Towards the algorithmic molecular self-assembly of fractals by cotranscriptional folding

Yusei Masuda, Shinnosuke Seki, and Yuki Ubukata

University of Electro-Communications Tokyo

July 30th, 2018

Cotranscriptional folding

- Cotranscriptional folding–folding occurs during transcription

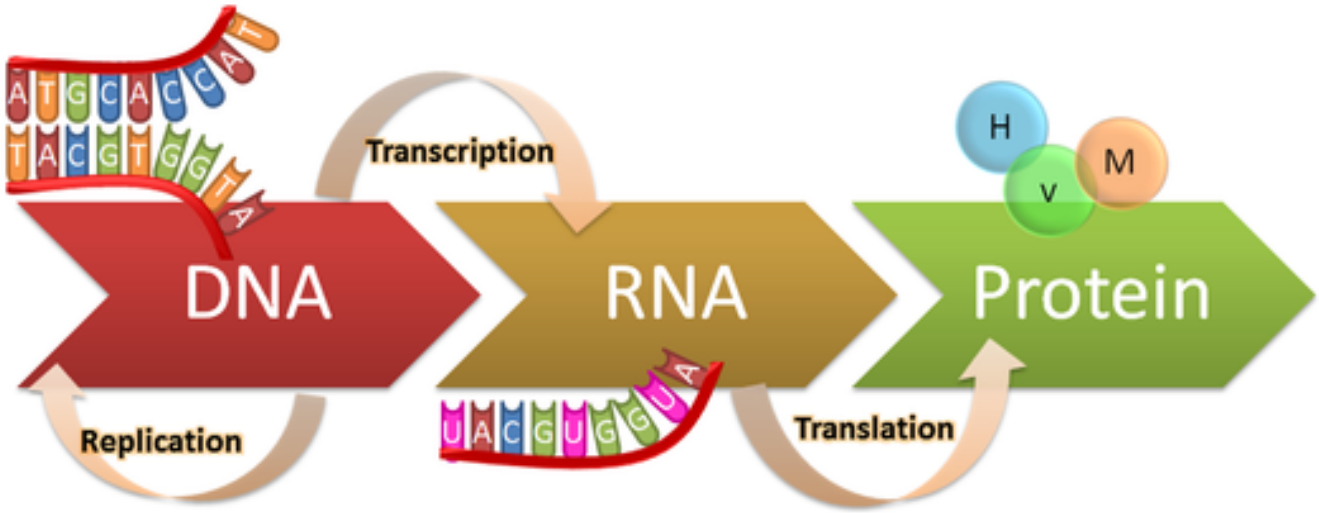


Figure: Central dogma–Information flow in biological systems.

Cotranscriptional folding

RNA polymerase

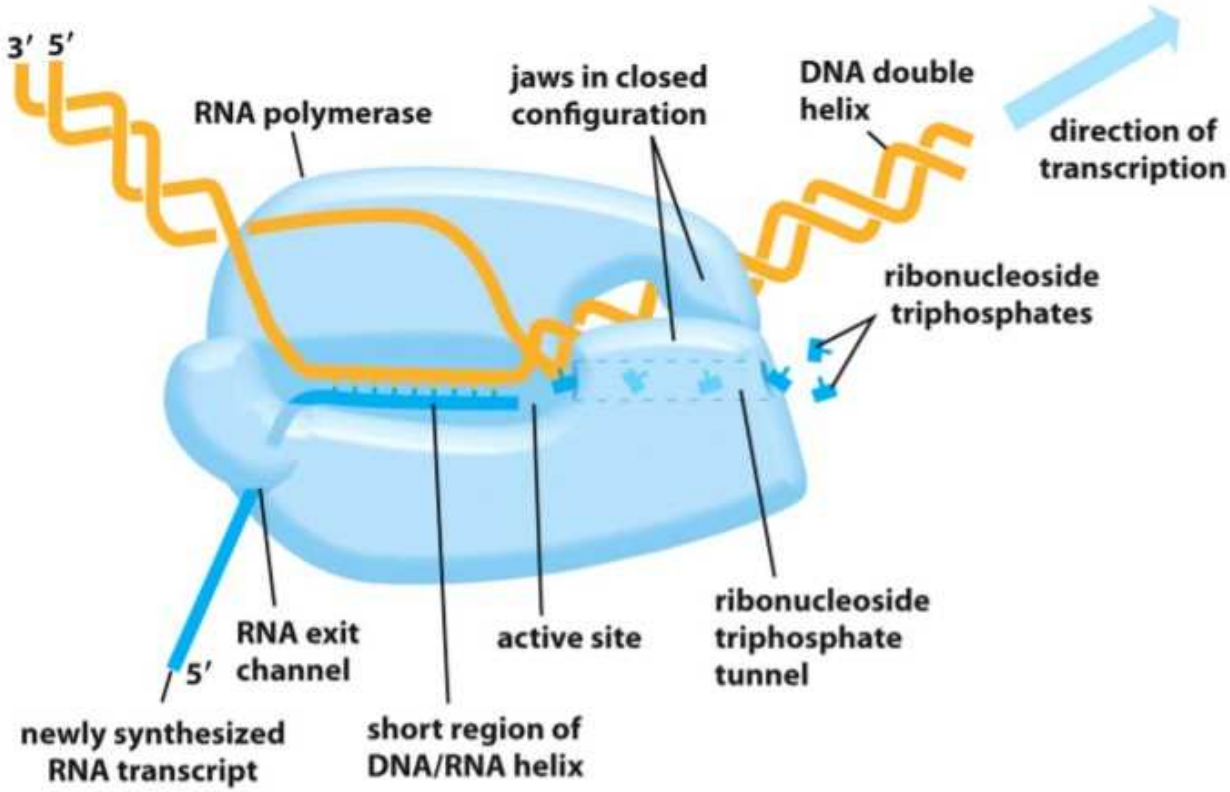


Figure 7-7 *Essential Cell Biology* (© Garland Science 2010)

Cotranscriptional folding

RNA Origami [Geary, Rothemund, Andersen ,*Science* 345 (2014) 799-804]

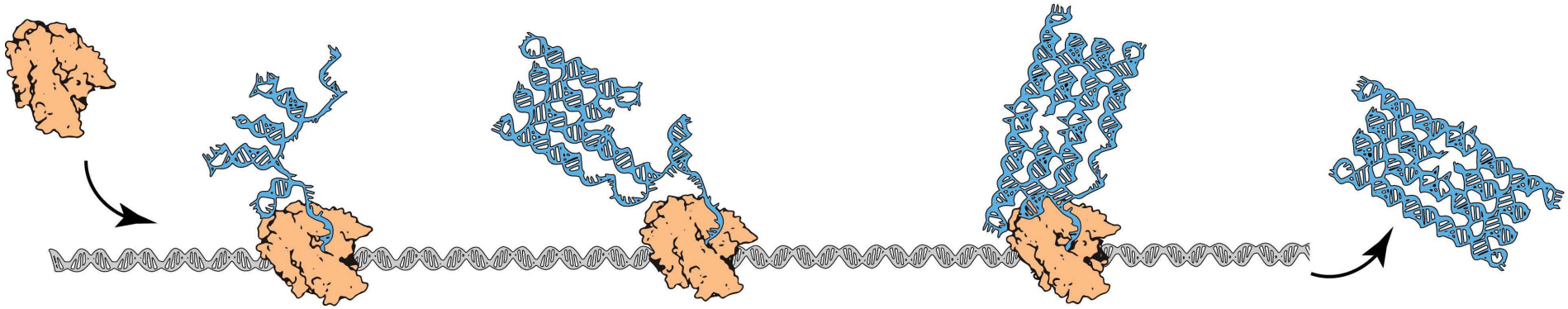


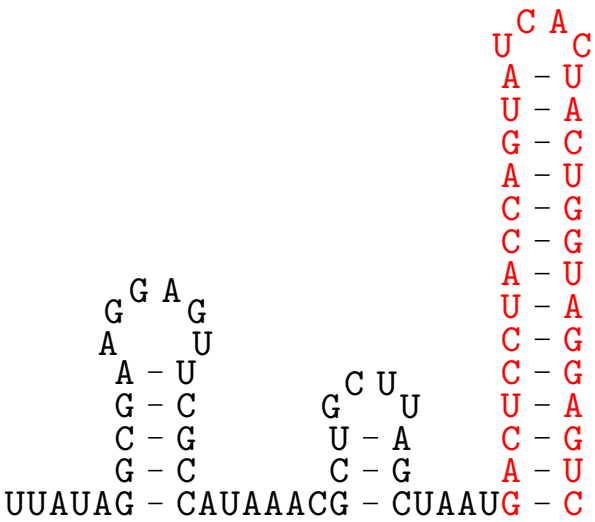
Figure: Design of ssRNA origami that self-assembles RNA tiles.

Cotranscriptional folding

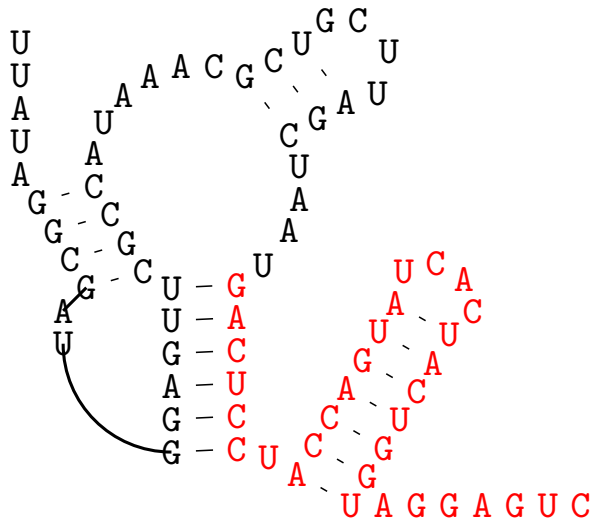
Regulation in gene expression by CF [Watters et al. 2016]

Transcript: UUAUAGGCGAUGGAGUUCGCCAUAAACGCUGCUUAGCUAAUGACUCCUACCAGUAUCACUACUGGUAGGAGUC

Concentration of NaF may promote or inhibit the transcript above to fold into the terminator stem cotranscriptionally.



Terminated (0 mM NaF)



Antiterminated (10 mM NaF)

RNA Origami [Geary, Rothemund, Andersen ,*Science* 345 (2014) 799-804]

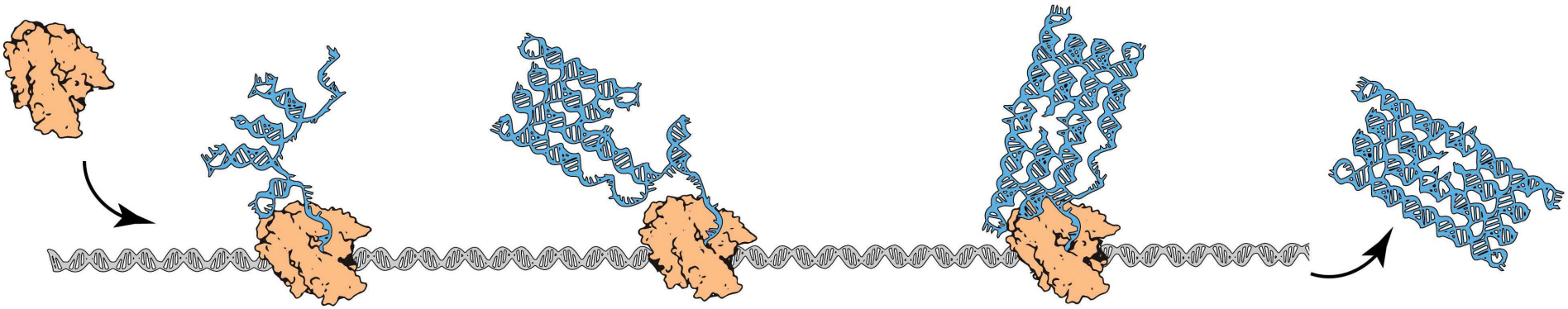
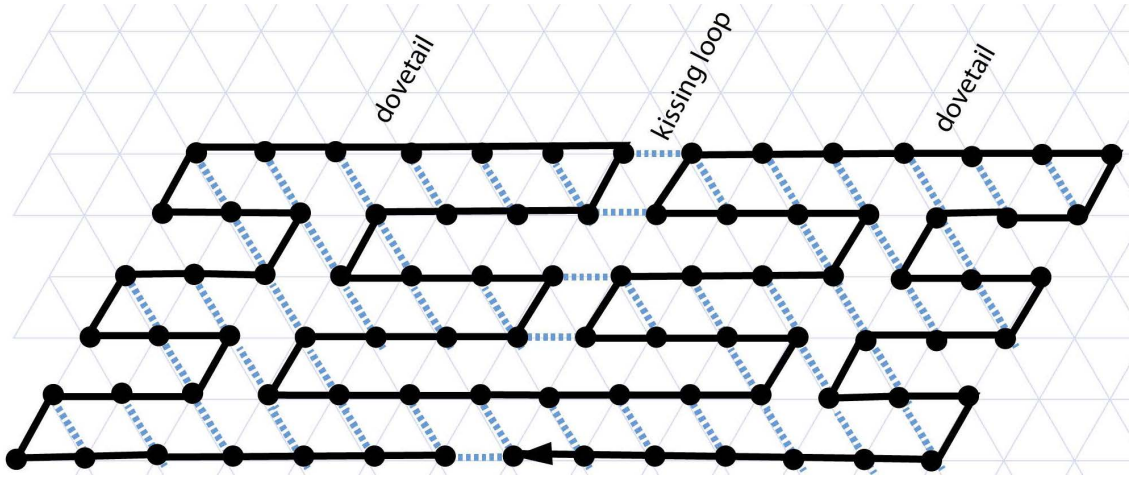


Figure: Design of ssRNA origami that self-assembles RNA tiles.

Oritatami System

- Oritatami system–a mathematical model of computation by cotranscriptional folding

abstract



Algorithmic shape self-assembly of fractal by CF

Motivations

- The initiation of algorithmic self-assembly of shape by CF
- A fractal is the benchmark of molecular self-assembly
- Traversable somehow algorithmically

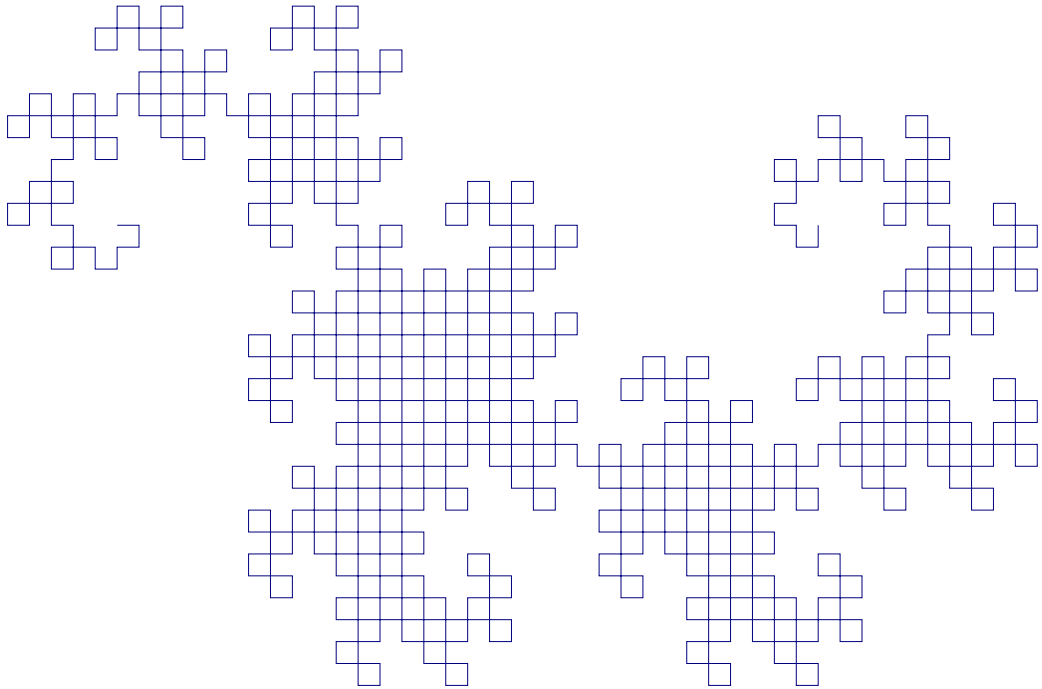


Figure: Heighway dragon.

Oritatami System

Definition

An oritatami system (OS) is a tuple $\Xi = (\Sigma, w, \mathcal{R}, \delta, \alpha)$, where

- Σ finite set of bead types
- $w \in \Sigma^*$ *transcript*, a string to be folded
- $\mathcal{R} \in \Sigma \times \Sigma$ *ruleset* to specify which types of beads can interact
- δ *delay* (transcription rate)
- α *arity*, max # of interactions per bead

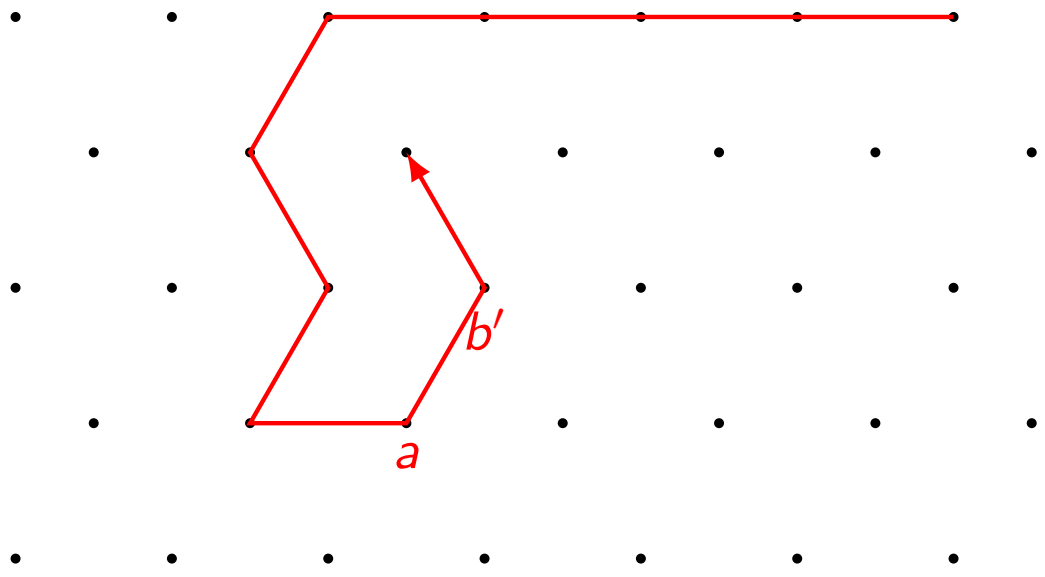
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



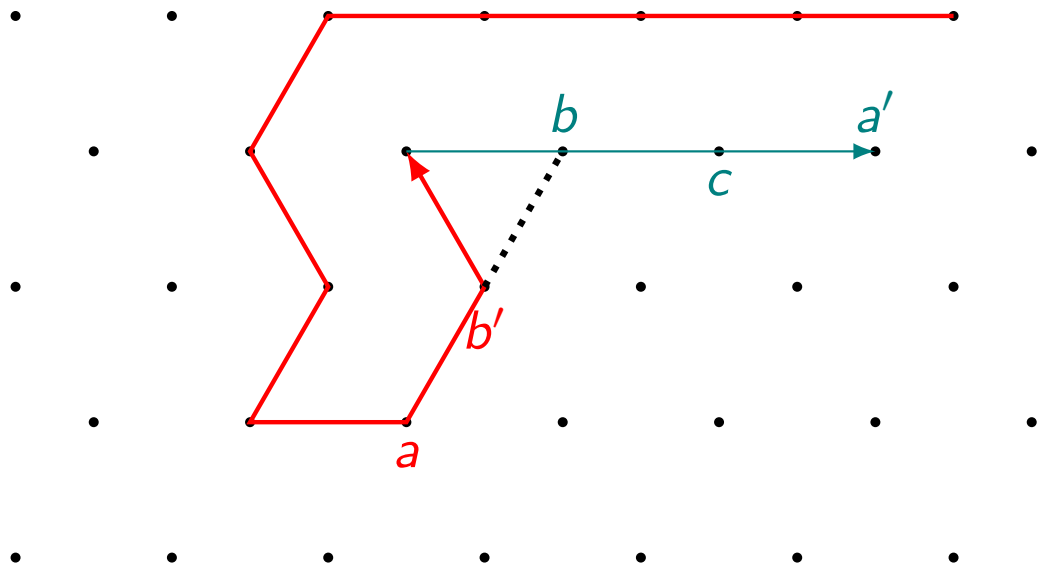
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



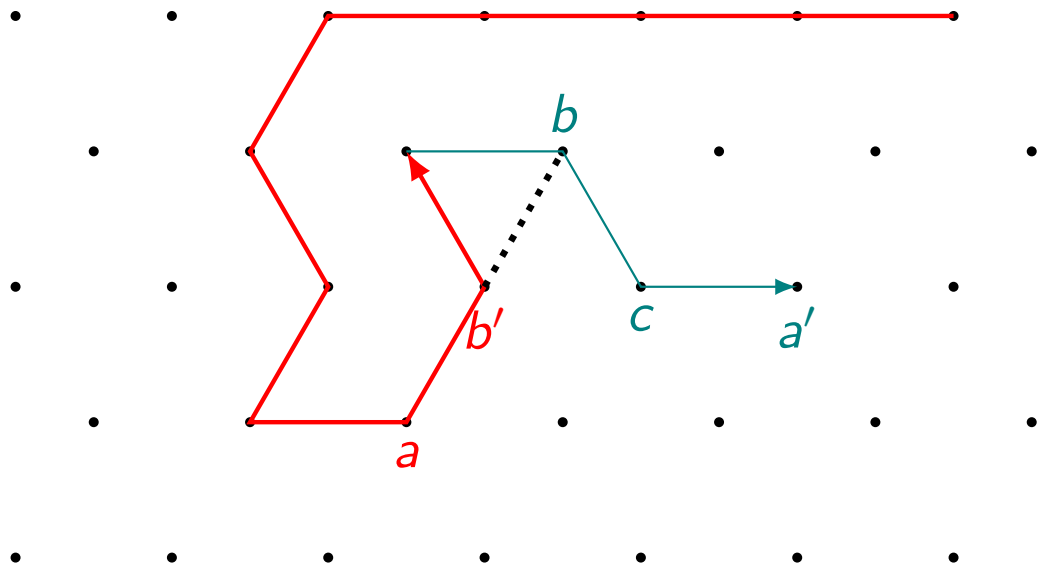
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



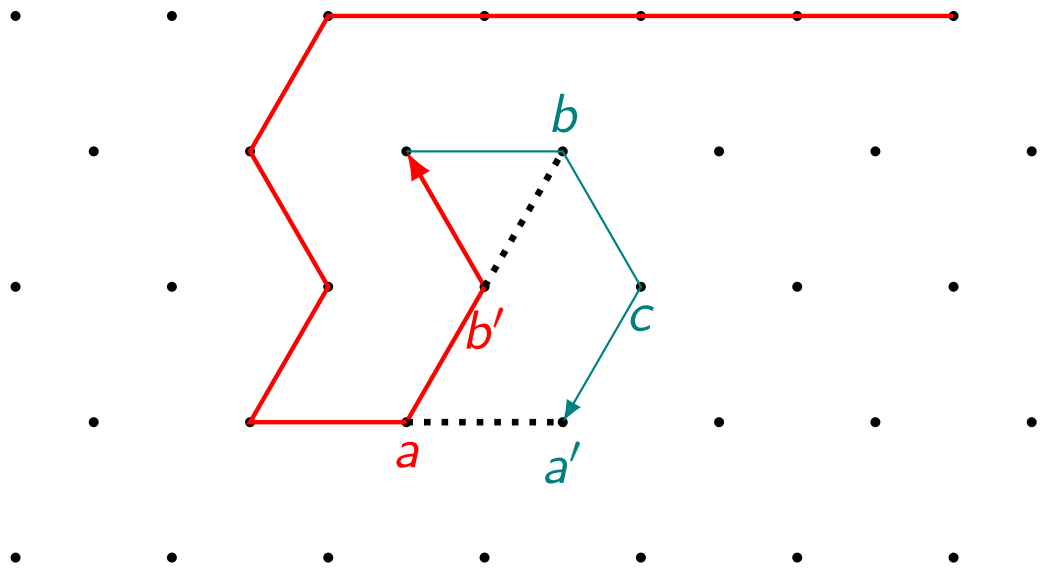
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



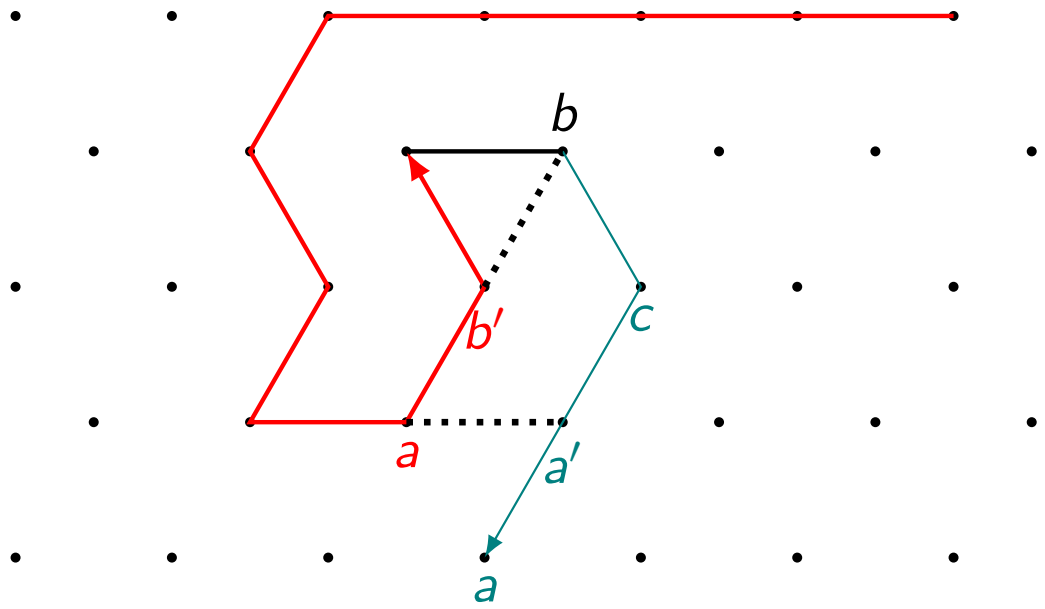
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



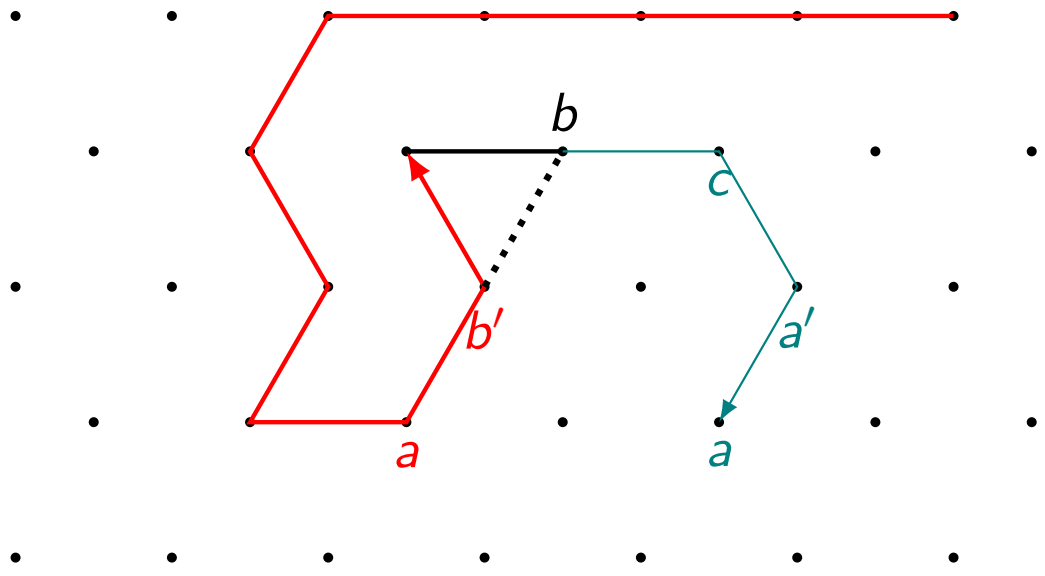
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



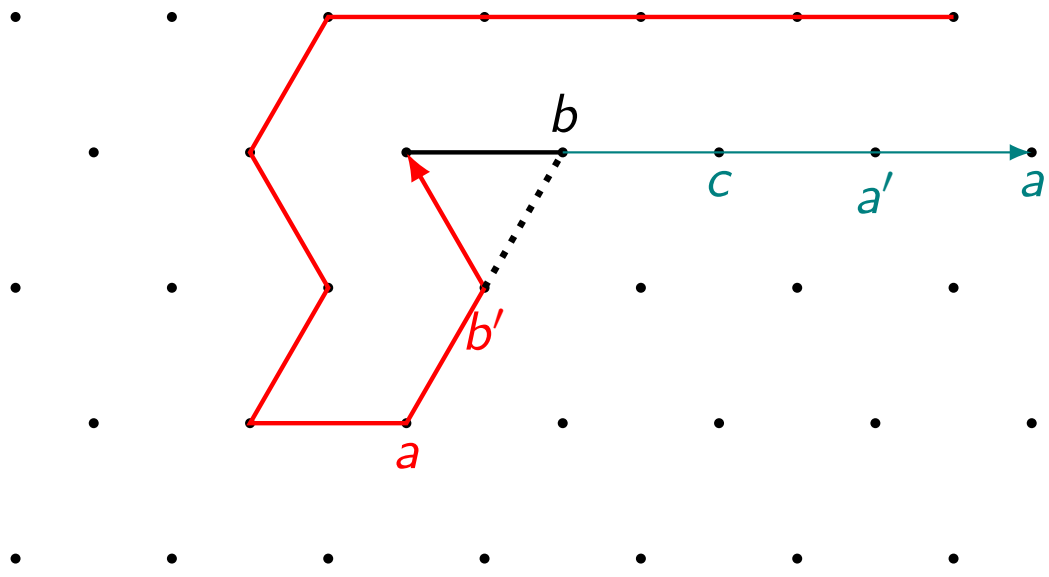
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



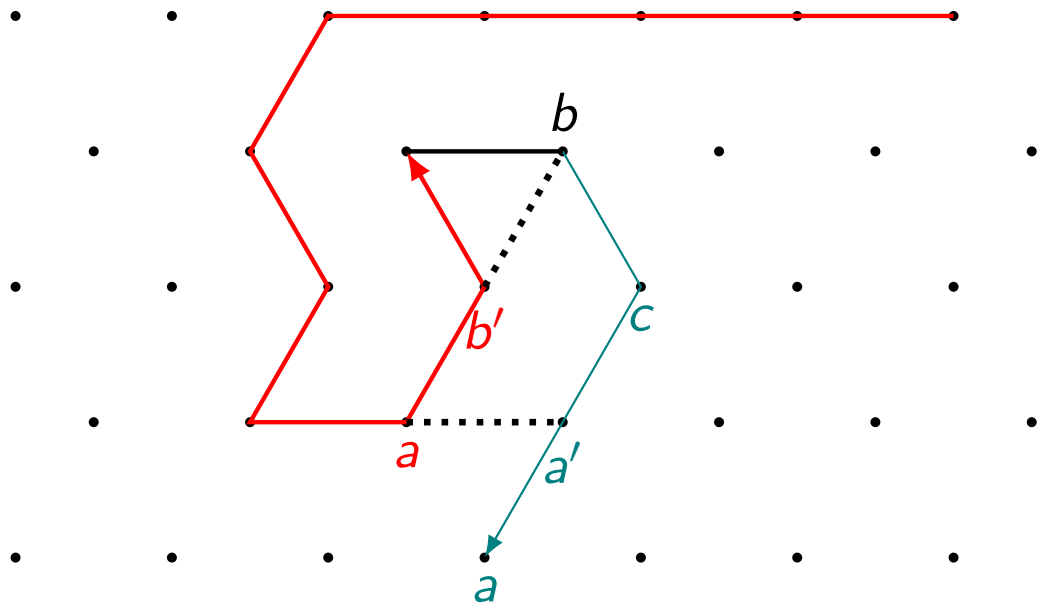
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



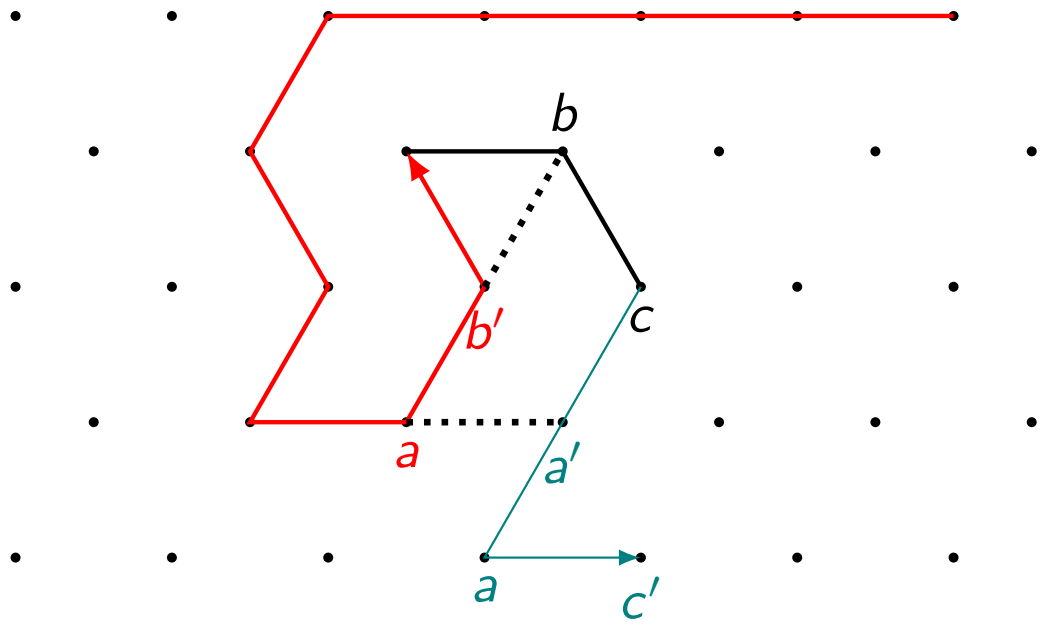
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



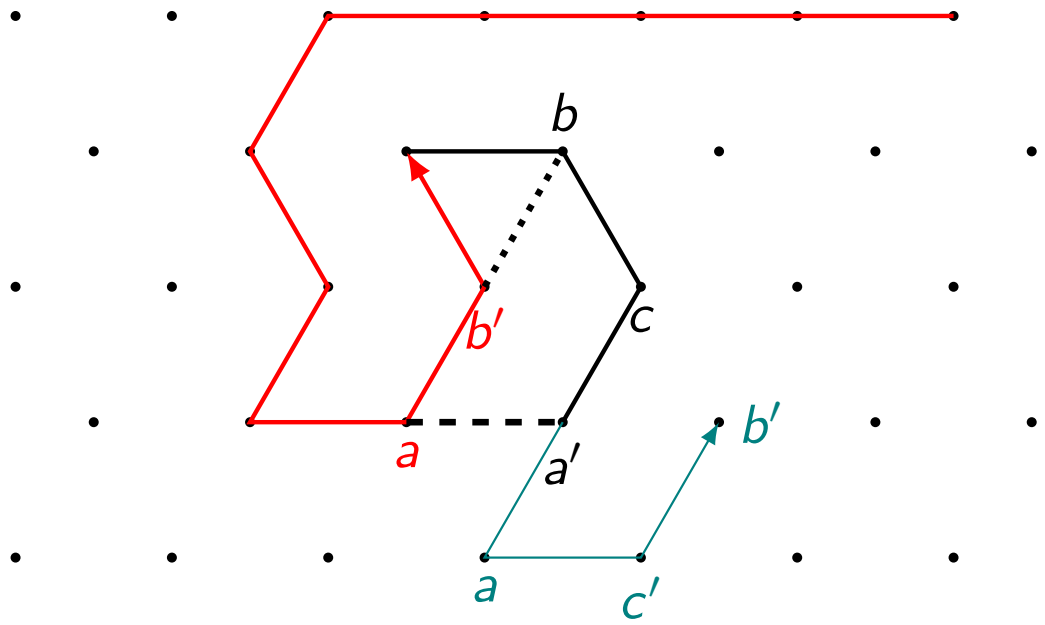
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



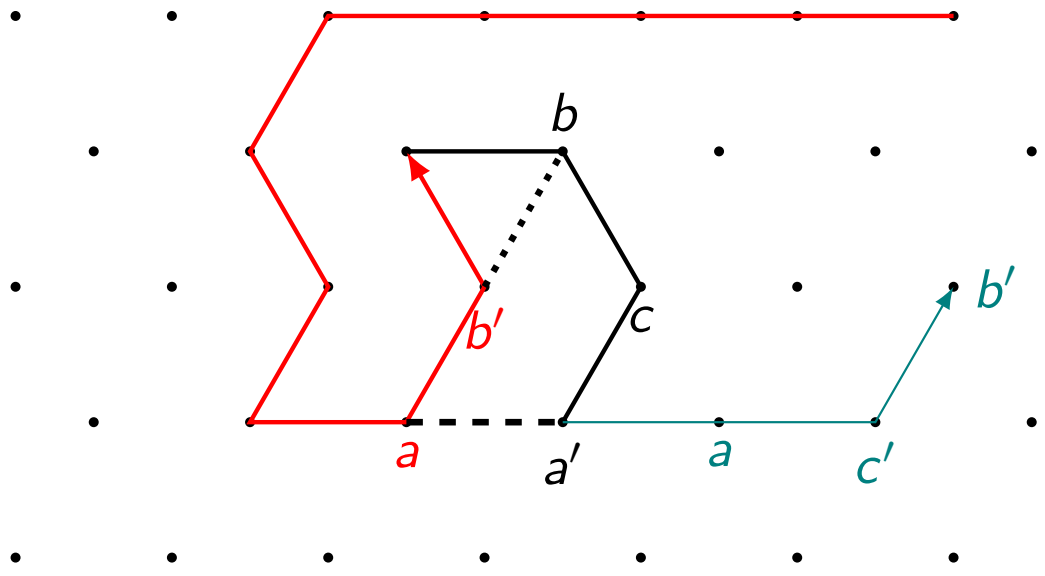
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



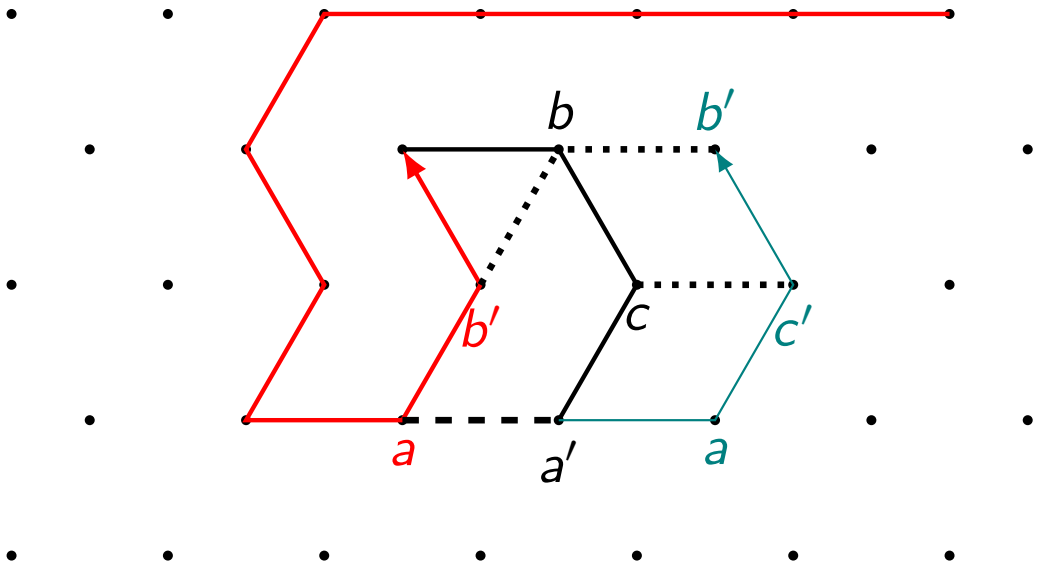
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



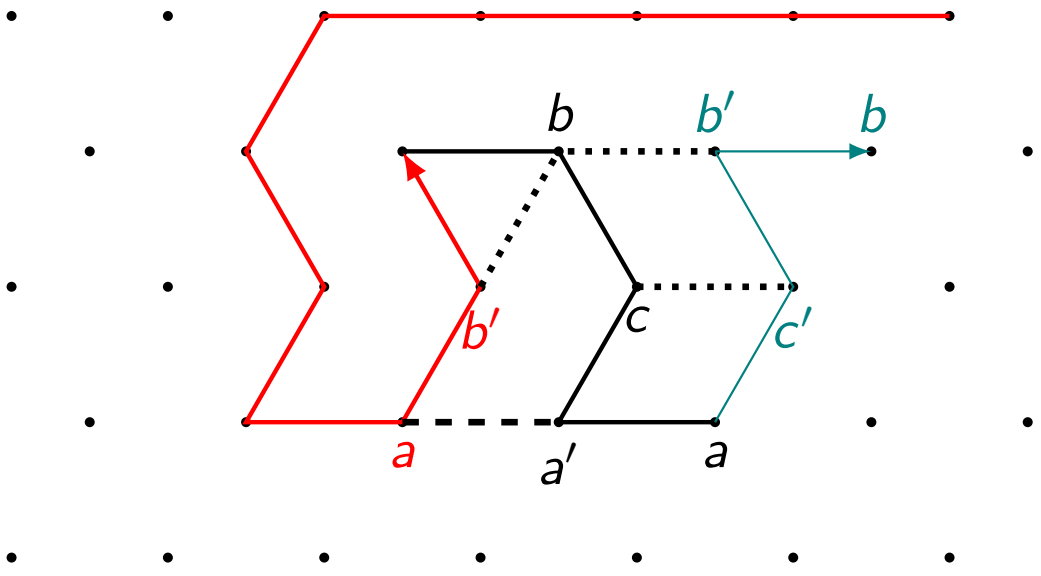
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



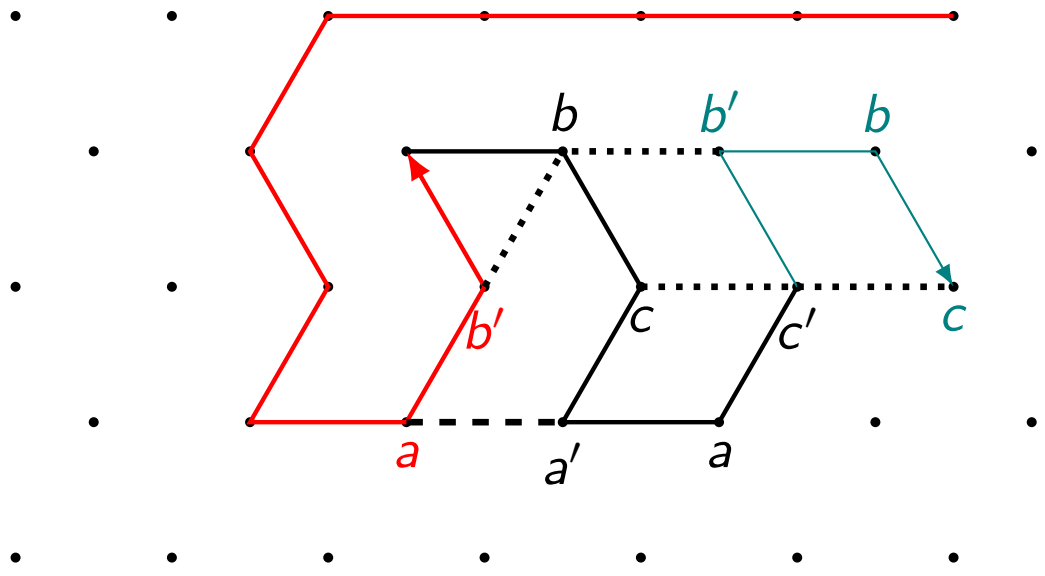
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



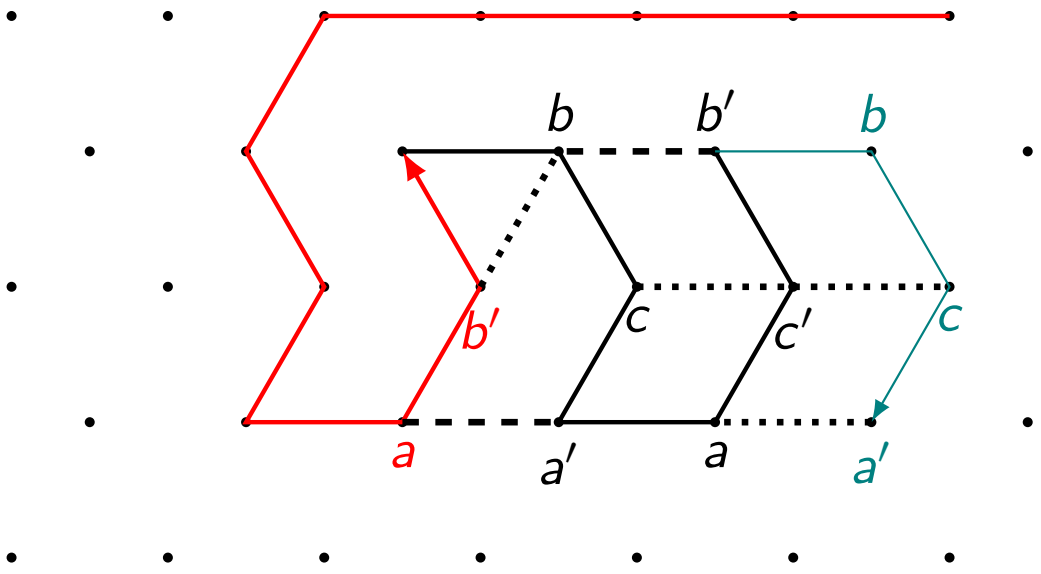
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



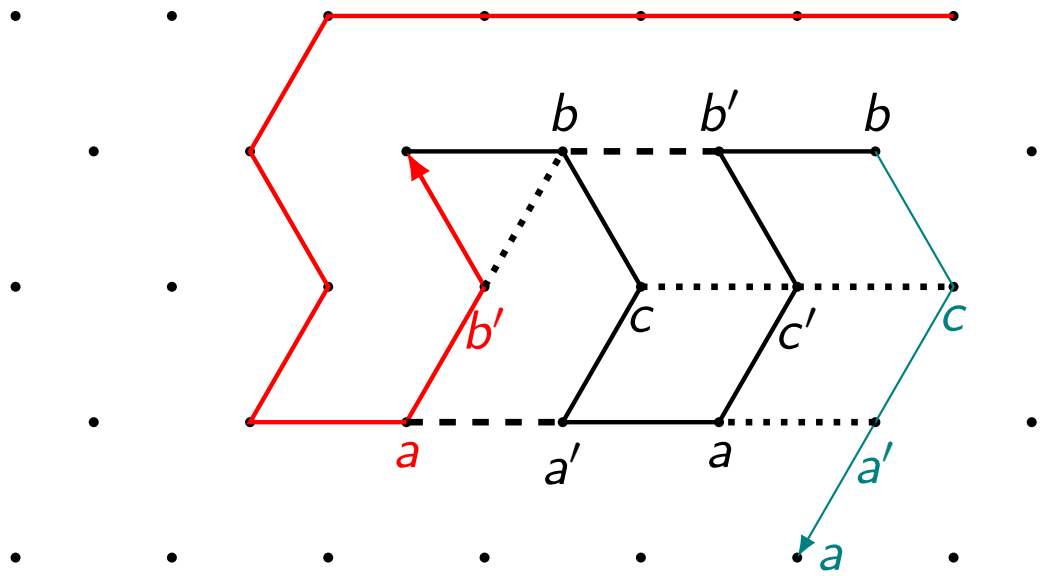
Oritatami System

Dynamics

Glider

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



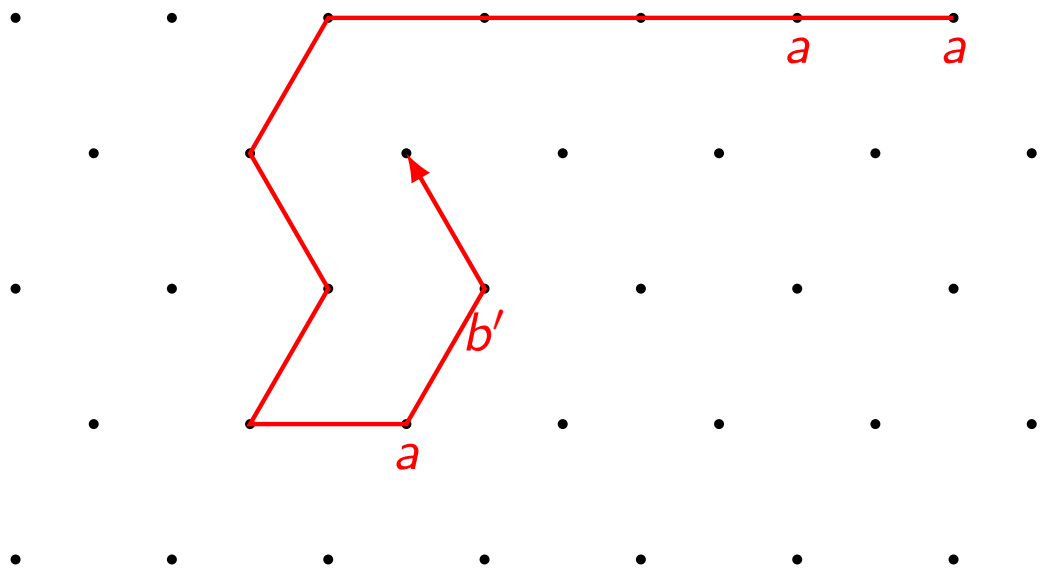
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for ($i = 0$, $i \leq |w|$, $i = i + 1$) {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



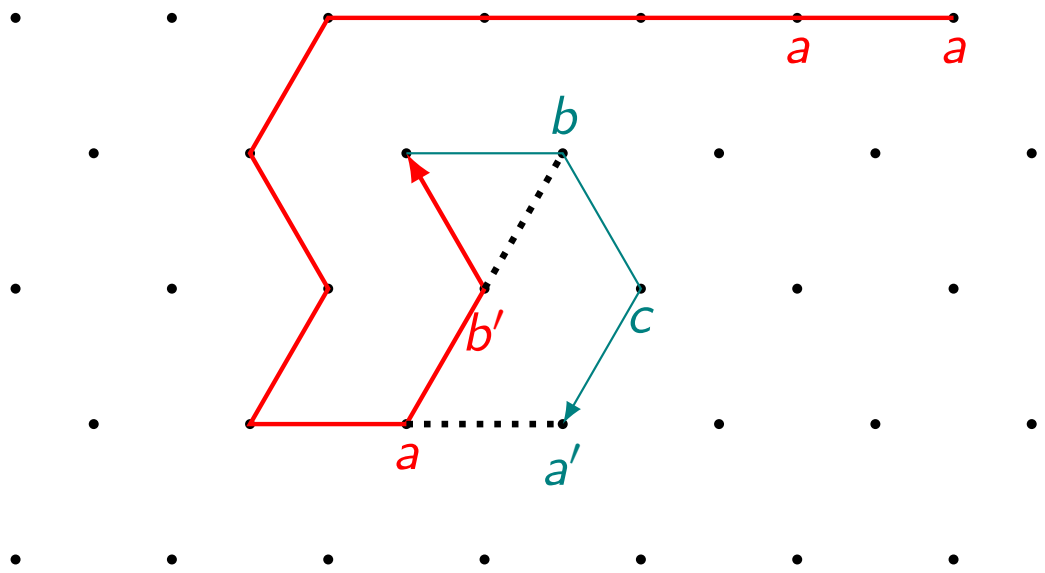
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



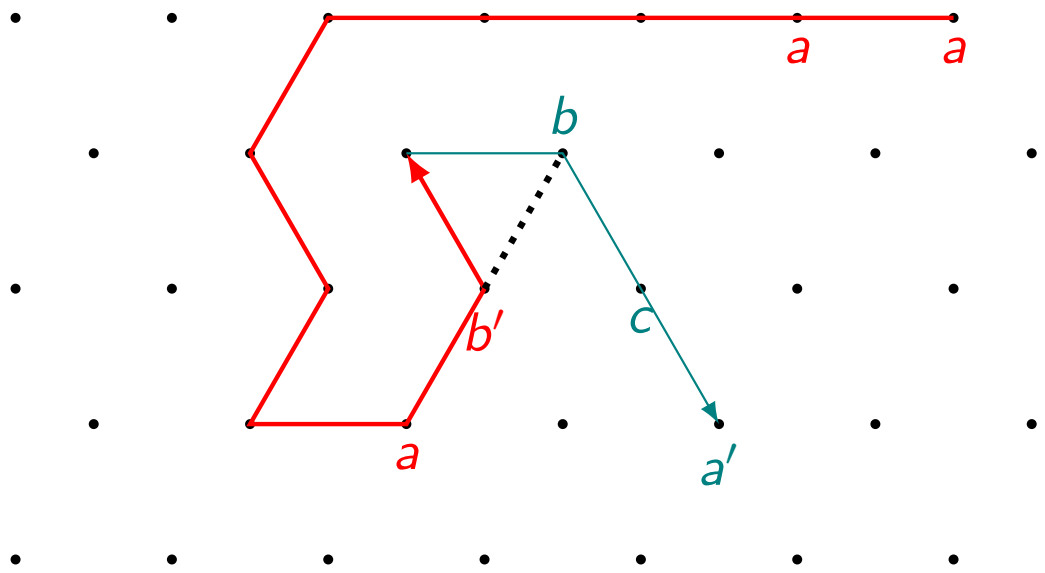
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



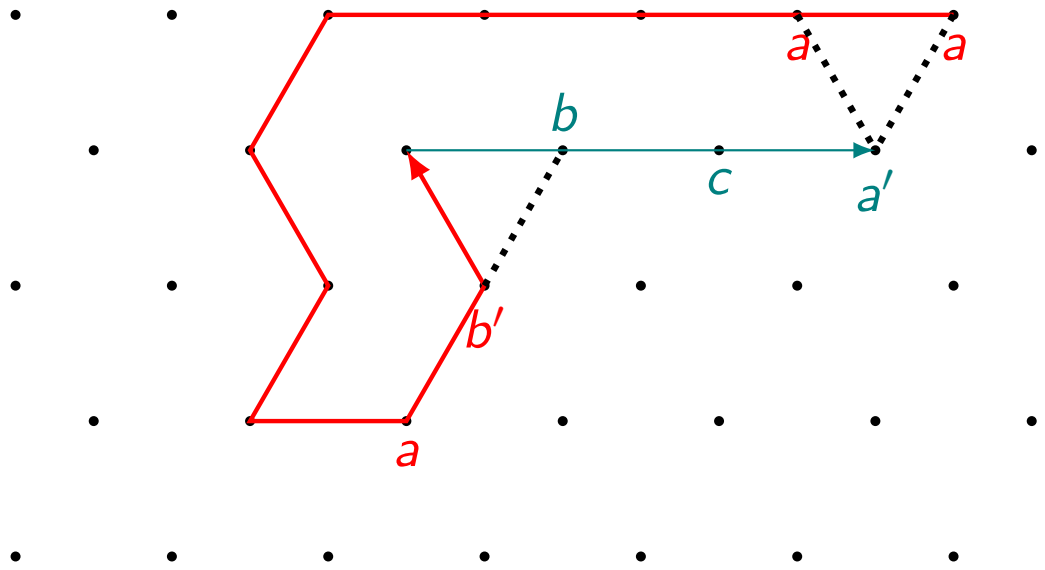
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



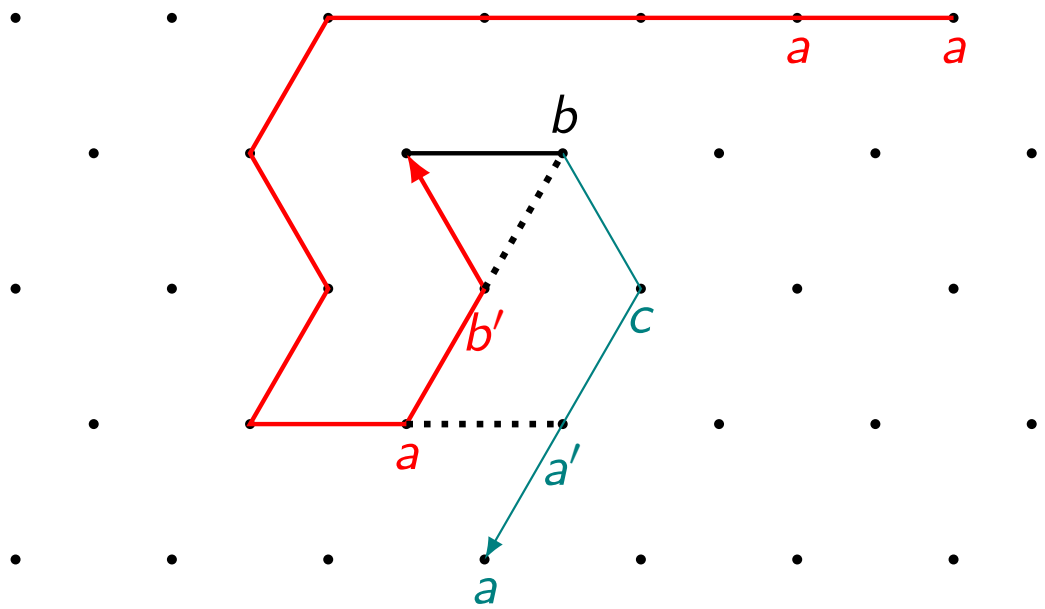
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



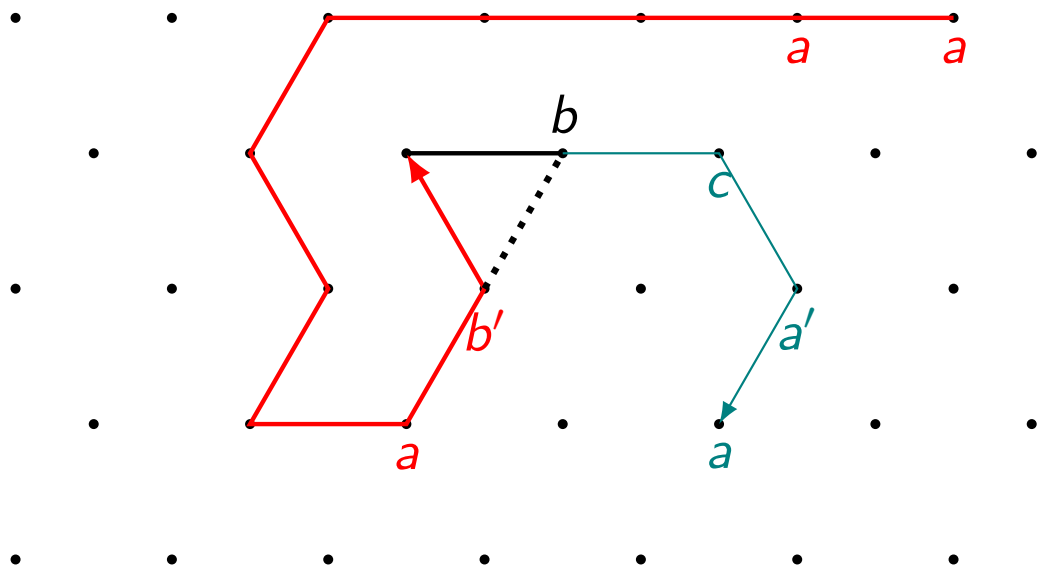
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



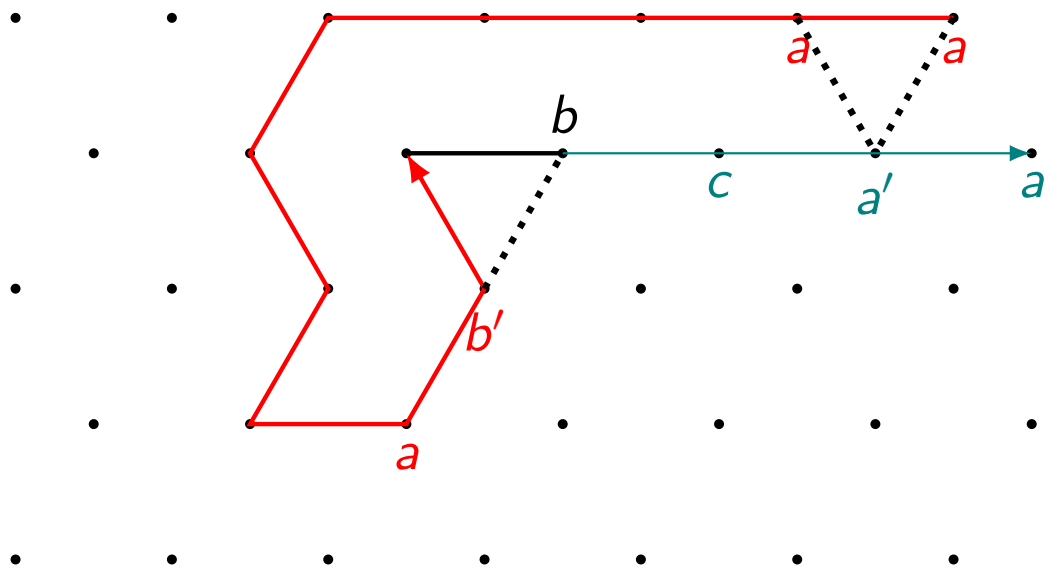
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



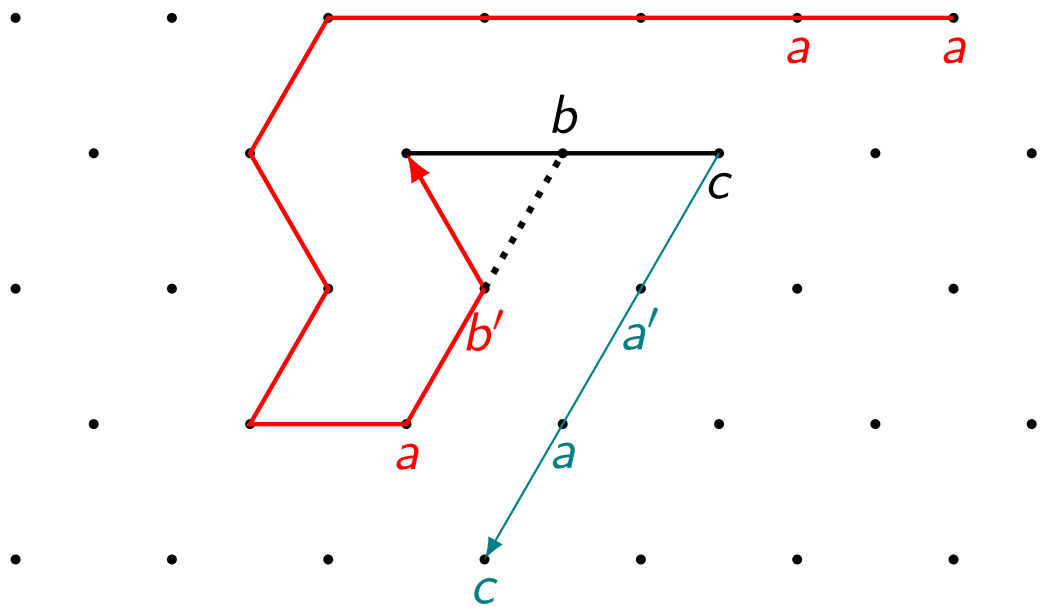
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



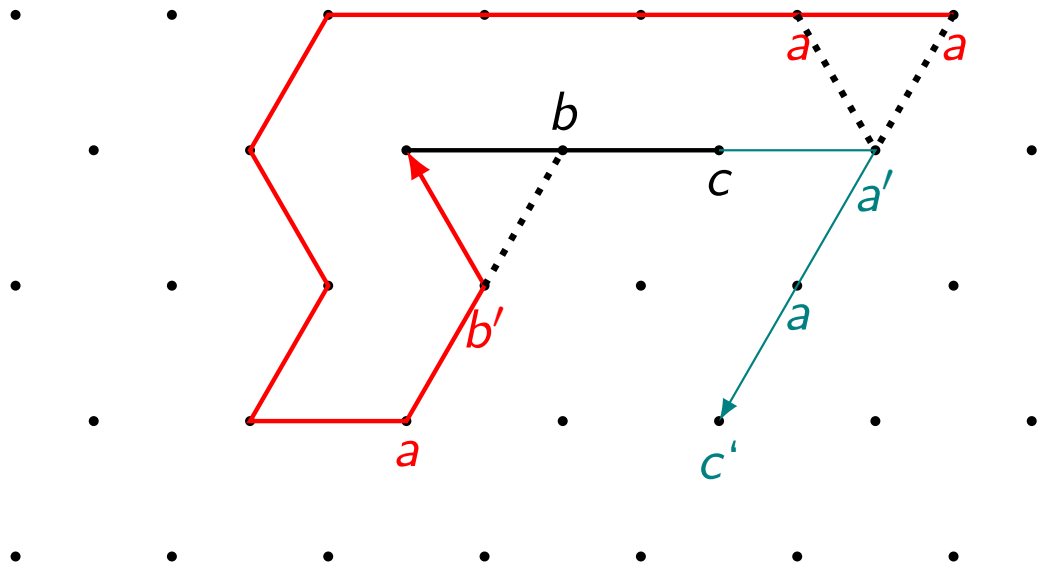
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



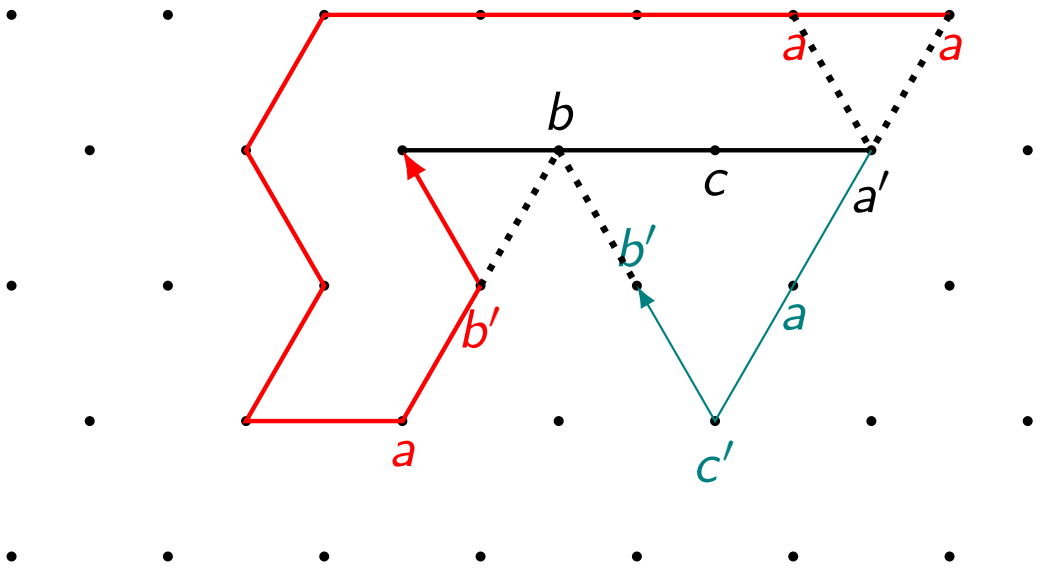
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



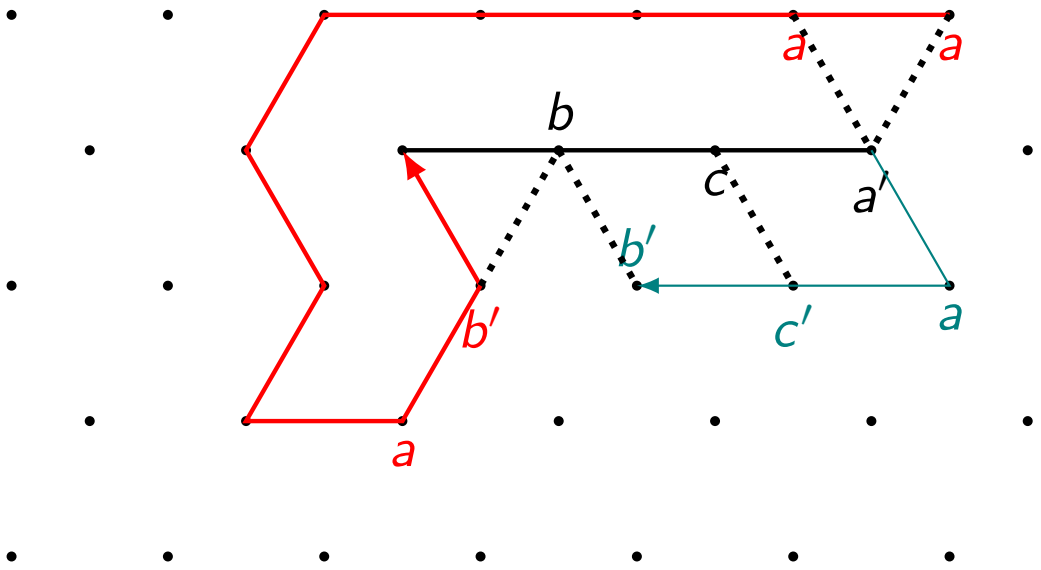
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



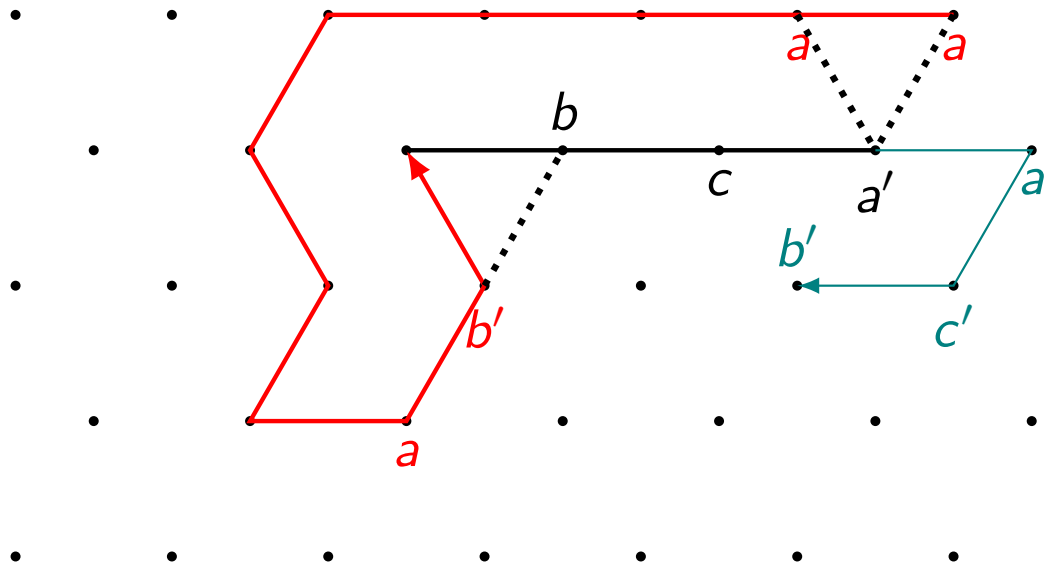
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



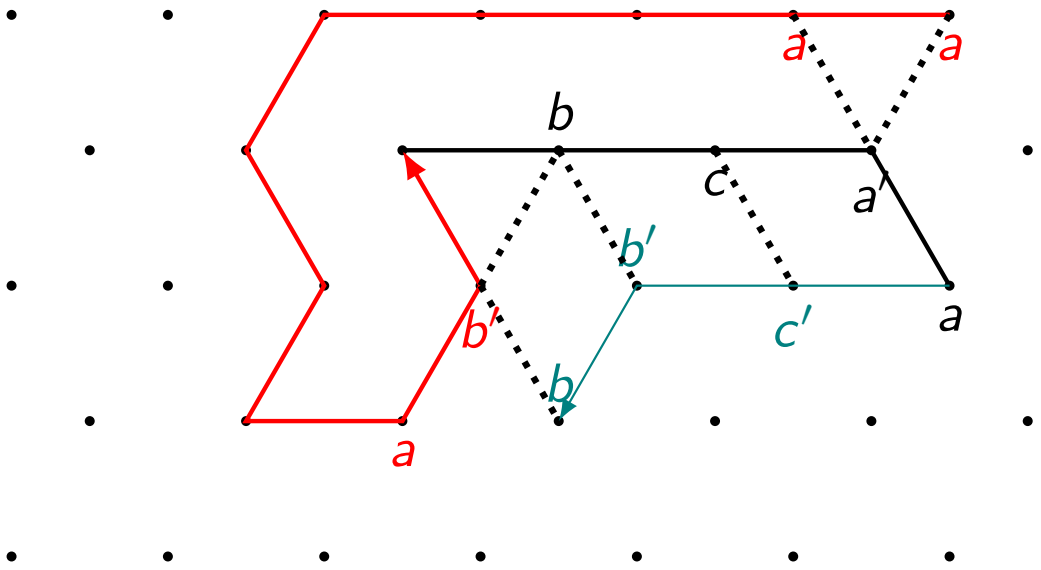
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



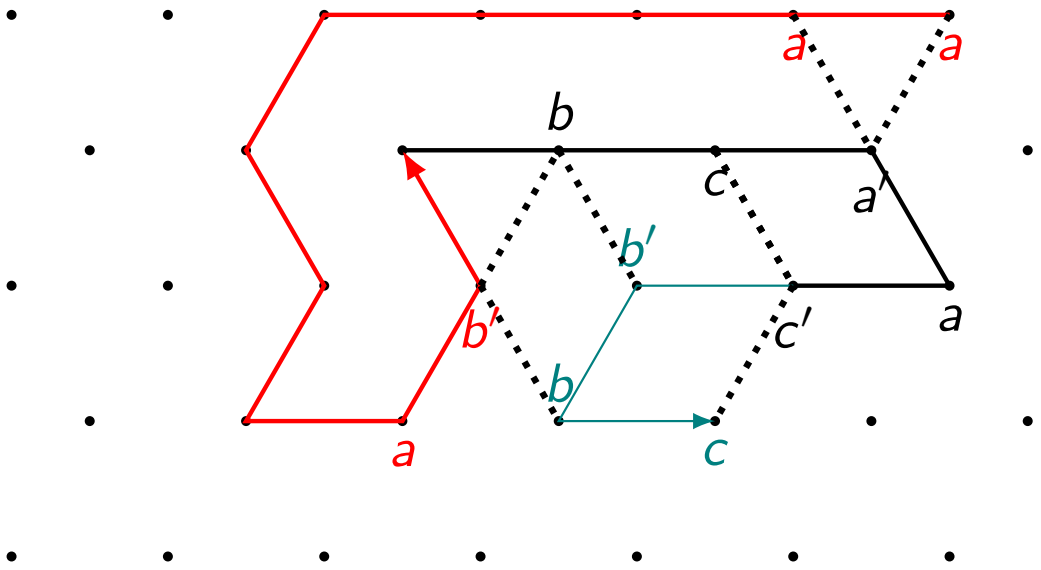
Oritatami System

Dynamics

Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



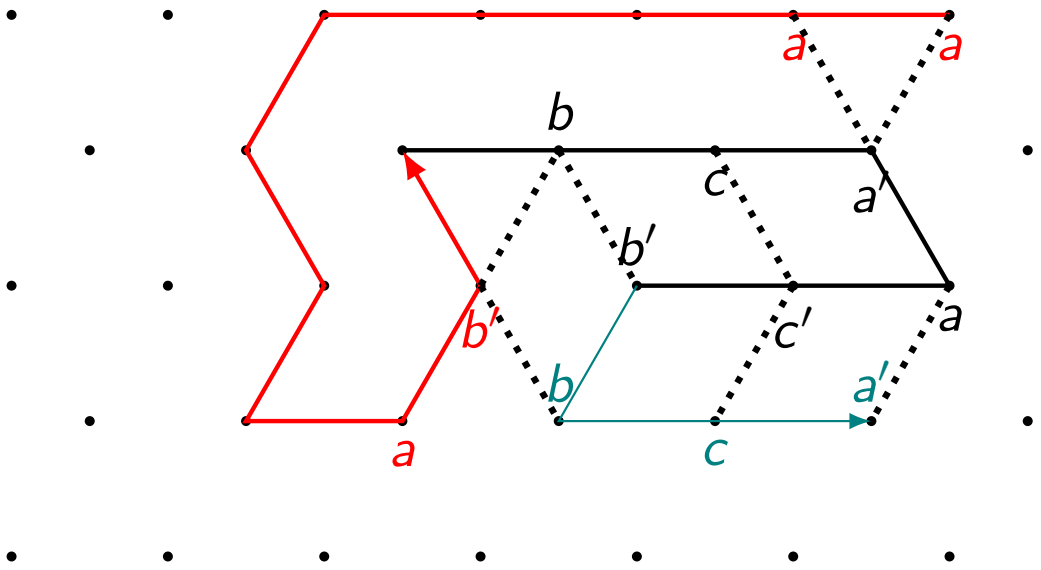
Oritatami System

Dynamics

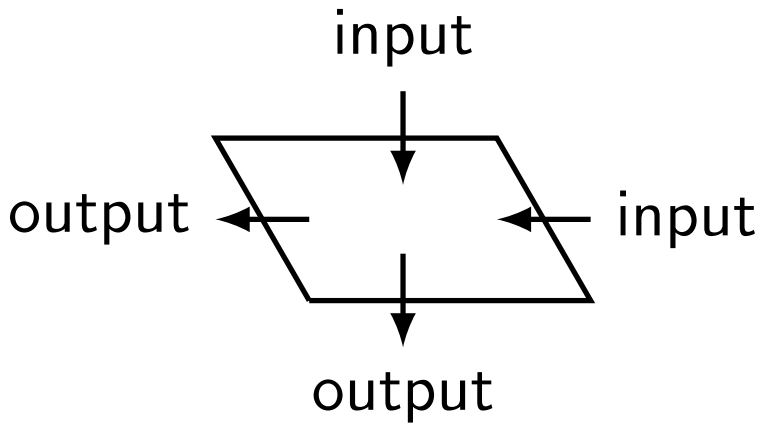
Glider (Context-sensitive)

Consider an OS with transcript $w = (bca'ac'b')^*$, rule set $R = \{(a, a'), (b, b'), (c, c')\}$, delay $\delta = 3$, and the seed below colored in red.

- Transcribe $w[1..\delta]$ at the 3'-end of the seed.
- for $(i = 0,\ i \leq |w|,\ i = i + 1)$ {
  1. Fold $w[i..i + \delta - 1]$ in all geometrically-possible manners.
  2. Stabilize $w[i]$ so as for $w[i..i + \delta - 1]$ to form as many bonds as possible.
  3. Transcribe $w[i + \delta]$ and $i = i + 1$.}



Module *[Geary et al. 2016]*



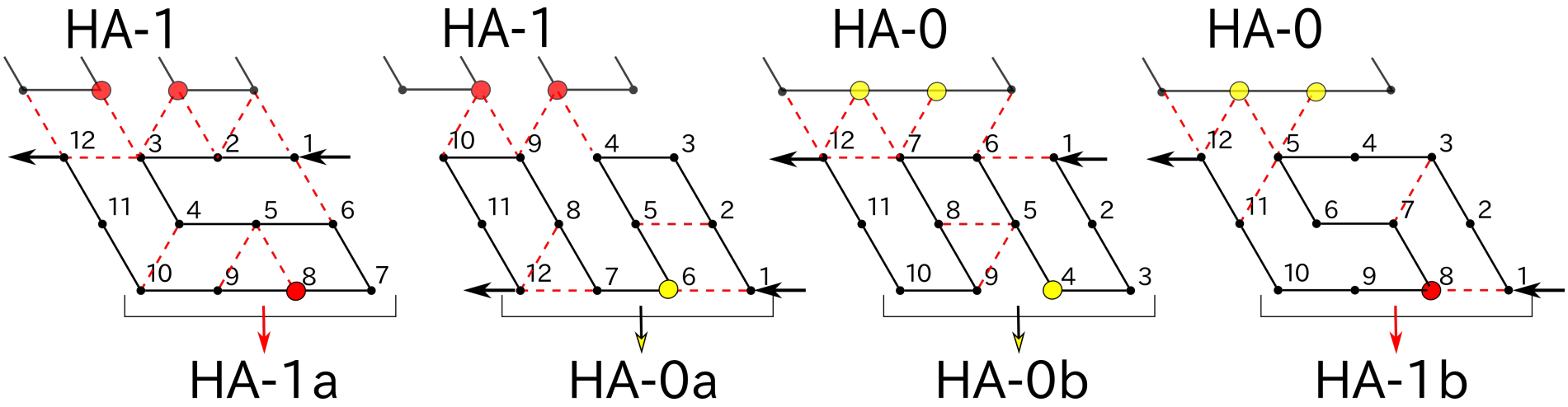
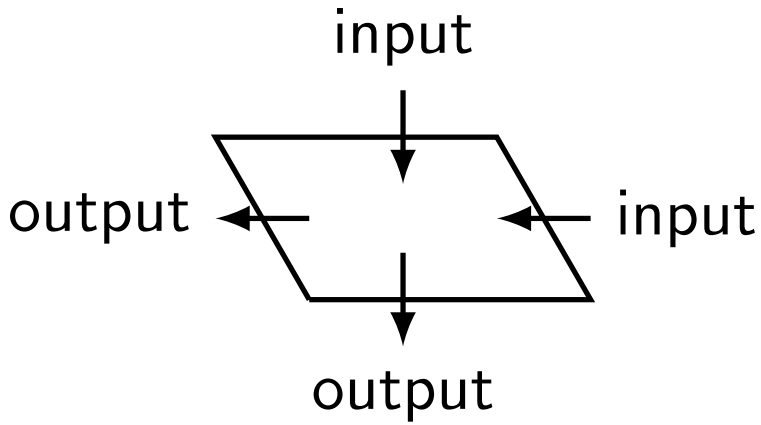


Figure: Four bricks of Half Adder *module*

Module *[Geary et al. 2016]*



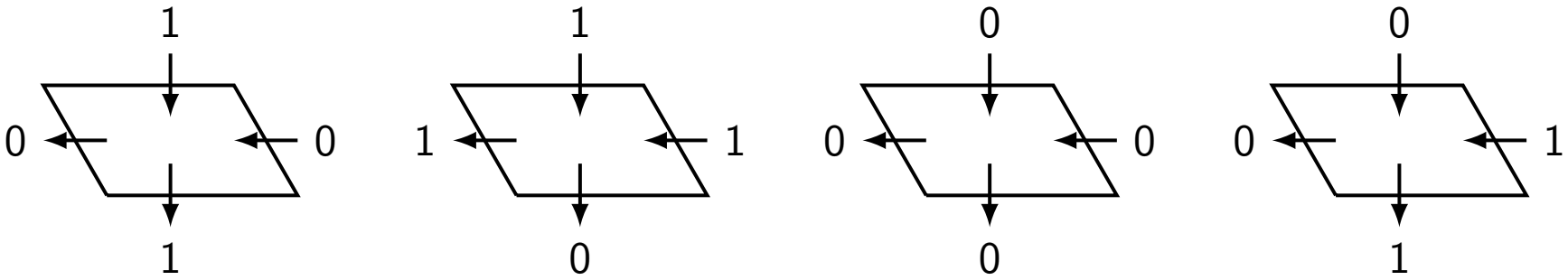
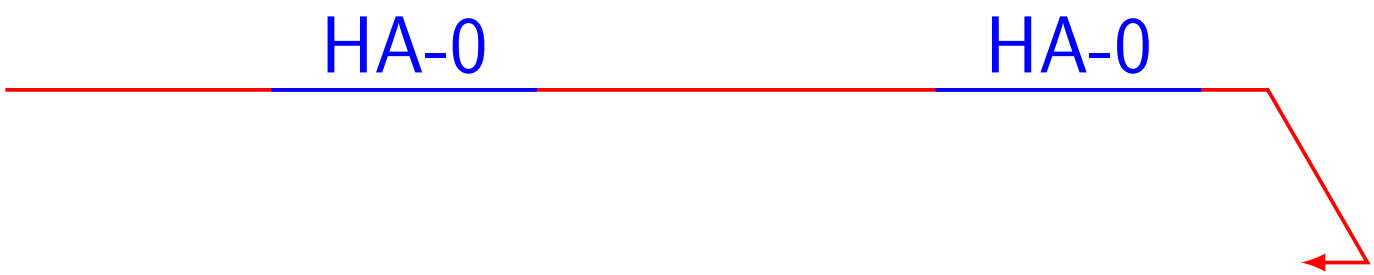
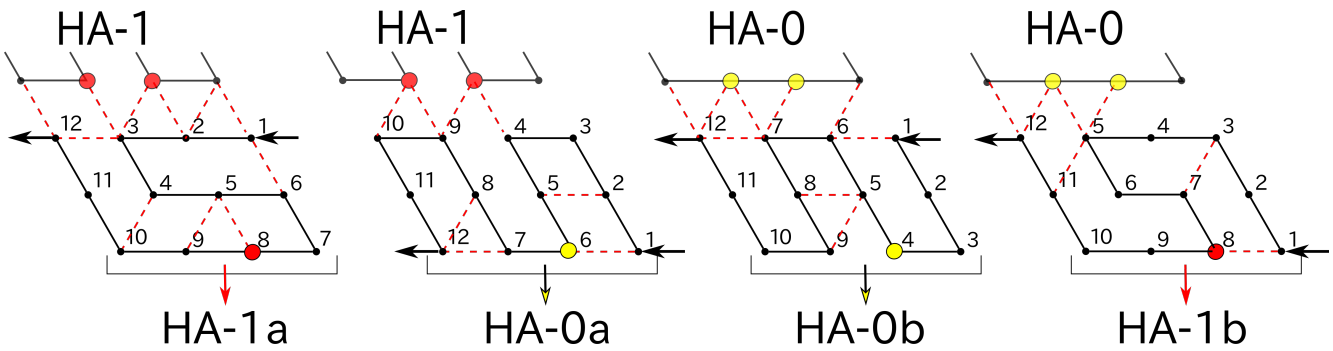


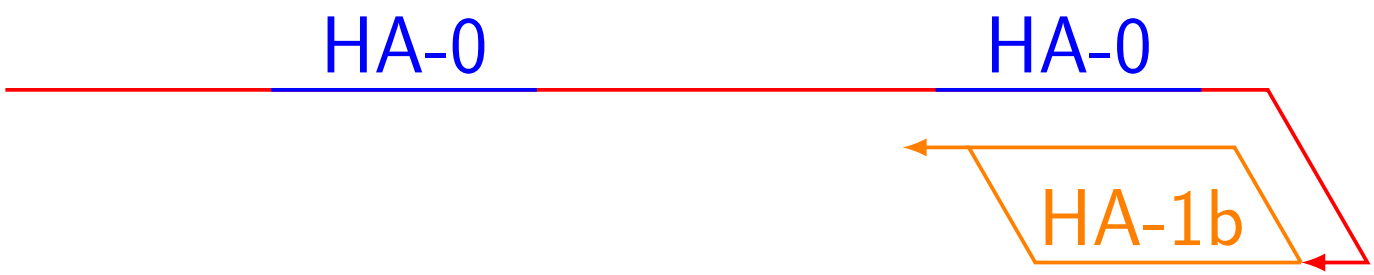
Figure: Four bricks of Half Adder *module*

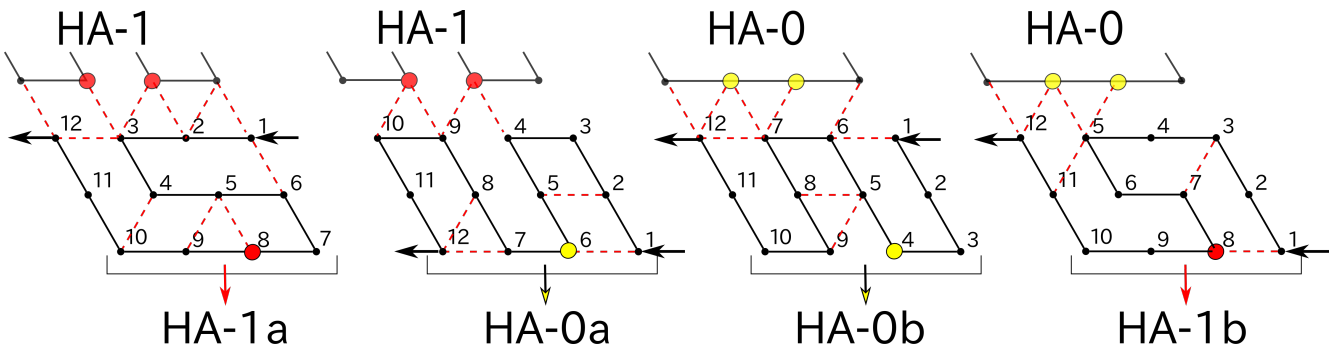
Module *[Geary et al. 2016]*



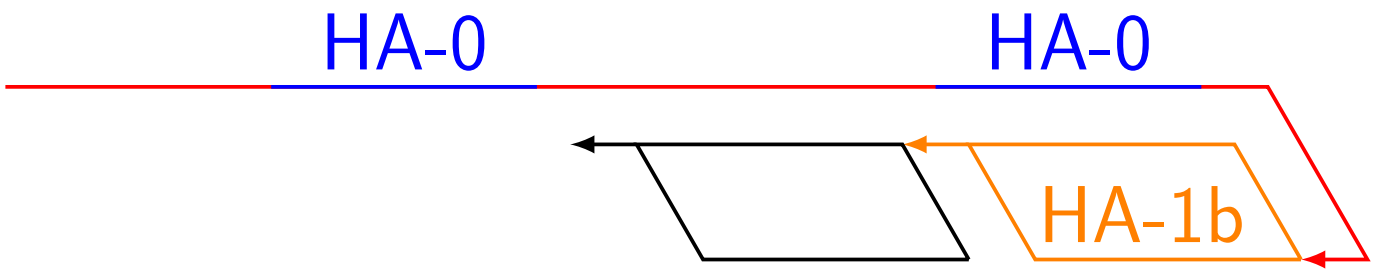


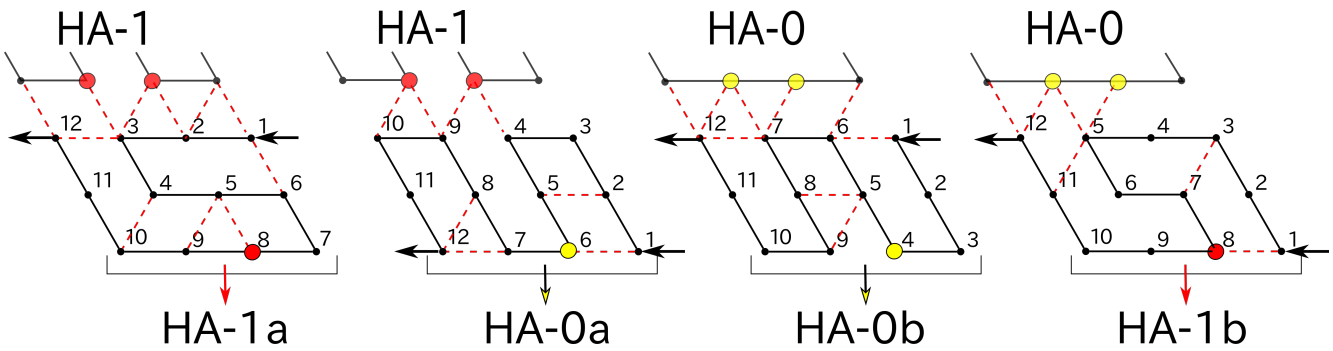
Module *[Geary et al. 2016]*



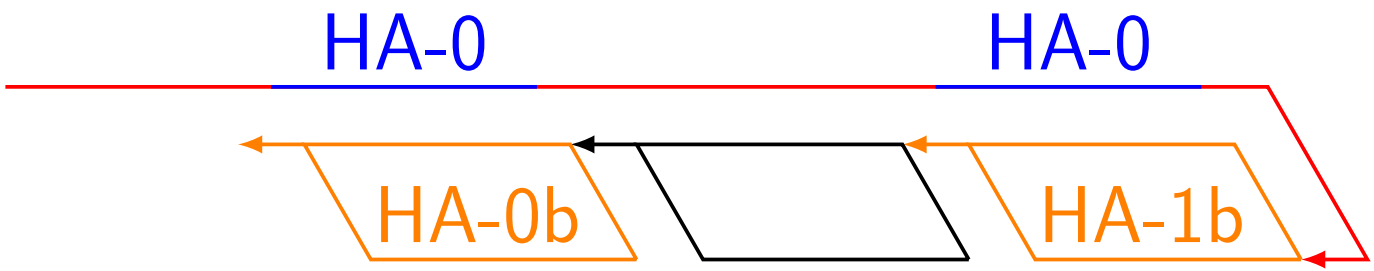


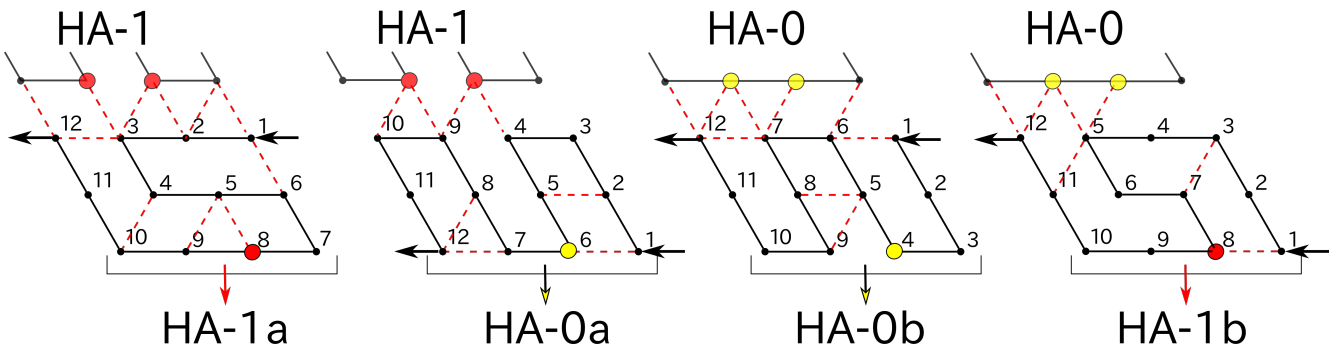
Module *[Geary et al. 2016]*



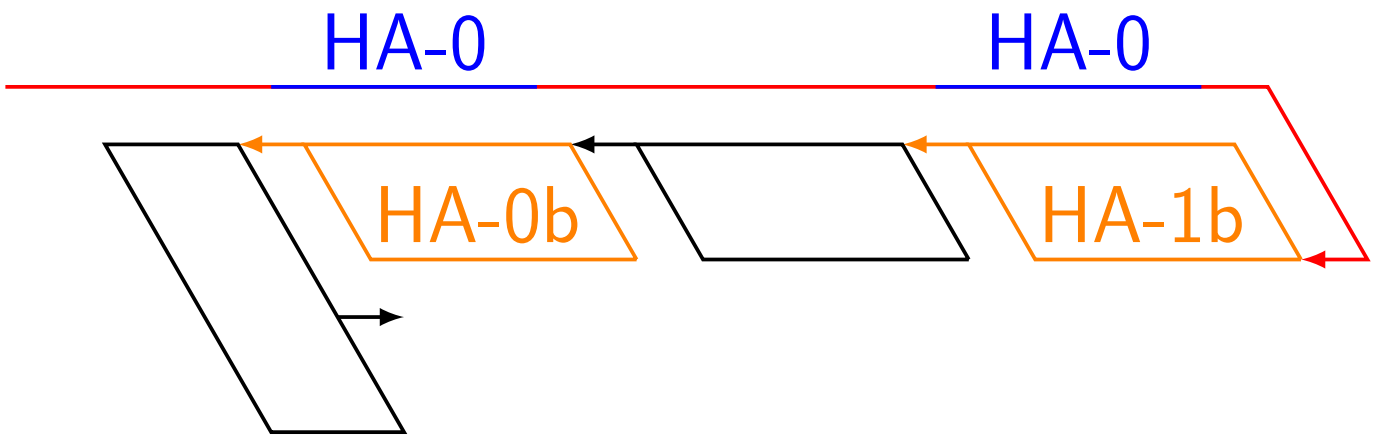


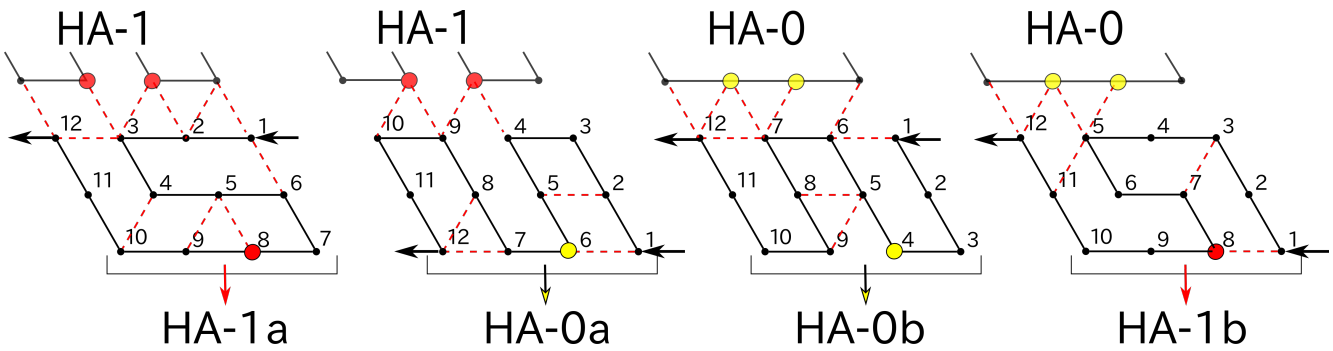
Module *[Geary et al. 2016]*



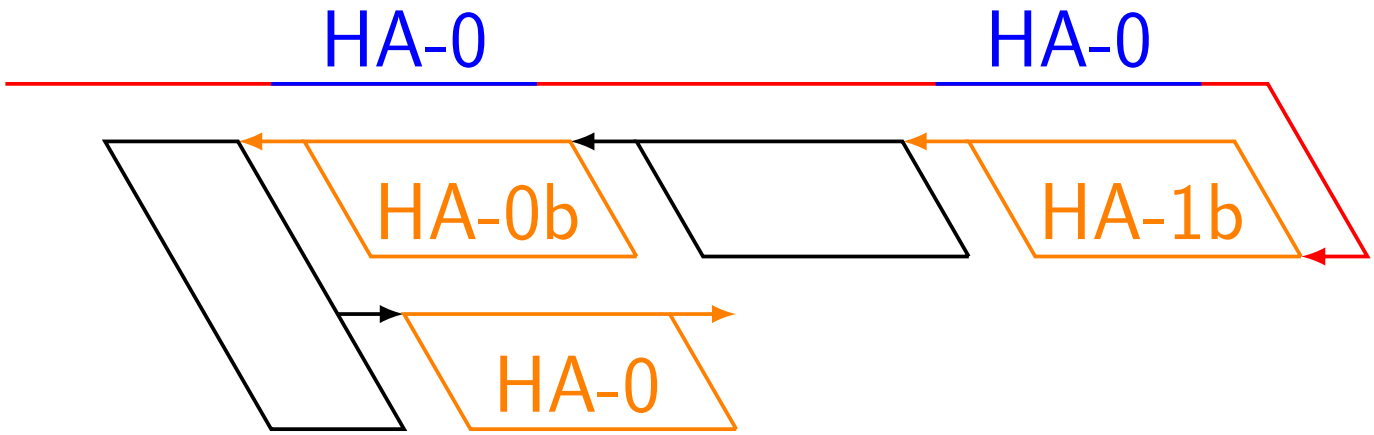


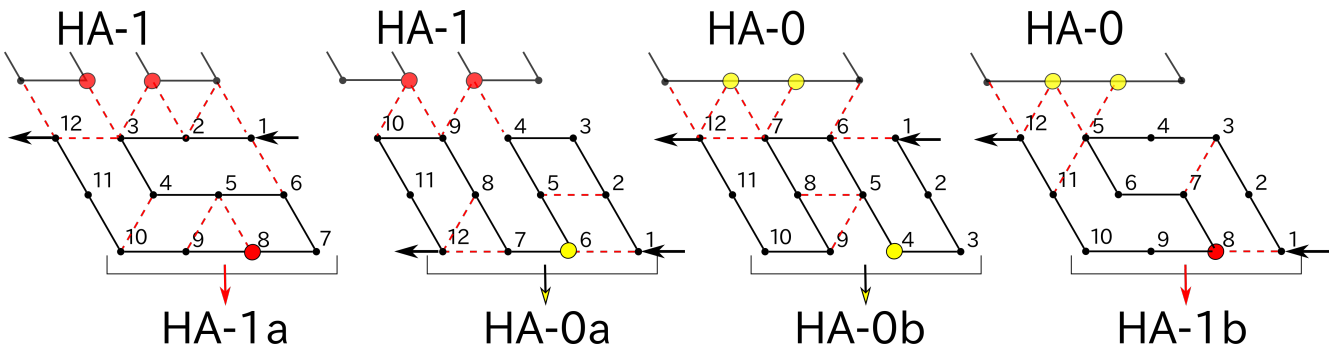
Module *[Geary et al. 2016]*



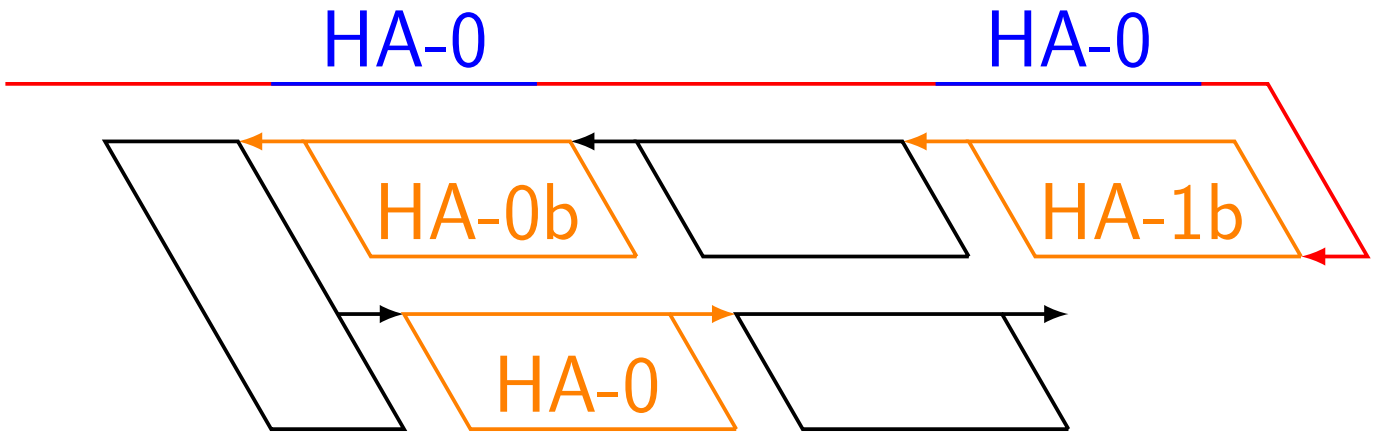


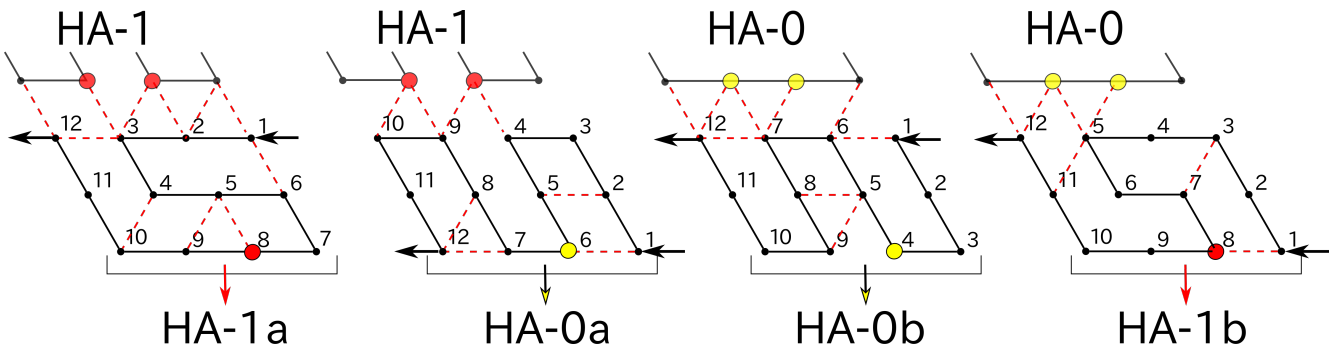
Module *[Geary et al. 2016]*



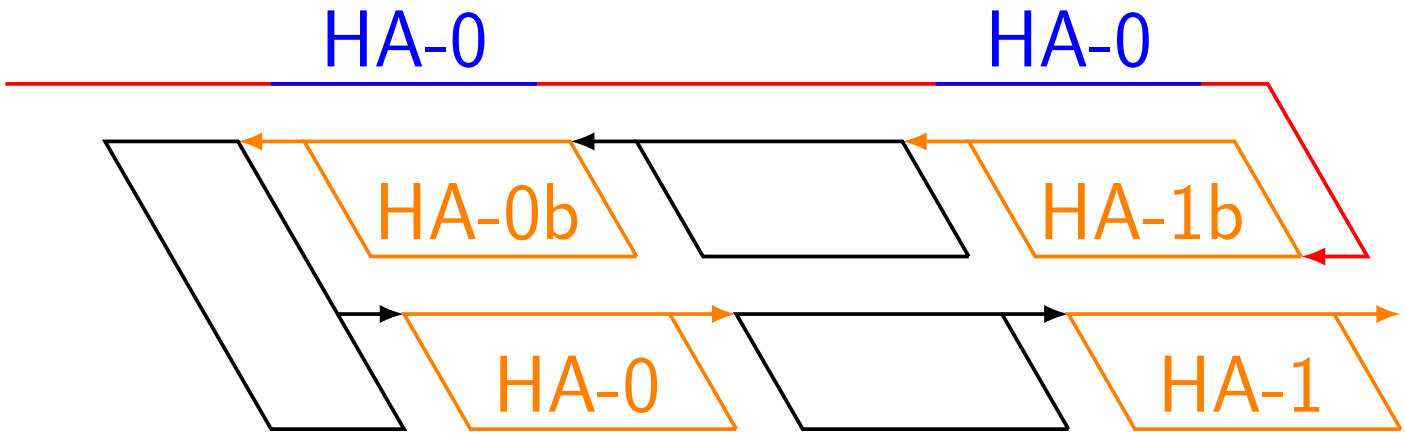


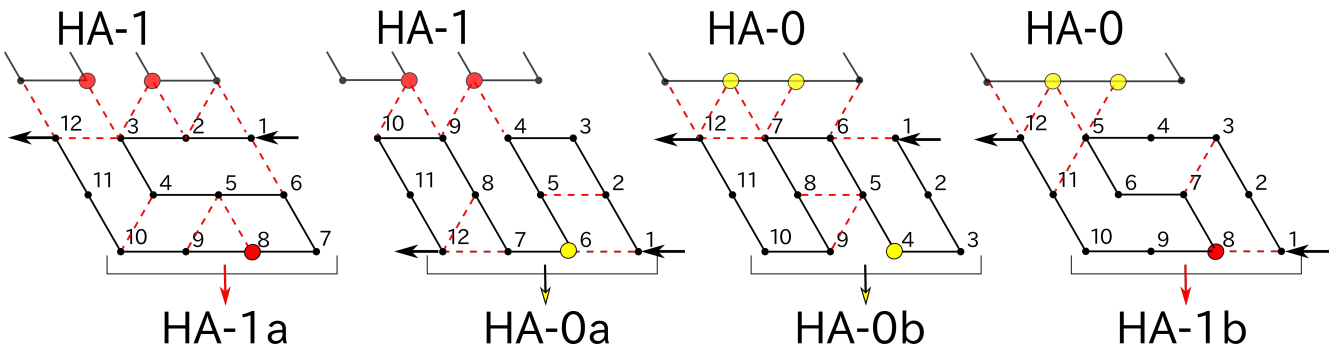
Module *[Geary et al. 2016]*



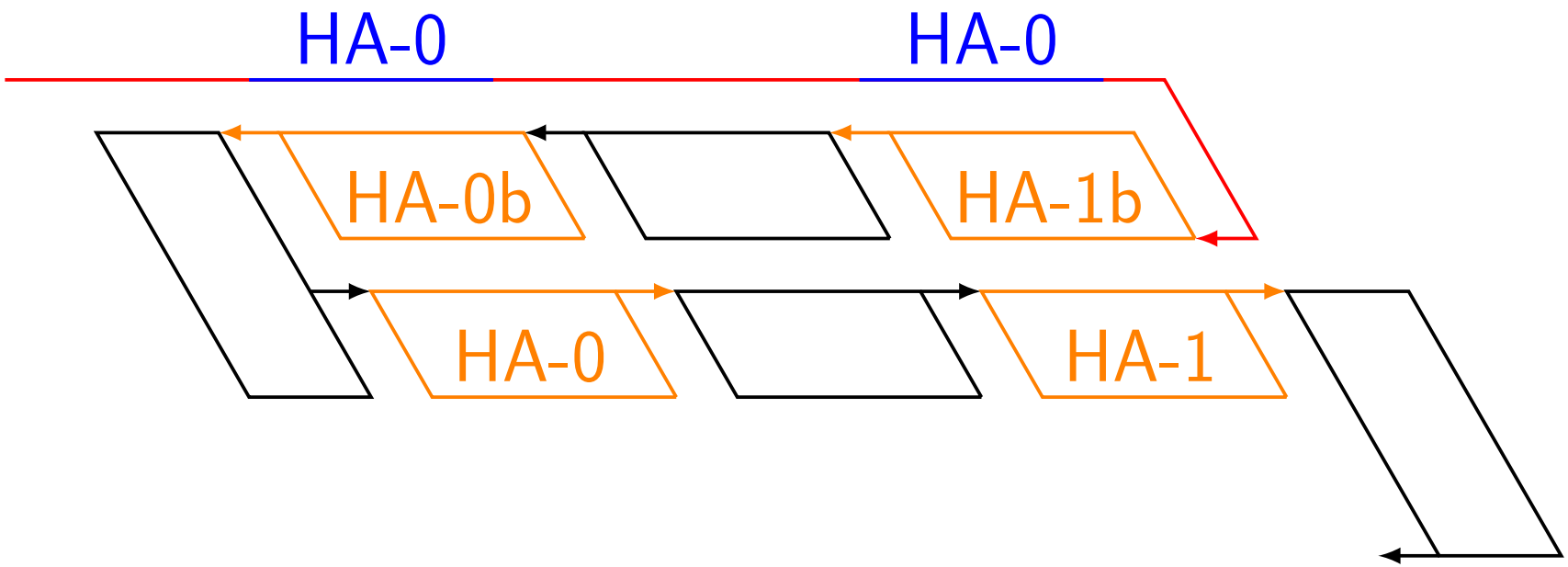


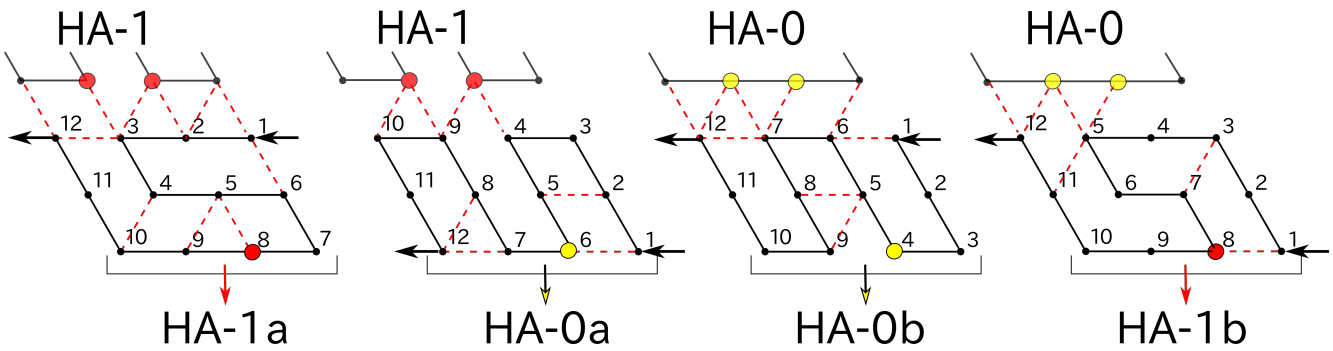
Module *[Geary et al. 2016]*



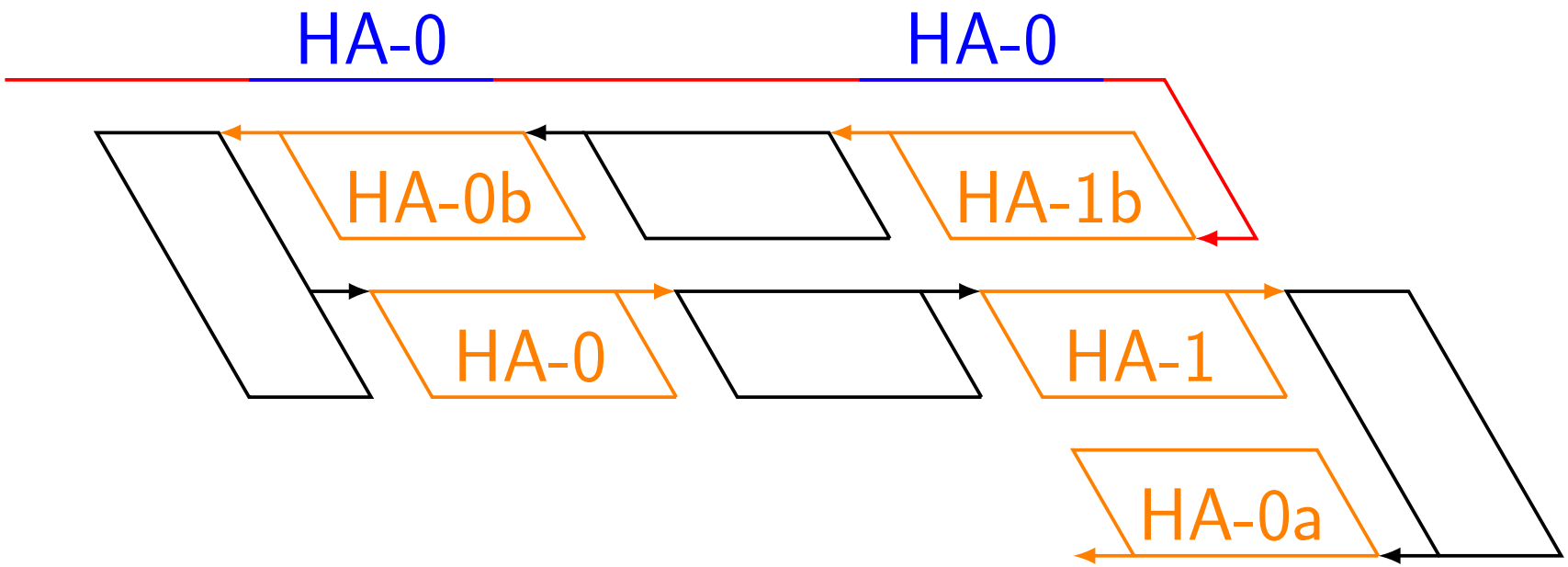


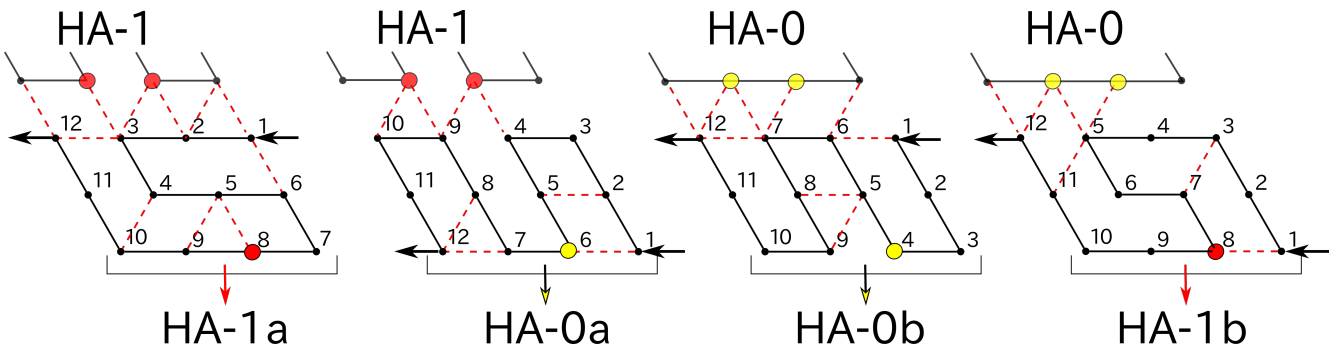
Module *[Geary et al. 2016]*



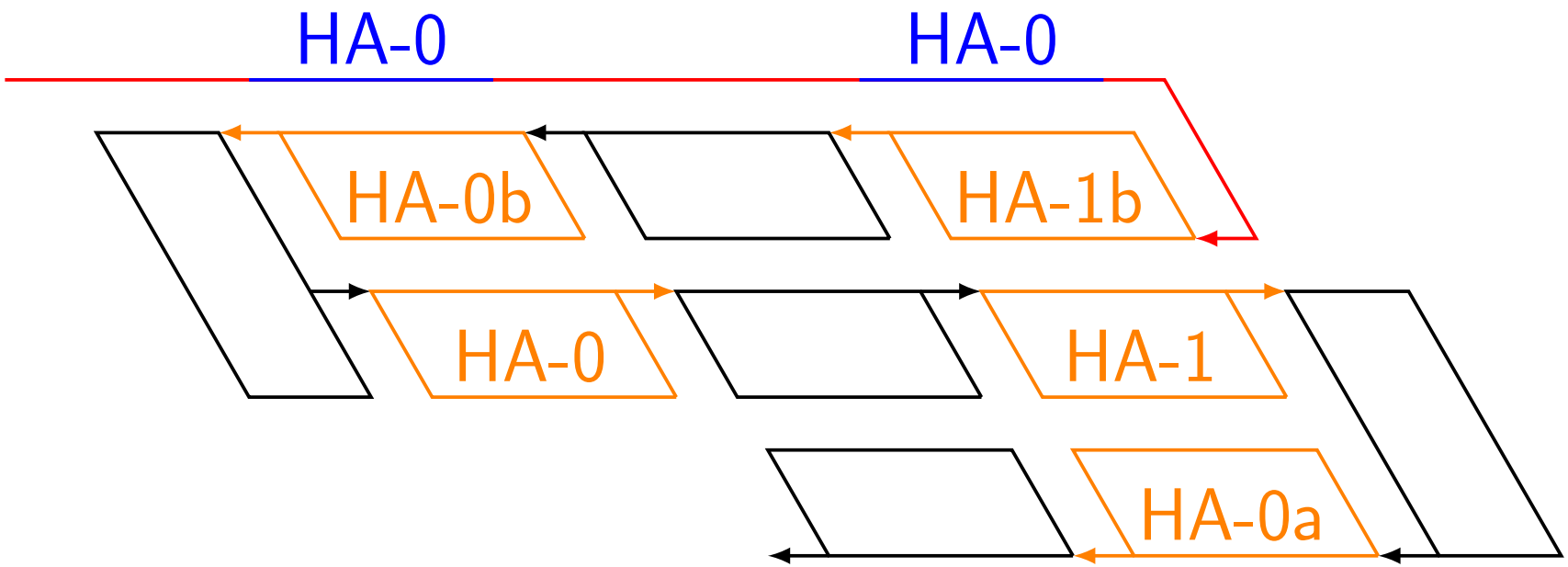


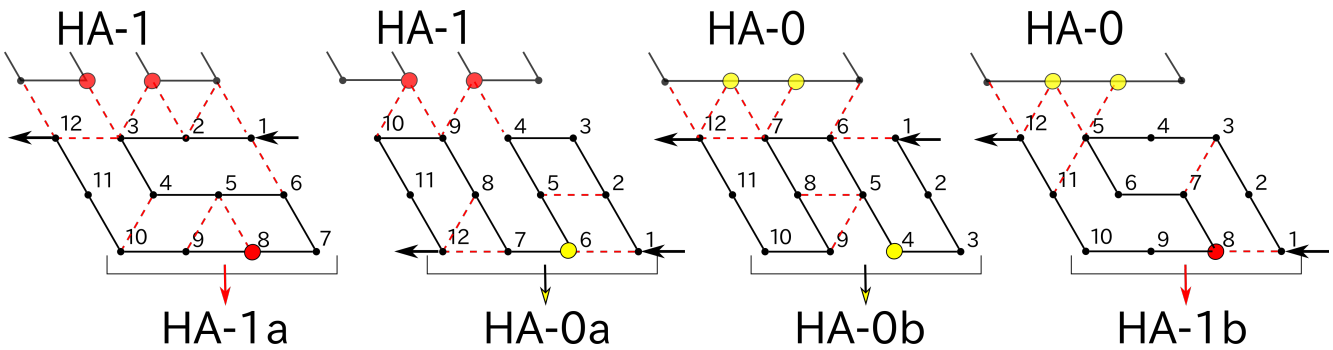
Module *[Geary et al. 2016]*



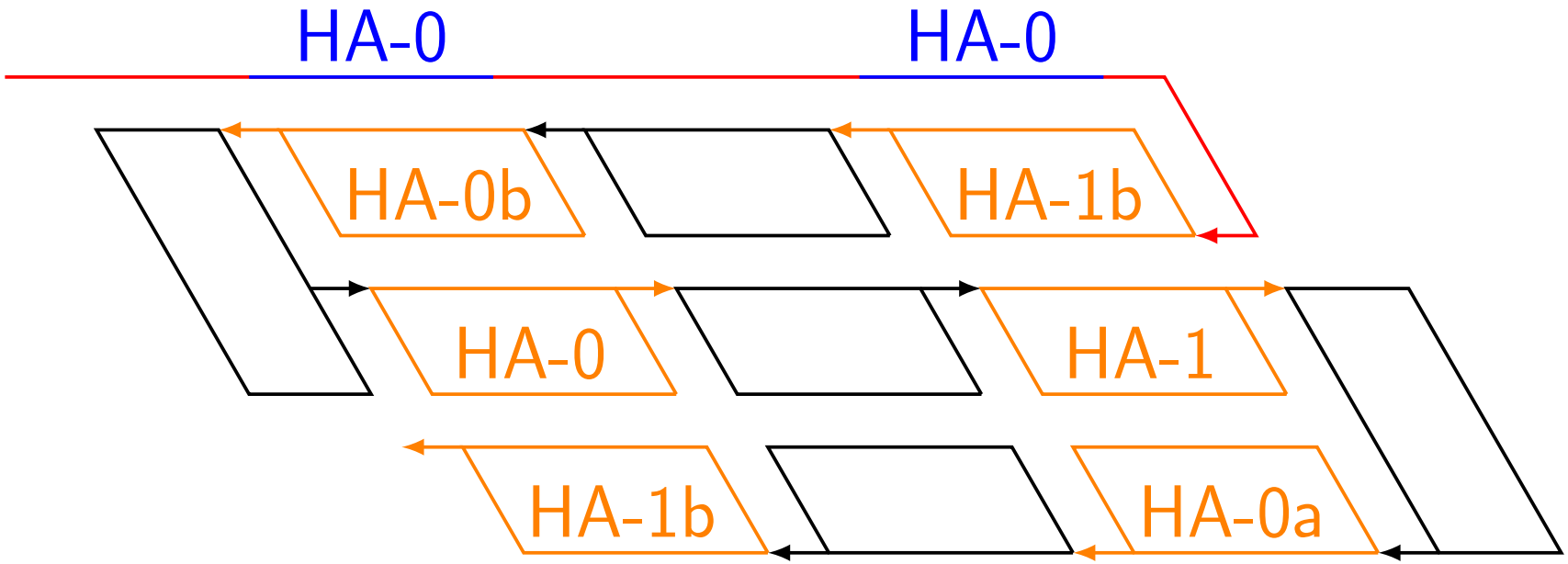


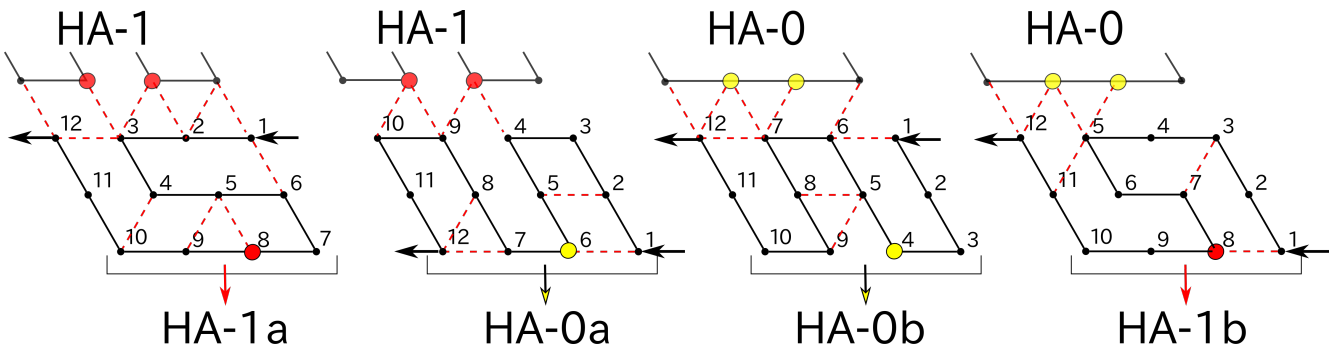
Module *[Geary et al. 2016]*



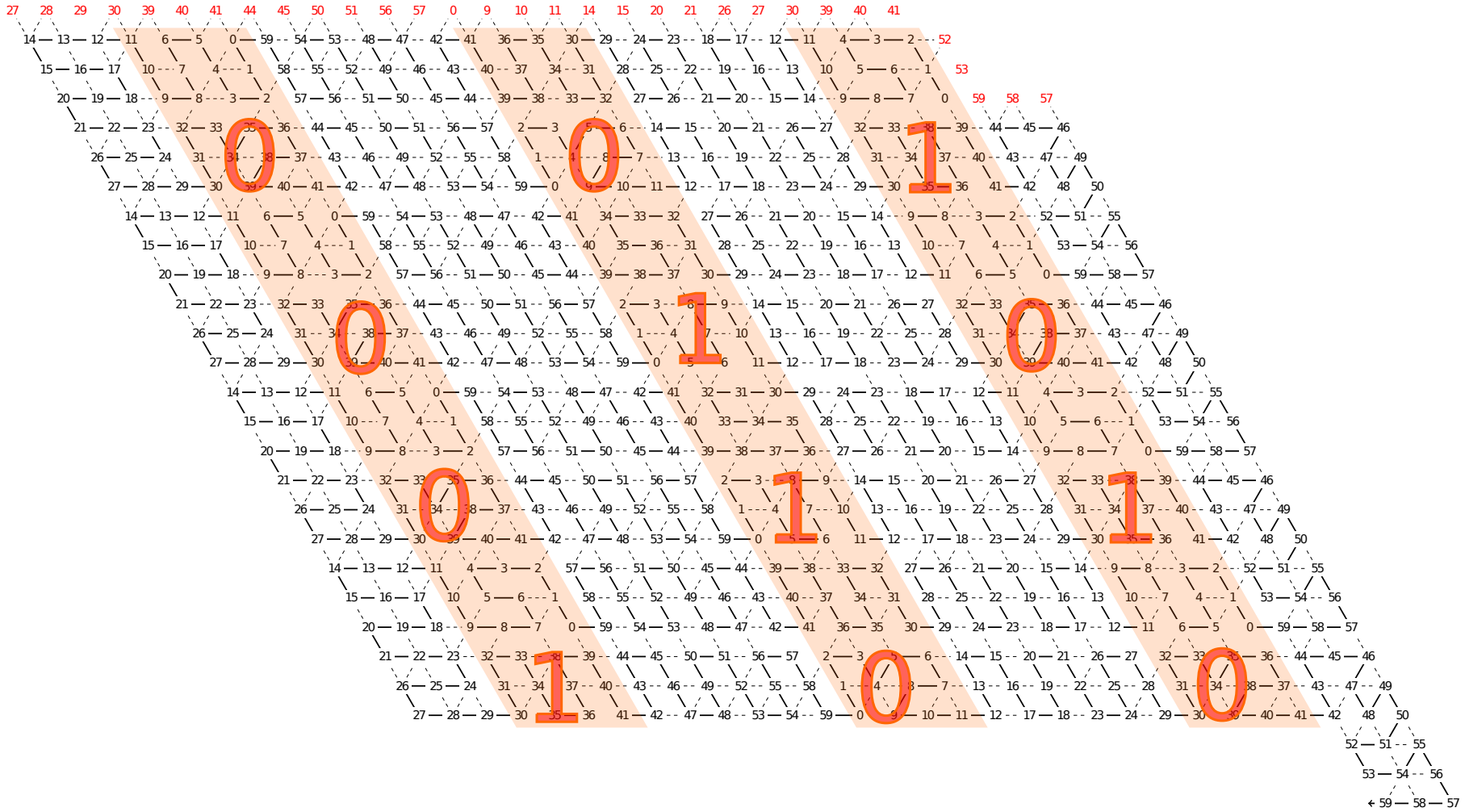


Module *[Geary et al. 2016]*





Module *[Geary et al. 2016]*



Implementation

- We design an oritatami system that self-assembles an n-bit fraction of the Heighway dragon.

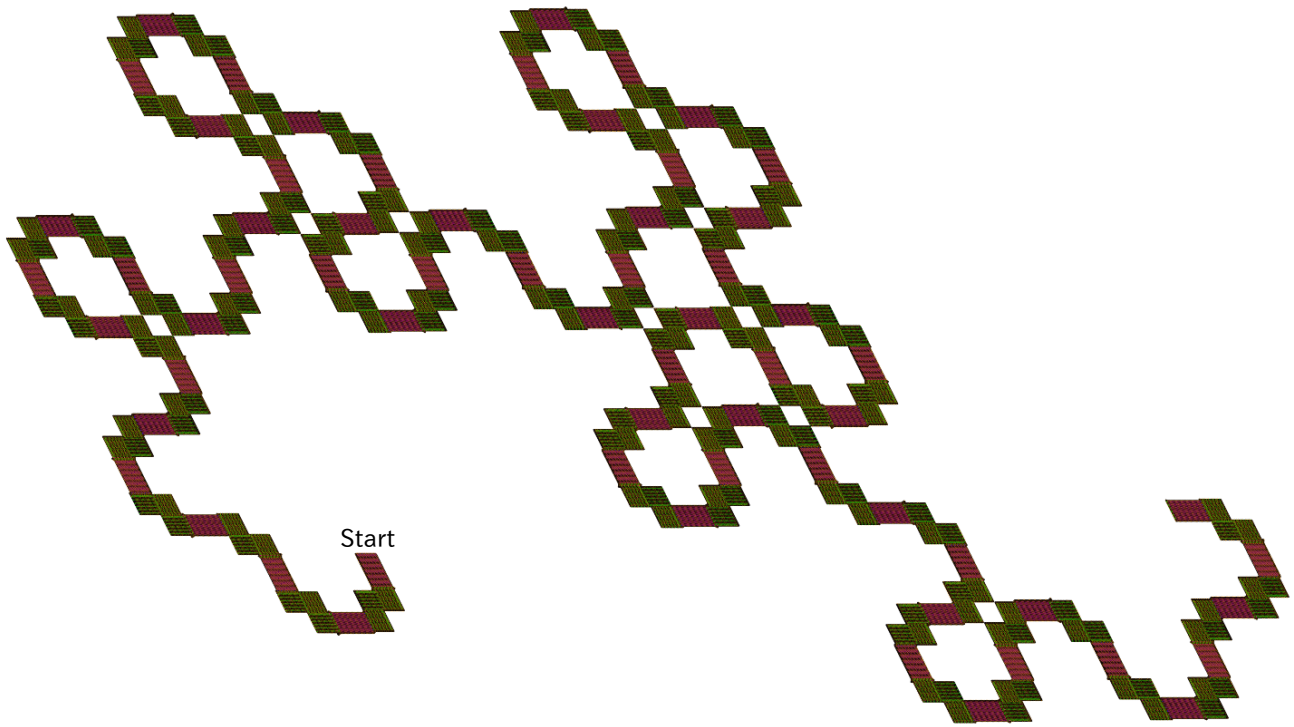


Figure: 6-bit Heighway dragon by the proposed Oritatami system.

Fractals and DFAO

Automatic sequences

Heighway dragon can be represented by the Paperfolding sequence P, which can be output by a DFAO.

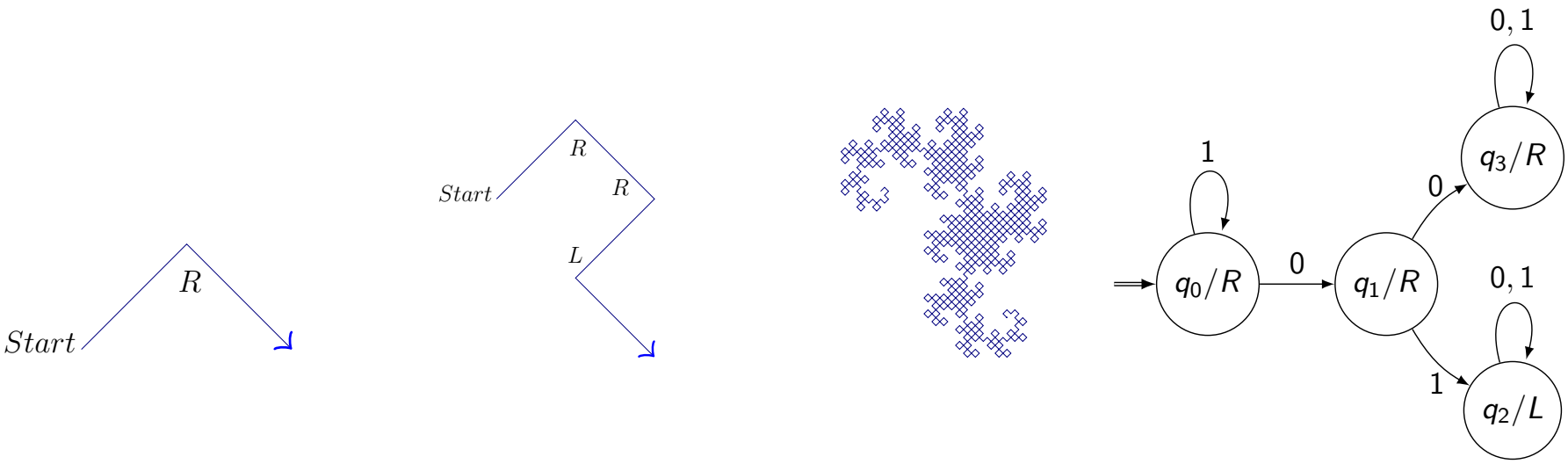


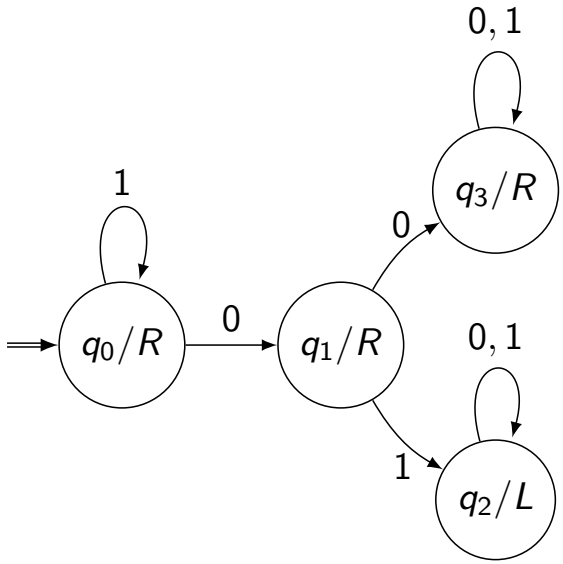
Figure: Paperfolding sequence $P = \mathrm{RRLRRLLR}\cdots$

Fractals and DFAO

- Paperfolding sequence $P = \mathrm{RRLRRLLRRRLLRLLRRRLRRLLLRRLLRLL}\cdots$

The i-th element of P ($i \geq 0$) can be computed by

1. feeding the DFA shown right with the binary representation of i from its LSB, and
2. reading R/L assigned to the reached state.



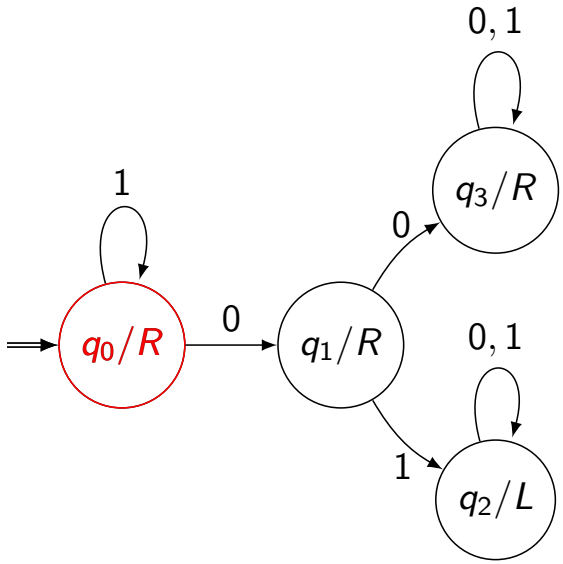
Fractals and DFAO

- Paperfolding sequence $P = \mathrm{RRLRRLLRRRLLRLLRRRLRRLLLRRLLRLL}\cdots$

$i = 0$

The i-th element of P ($i \geq 0$) can be computed by

1. feeding the DFA shown right with the binary representation of i from its LSB, and
2. reading R/L assigned to the reached state.



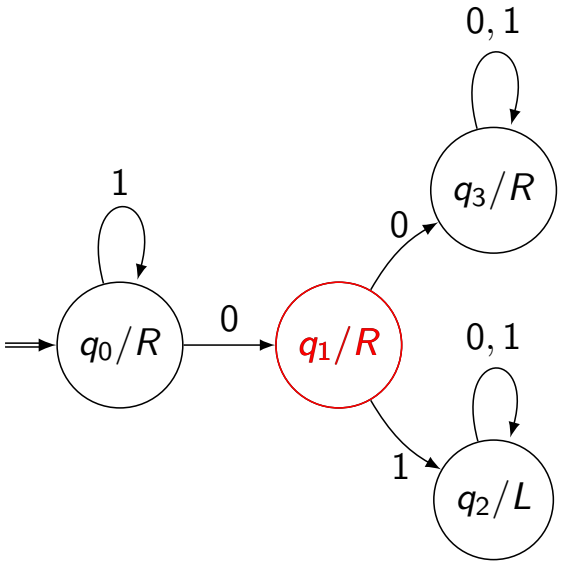
Fractals and DFAO

- Paperfolding sequence $P = \mathrm{R}\mathrm{RLRRLLRRRLLRLLRRRLRRLLLRRLLRLL}\cdots$

$i = 0$

The i-th element of P ($i \geq 0$) can be computed by

1. feeding the DFA shown right with the binary representation of i from its LSB, and
2. reading R/L assigned to the reached state.



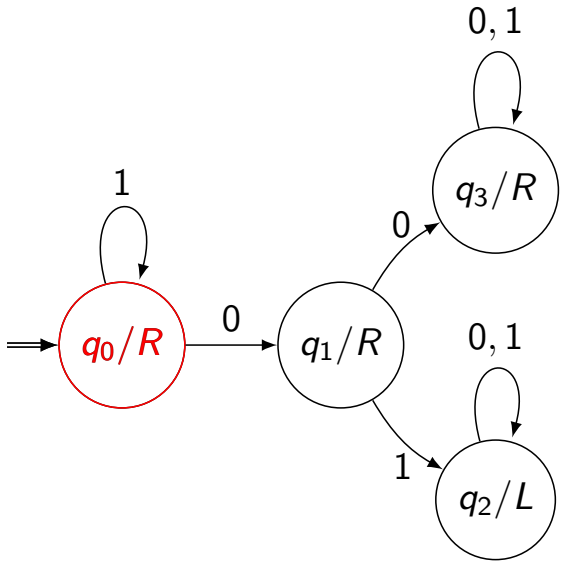
Fractals and DFAO

- Paperfolding sequence $P = \mathrm{RRLRRLLRRRLLRLLRRRLRRLLLRRLLRLL}\cdots$

$i = 1$

The i-th element of P ($i \geq 0$) can be computed by

1. feeding the DFA shown right with the binary representation of i from its LSB, and
2. reading R/L assigned to the reached state.



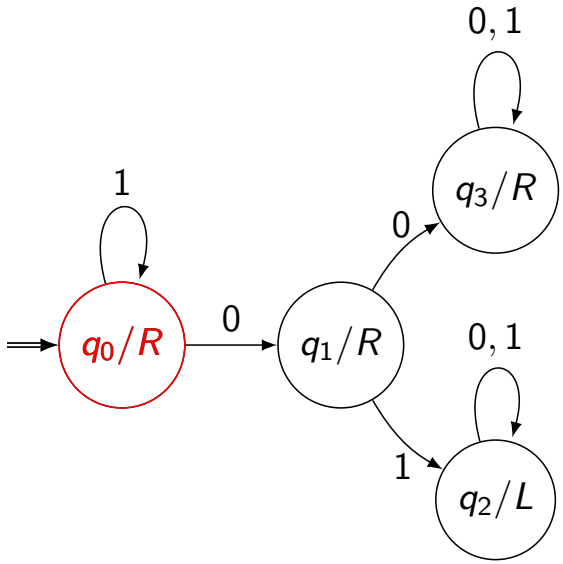
Fractals and DFAO

- Paperfolding sequence $P = \mathrm{RRLRRLLRRRLLRLLRRRLRRLLLRRLLRLL}\cdots$

$i = \quad 2 \to 10$

The i-th element of P ($i \geq 0$) can be computed by

1. feeding the DFA shown right with the binary representation of i from its LSB, and
2. reading R/L assigned to the reached state.



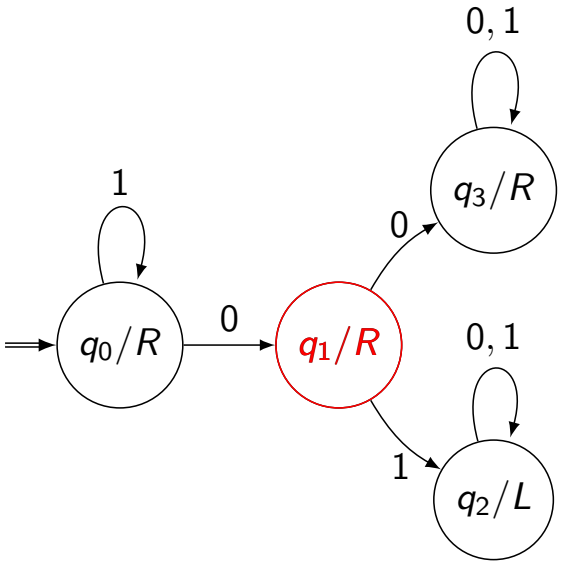
Fractals and DFAO

- Paperfolding sequence $P = \mathrm{RRLRRLLRRRLLRLLRRRLRRLLLRRLLRLL}\cdots$

$i = \quad 2 \to 10$

The i-th element of P ($i \geq 0$) can be computed by

1. feeding the DFA shown right with the binary representation of i from its LSB, and
2. reading R/L assigned to the reached state.



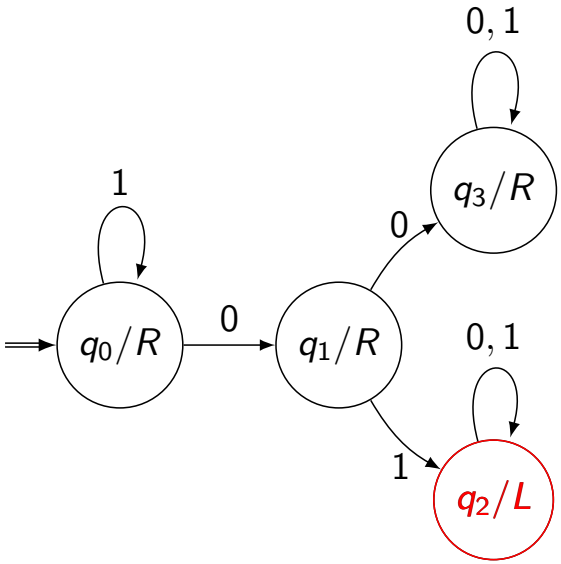
Fractals and DFAO

- Paperfolding sequence $P = \mathrm{RRLRRLLRRRLLRLLRRRLRRLLLRRLLRLL}\cdots$

$i = \quad 2 \to 10$

The i-th element of P ($i \geq 0$) can be computed by

1. feeding the DFA shown right with the binary representation of i from its LSB, and
2. reading R/L assigned to the reached state.

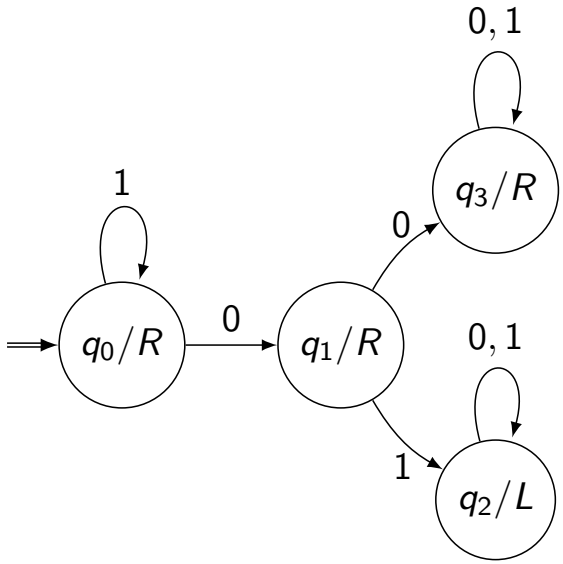


Fractals and DFAO

- Paperfolding sequence $P = \mathrm{RRLRRLLRRRLLRLLRRRLRRLLLRRLLRLL}\cdots$

The i-th element of P ($i \geq 0$) can be computed by

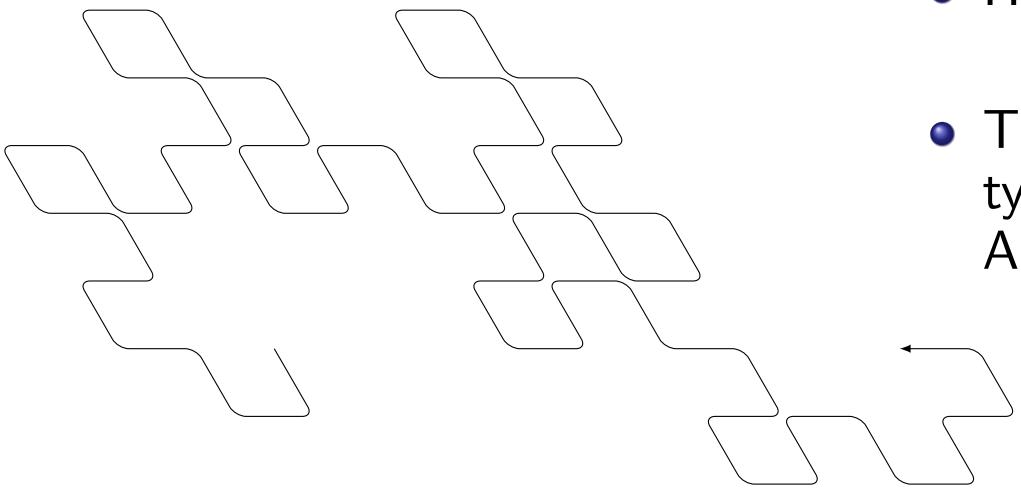
1. feeding the DFA shown right with the binary representation of i from its LSB, and
2. reading R/L assigned to the reached state.



Implementation

Problems

Over the triangular grid, it is more natural to consider the slanted Heighway dragon.

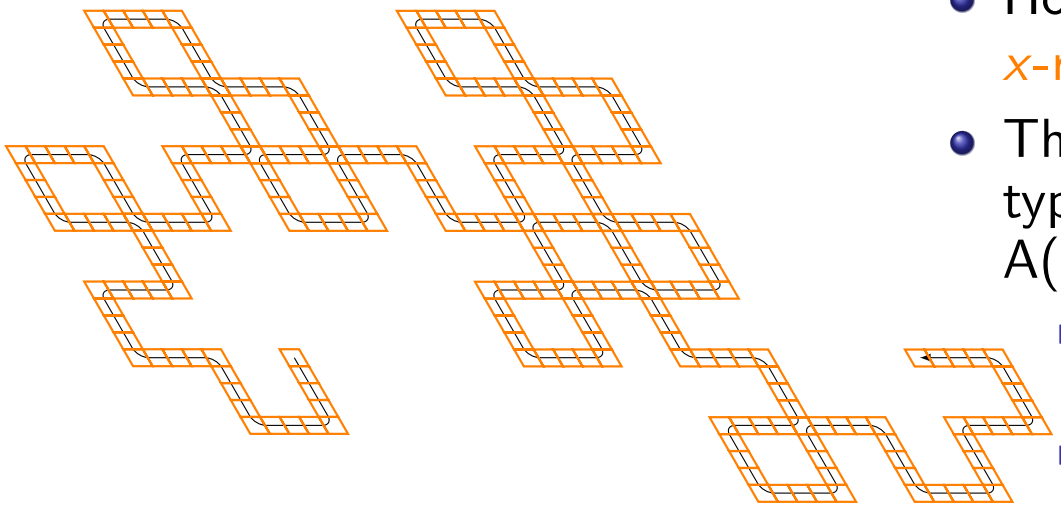


- How to avoid collisions?
- The slanted dragon involves four types of turns: L/R * A(cute)/O(btuse).
  - After the vertical line, L $\to$ O, R $\to$ A.
  - After the horizontal line, L $\to$ A, R $\to$ O.

Implementation

Problems

Over the triangular grid, it is more natural to consider the slanted Heighway dragon.



- How to avoid collisions?
  x-rhombus scaling
- The slanted dragon involves four types of turns: L/R * A(cute)/O(btuse).
  - After the vertical line, L $\to$ O, R $\to$ A.
  - After the horizontal line, L $\to$ A, R $\to$ O.

Implementation

Algorithm:

- Set $i = 0$.
- Repeat the following tasks while propagating i:
  1. Draw a vertical line segment.
  2. Compute $P[i]$ by the DFAO and interpret it as $\underline{L \to O, R \to A}$.
  3. Turn accordingly and $i := i + 1$.
  4. Draw a horizontal line segment.
  5. Compute $P[i]$ by the DFAO and interpret it as $\underline{L \to A, R \to O}$.
  6. Turn accordingly and $i := i + 1$.

Implementation

Algorithm:

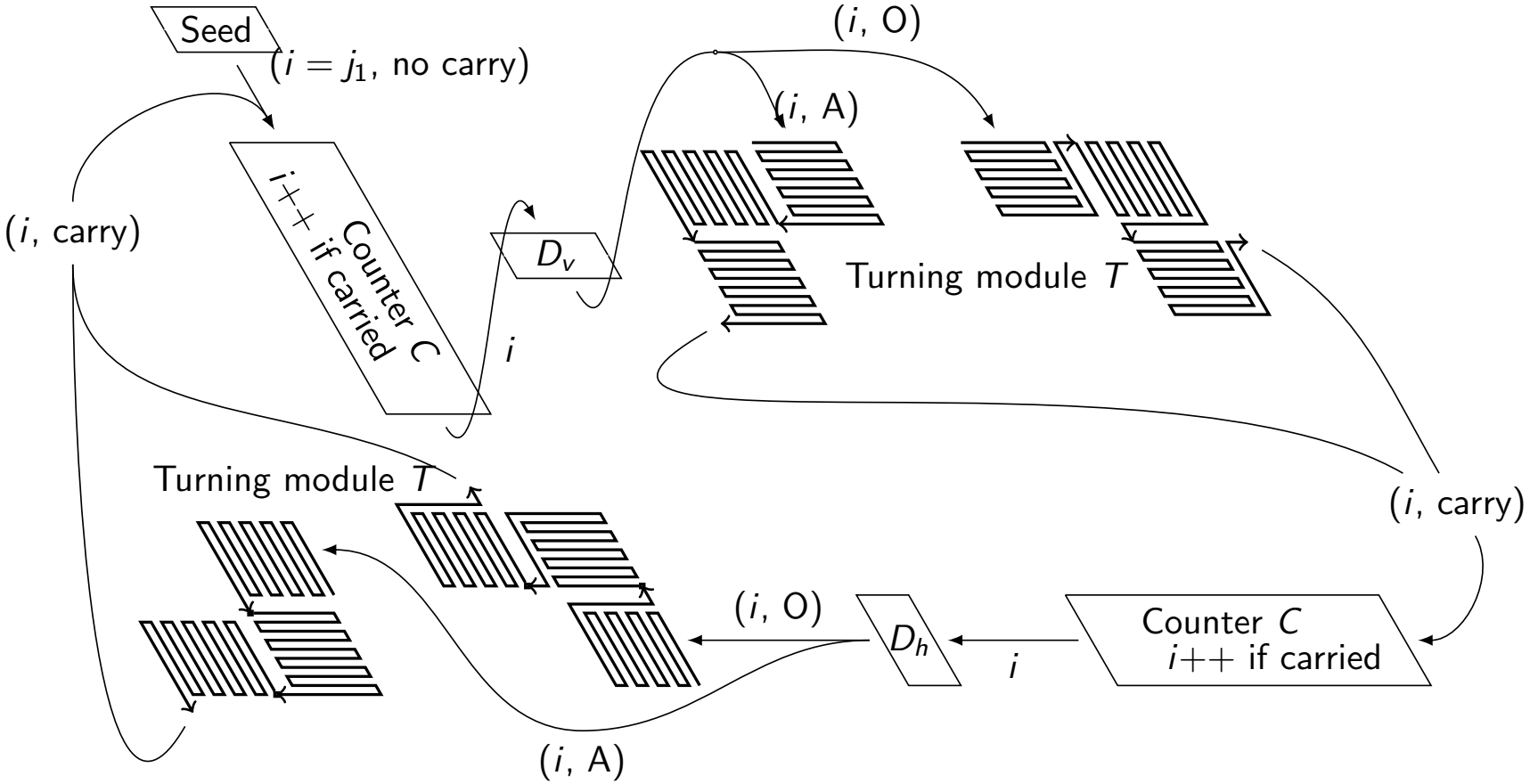
- Set $i = 0$.
- Repeat the following tasks while propagating i:
  1. Draw a vertical line segment.
  2. Compute $P[i]$ by the DFAO and interpret it as $\underline{L \to O, R \to A}$.
  3. Turn accordingly and $i := i + 1$.
  4. Draw a horizontal line segment.
  5. Compute $P[i]$ by the DFAO and interpret it as $\underline{L \to A, R \to O}$.
  6. Turn accordingly and $i := i + 1$.

Transcript: $(CD_vTCD_hT)^*$, where

- C is the counter module for 1, 4 ,
- D_v, D_h are the DFAO module for 2 and 5, respectively.
- T is the turning module for 3, 6

Implementation

Module automaton



Implementation

Seed

The initial count of i is encoded on the seed. For instance $i = (100)_2$ is encode as the following sequence of bead types:

$W_{t,0} = 338 \to 339 \to 344 \to 345$ and $W_{t,1} = 346 \to 347 \to 348 \to 349$

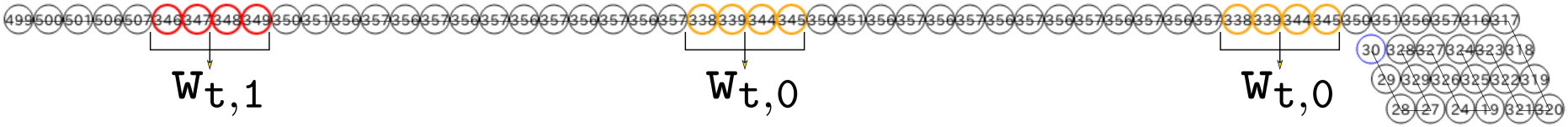
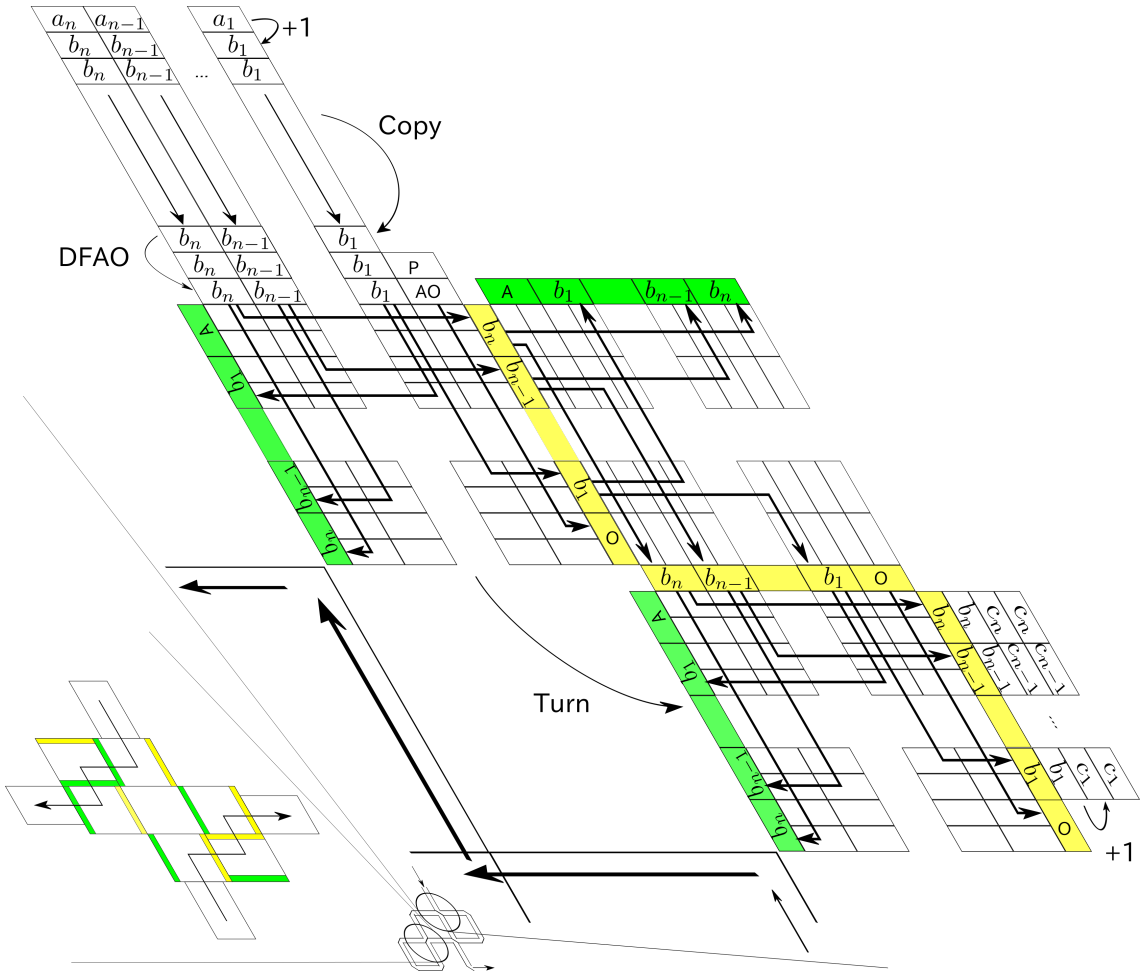


Figure: The seed for the 3-bit Heighway dragon that starts at $i = 100_2$.

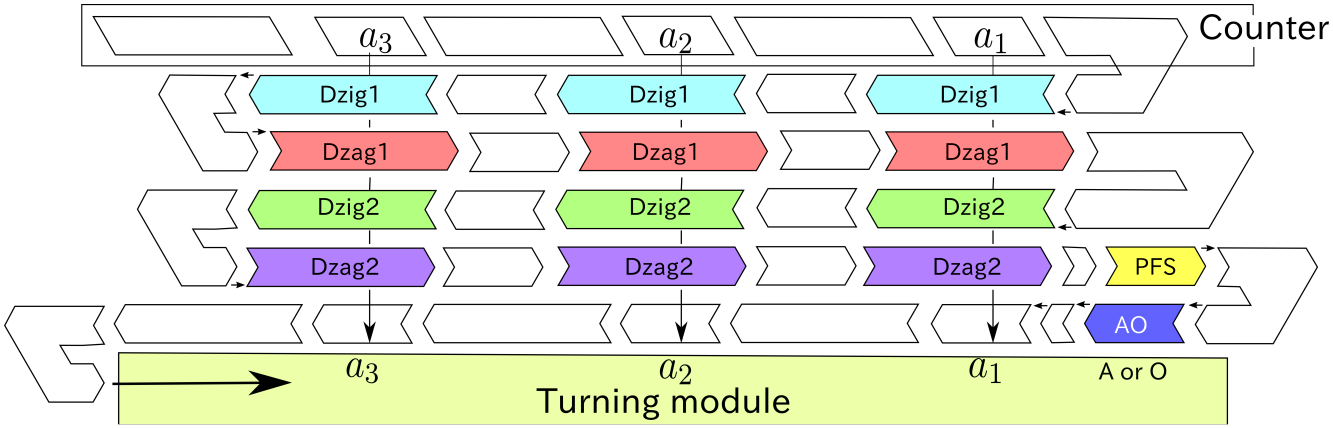
Implementation

Information flow

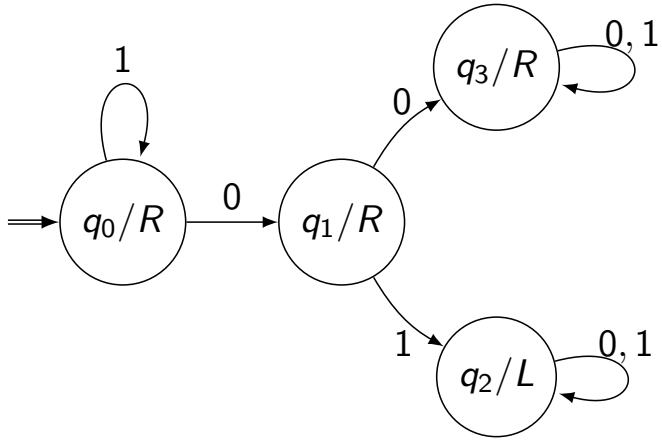


Implementation

DFAO module



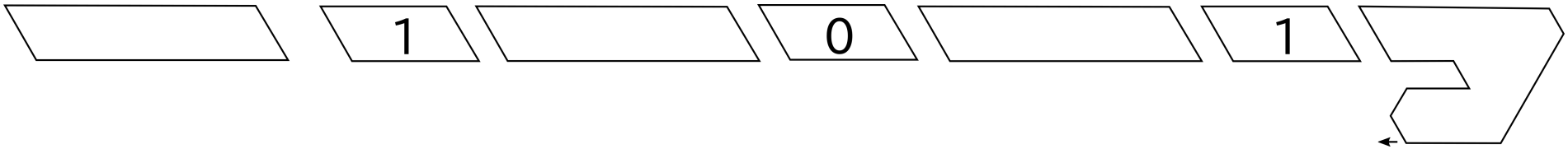
- 1st zigzag (Dzig1 &Dzag1) searches the first 0 (from LSB) and mark it as f0.
- 2nd zigzag (Dzig2 & Dzag2) checks if the f0 is followed by 0 or 1.

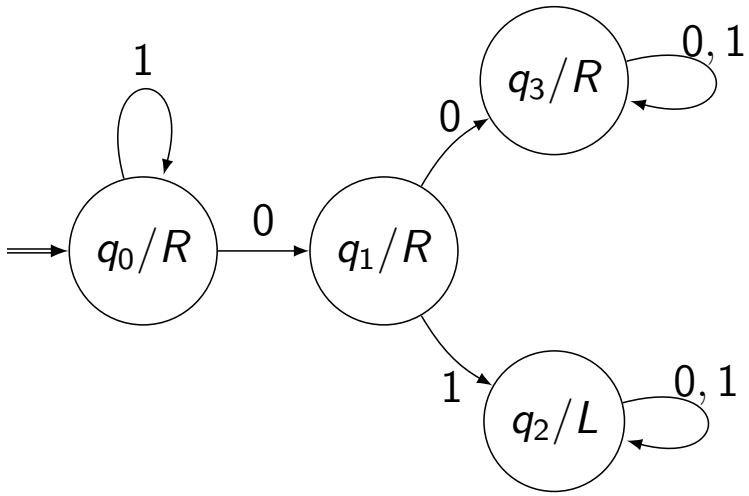


Implementation

DFAO module

Current count = 101

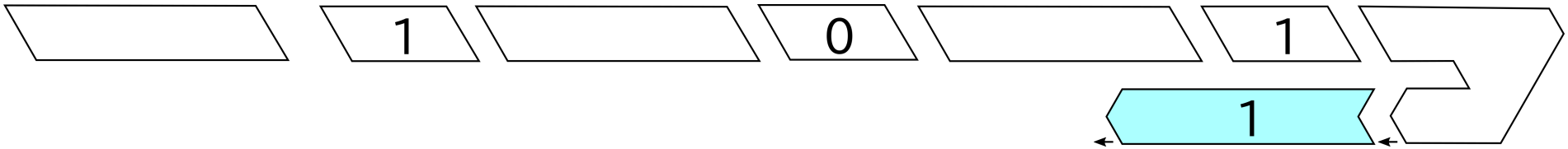


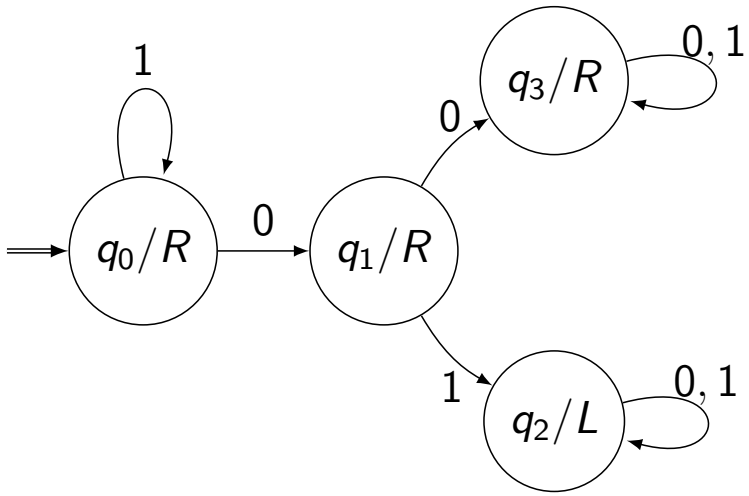


Implementation

DFAO module

Current count = 101

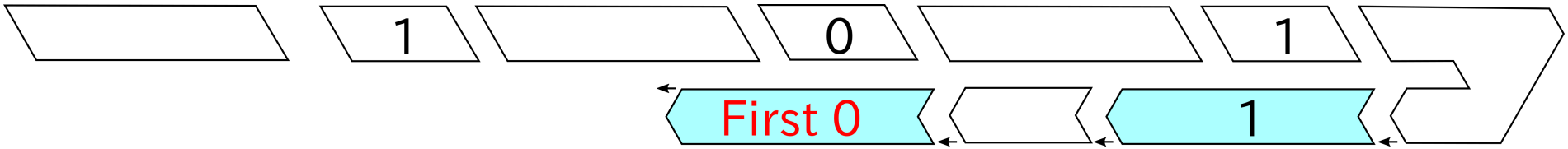


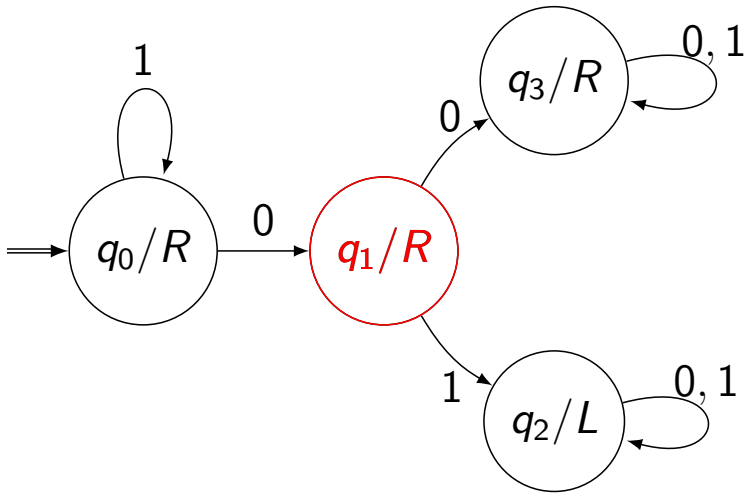


Implementation

DFAO module

Current count = 101

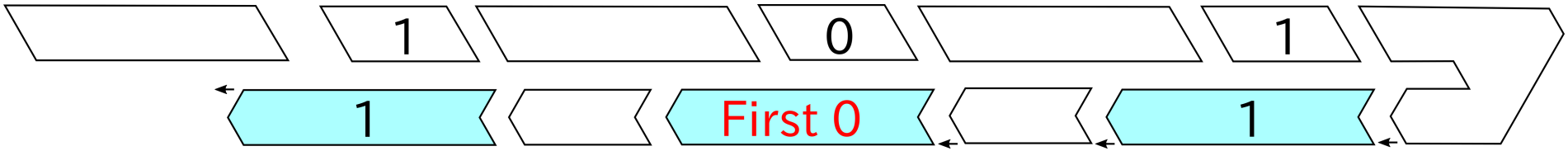


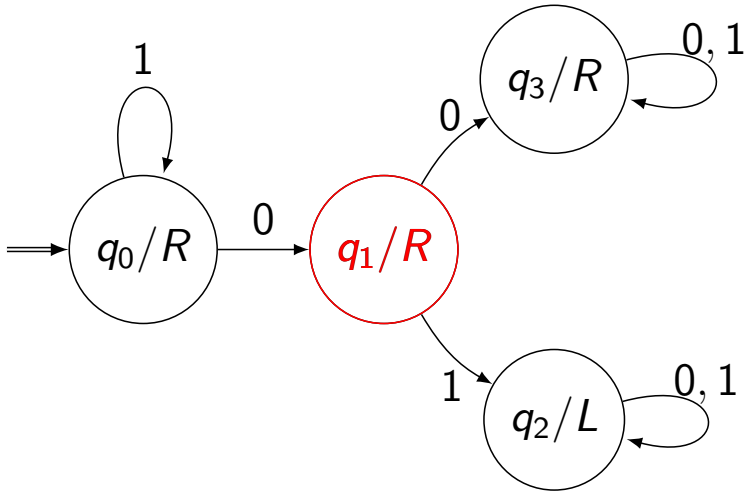


Implementation

DFAO module

Current count = 101

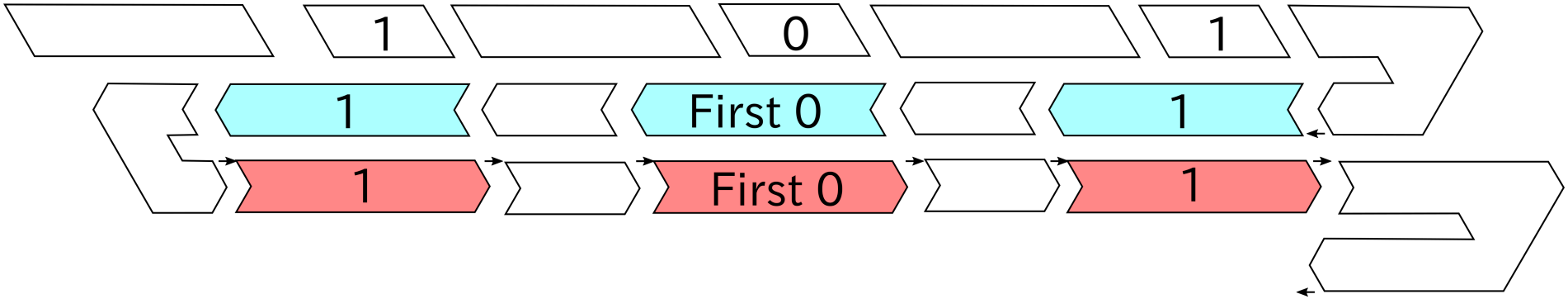


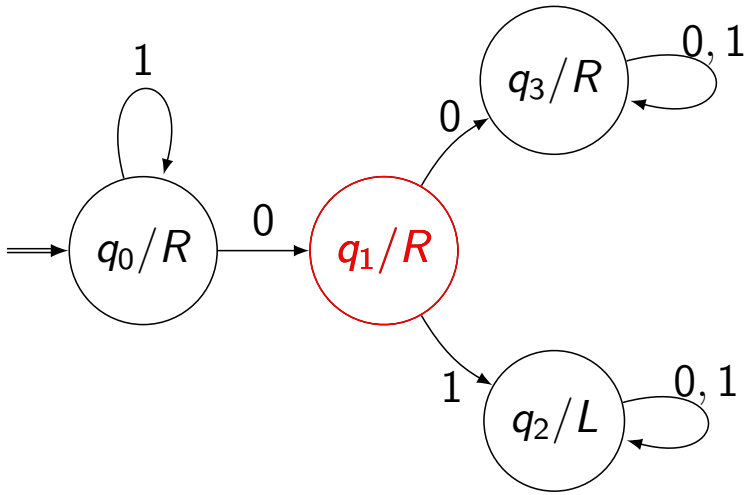


Implementation

DFAO module

Current count = 101

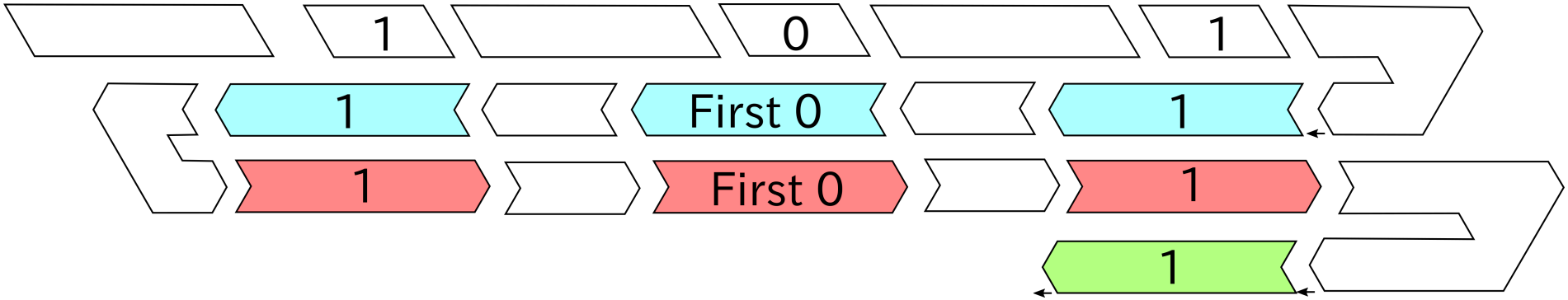


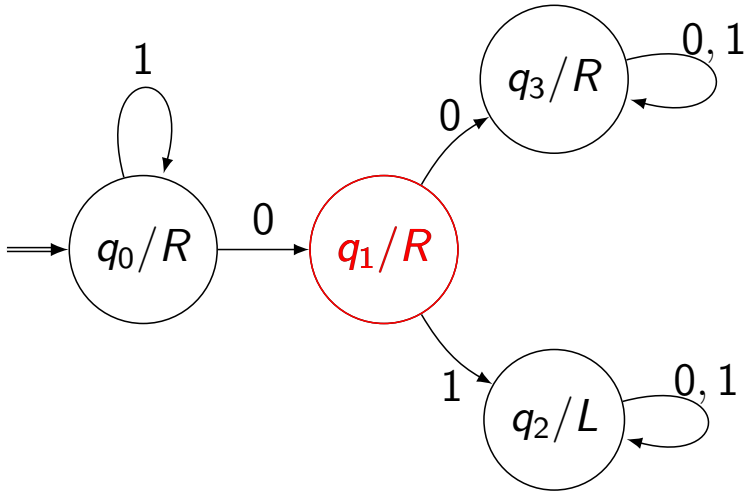


Implementation

DFAO module

Current count = 101

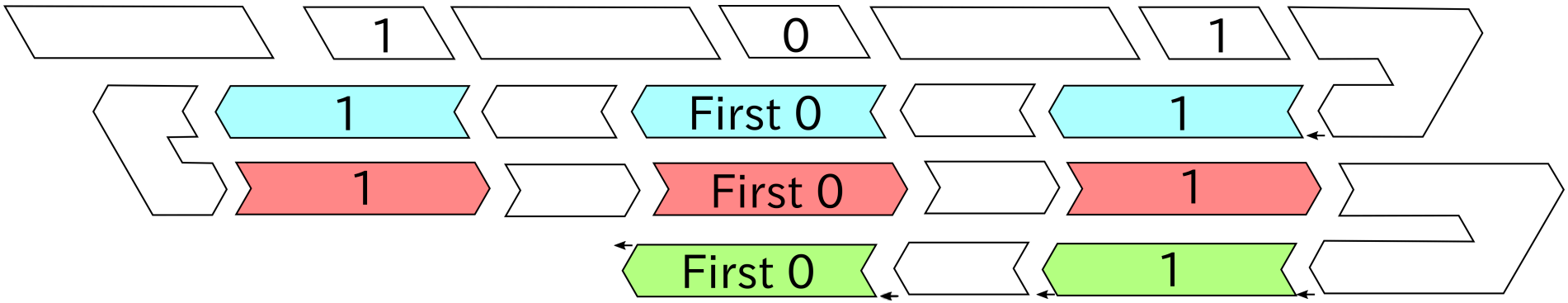


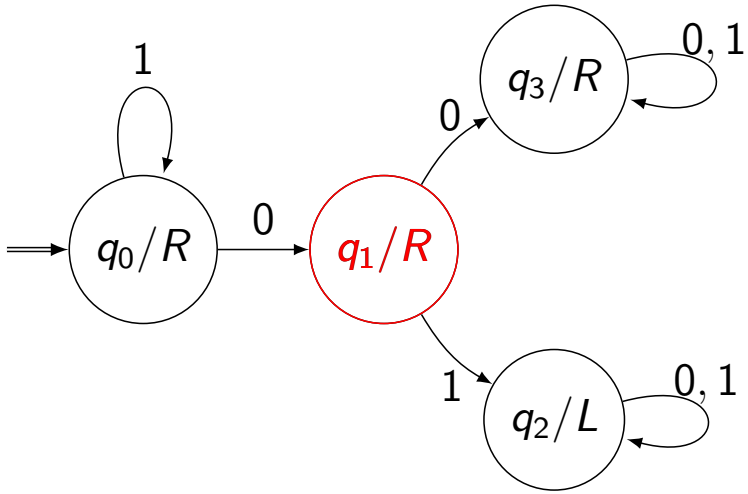


Implementation

DFAO module

Current count = 101

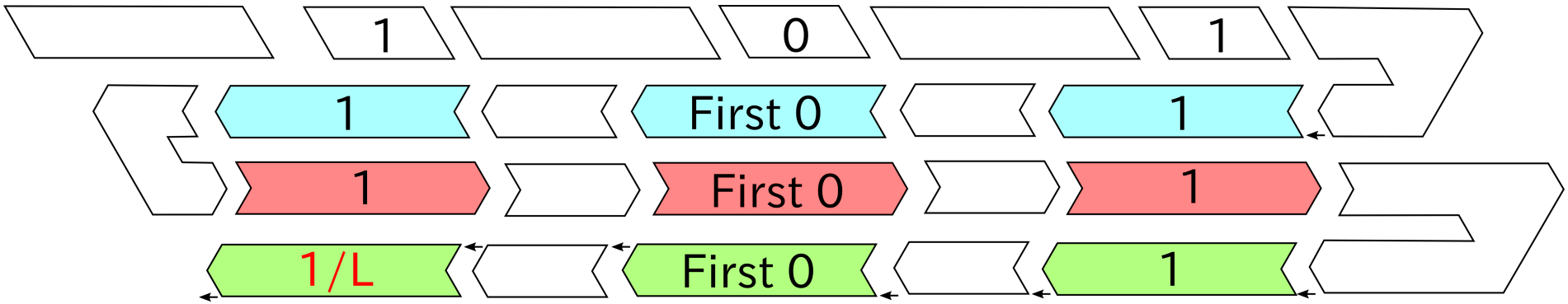


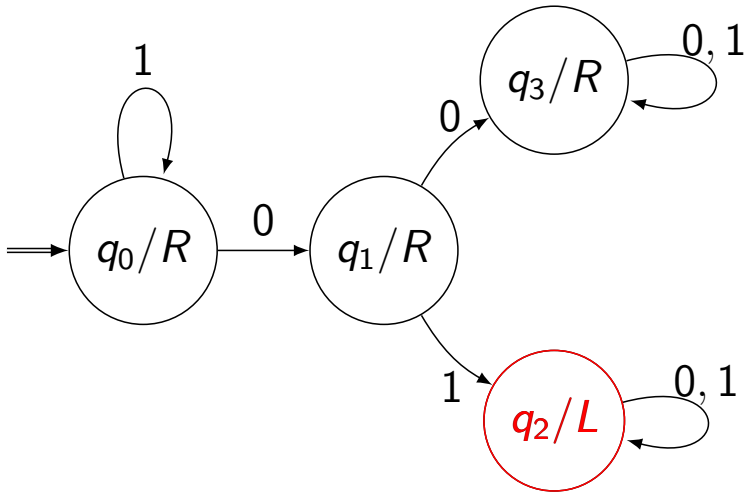


Implementation

DFAO module

Current count = 101

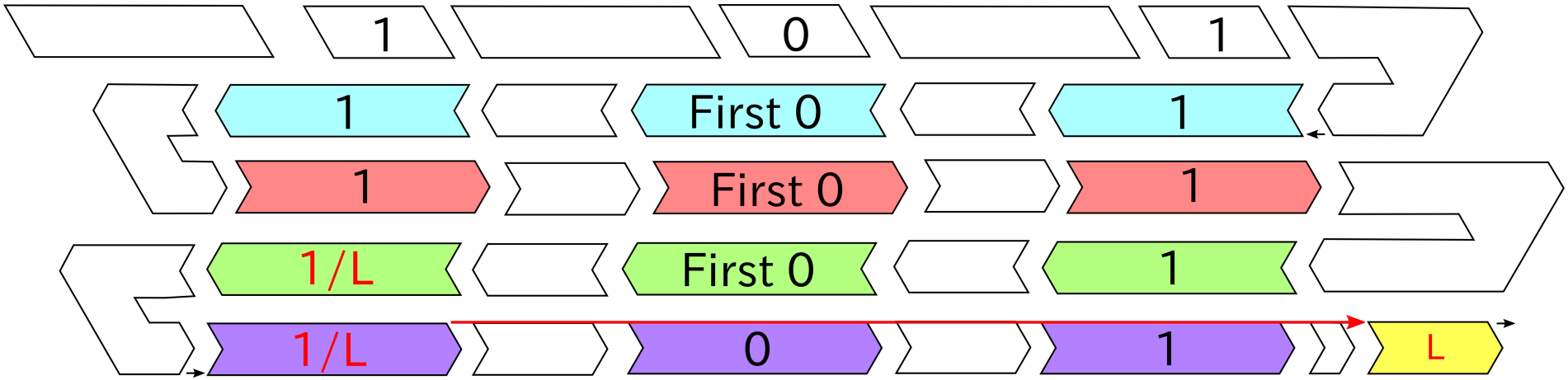


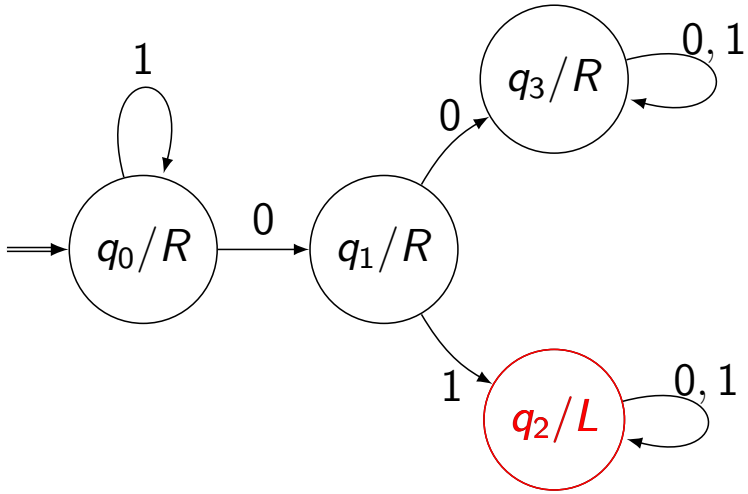


Implementation

DFAO module

Current count = 101

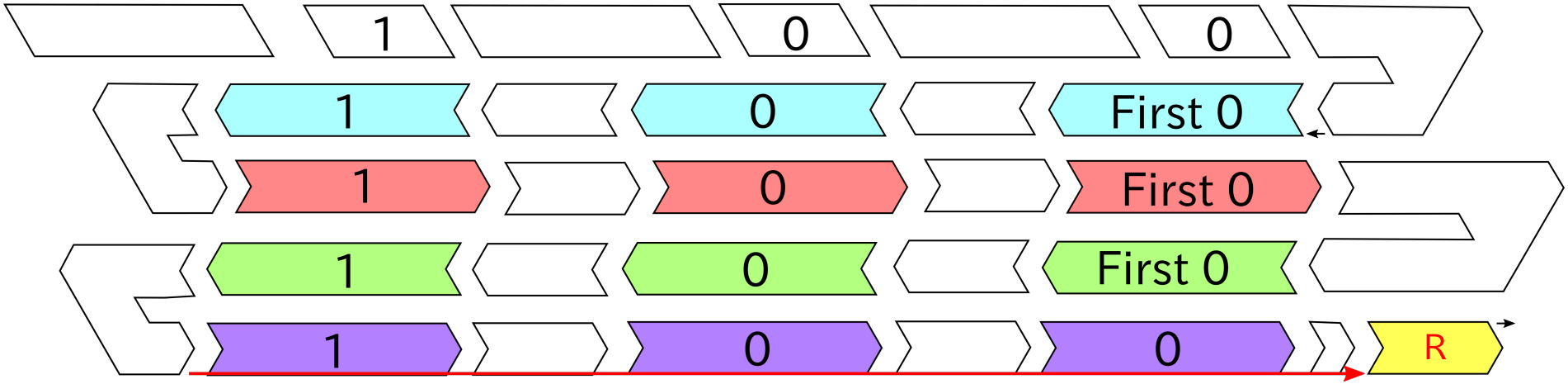


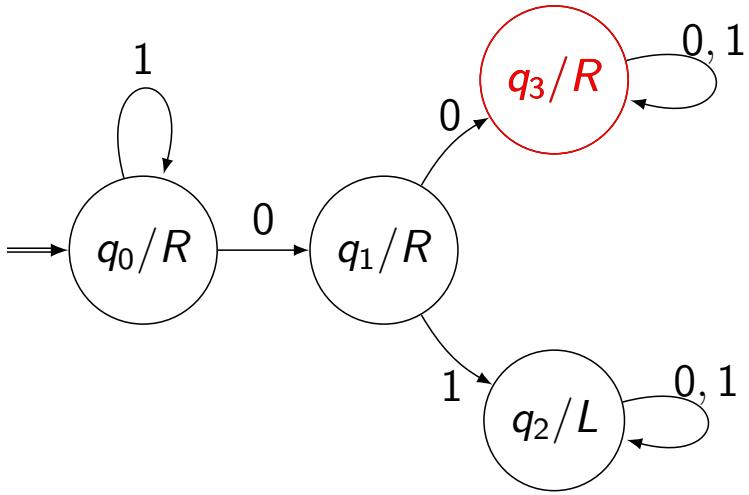


Implementation

DFAO module

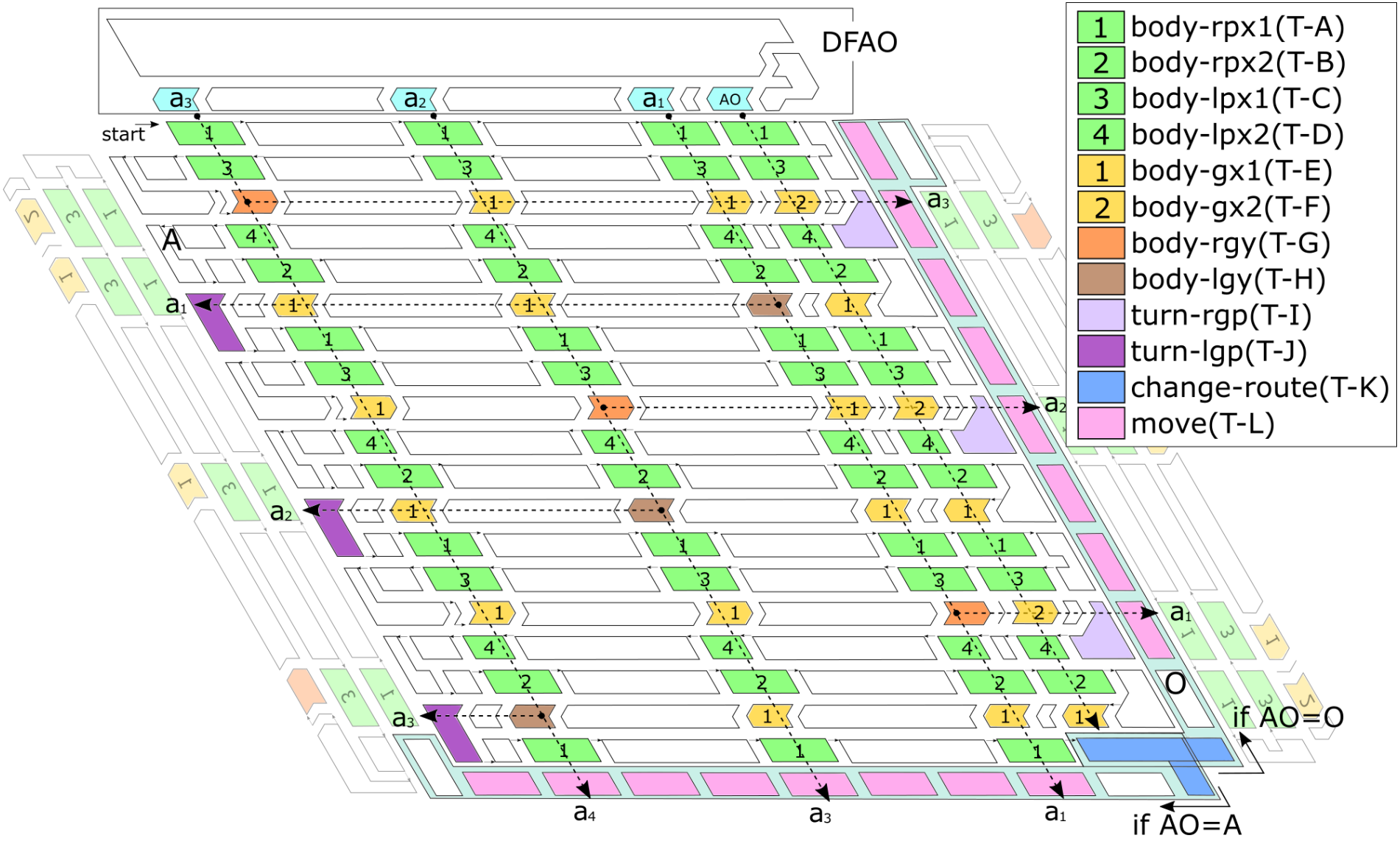
Current count = 100





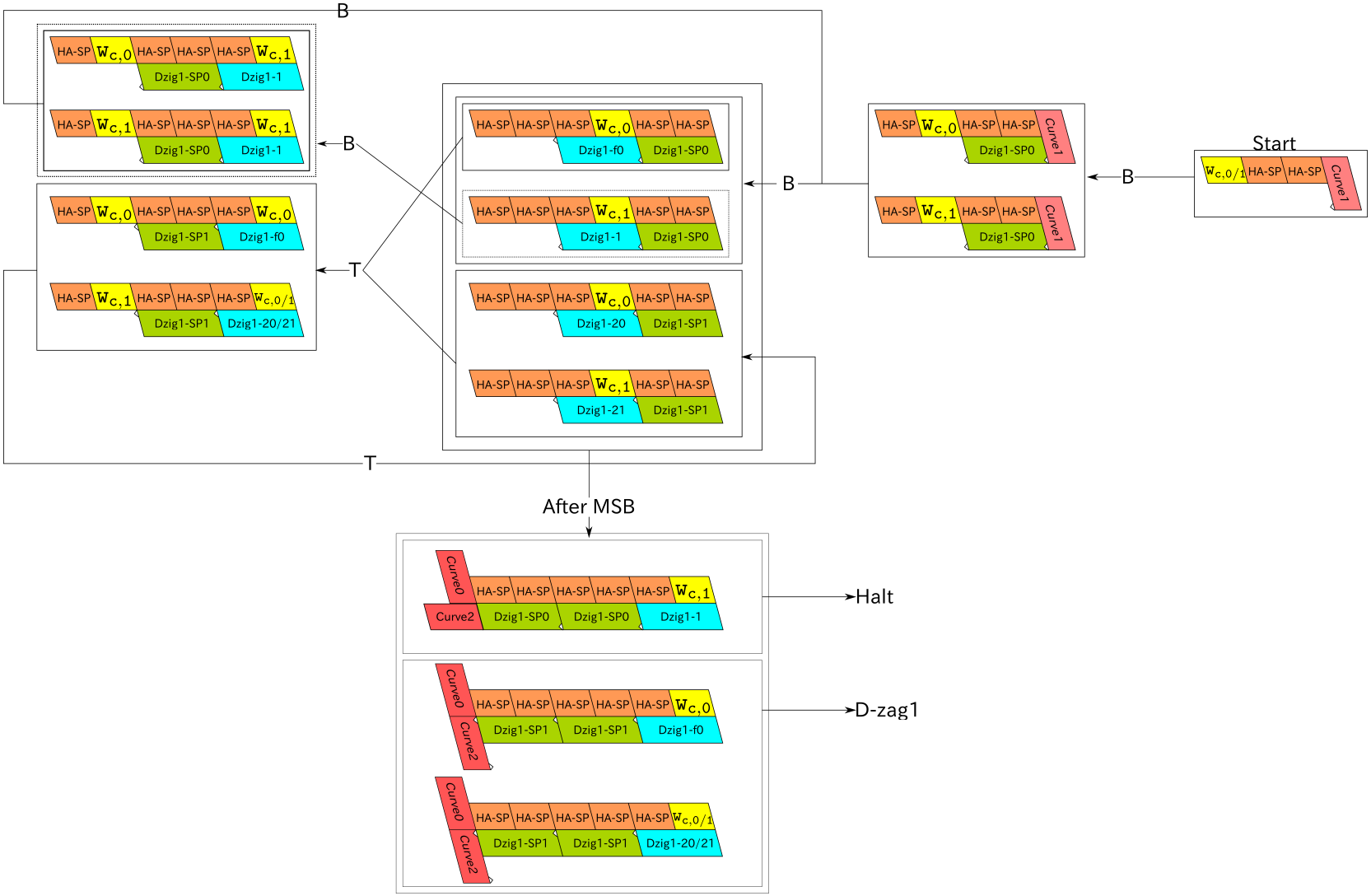
Implementation

Turning module



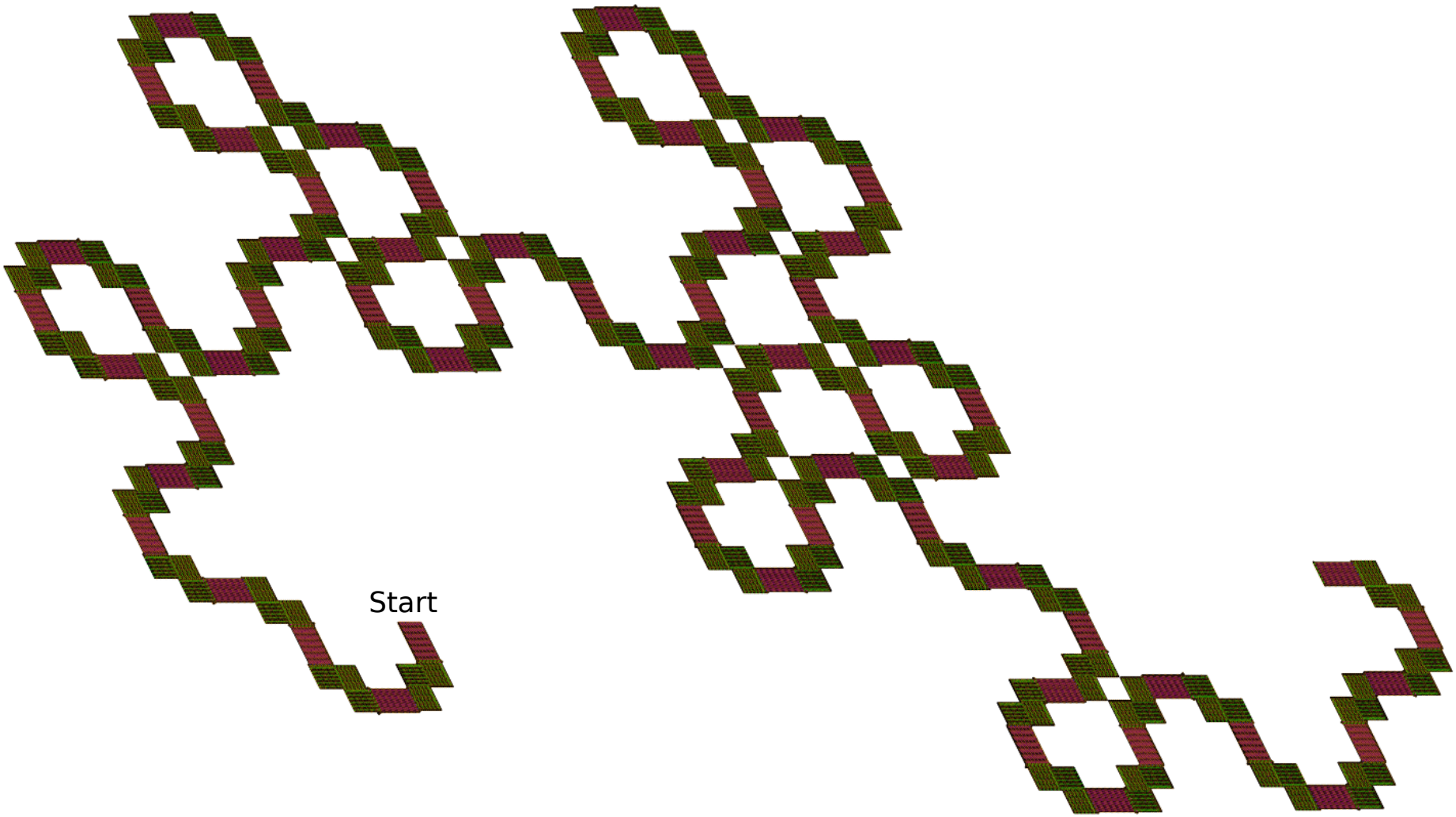
Implementation

Brick automata



Implementation

Result



Acknowledgements

Thanks a lot for your attention!!

Supported in part by

- JSPS KAKENHI Grant-in-Aid for Young Scientists (A) 16H05854
- JSPS KAKENHI Grant-in-Aid for Challenging Research (Exploratory) 18K19779
- JSPS and NRF under the Japan-Korea Basic Scientific Cooperation Program YB29004.

日本学術振興会
Japan Society for the Promotion of Science

Acknowledgements

Thanks a lot for your attention!!

Nicolas Schabanel (LIAFA, ENS de Lyon)

Yo-Sub Han (Yonsei Univ.)

Hwee Kim (Univ. of South Florida)

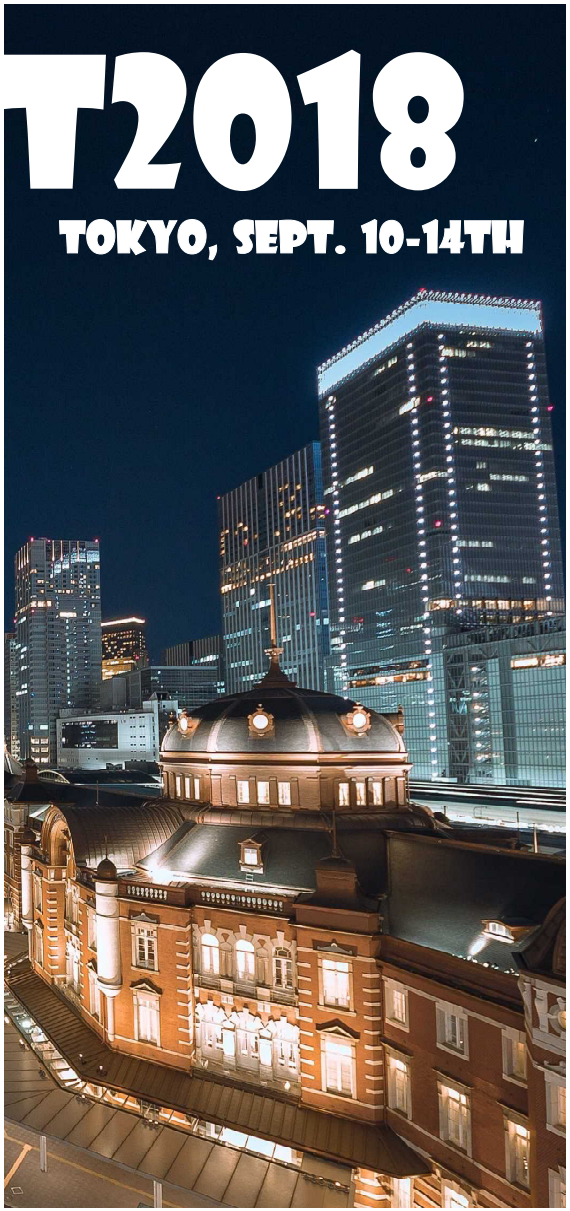
References I


C. Geary, P-E. Meunier, N. Schabanel, S. Seki
Programming biomolecules that fold greedily during translation.
Proc. MFCS 2016, LIPIcs 58, 43:1-43:14, 2016


C. Geary, P. W. K. Rothemund, E. S. Andersen
A single-stranded architecture for cotranscriptional folding of RNA nanostructures.
Science 345 (2014) 799-804


K. E. Watters, E. J .Strobel, A. M. Yu, J. T. Lis, and J. B. Lucks
Cotranscriptional folding of a riboswitch at nucleotide resolution.
Nat. Struct. Mol. Biol. 23(12) (2016) 1124-1133

Advertisement 1



https://dlt2018.uec.ac.jp

- Venue: Baltic Hall (5-6th floor at *Le Signe* in front of Fuchu station).
- Registration fee 30,000JPY.
- Pre-workshop to celebrate the birthdays of Masami Ito and Pal Dömösi will be held on September 5-7 @ Kyoto Sangyo University.