥ E22	F30 ♥ I09	F34 ♥ I02	F36 ♥ H11	F39 ♥ I11	F42 ♥ G26	F45 ♥ I03
F23 ♥ H12	F25 ♥ I06	F29 ♥ F18	F30 ♥ I10	F34 ♥ I03	F36 ♥ H15	F39 ♥ I12	F42 ♥ G30	F45 ♥ I04
F23 ♥ H16	F25 ♥ I07	F29 ♥ F24	F30 ♥ I11	F34 ♥ I04	F36 ♥ H17	F39 ♥ I13	F42 ♥ G34	F45 ♥ I05
F23 ♥ H18	F25 ♥ I08	F29 ♥ G02	F30 ♥ I12	F34 ♥ I05	F36 ♥ I00	F39 ♥ I14	F42 ♥ G37	F45 ♥ I06
F23 ♥ I00	F25 ♥ I09	F29 ♥ G06	F30 ♥ I13	F34 ♥ I06	F36 ♥ I01	F40 ♥ F31	F42 ♥ G38	F45 ♥ I07
F23 ♥ I01	F25 ♥ I10	F29 ♥ G10	F30 ♥ I14	F34 ♥ I07	F36 ♥ I02	F40 ♥ G06	F42 ♥ G42	F45 ♥ I08
F23 ♥ I02	F25 ♥ I11	F29 ♥ G14	F31 ♥ F15	F34 ♥ I08	F36 ♥ I03	F40 ♥ I00	F42 ♥ G46	F45 ♥ I09
F23 ♥ I03	F25 ♥ I12	F29 ♥ G18	F31 ♥ F40	F34 ♥ I09	F36 ♥ I04	F40 ♥ I01	F42 ♥ H01	F45 ♥ I10
F23 ♥ I04	F25 ♥ I13	F29 ♥ G22	F31 ♥ F51	F34 ♥ I10	F36 ♥ I05	F40 ♥ I02	F42 ♥ H05	F45 ♥ I11
F23 ♥ I05	F25 ♥ I14	F29 ♥ G26	F31 ♥ G40	F34 ♥ I11	F36 ♥ I06	F40 ♥ I03	F42 ♥ H09	F45 ♥ I12
F23 ♥ I06	F26 ♥ D20	F29 ♥ G30	F31 ♥ I00	F34 ♥ I12	F36 ♥ I07	F40 ♥ I04	F42 ♥ H13	F45 ♥ I13
F23 ♥ I07	F26 ♥ E13	F29 ♥ G34	F31 ♥ I01	F34 ♥ I13	F36 ♥ I08	F40 ♥ I05	F42 ♥ H19	F45 ♥ I14

F46 ♥ F00	F51 ♥ G07	G09 ♥ F37	G20 ♥ F00	G30 ♥ F16	G40 ♥ F12	H03 ♥ F12	H14 ♥ F41	I00 ♥ F31
F46 ♥ F01	F51 ♥ G19	G09 ♥ G14	G20 ♥ F03	G30 ♥ F17	G40 ♥ F23	H03 ♥ F24	H14 ♥ H07	I00 ♥ F32
F46 ♥ F25	F51 ♥ G31	G10 ♥ F05	G20 ♥ F11	G30 ♥ F18	G40 ♥ F24	H03 ♥ F36	H15 ♥ F00	I00 ♥ F33
F46 ♥ G00	F51 ♥ G43	G10 ♥ F06	G20 ♥ F12	G30 ♥ F29	G40 ♥ F31	H03 ♥ H01	H15 ♥ F12	I00 ♥ F34
F46 ♥ I00	G00 ♥ F00	G10 ♥ F17	G20 ♥ F23	G30 ♥ F30	G40 ♥ F35	H04 ♥ F00	H15 ♥ F24	I00 ♥ F35
F46 ♥ I01	G00 ♥ F11	G10 ♥ F18	G20 ♥ F24	G30 ♥ F41	G40 ♥ F36	H04 ♥ F11	H15 ♥ F36	I00 ♥ F36
F46 ♥ I02	G00 ♥ F12	G10 ♥ F29	G20 ♥ F35	G30 ♥ F42	G40 ♥ F47	H04 ♥ F12	H16 ♥ F11	I00 ♥ F37
F46 ♥ I03	G00 ♥ F23	G10 ♥ F30	G20 ♥ F36	G30 ♥ G16	G40 ♥ G06	H04 ♥ F23	H16 ♥ F23	I00 ♥ F38
F46 ♥ I04	G00 ♥ F24	G10 ♥ F36	G20 ♥ F47	G30 ♥ G41	G41 ♥ F30	H04 ♥ F24	H16 ♥ F35	I00 ♥ F39
F46 ♥ I05	G00 ♥ F35	G10 ♥ F41	G20 ♥ G26	G31 ♥ F15	G41 ♥ G30	H04 ♥ F35	H16 ♥ F47	I00 ♥ F40
F46 ♥ I06	G00 ♥ F36	G10 ♥ F42	G21 ♥ F02	G31 ♥ F51	G42 ♥ F05	H04 ♥ F36	H16 ♥ H11	I00 ♥ F41
F46 ♥ I07	G00 ♥ F46	G10 ♥ G13	G21 ♥ G02	G32 ♥ F00	G42 ♥ F06	H04 ♥ F47	H16 ♥ I00	I00 ♥ F42
F46 ♥ I08	G00 ♥ F47	G10 ♥ G36	G21 ♥ H02	G32 ♥ F11	G42 ♥ F17	H04 ♥ H01	H16 ♥ I02	I00 ♥ F43
F46 ♥ I09	G00 ♥ G23	G11 ♥ F36	G22 ♥ F01	G32 ♥ F12	G42 ♥ F18	H04 ♥ I18	H16 ♥ I04	I00 ♥ F44
F46 ♥ I10	G00 ♥ G46	G11 ♥ G13	G22 ♥ F05	G32 ♥ F14	G42 ♥ F29	H05 ♥ F06	H17 ♥ F00	I00 ♥ F45
F46 ♥ I11	G00 ♥ G48	G11 ♥ G36	G22 ♥ F06	G32 ♥ F23	G42 ♥ F30	H05 ♥ F18	H17 ♥ F12	I00 ♥ F46
F46 ♥ I12	G01 ♥ F06	G12 ♥ F00	G22 ♥ F17	G32 ♥ F24	G42 ♥ F41	H05 ♥ F30	H17 ♥ F24	I00 ♥ F47
F46 ♥ I13	G01 ♥ F18	G12 ♥ F11	G22 ♥ F18	G32 ♥ F35	G42 ♥ F42	H05 ♥ F42	H17 ♥ F36	I00 ♥ H10
F46 ♥ I14	G01 ♥ F30	G12 ♥ F12	G22 ♥ F29	G32 ♥ F36	G42 ♥ G04	H05 ♥ H10	H17 ♥ H22	I00 ♥ H16
F47 ♥ F24	G01 ♥ F42	G12 ♥ F23	G22 ♥ F30	G32 ♥ F47	G43 ♥ F28	H05 ♥ I01	H17 ♥ I03	I00 ♥ I07
F47 ♥ F42	G01 ♥ F45	G12 ♥ F24	G22 ♥ F41	G32 ♥ G14	G43 ♥ F51	H05 ♥ I03	H18 ♥ F11	I00 ♥ I08
F47 ♥ G00	G01 ♥ G46	G12 ♥ F35	G22 ♥ F42	G32 ♥ G39	G43 ♥ G28	H05 ♥ I05	H18 ♥ F23	I01 ♥ F00
F47 ♥ G04	G02 ♥ F05	G12 ♥ F36	G22 ♥ G24	G33 ♥ F13	G44 ♥ F00	H06 ♥ F05	H18 ♥ F35	I01 ♥ F01
F47 ♥ G08	G02 ♥ F06	G12 ♥ F47	G22 ♥ G25	G33 ♥ G38	G44 ♥ F11	H06 ♥ F17	H18 ♥ F47	I01 ♥ F02
F47 ♥ G12	G02 ♥ F17	G12 ♥ G34	G23 ♥ F00	G34 ♥ F05	G44 ♥ F12	H06 ♥ F29	H18 ♥ H20	I01 ♥ F03
F47 ♥ G16	G02 ♥ F18	G12 ♥ G35	G23 ♥ G00	G34 ♥ F06	G44 ♥ F23	H06 ♥ F41	H18 ♥ H21	I01 ♥ F04
F47 ♥ G20	G02 ♥ F29	G13 ♥ F06	G24 ♥ F00	G34 ♥ F12	G44 ♥ F24	H07 ♥ F00	H19 ♥ F06	I01 ♥ F05
F47 ♥ G24	G02 ♥ F30	G13 ♥ F10	G24 ♥ F11	G34 ♥ F17	G44 ♥ F27	H07 ♥ F12	H19 ♥ F18	I01 ♥ F06
F47 ♥ G28	G02 ♥ F41	G13 ♥ F18	G24 ♥ F12	G34 ♥ F18	G44 ♥ F35	H07 ♥ F24	H19 ♥ F30	I01 ♥ F07
F47 ♥ G32	G02 ♥ F42	G13 ♥ F30	G24 ♥ F22	G34 ♥ F29	G44 ♥ F36	H07 ♥ F36	H19 ♥ F42	I01 ♥ F08
F47 ♥ G36	G02 ♥ F44	G13 ♥ F42	G24 ♥ F23	G34 ♥ F30	G44 ♥ F47	H07 ♥ H02	H19 ♥ H08	I01 ♥ F09
F47 ♥ G40	G02 ♥ G21	G13 ♥ G10	G24 ♥ F24	G34 ♥ F41	G44 ♥ G02	H07 ♥ H14	H20 ♥ H18	I01 ♥ F10
F47 ♥ G44	G02 ♥ G44	G13 ♥ G11	G24 ♥ F35	G34 ♥ F42	G45 ♥ F26	H08 ♥ F11	H21 ♥ H18	I01 ♥ F11
F47 ♥ H00	G03 ♥ F43	G14 ♥ F05	G24 ♥ F36	G34 ♥ G12	G45 ♥ G26	H08 ♥ F23	H21 ♥ H23	I01 ♥ F12
F47 ♥ H04	G04 ♥ F00	G14 ♥ F06	G24 ♥ F47	G34 ♥ G37	G45 ♥ H02	H08 ♥ F35	H21 ♥ H24	I01 ♥ F13
F47 ♥ H08	G04 ♥ F11	G14 ♥ F09	G24 ♥ G22	G35 ♥ F12	G46 ♥ F05	H08 ♥ F47	H22 ♥ H17	I01 ♥ F14
F47 ♥ H12	G04 ♥ F12	G14 ♥ F17	G24 ♥ G47	G35 ♥ G12	G46 ♥ F06	H08 ♥ H13	H22 ♥ I02	I01 ♥ F15
F47 ♥ H16	G04 ♥ F23	G14 ♥ F18	G24 ♥ G48	G35 ♥ G37	G46 ♥ F17	H08 ♥ H19	H23 ♥ H21	I01 ♥ F16
F47 ♥ H18	G04 ♥ F24	G14 ♥ F29	G25 ♥ F06	G36 ♥ F00	G46 ♥ F18	H09 ♥ F06	H24 ♥ H21	I01 ♥ F17
F47 ♥ I00	G04 ♥ F35	G14 ♥ F30	G25 ♥ F18	G36 ♥ F11	G46 ♥ F25	H09 ♥ F18	I00 ♥ F00	I01 ♥ F18
F47 ♥ I01	G04 ♥ F36	G14 ♥ F41	G25 ♥ F21	G36 ♥ F12	G46 ♥ F29	H09 ♥ F30	I00 ♥ F01	I01 ♥ F19
F47 ♥ I02	G04 ♥ F42	G14 ♥ F42	G25 ♥ F30	G36 ♥ F23	G46 ♥ F30	H09 ♥ F42	I00 ♥ F02	I01 ♥ F20
F47 ♥ I03	G04 ♥ F47	G14 ♥ G09	G25 ♥ F42	G36 ♥ F24	G46 ♥ F41	H10 ♥ F05	I00 ♥ F03	I01 ♥ F21
F47 ♥ I04	G04 ♥ G19	G14 ♥ G32	G25 ♥ G22	G36 ♥ F35	G46 ♥ F42	H10 ♥ F17	I00 ♥ F04	I01 ♥ F22
F47 ♥ I05	G04 ♥ G42	G15 ♥ F08	G26 ♥ F05	G36 ♥ F36	G46 ♥ G00	H10 ♥ F29	I00 ♥ F05	I01 ♥ F23
F47 ♥ I06	G05 ♥ F41	G15 ♥ G08	G26 ♥ F06	G36 ♥ F47	G46 ♥ G01	H10 ♥ F41	I00 ♥ F06	I01 ♥ F24
F47 ♥ I07	G06 ♥ F05	G16 ♥ F00	G26 ♥ F17	G36 ♥ G10	G47 ♥ F24	H10 ♥ H05	I00 ♥ F07	I01 ♥ F25
F47 ♥ I08	G06 ♥ F06	G16 ♥ F07	G26 ♥ F18	G36 ♥ G11	G47 ♥ G24	H10 ♥ I00	I00 ♥ F08	I01 ♥ F26
F47 ♥ I09	G06 ♥ F17	G16 ♥ F11	G26 ♥ F20	G37 ♥ F06	G48 ♥ G00	H10 ♥ I02	I00 ♥ F09	I01 ♥ F27
F47 ♥ I10	G06 ♥ F18	G16 ♥ F12	G26 ♥ F29	G37 ♥ F18	G48 ♥ G24	H10 ♥ I04	I00 ♥ F10	I01 ♥ F28
F47 ♥ I11	G06 ♥ F29	G16 ♥ F23	G26 ♥ F30	G37 ♥ F30	H00 ♥ F11	H11 ♥ F00	I00 ♥ F11	I01 ♥ F29
F47 ♥ I12	G06 ♥ F30	G16 ♥ F24	G26 ♥ F41	G37 ♥ F34	H00 ♥ F23	H11 ♥ F12	I00 ♥ F12	I01 ♥ F30
F47 ♥ I13	G06 ♥ F40	G16 ♥ F35	G26 ♥ F42	G37 ♥ F42	H00 ♥ F35	H11 ♥ F24	I00 ♥ F13	I01 ♥ F31
F47 ♥ I14	G06 ♥ F41	G16 ♥ F36	G26 ♥ G20	G37 ♥ G34	H00 ♥ F47	H11 ♥ F36	I00 ♥ F14	I01 ♥ F32
F48 ♥ B04	G06 ♥ F42	G16 ♥ F47	G26 ♥ G45	G37 ♥ G35	H00 ♥ H02	H11 ♥ H16	I00 ♥ F15	I01 ♥ F33
F49 ♥ F22	G06 ♥ G17	G16 ♥ G30	G27 ♥ F19	G38 ♥ F05	H01 ♥ F06	H11 ♥ I01	I00 ♥ F16	I01 ♥ F34
F49 ♥ F23	G06 ♥ G40	G17 ♥ F06	G28 ♥ F00	G38 ♥ F06	H01 ♥ F18	H11 ♥ I03	I00 ♥ F17	I01 ♥ F35
F50 ♥ F06	G07 ♥ F39	G17 ♥ G06	G28 ♥ F11	G38 ♥ F17	H01 ♥ F30	H11 ♥ I05	I00 ♥ F18	I01 ♥ F36
F51 ♥ F00	G07 ♥ F51	G18 ♥ F05	G28 ♥ F12	G38 ♥ F18	H01 ♥ F42	H12 ♥ F11	I00 ♥ F19	I01 ♥ F37
F51 ♥ F06	G08 ♥ F00	G18 ♥ F06	G28 ♥ F18	G38 ♥ F29	H01 ♥ H03	H12 ♥ F23	I00 ♥ F20	I01 ♥ F38
F51 ♥ F07	G08 ♥ F11	G18 ♥ F17	G28 ♥ F23	G38 ♥ F30	H01 ♥ H04	H12 ♥ F35	I00 ♥ F21	I01 ♥ F39
F51 ♥ F12	G08 ♥ F12	G18 ♥ F18	G28 ♥ F24	G38 ♥ F33	H02 ♥ F05	H12 ♥ F47	I00 ♥ F22	I01 ♥ F40
F51 ♥ F17	G08 ♥ F23	G18 ♥ F29	G28 ♥ F35	G38 ♥ F41	H02 ♥ F17	H13 ♥ F06	I00 ♥ F23	I01 ♥ F41
F51 ♥ F18	G08 ♥ F24	G18 ♥ F30	G28 ♥ F36	G38 ♥ F42	H02 ♥ F29	H13 ♥ F18	I00 ♥ F24	I01 ♥ F42
F51 ♥ F24	G08 ♥ F35	G18 ♥ F41	G28 ♥ F47	G38 ♥ G08	H02 ♥ F41	H13 ♥ F30	I00 ♥ F25	I01 ♥ F43
F51 ♥ F30	G08 ♥ F36	G18 ♥ F42	G28 ♥ G18	G38 ♥ G33	H02 ♥ G21	H13 ♥ F42	I00 ♥ F26	I01 ♥ F44
F51 ♥ F31	G08 ♥ F38	G18 ♥ G28	G28 ♥ G43	G39 ♥ F32	H02 ♥ G45	H13 ♥ H08	I00 ♥ F27	I01 ♥ F45
F51 ♥ F36	G08 ♥ F47	G19 ♥ F04	G29 ♥ F17	G39 ♥ G32	H02 ♥ H00	H14 ♥ F05	I00 ♥ F28	I01 ♥ F46
F51 ♥ F41	G08 ♥ G15	G19 ♥ F51	G30 ♥ F05	G40 ♥ F00	H02 ♥ H07	H14 ♥ F17	I00 ♥ F29	I01 ♥ F47
F51 ♥ F42	G08 ♥ G38	G19 ♥ G04	G30 ♥ F06	G40 ♥ F11	H03 ♥ F00	H14 ♥ F29	I00 ♥ F30	I01 ♥ H05

I01 ♥ H11
I01 ♥ I07
I01 ♥ I13
I01 ♥ I15
I01 ♥ I16
I01 ♥ I19
I01 ♥ J00
I01 ♥ J02
I02 ♥ F00
I02 ♥ F01
I02 ♥ F02
I02 ♥ F03
I02 ♥ F04
I02 ♥ F05
I02 ♥ F06
I02 ♥ F07
I02 ♥ F08
I02 ♥ F09
I02 ♥ F10
I02 ♥ F11
I02 ♥ F12
I02 ♥ F13
I02 ♥ F14
I02 ♥ F15
I02 ♥ F16
I02 ♥ F17
I02 ♥ F18
I02 ♥ F19
I02 ♥ F20
I02 ♥ F21
I02 ♥ F22
I02 ♥ F23
I02 ♥ F24
I02 ♥ F25
I02 ♥ F26
I02 ♥ F27
I02 ♥ F28
I02 ♥ F29
I02 ♥ F30
I02 ♥ F31
I02 ♥ F32
I02 ♥ F33
I02 ♥ F34
I02 ♥ F35
I02 ♥ F36
I02 ♥ F37
I02 ♥ F38
I02 ♥ F39
I02 ♥ F40
I02 ♥ F41
I02 ♥ F42
I02 ♥ F43
I02 ♥ F44
I02 ♥ F45
I02 ♥ F46
I02 ♥ F47
I02 ♥ H10
I02 ♥ H16
I02 ♥ H22
I02 ♥ I12
I02 ♥ I13
I02 ♥ I18
I03 ♥ F00
I03 ♥ F01
I03 ♥ F02
I03 ♥ F03
I03 ♥ F04
I03 ♥ F05
I03 ♥ F06
I03 ♥ F07
I03 ♥ F08
I03 ♥ F09
I03 ♥ F10
I03 ♥ F11
I03 ♥ F12
I03 ♥ F13
I03 ♥ F14
I03 ♥ F15
I03 ♥ F16
I03 ♥ F17
I03 ♥ F18
I03 ♥ F19
I03 ♥ F20
I03 ♥ F21
I03 ♥ F22
I03 ♥ F23
I03 ♥ F24
I03 ♥ F25
I03 ♥ F26
I03 ♥ F27
I03 ♥ F28
I03 ♥ F29
I03 ♥ F30
I03 ♥ F31
I03 ♥ F32
I03 ♥ F33
I03 ♥ F34
I03 ♥ F35
I03 ♥ F36
I03 ♥ F37
I03 ♥ F38
I03 ♥ F39
I03 ♥ F40
I03 ♥ F41
I03 ♥ F42
I03 ♥ F43
I03 ♥ F44
I03 ♥ F45
I03 ♥ F46
I03 ♥ F47
I03 ♥ H05
I03 ♥ H11
I03 ♥ H17
I03 ♥ I07
I03 ♥ I08
I04 ♥ F00
I04 ♥ F01
I04 ♥ F02
I04 ♥ F03
I04 ♥ F04
I04 ♥ F05
I04 ♥ F06
I04 ♥ F07
I04 ♥ F08
I04 ♥ F09
I04 ♥ F10
I04 ♥ F11
I04 ♥ F12
I04 ♥ F13
I04 ♥ F14
I04 ♥ F15
I04 ♥ F16
I04 ♥ F17
I04 ♥ F18
I04 ♥ F19
I04 ♥ F20
I04 ♥ F21
I04 ♥ F22
I04 ♥ F23
I04 ♥ F24
I04 ♥ F25
I04 ♥ F26
I04 ♥ F27
I04 ♥ F28
I04 ♥ F29
I04 ♥ F30
I04 ♥ F31
I04 ♥ F32
I04 ♥ F33
I04 ♥ F34
I04 ♥ F35
I04 ♥ F36
I04 ♥ F37
I04 ♥ F38
I04 ♥ F39
I04 ♥ F40
I04 ♥ F41
I04 ♥ F42
I04 ♥ F43
I04 ♥ F44
I04 ♥ F45
I04 ♥ F46
I04 ♥ F47
I04 ♥ H10
I04 ♥ H16
I05 ♥ F00
I05 ♥ F01
I05 ♥ F02
I05 ♥ F03
I05 ♥ F04
I05 ♥ F05
I05 ♥ F06
I05 ♥ F07
I05 ♥ F08
I05 ♥ F09
I05 ♥ F10
I05 ♥ F11
I05 ♥ F12
I05 ♥ F13
I05 ♥ F14
I05 ♥ F15
I05 ♥ F16
I05 ♥ F17
I05 ♥ F18
I05 ♥ F19
I05 ♥ F20
I05 ♥ F21
I05 ♥ F22
I05 ♥ F23
I05 ♥ F24
I05 ♥ F25
I05 ♥ F26
I05 ♥ F27
I05 ♥ F28
I05 ♥ F29
I05 ♥ F30
I05 ♥ F31
I05 ♥ F32
I05 ♥ F33
I05 ♥ F34
I05 ♥ F35
I05 ♥ F36
I05 ♥ F37
I05 ♥ F38
I05 ♥ F39
I05 ♥ F40
I05 ♥ F41
I05 ♥ F42
I05 ♥ F43
I05 ♥ F44
I05 ♥ F45
I05 ♥ F46
I05 ♥ F47
I05 ♥ H05
I05 ♥ H11
I06 ♥ F00
I06 ♥ F01
I06 ♥ F02
I06 ♥ F03
I06 ♥ F04
I06 ♥ F05
I06 ♥ F06
I06 ♥ F07
I06 ♥ F08
I06 ♥ F09
I06 ♥ F10
I06 ♥ F11
I06 ♥ F12
I06 ♥ F13
I06 ♥ F14
I06 ♥ F15
I06 ♥ F16
I06 ♥ F17
I06 ♥ F18
I06 ♥ F19
I06 ♥ F20
I06 ♥ F21
I06 ♥ F22
I06 ♥ F23
I06 ♥ F24
I06 ♥ F25
I06 ♥ F26
I06 ♥ F27
I06 ♥ F28
I06 ♥ F29
I06 ♥ F30
I06 ♥ F31
I06 ♥ F32
I06 ♥ F33
I06 ♥ F34
I06 ♥ F35
I06 ♥ F36
I06 ♥ F37
I06 ♥ F38
I06 ♥ F39
I06 ♥ F40
I06 ♥ F41
I06 ♥ F42
I06 ♥ F43
I06 ♥ F44
I06 ♥ F45
I06 ♥ F46
I06 ♥ F47
I06 ♥ I11
I06 ♥ I15
I07 ♥ F00
I07 ♥ F01
I07 ♥ F02
I07 ♥ F03
I07 ♥ F04
I07 ♥ F05
I07 ♥ F06
I07 ♥ F07
I07 ♥ F08
I07 ♥ F09
I07 ♥ F10
I07 ♥ F11
I07 ♥ F12
I07 ♥ F13
I07 ♥ F14
I07 ♥ F15
I07 ♥ F16
I07 ♥ F17
I07 ♥ F18
I07 ♥ F19
I07 ♥ F20
I07 ♥ F21
I07 ♥ F22
I07 ♥ F23
I07 ♥ F24
I07 ♥ F25
I07 ♥ F26
I07 ♥ F27
I07 ♥ F28
I07 ♥ F29
I07 ♥ F30
I07 ♥ F31
I07 ♥ F32
I07 ♥ F33
I07 ♥ F34
I07 ♥ F35
I07 ♥ F36
I07 ♥ F37
I07 ♥ F38
I07 ♥ F39
I07 ♥ F40
I07 ♥ F41
I07 ♥ F42
I07 ♥ F43
I07 ♥ F44
I07 ♥ F45
I07 ♥ F46
I07 ♥ F47
I07 ♥ I00
I07 ♥ I01
I07 ♥ I03
I07 ♥ I10
I07 ♥ I11
I07 ♥ I16
I07 ♥ I19
I08 ♥ F00
I08 ♥ F01
I08 ♥ F02
I08 ♥ F03
I08 ♥ F04
I08 ♥ F05
I08 ♥ F06
I08 ♥ F07
I08 ♥ F08
I08 ♥ F09
I08 ♥ F10
I08 ♥ F11
I08 ♥ F12
I08 ♥ F13
I08 ♥ F14
I08 ♥ F15
I08 ♥ F16
I08 ♥ F17
I08 ♥ F18
I08 ♥ F19
I08 ♥ F20
I08 ♥ F21
I08 ♥ F22
I08 ♥ F23
I08 ♥ F24
I08 ♥ F25
I08 ♥ F26
I08 ♥ F27
I08 ♥ F28
I08 ♥ F29
I08 ♥ F30
I08 ♥ F31
I08 ♥ F32
I08 ♥ F33
I08 ♥ F34
I08 ♥ F35
I08 ♥ F36
I08 ♥ F37
I08 ♥ F38
I08 ♥ F39
I08 ♥ F40
I08 ♥ F41
I08 ♥ F42
I08 ♥ F43
I08 ♥ F44
I08 ♥ F45
I08 ♥ F46
I08 ♥ F47
I08 ♥ I00
I08 ♥ I03
I09 ♥ F00
I09 ♥ F01
I09 ♥ F02
I09 ♥ F03
I09 ♥ F04
I09 ♥ F05
I09 ♥ F06
I09 ♥ F07
I09 ♥ F08
I09 ♥ F09
I09 ♥ F10
I09 ♥ F11
I09 ♥ F12
I09 ♥ F13
I09 ♥ F14
I09 ♥ F15
I09 ♥ F16
I09 ♥ F17
I09 ♥ F18
I09 ♥ F19
I09 ♥ F20
I09 ♥ F21
I09 ♥ F22
I09 ♥ F23
I09 ♥ F24
I09 ♥ F25
I09 ♥ F26
I09 ♥ F27
I09 ♥ F28
I09 ♥ F29
I09 ♥ F30
I09 ♥ F31
I09 ♥ F32
I09 ♥ F33
I09 ♥ F34
I09 ♥ F35
I09 ♥ F36
I09 ♥ F37
I09 ♥ F38
I09 ♥ F39
I09 ♥ F40
I09 ♥ F41
I09 ♥ F42
I09 ♥ F43
I09 ♥ F44
I09 ♥ F45
I09 ♥ F46
I09 ♥ F47
I09 ♥ I13
I09 ♥ I14
I10 ♥ F00
I10 ♥ F01
I10 ♥ F02
I10 ♥ F03
I10 ♥ F04
I10 ♥ F05
I10 ♥ F06
I10 ♥ F07
I10 ♥ F08
I10 ♥ F09
I10 ♥ F10
I10 ♥ F11
I10 ♥ F12
I10 ♥ F13
I10 ♥ F14
I10 ♥ F15
I10 ♥ F16
I10 ♥ F17
I10 ♥ F18
I10 ♥ F19
I10 ♥ F20
I10 ♥ F21
I10 ♥ F22
I10 ♥ F23
I10 ♥ F24
I10 ♥ F25
I10 ♥ F26
I10 ♥ F27
I10 ♥ F28
I10 ♥ F29
I10 ♥ F30
I10 ♥ F31
I10 ♥ F32
I10 ♥ F33
I10 ♥ F34
I10 ♥ F35
I10 ♥ F36
I10 ♥ F37
I10 ♥ F38
I10 ♥ F39
I10 ♥ F40
I10 ♥ F41
I10 ♥ F42
I10 ♥ F43
I10 ♥ F44
I10 ♥ F45
I10 ♥ F46
I10 ♥ F47
I10 ♥ I07
I10 ♥ I13
I10 ♥ I16
I10 ♥ I18
I11 ♥ F00
I11 ♥ F01
I11 ♥ F02
I11 ♥ F03
I11 ♥ F04
I11 ♥ F05
I11 ♥ F06
I11 ♥ F07
I11 ♥ F08
I11 ♥ F09
I11 ♥ F10
I11 ♥ F11
I11 ♥ F12
I11 ♥ F13
I11 ♥ F14
I11 ♥ F15
I11 ♥ F16
I11 ♥ F17
I11 ♥ F18
I11 ♥ F19
I11 ♥ F20
I11 ♥ F21
I11 ♥ F22
I11 ♥ F23
I11 ♥ F24
I11 ♥ F25
I11 ♥ F26
I11 ♥ F27
I11 ♥ F28
I11 ♥ F29
I11 ♥ F30
I11 ♥ F31
I11 ♥ F32
I11 ♥ F33
I11 ♥ F34
I11 ♥ F35
I11 ♥ F36
I11 ♥ F37
I11 ♥ F38
I11 ♥ F39
I11 ♥ F40
I11 ♥ F41
I11 ♥ F42
I11 ♥ F43
I11 ♥ F44
I11 ♥ F45
I11 ♥ F46
I11 ♥ F47
I11 ♥ I06
I11 ♥ I07
I12 ♥ F00
I12 ♥ F01
I12 ♥ F02
I12 ♥ F03
I12 ♥ F04
I12 ♥ F05
I12 ♥ F06
I12 ♥ F07
I12 ♥ F08
I12 ♥ F09
I12 ♥ F10
I12 ♥ F11
I12 ♥ F12
I12 ♥ F13
I12 ♥ F14
I12 ♥ F15
I12 ♥ F16
I12 ♥ F17
I12 ♥ F18
I12 ♥ F19
I12 ♥ F20
I12 ♥ F21
I12 ♥ F22
I12 ♥ F23
I12 ♥ F24
I12 ♥ F25
I12 ♥ F26
I12 ♥ F27
I12 ♥ F28
I12 ♥ F29
I12 ♥ F30
I12 ♥ F31
I12 ♥ F32
I12 ♥ F33
I12 ♥ F34
I12 ♥ F35
I12 ♥ F36
I12 ♥ F37
I12 ♥ F38
I12 ♥ F39
I12 ♥ F40
I12 ♥ F41
I12 ♥ F42
I12 ♥ F43
I12 ♥ F44
I12 ♥ F45
I12 ♥ F46
I12 ♥ F47
I12 ♥ I02
I13 ♥ F00
I13 ♥ F01
I13 ♥ F02
I13 ♥ F03
I13 ♥ F04
I13 ♥ F05
I13 ♥ F06
I13 ♥ F07
I13 ♥ F08
I13 ♥ F09
I13 ♥ F10
I13 ♥ F11
I13 ♥ F12
I13 ♥ F13
I13 ♥ F14
I13 ♥ F15
I13 ♥ F16
I13 ♥ F17
I13 ♥ F18
I13 ♥ F19
I13 ♥ F20
I13 ♥ F21
I13 ♥ F22
I13 ♥ F23
I13 ♥ F24
I13 ♥ F25
I13 ♥ F26
I13 ♥ F27
I13 ♥ F28
I13 ♥ F29
I13 ♥ F30
I13 ♥ F31
I13 ♥ F32
I13 ♥ F33
I13 ♥ F34
I13 ♥ F35
I13 ♥ F36
I13 ♥ F37
I13 ♥ F38
I13 ♥ F39
I13 ♥ F40
I13 ♥ F41
I13 ♥ F42
I13 ♥ F43
I13 ♥ F44
I13 ♥ F45
I13 ♥ F46
I13 ♥ F47
I13 ♥ I01
I13 ♥ I02
I13 ♥ I09
I13 ♥ I10
I13 ♥ I17
I14 ♥ F00
I14 ♥ F01
I14 ♥ F02
I14 ♥ F03
I14 ♥ F04
I14 ♥ F05

I14 ♥ F06
I14 ♥ F07
I14 ♥ F08
I14 ♥ F09
I14 ♥ F10
I14 ♥ F11
I14 ♥ F12
I14 ♥ F13
I14 ♥ F14
I14 ♥ F15
I14 ♥ F16
I14 ♥ F17
I14 ♥ F18
I14 ♥ F19
I14 ♥ F20
I14 ♥ F21
I14 ♥ F22
I14 ♥ F23
I14 ♥ F24
I14 ♥ F25
I14 ♥ F26
I14 ♥ F27
I14 ♥ F28
I14 ♥ F29
I14 ♥ F30
I14 ♥ F31
I14 ♥ F32
I14 ♥ F33
I14 ♥ F34
I14 ♥ F35
I14 ♥ F36
I14 ♥ F37
I14 ♥ F38
I14 ♥ F39
I14 ♥ F40
I14 ♥ F41
I14 ♥ F42
I14 ♥ F43
I14 ♥ F44
I14 ♥ F45
I14 ♥ F46
I14 ♥ F47
I14 ♥ I09
I15 ♥ I01
I15 ♥ I06
I15 ♥ I15
I15 ♥ I15
I16 ♥ I01
I16 ♥ I07
I16 ♥ I10
I17 ♥ B02
I17 ♥ I13
I17 ♥ I19
I18 ♥ H04
I18 ♥ I02
I18 ♥ I10
I19 ♥ B02
I19 ♥ I01
I19 ♥ I07
I19 ♥ I17
I19 ♥ I19
I19 ♥ I19
I19 ♥ J12

F

J00 ♥ B01
J00 ♥ F00
J00 ♥ F05
J00 ♥ F24
J00 ♥ F29
J00 ♥ I01
J00 ♥ J02
J00 ♥ L02
J01 ♥ B01
J01 ♥ F00
J01 ♥ F24
J01 ♥ J03
J01 ♥ J04
J01 ♥ L01
J02 ♥ I01
J02 ♥ J00
J02 ♥ J07
J03 ♥ J01
J04 ♥ J01
J04 ♥ J48
J04 ♥ L00
J05 ♥ J10
J05 ♥ J14
J05 ♥ J21
J05 ♥ J23
J05 ♥ J25
J05 ♥ J27
J05 ♥ J29
J05 ♥ J31
J05 ♥ J33
J05 ♥ J35
J05 ♥ J37
J05 ♥ J39
J05 ♥ J41
J05 ♥ J43
J05 ♥ J45
J05 ♥ J47
J05 ♥ J48
J05 ♥ J49
J05 ♥ J51
J06 ♥ J11
J07 ♥ C05
J07 ♥ J02
J07 ♥ J08
J08 ♥ C03
J08 ♥ C04
J08 ♥ J07
J10 ♥ J05
J10 ♥ J22
J10 ♥ J24
J10 ♥ J26
J10 ♥ J28
J10 ♥ J30
J10 ♥ J32
J10 ♥ J34
J10 ♥ J36
J10 ♥ J38
J10 ♥ J40
J10 ♥ J42
J10 ♥ J44
J10 ♥ J46
J10 ♥ J48
J10 ♥ J49
J10 ♥ J50
J10 ♥ J52
J11 ♥ C04
J11 ♥ J06
J11 ♥ J13
J12 ♥ C03
J12 ♥ I19
J12 ♥ J16
J13 ♥ J11
J14 ♥ J05
J14 ♥ J20
J15 ♥ J17
J15 ♥ J19
J16 ♥ J12
J17 ♥ A12
J17 ♥ B00
J17 ♥ J15
J17 ♥ J19
J18 ♥ A00
J18 ♥ A01
J18 ♥ B01
J18 ♥ L16
J18 ♥ L17
J19 ♥ J15
J19 ♥ J17
J19 ♥ L15
J20 ♥ J14
J20 ♥ L14
J21 ♥ J05
J21 ♥ L16
J22 ♥ J10
J22 ♥ L12
J23 ♥ J05
J23 ♥ L11
J24 ♥ J10
J24 ♥ K03
J24 ♥ K04
J25 ♥ J05
J25 ♥ K00
J25 ♥ K05
J26 ♥ J10
J26 ♥ K01
J26 ♥ K06
J27 ♥ J05
J27 ♥ K02
J27 ♥ K07
J28 ♥ J10
J28 ♥ K08
J28 ♥ L07
J29 ♥ J05
J29 ♥ K09
J29 ♥ L08
J30 ♥ J10
J30 ♥ K10
J30 ♥ L09
J31 ♥ J05
J31 ♥ K11
J31 ♥ L10
J32 ♥ J10
J32 ♥ K12
J33 ♥ J05
J33 ♥ K13
J34 ♥ J10
J34 ♥ K14
J35 ♥ J05
J35 ♥ K15
J36 ♥ J10
J36 ♥ K16
J37 ♥ J05
J37 ♥ K17
J38 ♥ J10
J38 ♥ K18
J39 ♥ J05
J39 ♥ K00
J39 ♥ K19
J40 ♥ J10
J40 ♥ K01
J40 ♥ K20
J41 ♥ J05
J41 ♥ K02
J41 ♥ K21
J42 ♥ J10
J42 ♥ K22
J43 ♥ J05
J43 ♥ K23
J44 ♥ J10
J44 ♥ K24
J45 ♥ J05
J45 ♥ K25
J46 ♥ J10
J46 ♥ K26
J47 ♥ J05
J47 ♥ K27
J48 ♥ J04
J48 ♥ J05
J48 ♥ J10
J48 ♥ K03
J48 ♥ L03
J48 ♥ L06
J48 ♥ L08
J49 ♥ J05
J49 ♥ J10
J49 ♥ K00
J49 ♥ L00
J49 ♥ L04
J50 ♥ J10
J50 ♥ K01
J50 ♥ L05
J51 ♥ J05
J51 ♥ K02
J51 ♥ L02
J51 ♥ L06
J52 ♥ J10
J52 ♥ K01

G

K00 ♥ J25
K00 ♥ J39
K00 ♥ J49
K00 ♥ K29
K00 ♥ K31
K00 ♥ K33
K01 ♥ J26
K01 ♥ J40
K01 ♥ J50
K01 ♥ J52
K01 ♥ K28
K01 ♥ K30
K01 ♥ K32
K02 ♥ J27
K02 ♥ J41
K02 ♥ J51
K02 ♥ K29
K02 ♥ K31
K02 ♥ K33
K03 ♥ J24
K03 ♥ J48
K03 ♥ K28
K03 ♥ K30
K03 ♥ K32
K04 ♥ J24
K04 ♥ K32
K05 ♥ J25
K05 ♥ K06
K05 ♥ K12
K05 ♥ K18
K05 ♥ K24
K05 ♥ K31
K06 ♥ J26
K06 ♥ K05
K06 ♥ K11
K06 ♥ K17
K06 ♥ K23
K06 ♥ K30
K06 ♥ L11
K07 ♥ J27
K07 ♥ K29
K08 ♥ J28
K08 ♥ K09
K08 ♥ K15
K08 ♥ K21
K08 ♥ K27
K08 ♥ K28
K09 ♥ J29
K09 ♥ K08
K09 ♥ K14
K09 ♥ K20
K09 ♥ K26
K09 ♥ K33
K09 ♥ L14
K10 ♥ J30
K10 ♥ K32
K11 ♥ J31
K11 ♥ K06
K11 ♥ K12
K11 ♥ K18
K11 ♥ K24
K11 ♥ K31
K12 ♥ J32
K12 ♥ K05
K12 ♥ K11
K12 ♥ K17
K12 ♥ K23
K12 ♥ K30
K12 ♥ L11
K13 ♥ J33
K13 ♥ K29
K14 ♥ J34
K14 ♥ K09
K14 ♥ K15
K14 ♥ K21
K14 ♥ K27
K14 ♥ K28
K14 ♥ L08
K15 ♥ J35
K15 ♥ K08
K15 ♥ K14
K15 ♥ K20
K15 ♥ K26
K15 ♥ K33
K15 ♥ L14
K16 ♥ J36
K16 ♥ K32
K17 ♥ J37
K17 ♥ K06
K17 ♥ K12
K17 ♥ K18
K17 ♥ K24
K17 ♥ K31
K18 ♥ J38
K18 ♥ K05
K18 ♥ K11
K18 ♥ K17
K18 ♥ K23
K18 ♥ K30
K18 ♥ L11
K19 ♥ J39
K19 ♥ K29
K20 ♥ J40
K20 ♥ K09
K20 ♥ K15
K20 ♥ K21
K20 ♥ K27
K20 ♥ K28
K21 ♥ J41
K21 ♥ K08
K21 ♥ K14
K21 ♥ K20
K21 ♥ K26
K21 ♥ K33
K21 ♥ L14
K22 ♥ J42
K22 ♥ K32
K23 ♥ J43
K23 ♥ K06
K23 ♥ K12
K23 ♥ K18
K23 ♥ K24
K23 ♥ K31
K24 ♥ J44
K24 ♥ K05
K24 ♥ K11
K24 ♥ K17
K24 ♥ K23
K24 ♥ K30
K24 ♥ L11
K25 ♥ J45
K25 ♥ K29
K26 ♥ J46
K26 ♥ K09
K26 ♥ K15
K26 ♥ K21
K26 ♥ K27
K26 ♥ K28
K27 ♥ J47
K27 ♥ K08
K27 ♥ K14
K27 ♥ K20
K27 ♥ K26
K27 ♥ K33
K27 ♥ L14
K28 ♥ K01
K28 ♥ K03
K28 ♥ K08
K28 ♥ K14
K28 ♥ K20
K28 ♥ K26
K28 ♥ K33
K28 ♥ K38
K28 ♥ K44
K28 ♥ L07
K28 ♥ L25
K28 ♥ L40
K28 ♥ L41
K29 ♥ K00
K29 ♥ K02
K29 ♥ K07
K29 ♥ K13
K29 ♥ K19
K29 ♥ K25
K29 ♥ K37
K29 ♥ K43
K29 ♥ L39
K29 ♥ L40
K29 ♥ L41
K30 ♥ K01
K30 ♥ K03
K30 ♥ K06
K30 ♥ K12
K30 ♥ K18
K30 ♥ K24
K30 ♥ K31
K30 ♥ K36
K30 ♥ K42
K30 ♥ L24
K30 ♥ L39
K30 ♥ L40
K31 ♥ K00
K31 ♥ K02
K31 ♥ K05
K31 ♥ K11
K31 ♥ K17
K31 ♥ K23
K31 ♥ K30
K31 ♥ K35
K31 ♥ K41
K31 ♥ L10
K31 ♥ L28
K31 ♥ L39
K32 ♥ K01
K32 ♥ K03
K32 ♥ K04
K32 ♥ K10
K32 ♥ K16
K32 ♥ K22
K32 ♥ K34
K32 ♥ K40
K32 ♥ L09
K32 ♥ L38
K33 ♥ K00
K33 ♥ K02
K33 ♥ K09
K33 ♥ K15
K33 ♥ K21
K33 ♥ K27
K33 ♥ K28
K33 ♥ K39
K33 ♥ K45
K33 ♥ L08
K33 ♥ L21
K33 ♥ L41
K34 ♥ K32
K34 ♥ K39
K34 ♥ K45
K34 ♥ K50
K35 ♥ K31
K35 ♥ K45
K35 ♥ K49
K35 ♥ K50
K35 ♥ L40
K35 ♥ L41
K35 ♥ L56
K35 ♥ L57
K36 ♥ K30
K36 ♥ K37
K36 ♥ L35
K37 ♥ K29
K37 ♥ K36
K37 ♥ K47
K37 ♥ L39
K37 ♥ L40
K38 ♥ K28
K38 ♥ K46
K38 ♥ K47
K38 ♥ L39
K39 ♥ K33
K39 ♥ K34
K39 ♥ L38
K40 ♥ K32
K40 ♥ K45
K40 ♥ K46
K40 ♥ K50
K41 ♥ K31
K41 ♥ K44
K41 ♥ K49
K41 ♥ K50
K41 ♥ K51
K41 ♥ L24
K41 ♥ L41
K41 ♥ L55
K42 ♥ K30
K42 ♥ K43
K42 ♥ L40
K42 ♥ L44
K42 ♥ L46
K42 ♥ L48
K43 ♥ K29
K43 ♥ K42
K43 ♥ K47
K43 ♥ L42
K43 ♥ L43
K43 ♥ L45
K43 ♥ L47
K44 ♥ K28
K44 ♥ K41
K44 ♥ K46
K44 ♥ K47
K45 ♥ K33
K45 ♥ K34
K45 ♥ K35
K45 ♥ K40
K45 ♥ L17
K45 ♥ L82
K46 ♥ K38
K46 ♥ K40
K46 ♥ K44
K46 ♥ K47
K46 ♥ K56
K46 ♥ L41
K47 ♥ K37
K47 ♥ K38
K47 ♥ K43
K47 ♥ K44
K47 ♥ K46
K47 ♥ K55
K47 ♥ K57
K47 ♥ L40
K47 ♥ L41
K47 ♥ L75
K48 ♥ K51
K48 ♥ K54
K48 ♥ K56
K48 ♥ L72
K48 ♥ L73
K48 ♥ L74
K49 ♥ K35
K49 ♥ K41
K49 ♥ K50
K49 ♥ K53
K49 ♥ L58
K49 ♥ L60
K49 ♥ L62
K49 ♥ L64
K50 ♥ K34
K50 ♥ K35
K50 ♥ K40
K50 ♥ K41
K50 ♥ K49
K50 ♥ K52
K50 ♥ L56
K50 ♥ L57
K50 ♥ L59
K50 ♥ L61
K50 ♥ L63
K51 ♥ K41
K51 ♥ K48
K51 ♥ K57
K52 ♥ K50
K52 ♥ K57
K52 ♥ K59
K52 ♥ K63
K52 ♥ K69
K52 ♥ L92
K52 ♥ L98
K52 ♥ M04
K52 ♥ M10
K52 ♥ M16
K53 ♥ K49
K53 ♥ K56
K53 ♥ K58
K53 ♥ K62
K53 ♥ K63
K53 ♥ K68
K53 ♥ K69
K53 ♥ L55
K53 ♥ L90
K53 ♥ L91
K53 ♥ L92
K53 ♥ L97
K53 ♥ L98
K53 ♥ M03
K53 ♥ M04
K53 ♥ M09
K53 ♥ M10
K53 ♥ M15
K53 ♥ M16
K54 ♥ K48
K54 ♥ K55
K54 ♥ L75
K54 ♥ L78
K54 ♥ L80
K55 ♥ K47
K55 ♥ K54
K55 ♥ K60
K55 ♥ K66
K55 ♥ L74
K55 ♥ L76
K55 ♥ L77
K55 ♥ L79
K55 ♥ L95
K55 ♥ M01
K55 ♥ M07
K55 ♥ M13
K55 ♥ M19
K55 ♥ M22
K56 ♥ K46
K56 ♥ K48
K56 ♥ K53
K56 ♥ K59
K56 ♥ K60
K56 ♥ K65
K56 ♥ K66
K56 ♥ L94
K56 ♥ L95
K56 ♥ M00
K56 ♥ M01
K56 ♥ M06
K56 ♥ M07

K56 ♥ M12	K69 ♥ K52	L13 ♥ L22	L28 ♥ L03	L41 ♥ K33	L56 ♥ K35	L74 ♥ L82	L94 ♥ K56	M14 ♥ L95
K56 ♥ M13	K69 ♥ K53	L14 ♥ J20	L28 ♥ L33	L41 ♥ K35	L56 ♥ K50	L75 ♥ K47	L94 ♥ K63	M15 ♥ K53
K56 ♥ M18	K69 ♥ K60	L14 ♥ K09	L28 ♥ L34	L41 ♥ K41	L56 ♥ L72	L75 ♥ K54	L94 ♥ M15	M15 ♥ L94
K56 ♥ M19	K69 ♥ K64	L14 ♥ K15	L29 ♥ L02	L41 ♥ K46	L56 ♥ L73	L75 ♥ K60	L95 ♥ K55	M16 ♥ K52
K56 ♥ M21	L00 ♥ J04	L14 ♥ K21	L29 ♥ L33	L41 ♥ K47	L57 ♥ K35	L75 ♥ L81	L95 ♥ K56	M16 ♥ K53
K56 ♥ M22	L00 ♥ J49	L14 ♥ K27	L29 ♥ L34	L41 ♥ L18	L57 ♥ K50	L76 ♥ K55	L95 ♥ K62	M16 ♥ L93
K57 ♥ K47	L00 ♥ L04	L15 ♥ J19	L30 ♥ L01	L41 ♥ L39	L57 ♥ L38	L76 ♥ L54	L95 ♥ M14	M17 ♥ L92
K57 ♥ K51	L00 ♥ L05	L15 ♥ L20	L30 ♥ L32	L41 ♥ L47	L57 ♥ L71	L76 ♥ L90	L96 ♥ K63	M18 ♥ K56
K57 ♥ K52	L00 ♥ L31	L16 ♥ J18	L30 ♥ L33	L42 ♥ K43	L58 ♥ K49	L77 ♥ K55	L96 ♥ M13	M19 ♥ K55
K57 ♥ M05	L01 ♥ J01	L16 ♥ J21	L31 ♥ D57	L42 ♥ L23	L58 ♥ L70	L77 ♥ L53	L97 ♥ K53	M19 ♥ K56
K58 ♥ K53	L01 ♥ L30	L16 ♥ L19	L31 ♥ D58	L42 ♥ L55	L59 ♥ K50	L77 ♥ L88	L97 ♥ K62	M19 ♥ L91
K58 ♥ K61	L02 ♥ A07	L16 ♥ L20	L31 ♥ L00	L43 ♥ K43	L59 ♥ L36	L78 ♥ K54	L97 ♥ M12	M20 ♥ L82
K59 ♥ K52	L02 ♥ J00	L17 ♥ A00	L31 ♥ L26	L43 ♥ L22	L59 ♥ L69	L78 ♥ L52	L98 ♥ K52	M20 ♥ L83
K59 ♥ K56	L02 ♥ J51	L17 ♥ C11	L31 ♥ L27	L43 ♥ L53	L60 ♥ K49	L78 ♥ L87	L98 ♥ K53	M20 ♥ M25
K59 ♥ K60	L02 ♥ L05	L17 ♥ C12	L31 ♥ L48	L44 ♥ K42	L60 ♥ L35	L78 ♥ L88	L98 ♥ K63	M21 ♥ K56
K60 ♥ K55	L02 ♥ L06	L17 ♥ E09	L31 ♥ L49	L44 ♥ L21	L60 ♥ L36	L79 ♥ K55	L98 ♥ M11	M21 ♥ K68
K60 ♥ K56	L02 ♥ L29	L17 ♥ E33	L32 ♥ L30	L44 ♥ L52	L60 ♥ L68	L79 ♥ L51	L99 ♥ K62	M21 ♥ L48
K60 ♥ K59	L03 ♥ A07	L17 ♥ J18	L32 ♥ L37	L44 ♥ L53	L61 ♥ K50	L80 ♥ K54	L99 ♥ M10	M21 ♥ L82
K60 ♥ K69	L03 ♥ A08	L17 ♥ K45	L32 ♥ L63	L45 ♥ K43	L61 ♥ L67	L80 ♥ L50	M00 ♥ K56	M21 ♥ M24
K60 ♥ L75	L03 ♥ J48	L17 ♥ L12	L32 ♥ L83	L45 ♥ L20	L62 ♥ K49	L80 ♥ L85	M00 ♥ K63	M22 ♥ K55
K61 ♥ K58	L03 ♥ L28	L17 ♥ L65	L33 ♥ L28	L46 ♥ K42	L62 ♥ L33	L81 ♥ L49	M00 ♥ M09	M22 ♥ K56
K61 ♥ K68	L04 ♥ J49	L18 ♥ D56	L33 ♥ L29	L46 ♥ L19	L62 ♥ L66	L81 ♥ L75	M01 ♥ K55	M22 ♥ K67
K62 ♥ K53	L04 ♥ L00	L18 ♥ E20	L33 ♥ L30	L46 ♥ L50	L63 ♥ K50	L81 ♥ L83	M01 ♥ K56	M22 ♥ L18
K62 ♥ K63	L04 ♥ L27	L18 ♥ E44	L33 ♥ L62	L46 ♥ L51	L63 ♥ L32	L81 ♥ L84	M01 ♥ K62	M22 ♥ L48
K62 ♥ L93	L05 ♥ J50	L18 ♥ L23	L33 ♥ L63	L47 ♥ K43	L63 ♥ L33	L81 ♥ L85	M01 ♥ M08	M23 ♥ M28
K62 ♥ L95	L05 ♥ L00	L18 ♥ L41	L34 ♥ L28	L47 ♥ L18	L64 ♥ D57	L82 ♥ K45	M02 ♥ K63	M23 ♥ M29
K62 ♥ L97	L05 ♥ L02	L18 ♥ L47	L34 ♥ L29	L47 ♥ L41	L64 ♥ K49	L82 ♥ L48	M02 ♥ M07	M24 ♥ L72
K62 ♥ L99	L06 ♥ J48	L18 ♥ L64	L35 ♥ K36	L47 ♥ L49	L64 ♥ L18	L82 ♥ L49	M03 ♥ K53	M24 ♥ M21
K62 ♥ M01	L06 ♥ J51	L18 ♥ L65	L35 ♥ L27	L47 ♥ L50	L64 ♥ L48	L82 ♥ L74	M03 ♥ K62	M24 ♥ M28
K62 ♥ M03	L06 ♥ L02	L18 ♥ M22	L35 ♥ L60	L48 ♥ K42	L65 ♥ L17	L82 ♥ L84	M03 ♥ M06	M25 ♥ L71
K63 ♥ K52	L06 ♥ L25	L19 ♥ L16	L36 ♥ L26	L48 ♥ L31	L65 ♥ L18	L82 ♥ M20	M04 ♥ K52	M25 ♥ L72
K63 ♥ K53	L07 ♥ J28	L19 ♥ L46	L36 ♥ L59	L48 ♥ L64	L65 ♥ M30	L82 ♥ M21	M04 ♥ K53	M25 ♥ M20
K63 ♥ K62	L07 ♥ K28	L20 ♥ L15	L36 ♥ L60	L48 ♥ L82	L66 ♥ L62	L83 ♥ L32	M04 ♥ K63	M25 ♥ M27
K63 ♥ L91	L07 ♥ L09	L20 ♥ L16	L37 ♥ L25	L48 ♥ M21	L66 ♥ M30	L83 ♥ L81	M04 ♥ M06	M25 ♥ M28
K63 ♥ L92	L07 ♥ L10	L20 ♥ L45	L37 ♥ L32	L48 ♥ M22	L67 ♥ L61	L83 ♥ M20	M05 ♥ K57	M26 ♥ A02
K63 ♥ L94	L08 ♥ J29	L21 ♥ K33	L38 ♥ K32	L49 ♥ L31	L67 ♥ M29	L84 ♥ L81	M06 ♥ K56	M27 ♥ L69
K63 ♥ L96	L08 ♥ J48	L21 ♥ L44	L38 ♥ K39	L49 ♥ L47	L68 ♥ L60	L84 ♥ L82	M06 ♥ M03	M27 ♥ L70
K63 ♥ L98	L08 ♥ K14	L22 ♥ L13	L38 ♥ L57	L49 ♥ L81	L68 ♥ M28	L85 ♥ L80	M06 ♥ M04	M27 ♥ M25
K63 ♥ M00	L08 ♥ K33	L22 ♥ L43	L39 ♥ K29	L49 ♥ L82	L69 ♥ L59	L85 ♥ L81	M07 ♥ K55	M27 ♥ M30
K63 ♥ M02	L09 ♥ J30	L23 ♥ L12	L39 ♥ K30	L50 ♥ L46	L69 ♥ M27	L87 ♥ L78	M07 ♥ K56	M28 ♥ L68
K63 ♥ M04	L09 ♥ K32	L23 ♥ L18	L39 ♥ K31	L50 ♥ L47	L69 ♥ M28	L88 ♥ L77	M07 ♥ M02	M28 ♥ L69
K64 ♥ K69	L09 ♥ L07	L23 ♥ L42	L39 ♥ K37	L50 ♥ L80	L70 ♥ L58	L88 ♥ L78	M08 ♥ M01	M28 ♥ M23
K65 ♥ K56	L10 ♥ J31	L24 ♥ K30	L39 ♥ K38	L51 ♥ L46	L70 ♥ M27	L90 ♥ K53	M09 ♥ K53	M28 ♥ M24
K65 ♥ K68	L10 ♥ K31	L24 ♥ K41	L39 ♥ L41	L51 ♥ L79	L71 ♥ L57	L90 ♥ L76	M09 ♥ M00	M28 ♥ M25
K66 ♥ K55	L10 ♥ L07	L25 ♥ K28	L40 ♥ K28	L52 ♥ L44	L71 ♥ M25	L91 ♥ K53	M10 ♥ K52	M28 ♥ M30
K66 ♥ K56	L11 ♥ J23	L25 ♥ L06	L40 ♥ K29	L52 ♥ L78	L72 ♥ K48	L91 ♥ K63	M10 ♥ K53	M29 ♥ L67
K66 ♥ K67	L11 ♥ K06	L25 ♥ L37	L40 ♥ K30	L53 ♥ L43	L72 ♥ L56	L91 ♥ M19	M10 ♥ L99	M29 ♥ M23
K67 ♥ K66	L11 ♥ K12	L26 ♥ L31	L40 ♥ K35	L53 ♥ L44	L72 ♥ M24	L92 ♥ K52	M11 ♥ L98	M30 ♥ L65
K67 ♥ M22	L11 ♥ K18	L26 ♥ L36	L40 ♥ K37	L53 ♥ L77	L72 ♥ M25	L92 ♥ K53	M12 ♥ K56	M30 ♥ L66
K68 ♥ K53	L11 ♥ K24	L27 ♥ L04	L40 ♥ K42	L54 ♥ L76	L73 ♥ K48	L92 ♥ K63	M12 ♥ L97	M30 ♥ M27
K68 ♥ K61	L12 ♥ J22	L27 ♥ L31	L40 ♥ K47	L55 ♥ K41	L73 ♥ L56	L92 ♥ M17	M13 ♥ K55	M30 ♥ M28
K68 ♥ K65	L12 ♥ L17	L27 ♥ L35	L41 ♥ K28	L55 ♥ K53	L74 ♥ K48	L93 ♥ K62	M13 ♥ K56	
K68 ♥ M21	L12 ♥ L23	L28 ♥ K31	L41 ♥ K29	L55 ♥ L42	L74 ♥ K55	L93 ♥ M16	M13 ♥ L96	

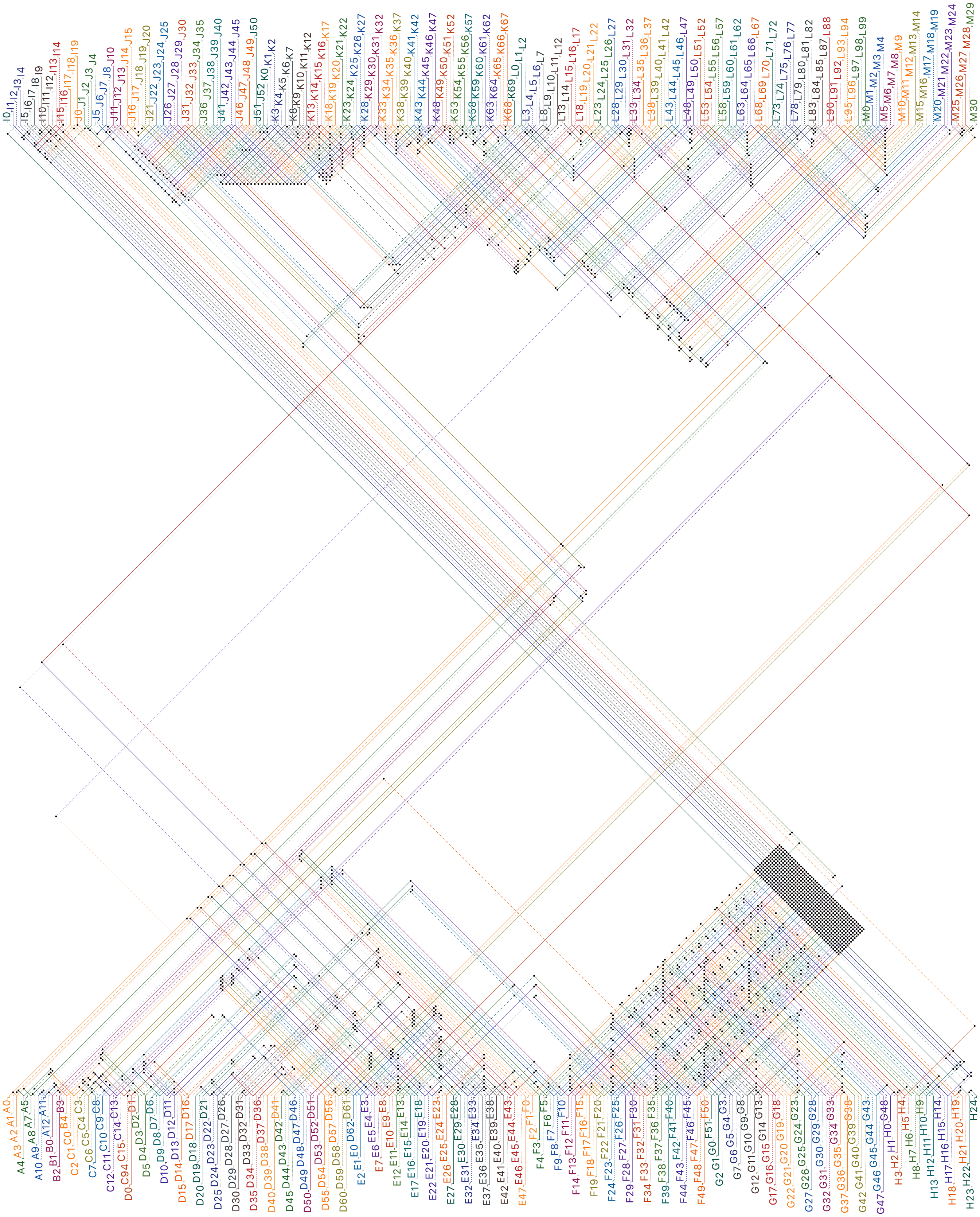


Figure 52 The rule ♥ of the SCTS Oritatami system: in this diagram, we have b ♥ b' iff there is a bullet • at the intersection of one the two lines coming from b and from b'; for instance, we have `A0`♥`A2` but not `A0`♥`A5`.