Oritatami systems doodle uncomputably

Daria **Pchelina (LIPN)**
Nicolas **Schabanel (LIP)**
Shinnosuke **Seki (UEC, Tokyo)**
Guillaume **Theyssier (I2M)**

Oritatami:
A model for co-transcriptional folding

The program:

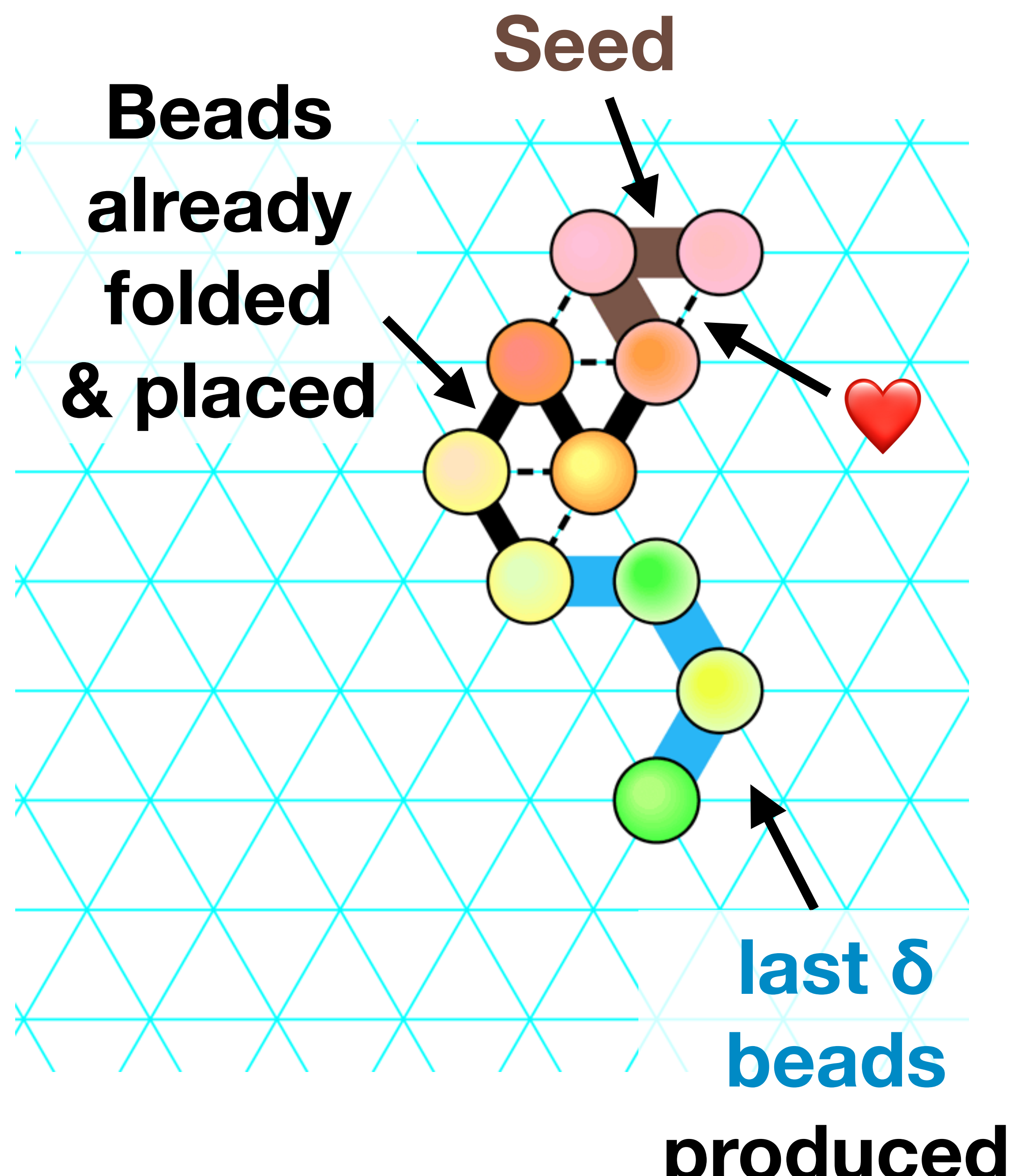
- a **sequence** of **bead types** (the **transcript**)

The instructions:

- the **rule** **a**❤️**b** if bead types **a** and **b** attract each other

The input configuration:

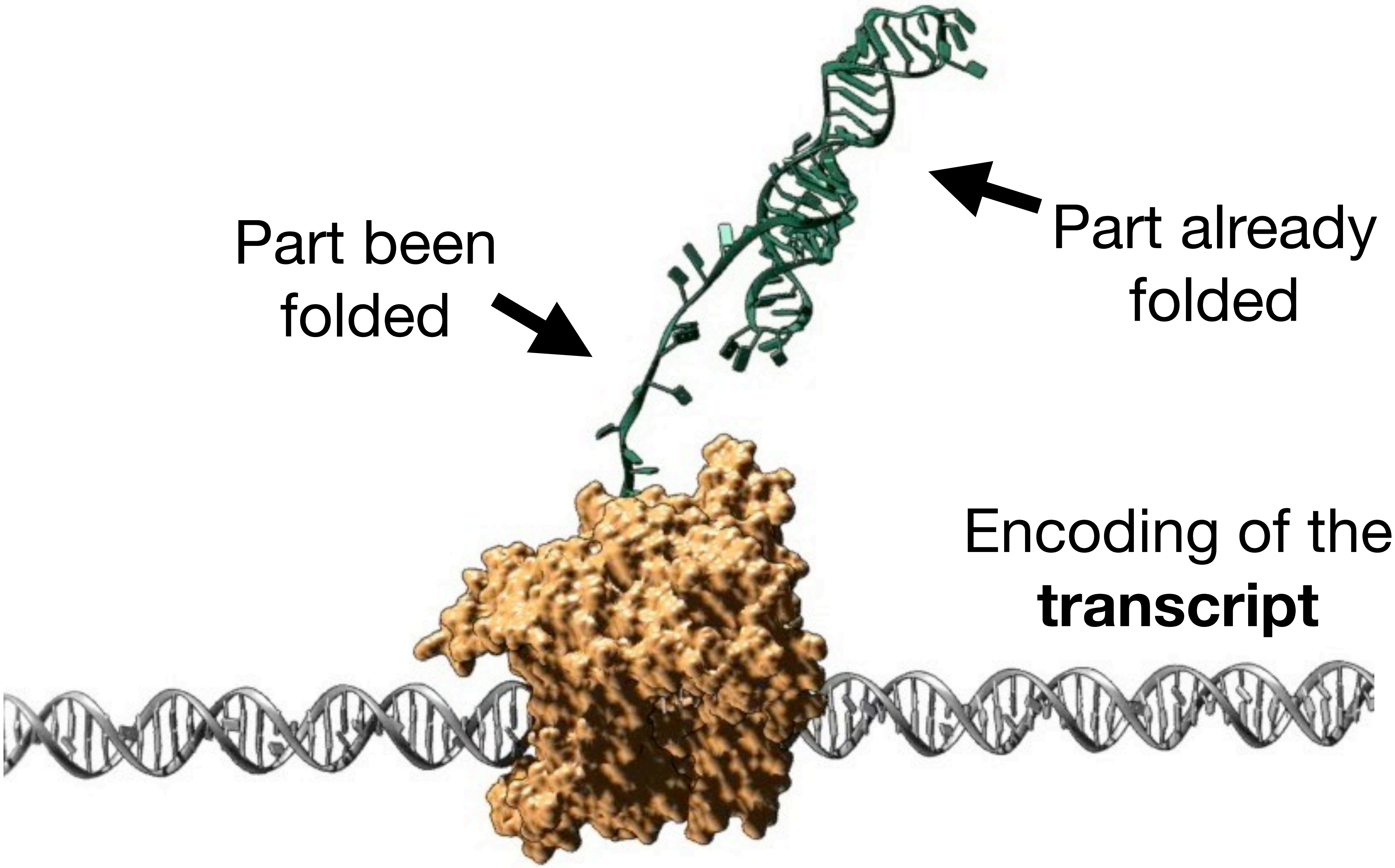
- Some beads placed beforehand (the **seed**)



Geary, Meunier, Schabanel, Seki MFCS 2016

RNA Folding

(Real time: ~1 second)



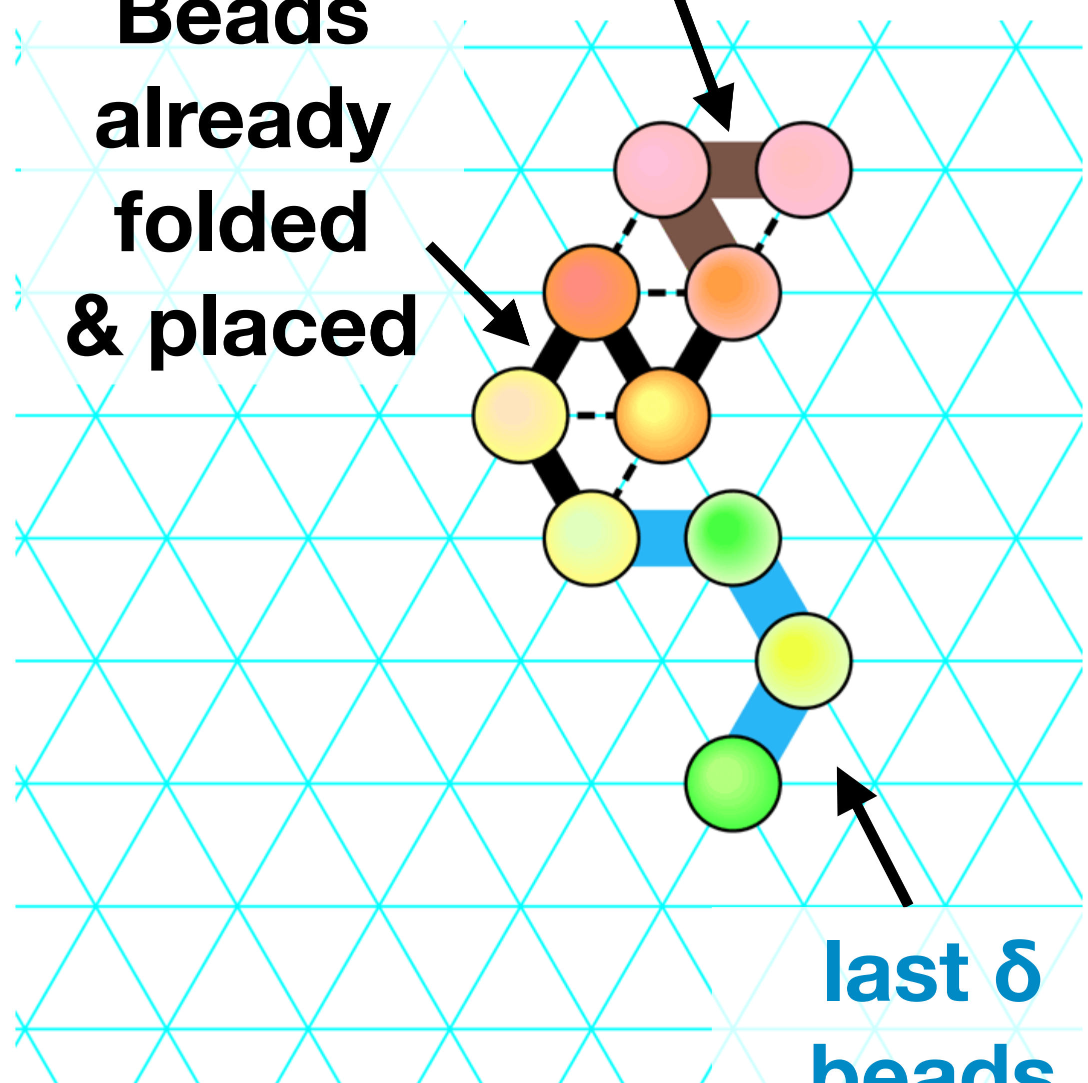
Video: Geary

Oritatami:
A model for co-transcriptional folding

The dynamics

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

here, delay **δ = 3**



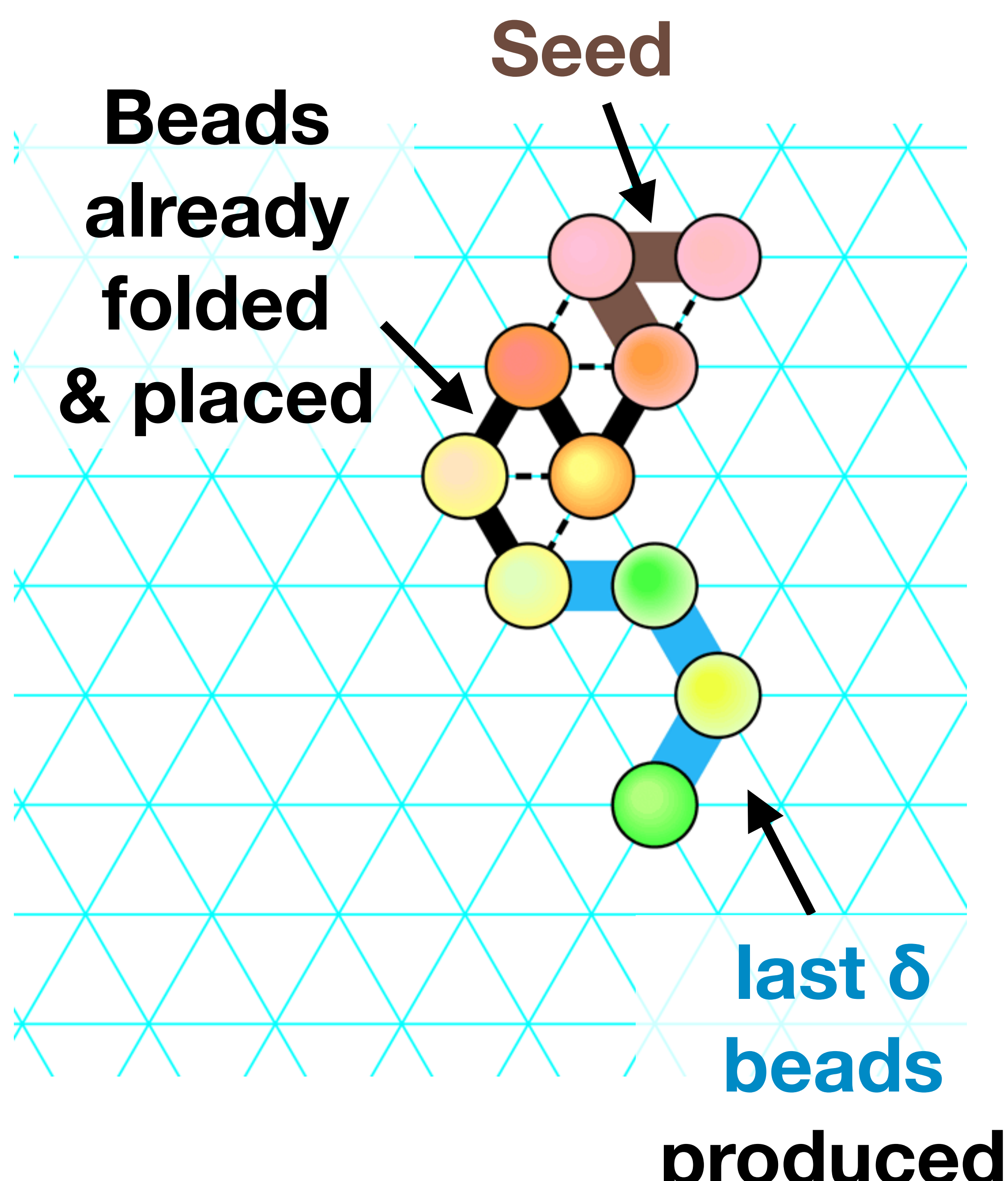
Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami:
A model for co-transcriptional folding

The dynamics

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

here, delay **δ = 3**

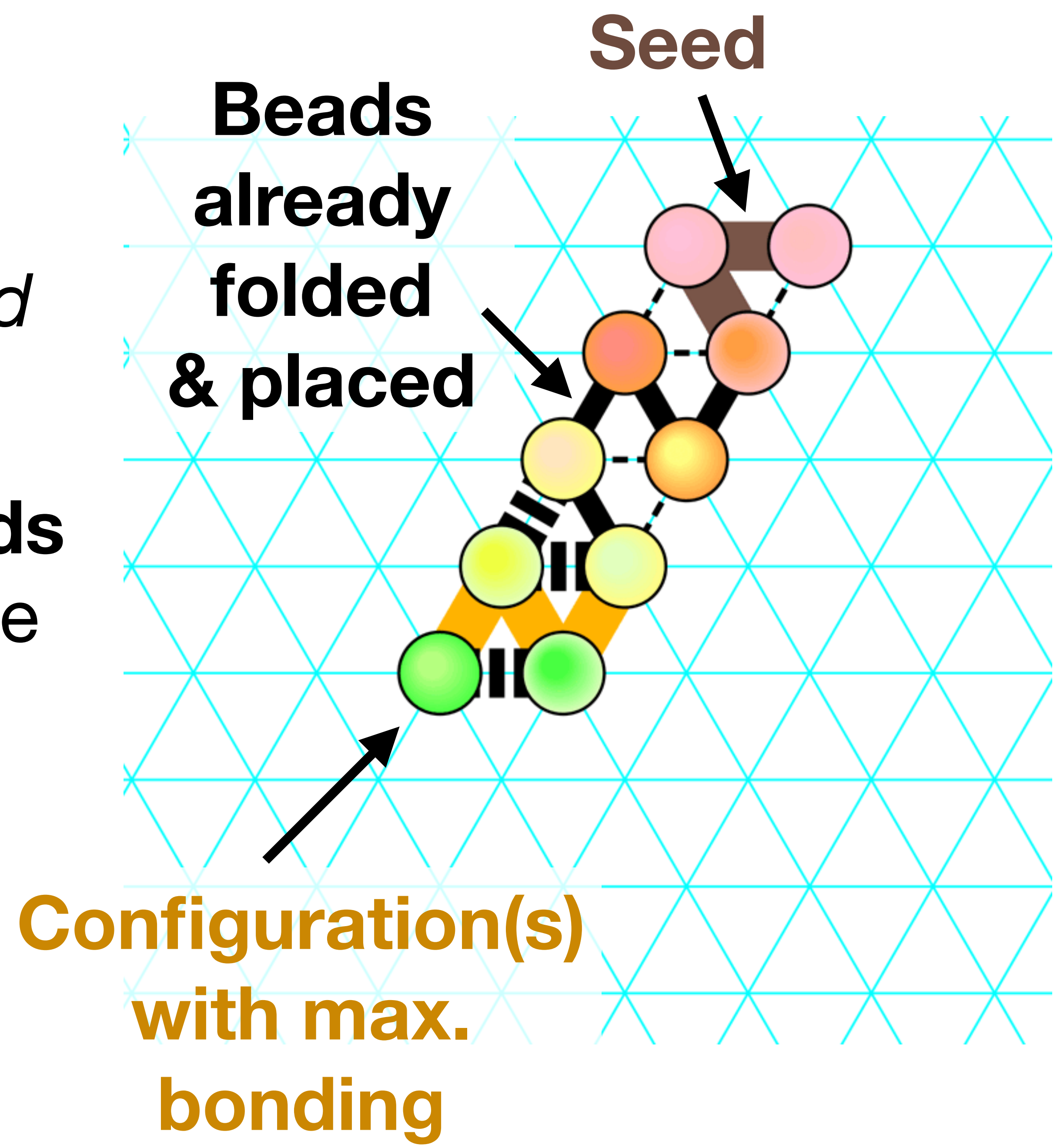


Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami:
A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

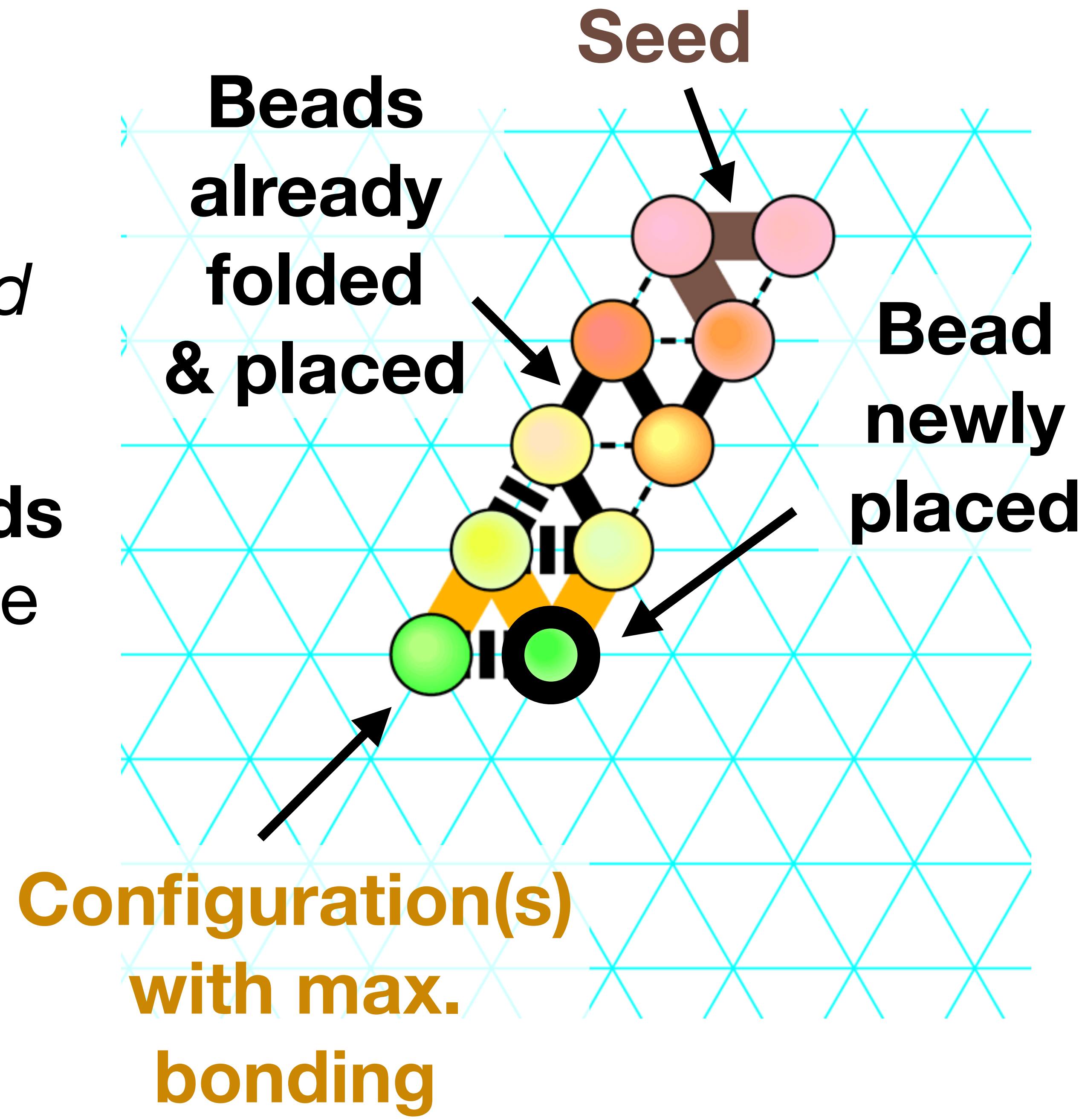


Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami: A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

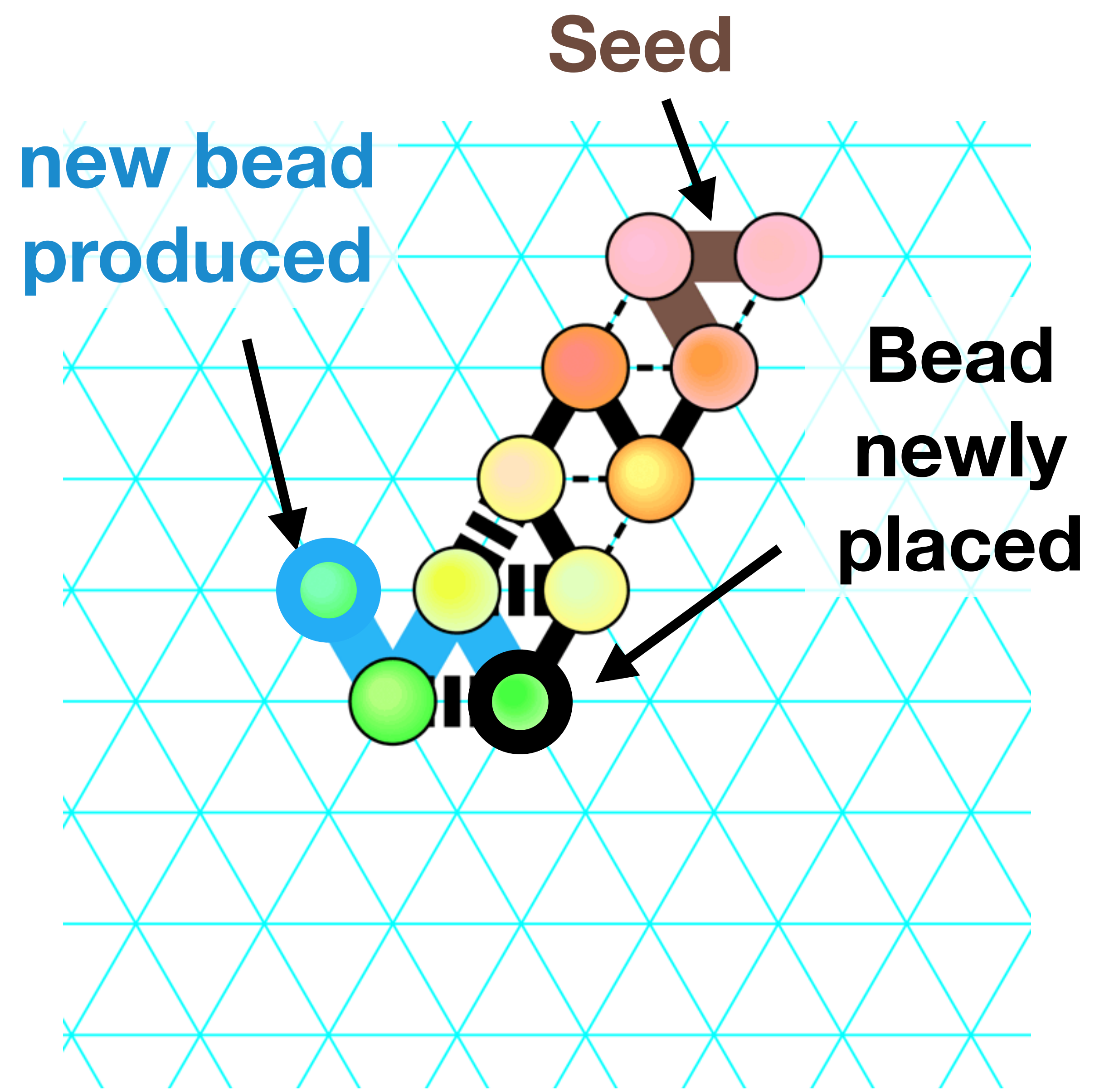


Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami:
A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

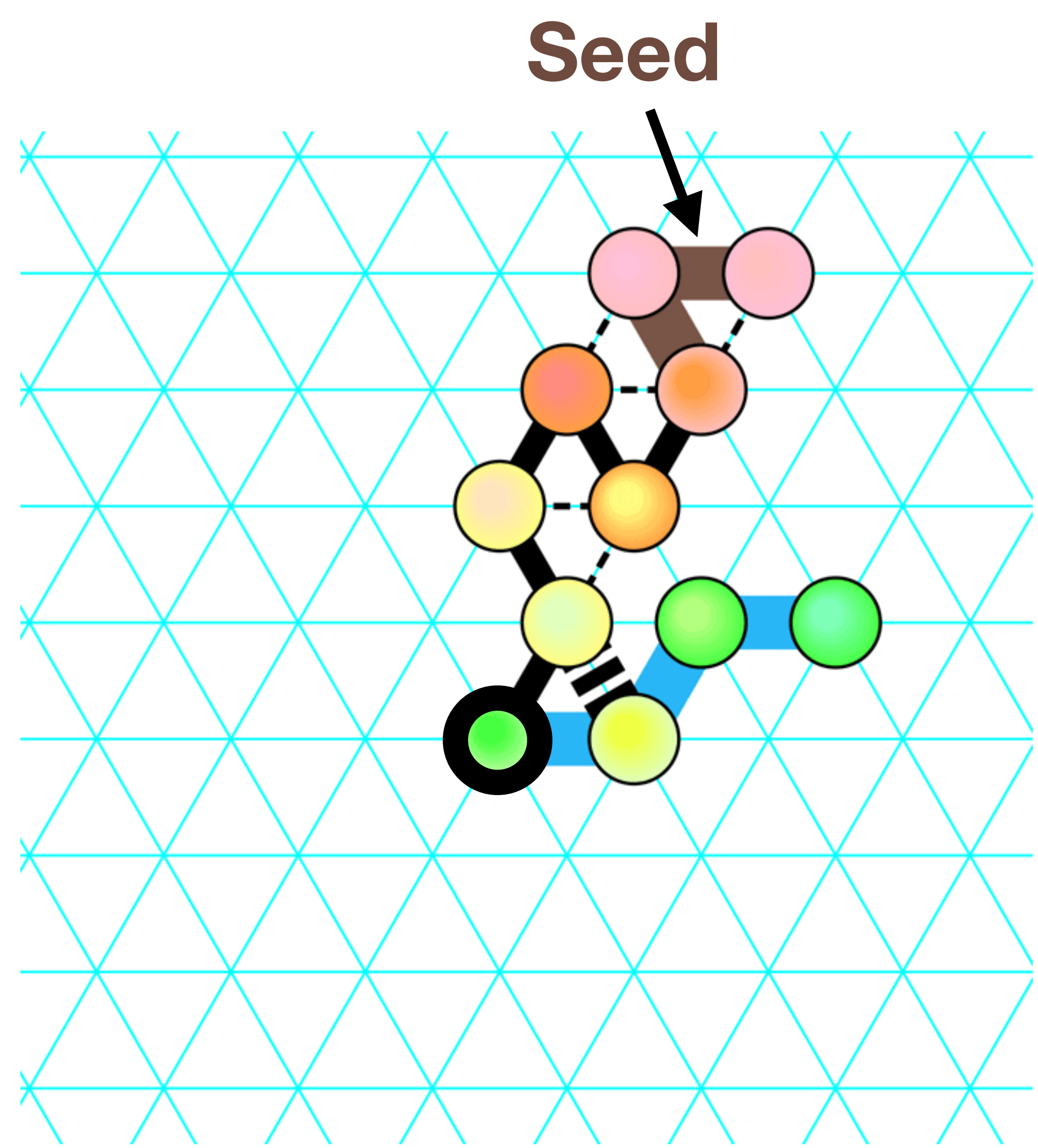


Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami:
A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

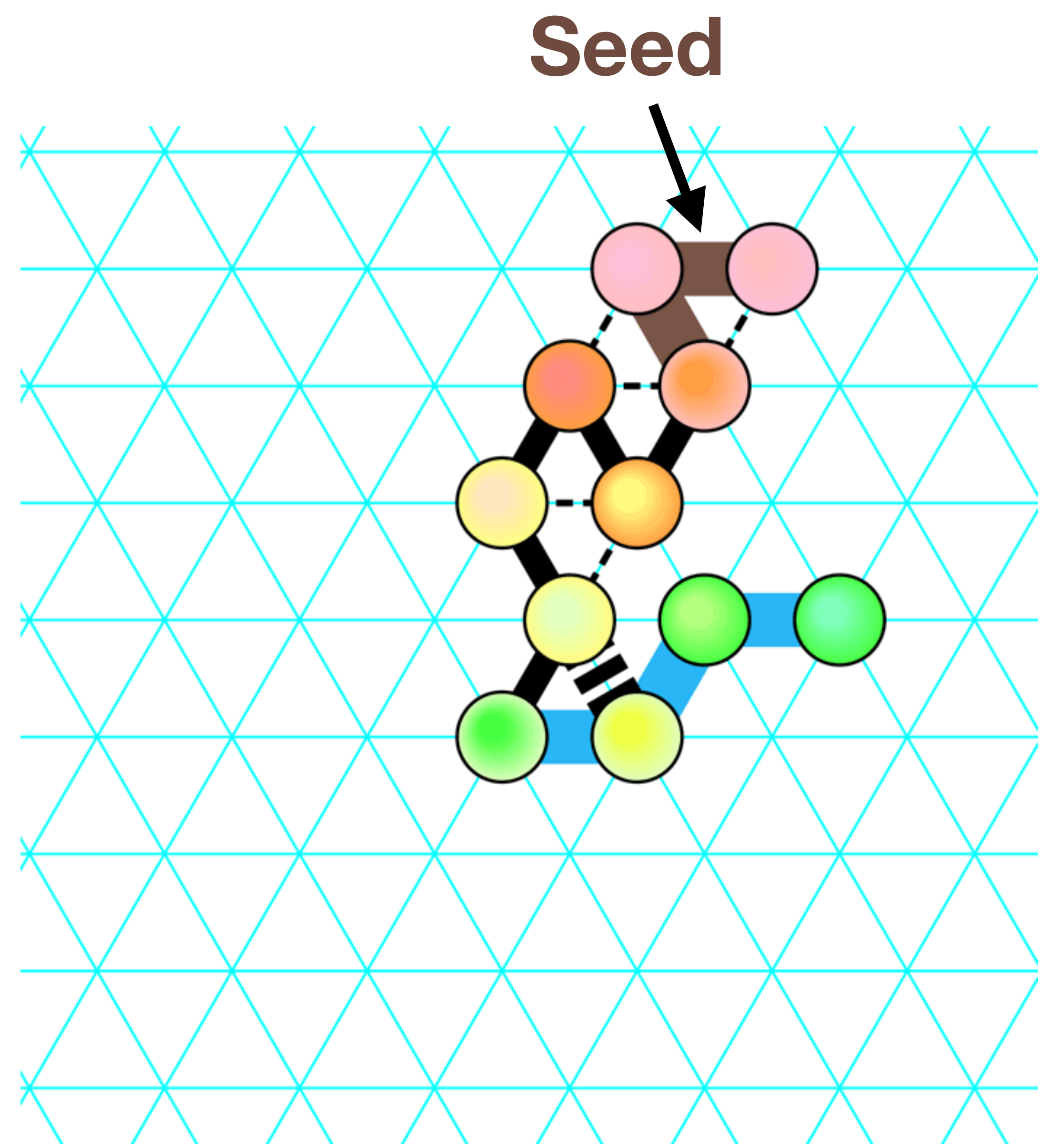


Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami:
A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

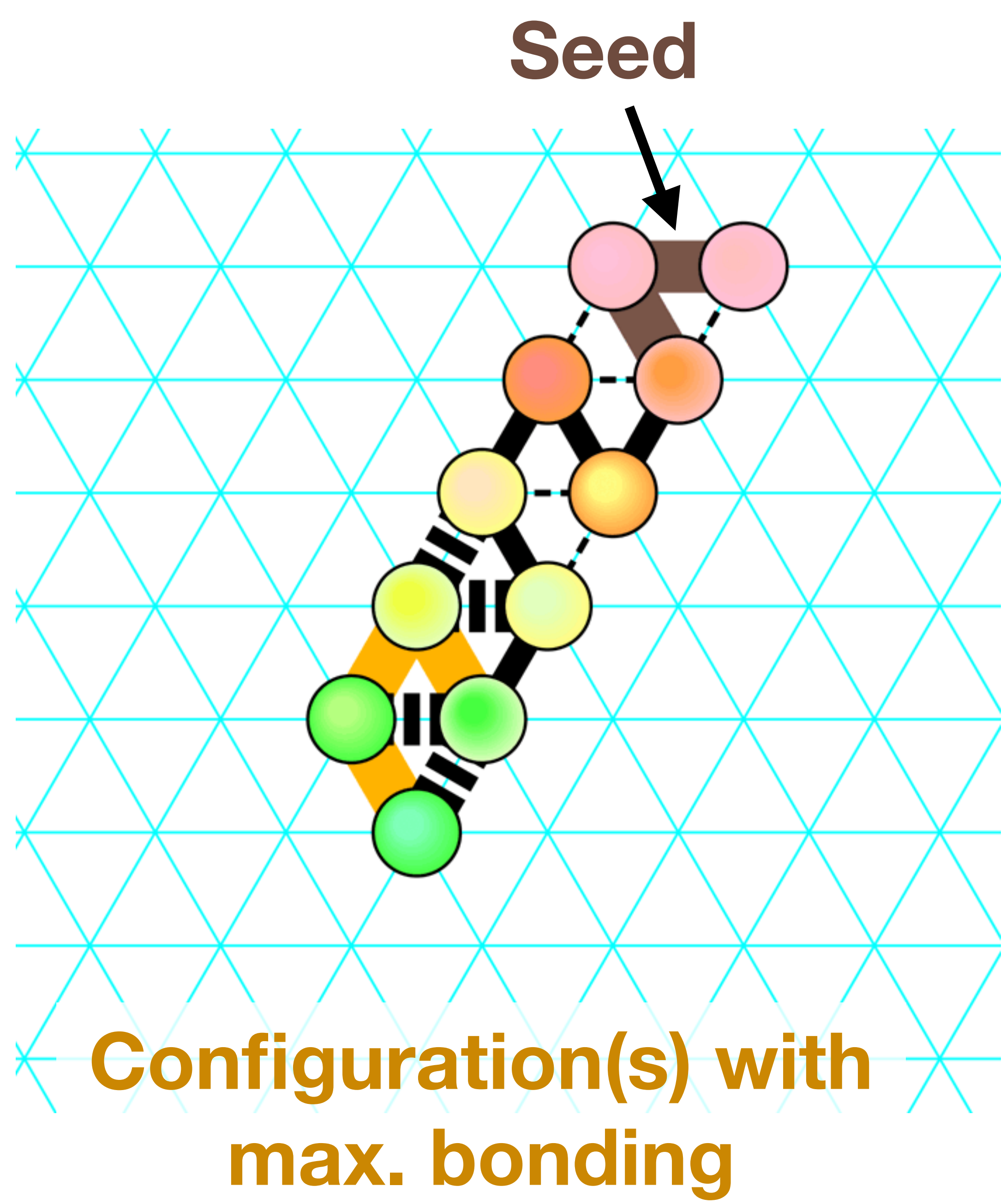


Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami: A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

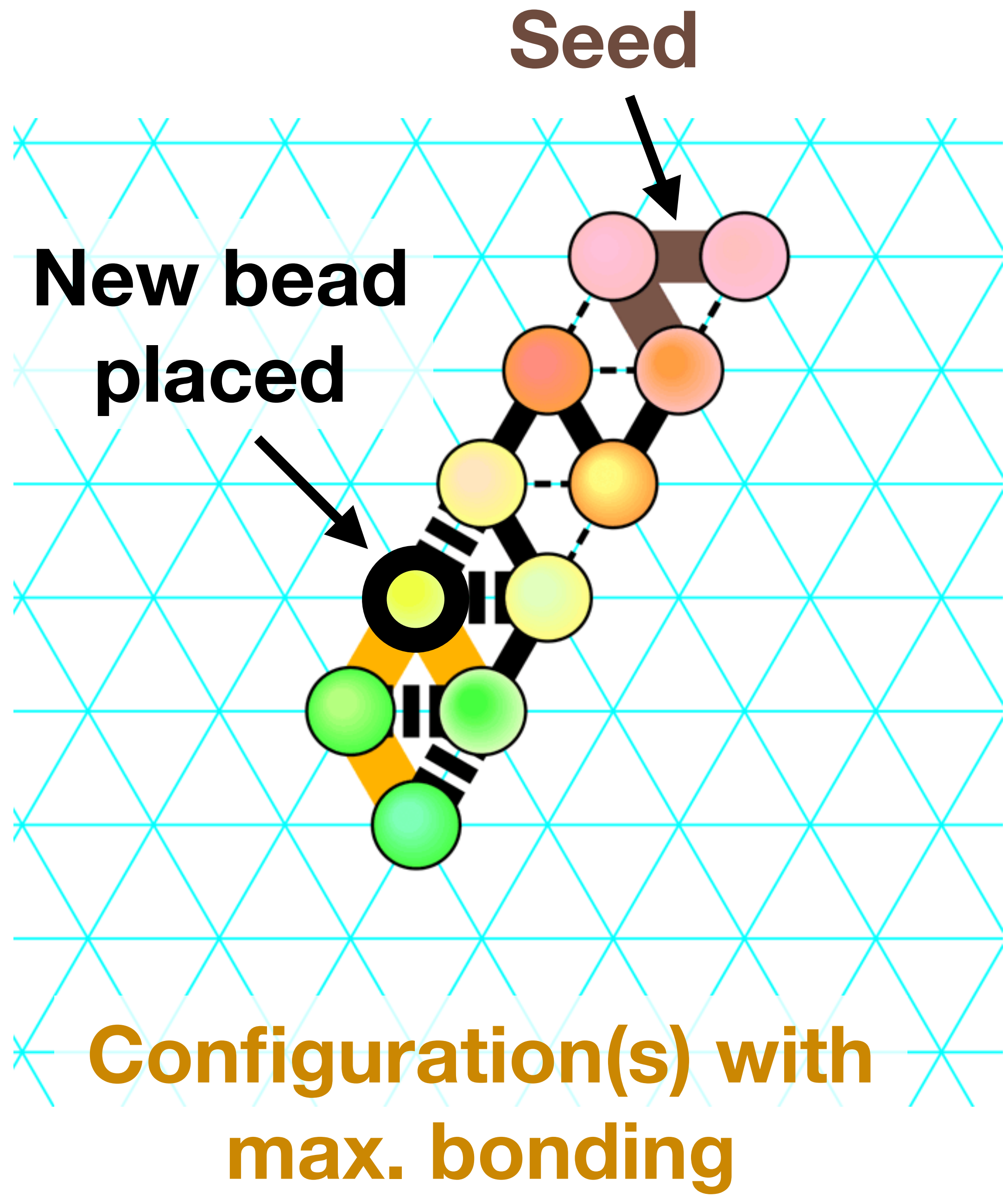


Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami:
A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations



Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami:
A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

There might be <u>several</u> configurations with max. bonding

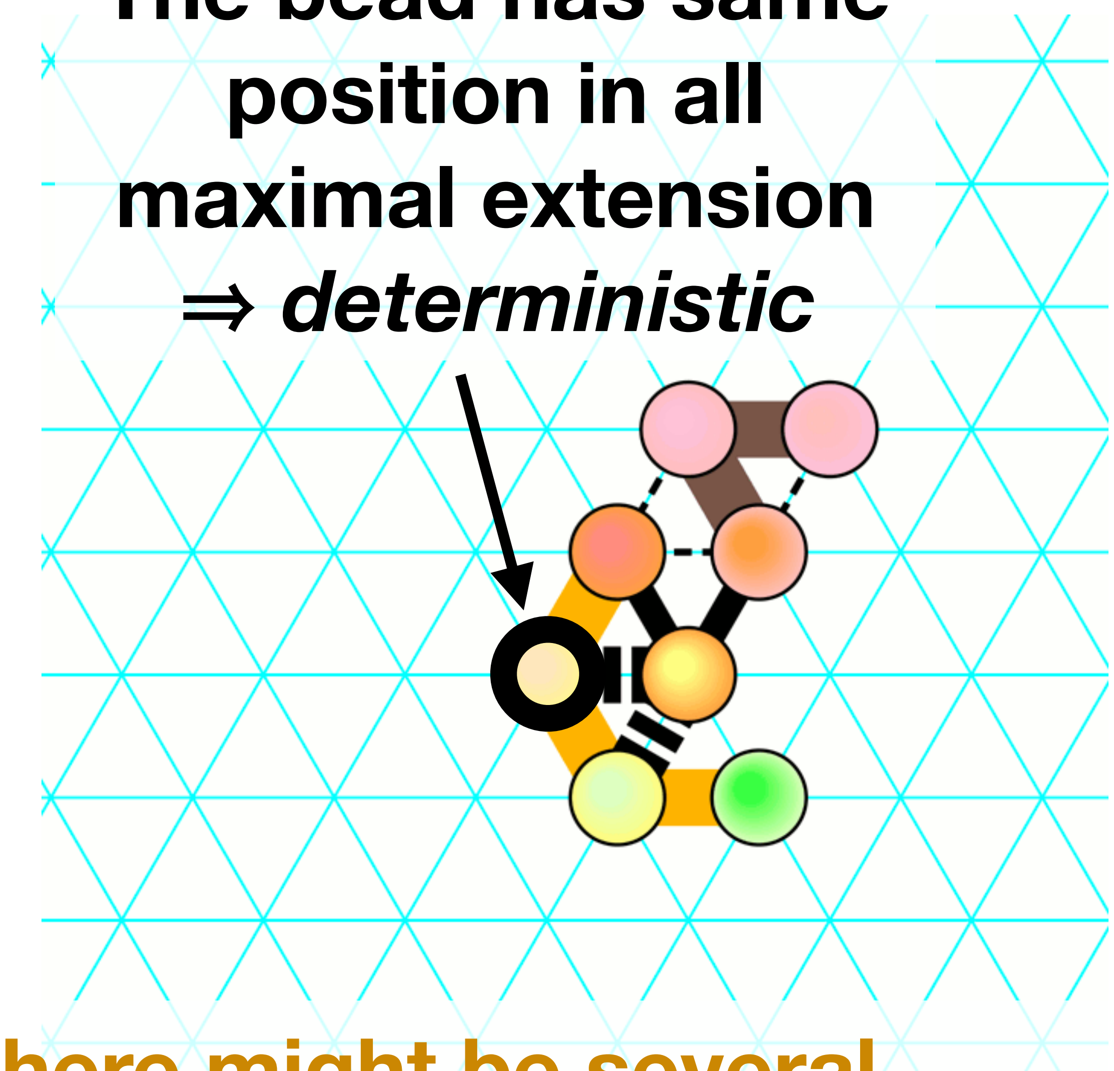
Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami: A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

The bead has same position in all maximal extension ⇒ *deterministic*



There might be <u>several</u> configurations with max. bonding

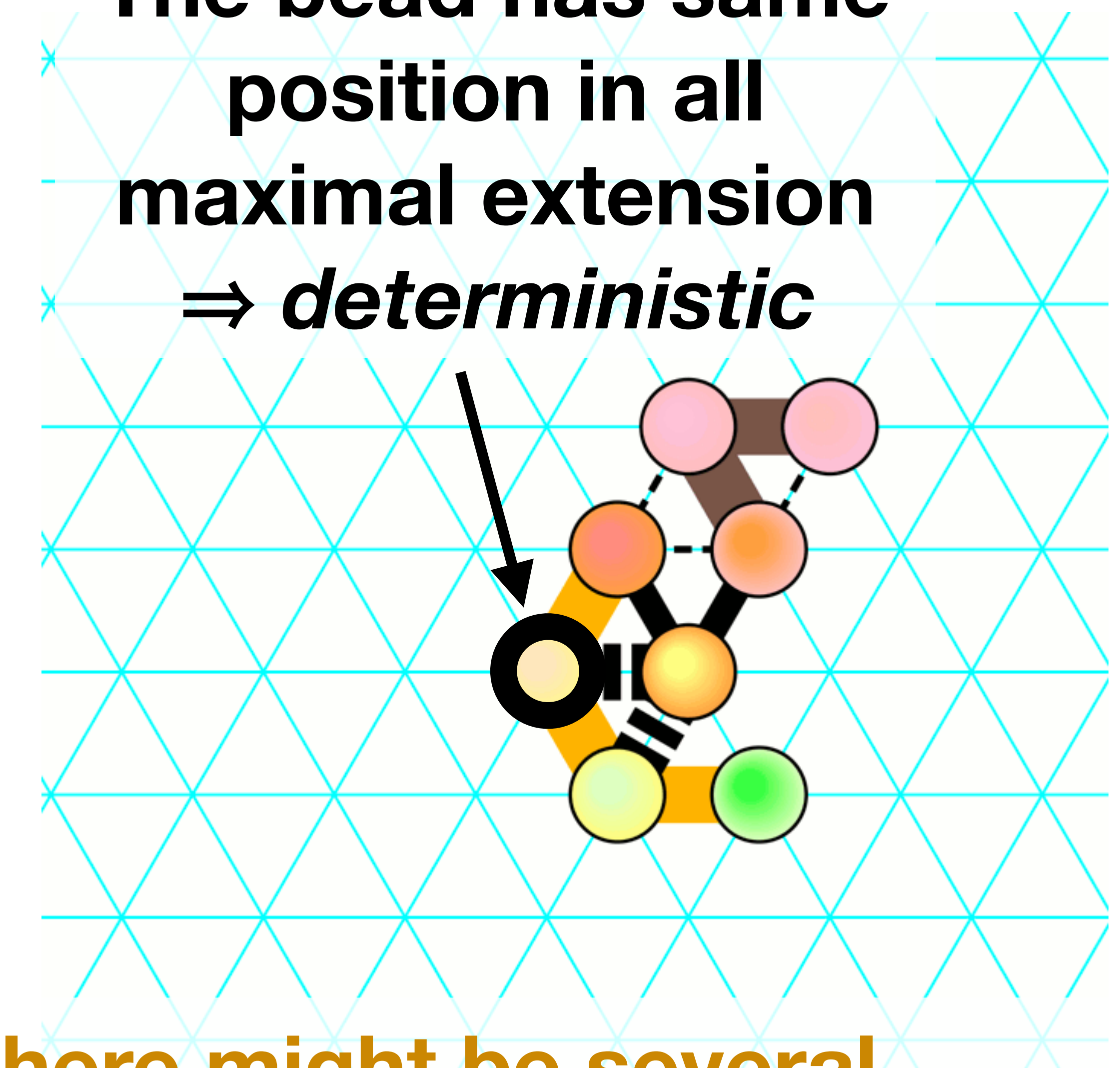
Geary, Meunier, Schabanel, Seki MFCS 2016

Oritatami:
A model for co-transcriptional folding

The dynamics.

- Starting from the seed, the sequence is *produced one bead at a time*
- **Only the δ last produced beads** are free to move and explore the accessible positions to settle in the ones **maximizing the number of bonds**
- All other beads remain in their last locations

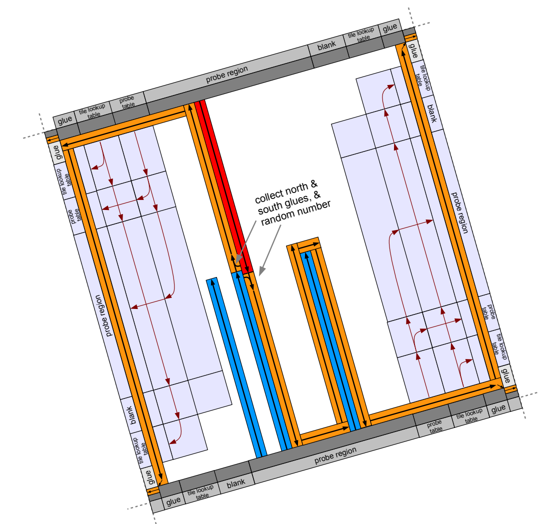
The bead has same position in all maximal extension ⇒ *deterministic*



There might be several configurations with max. bonding

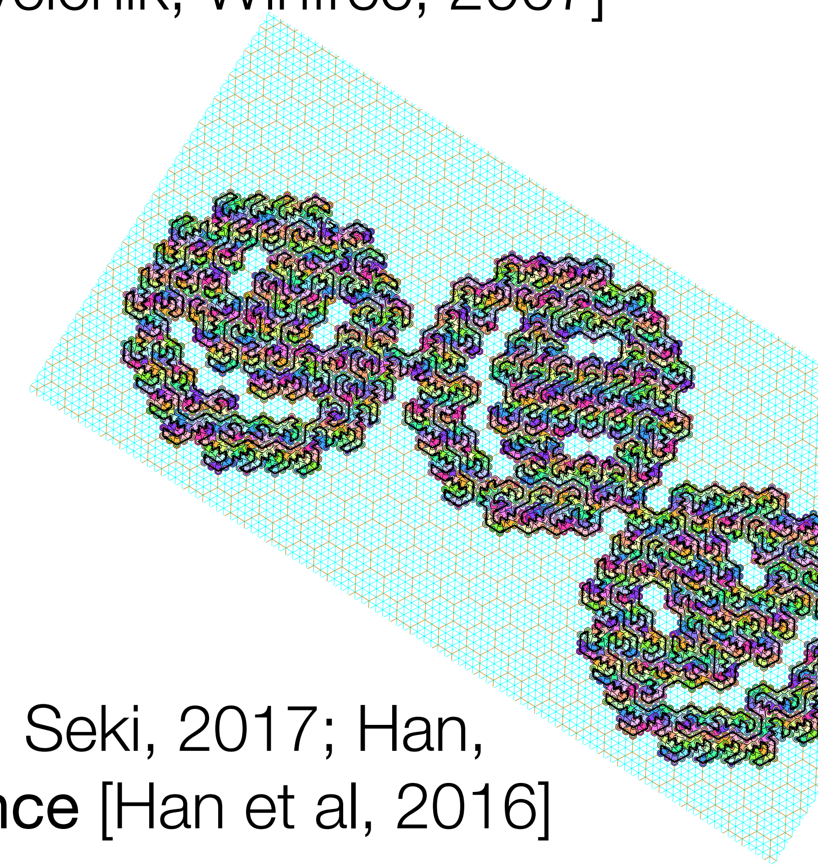
Geary, Meunier, Schabanel, Seki MFCS 2016

Previous work



Some abstract Tile Assembly seminal work

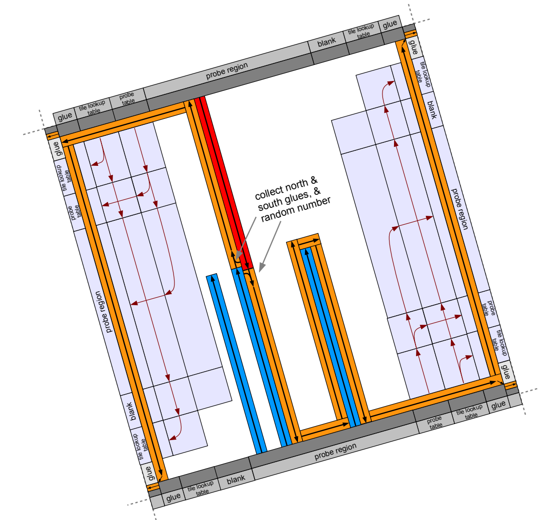
- Tile assembly systems are Turing universal [Winfree, 1998]
- Arbitrary shape assembly with optimal tile set size [Soloveichik, Winfree, 2007]
- Intrinsic universality [Doty et al, 2012]
- Uncomputable limit configuration [Lathrop et al, 2011]



Oritatami

- A binary counter [Geary, Meunier, S., Seki, 2016]
- Heighdragon fractal [Masuda, Seki, Ubukata, 2018]
- Folding arbitrary shapes [Demaine et al, 2018]
- NP-hardness for oritatami design [Geary et al, 2016; Ota, Seki, 2017; Han, Kim, 2017] and for non-determinisitic oritatami equivalence [Han et al, 2016]
- Efficient Turing Machine simulation through tag-systems [Geary et al, 2018]
- Intrinsic 1D Cellular Automata simulation [Pchelina et al, 2020]

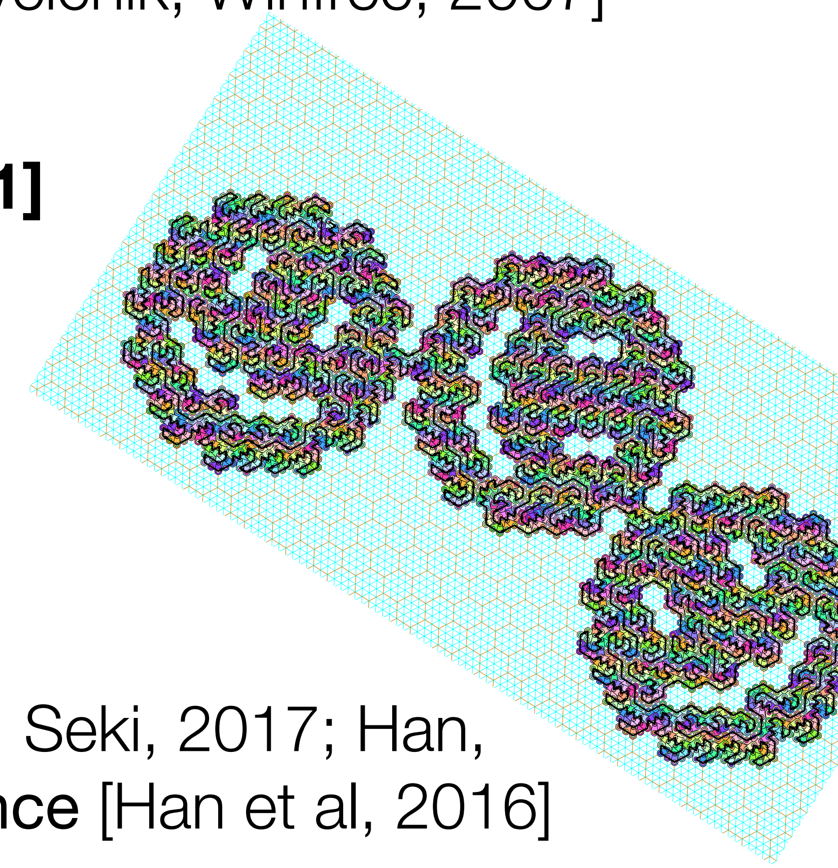
Previous work



Some abstract Tile Assembly seminal work

- Tile assembly systems are Turing universal [Winfree, 1998]
- Arbitrary shape assembly with optimal tile set size [Soloveichik, Winfree, 2007]
- Intrinsic universality [Doty et al, 2012]

★ **Uncomputable limit configuration [Lathrop et al, 2011]**



Oritatami

- A binary counter [Geary, Meunier, S., Seki, 2016]
- Heighdragon fractal [Masuda, Seki, Ubukata, 2018]
- Folding arbitrary shapes [Demaine et al, 2018]
- NP-hardness for oritatami design [Geary et al, 2016; Ota, Seki, 2017; Han, Kim, 2017] and for non-determinisitic oritatami equivalence [Han et al, 2016]
- Efficient Turing Machine simulation through tag-systems [Geary et al, 2018]
- Intrinsic 1D Cellular Automata simulation [Pchelina et al, 2020]

★ **TODAY: Uncomputable limit configuration & Turmite intrinsinc simulation**

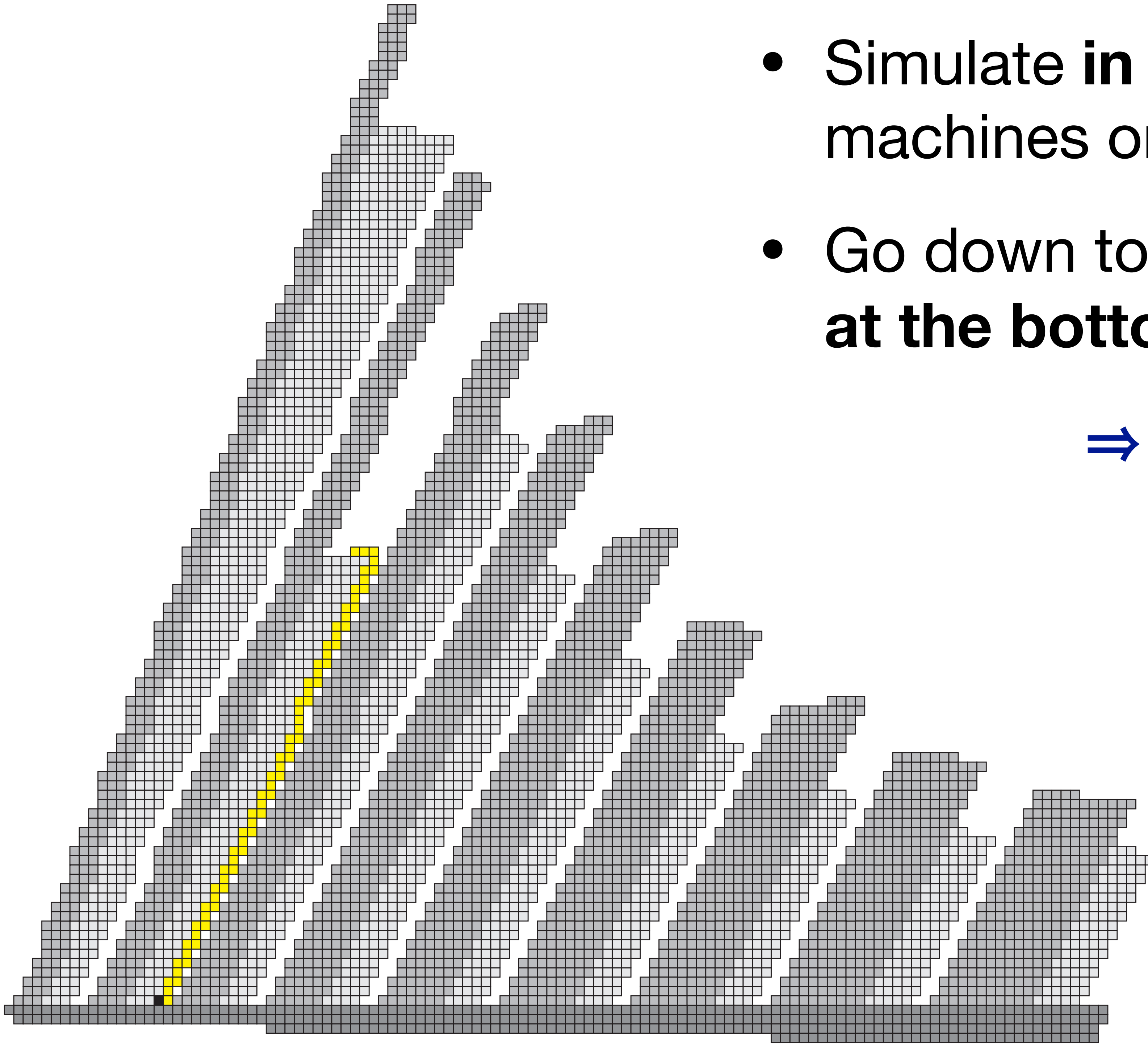
Uncomputable? Limit configuration?

- The **limit configuration c∞** is the configuration at the end of time:

$$c^0 \subset \cdots \subset c^t \subset c^{t+1} \subset \cdots \subset c^\infty = \bigcup_{t=0}^{\infty} c^t$$

- c∞ is **uncomputable** is the function $(i,j) \mapsto c^\infty_{i,j}$ is uncomputable

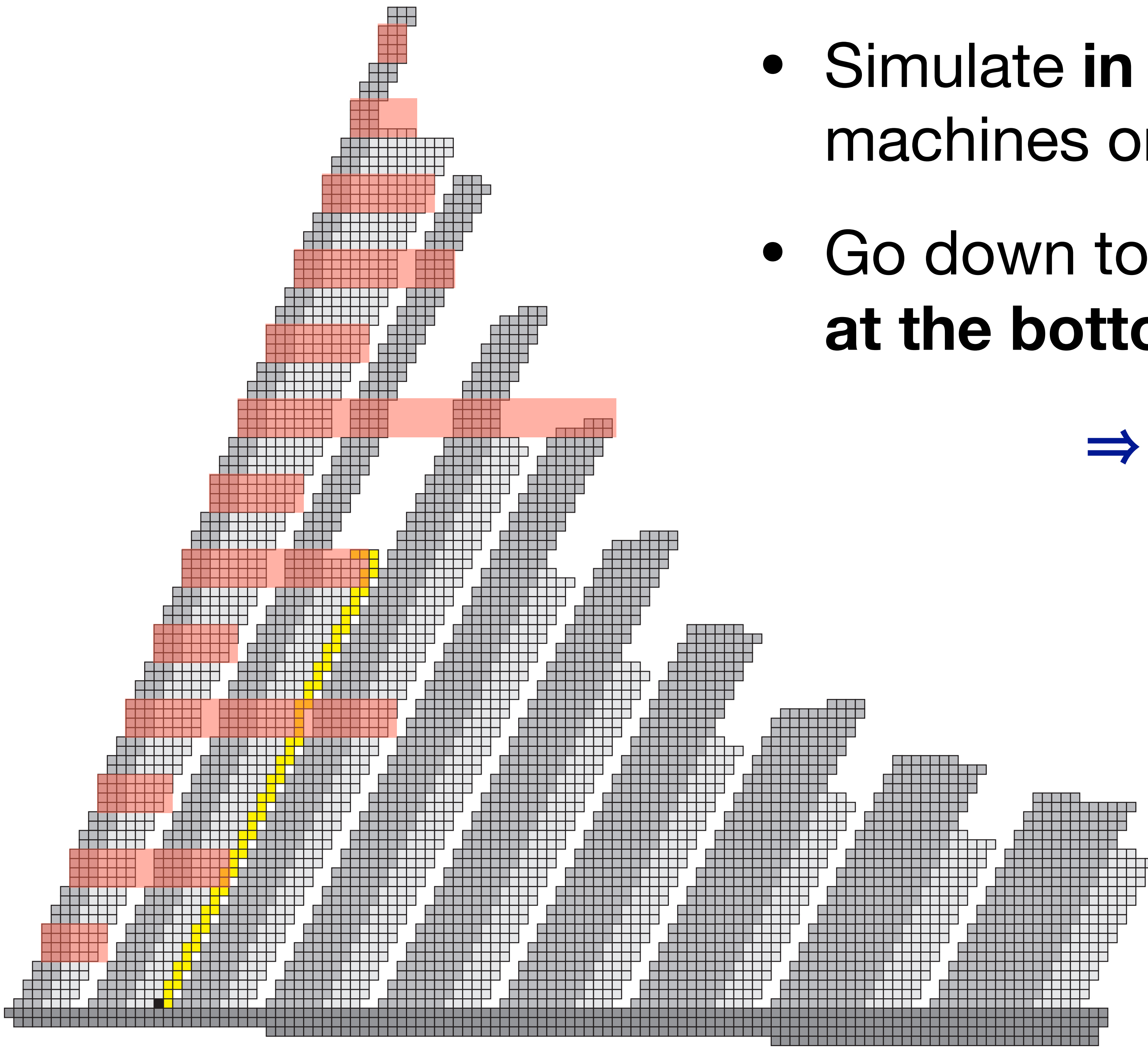
Uncomputable limit configuration in aTAM



- Simulate **in parallel** all Turing machines on an empty input
- Go down to **place a (black) tile at the bottom if the TM halts**

⇒ **The bottom row is uncomputable**

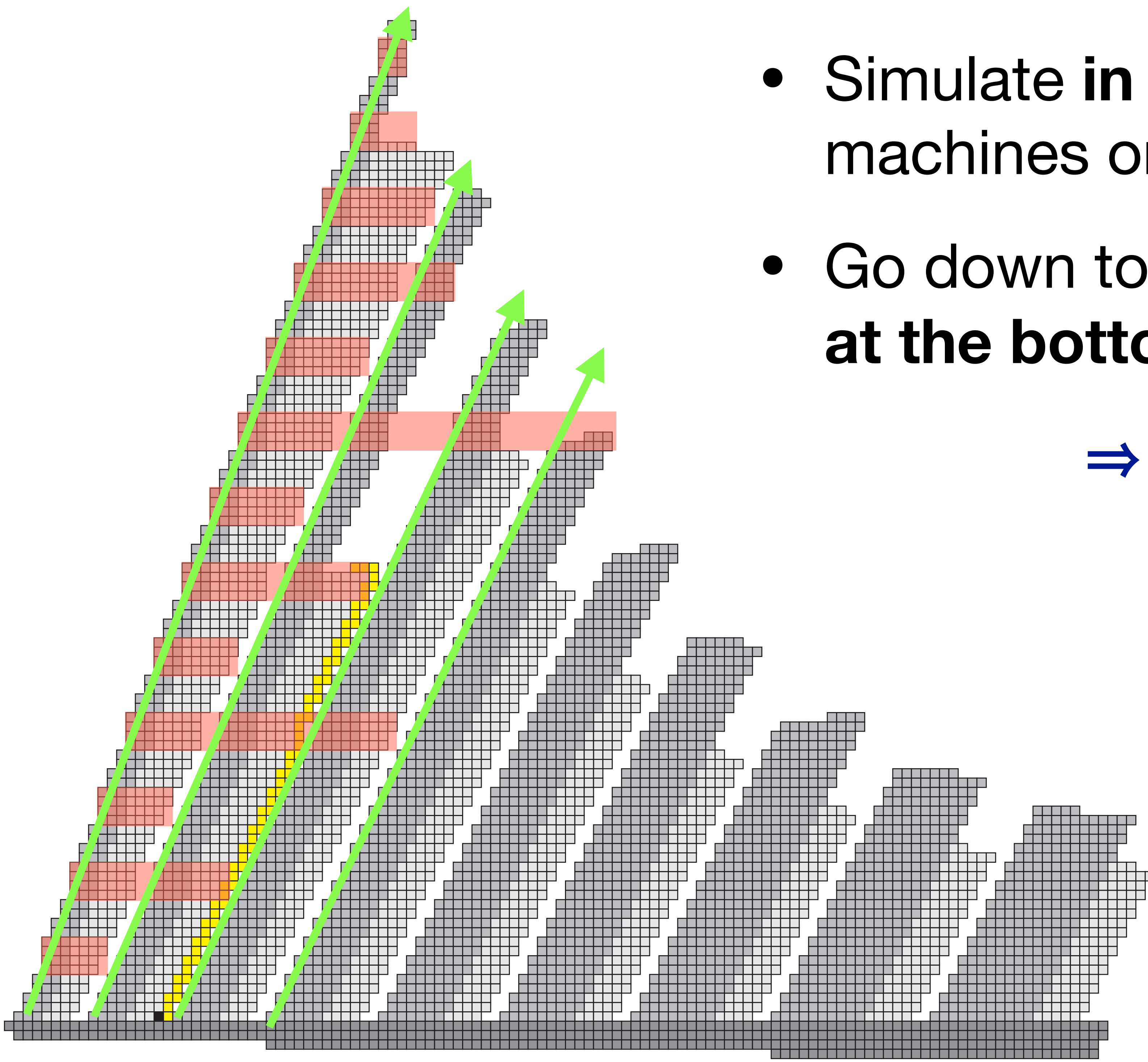
Uncomputable limit configuration in aTAM



- Simulate **in parallel** all Turing machines on an empty input
- Go down to **place a (black) tile at the bottom if the TM halts**

⇒ **The bottom row is uncomputable**

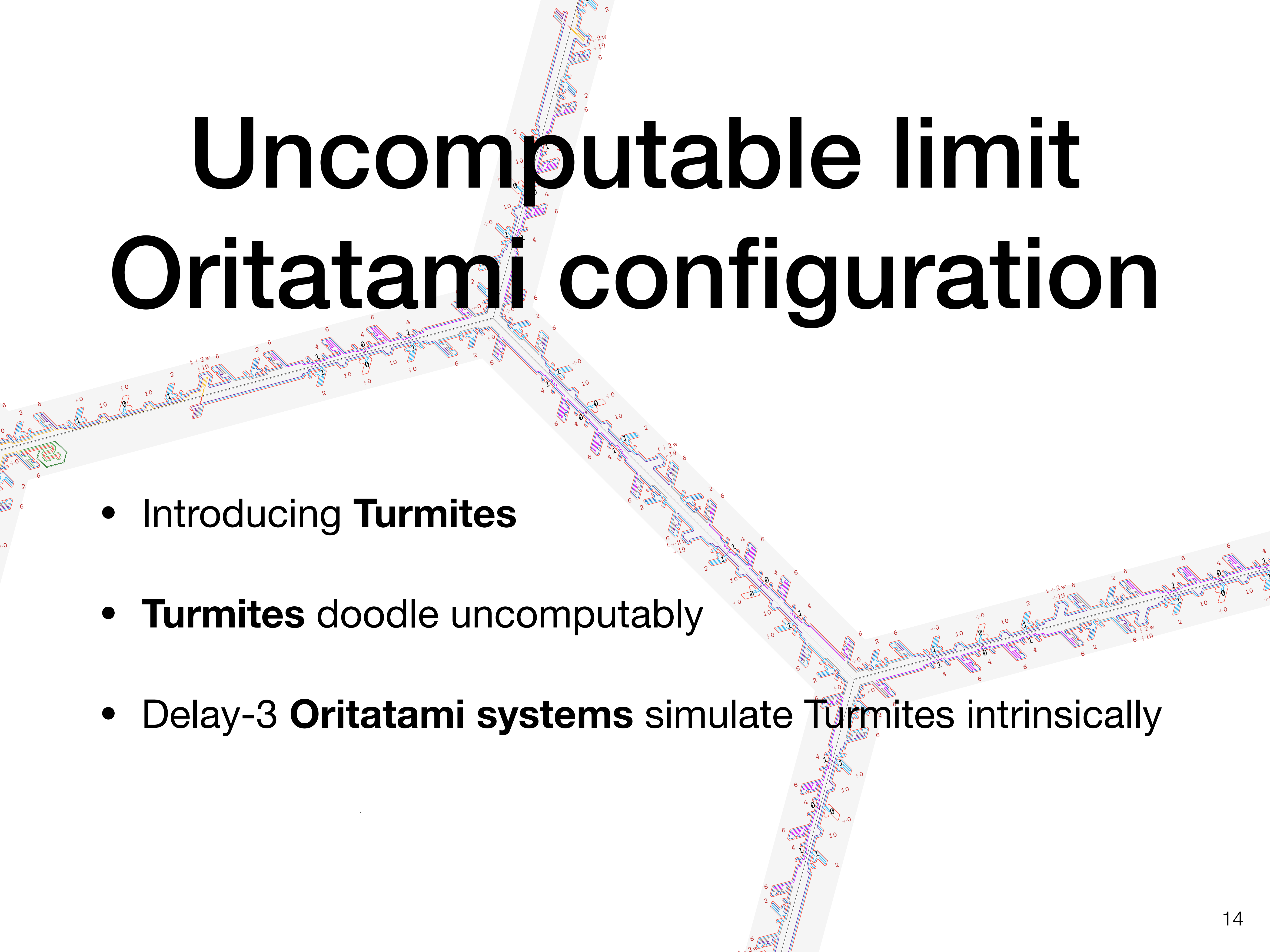
Uncomputable limit configuration in aTAM



- Simulate **in parallel** all Turing machines on an empty input
- Go down to **place a (black) tile at the bottom if the TM halts**

⇒ The bottom row is uncomputable

Uncomputable limit Oritatami configuration



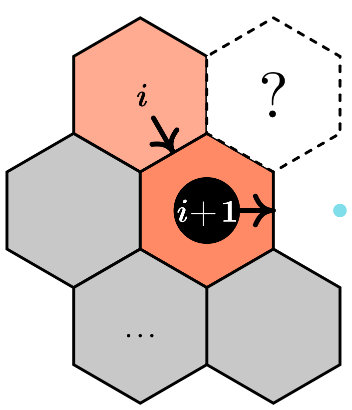
- Introducing **Turmites**
- **Turmites** doodle uncomputably
- Delay-3 **Oritatami systems** simulate Turmites intrinsically

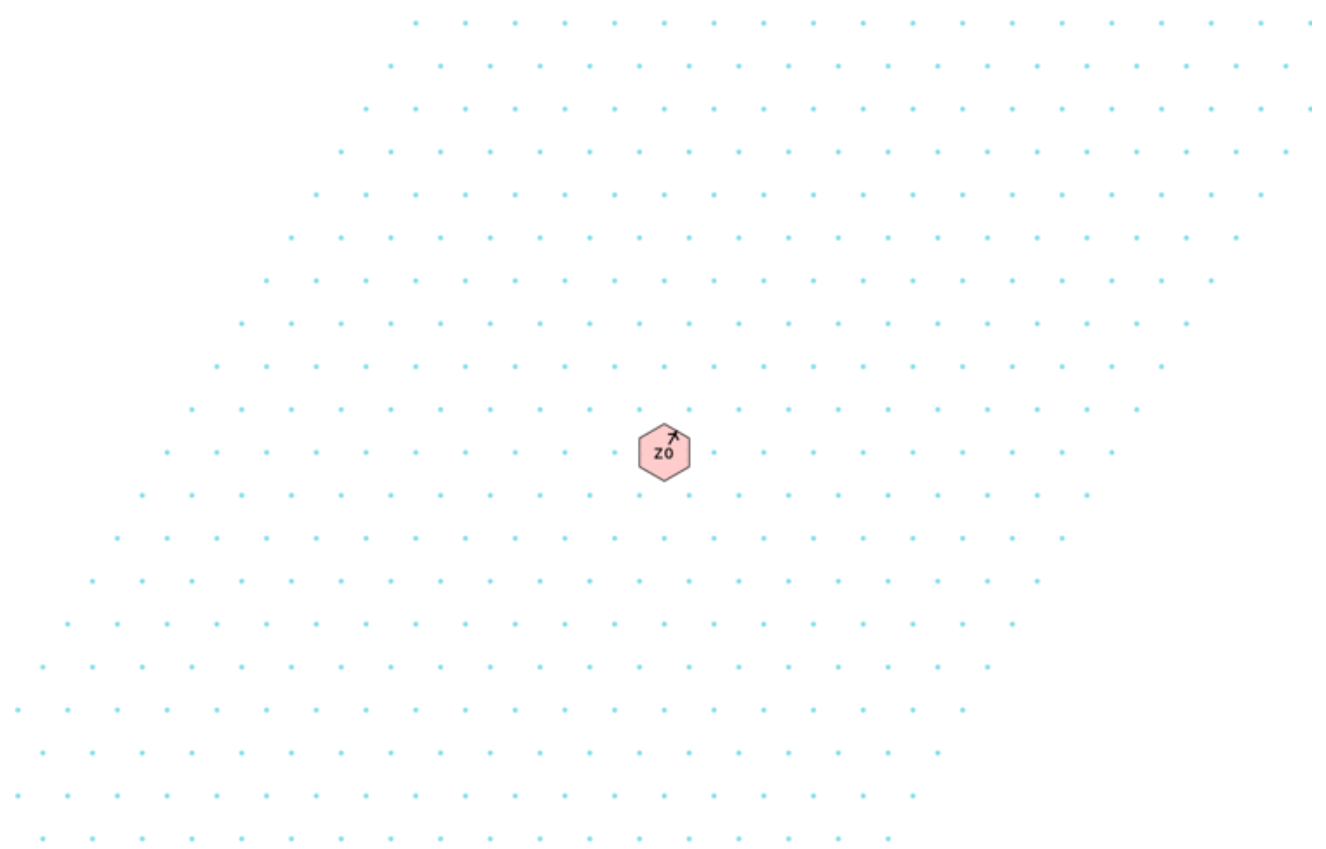
Turmites

A **finite automata** follows a **self-avoiding path**, **moving** and **writing** a state according to a **uniform local rule**

A clockwise walker

The rule:



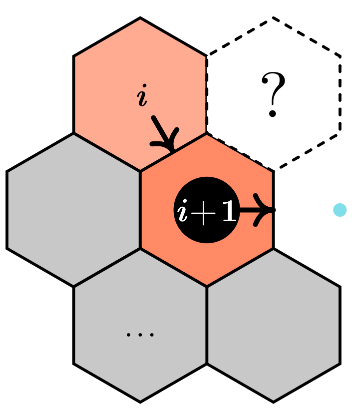


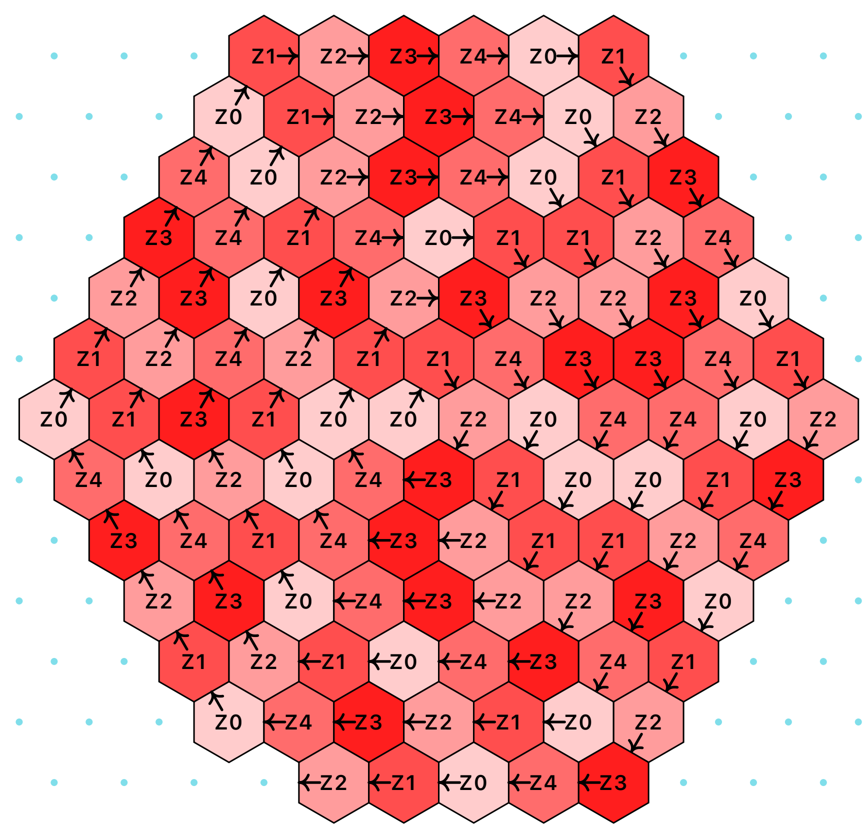
Turmites

A **finite automata** follows a **self-avoiding path, moving** and **writing** a state according to a **uniform local rule**

A clockwise walker

The rule:

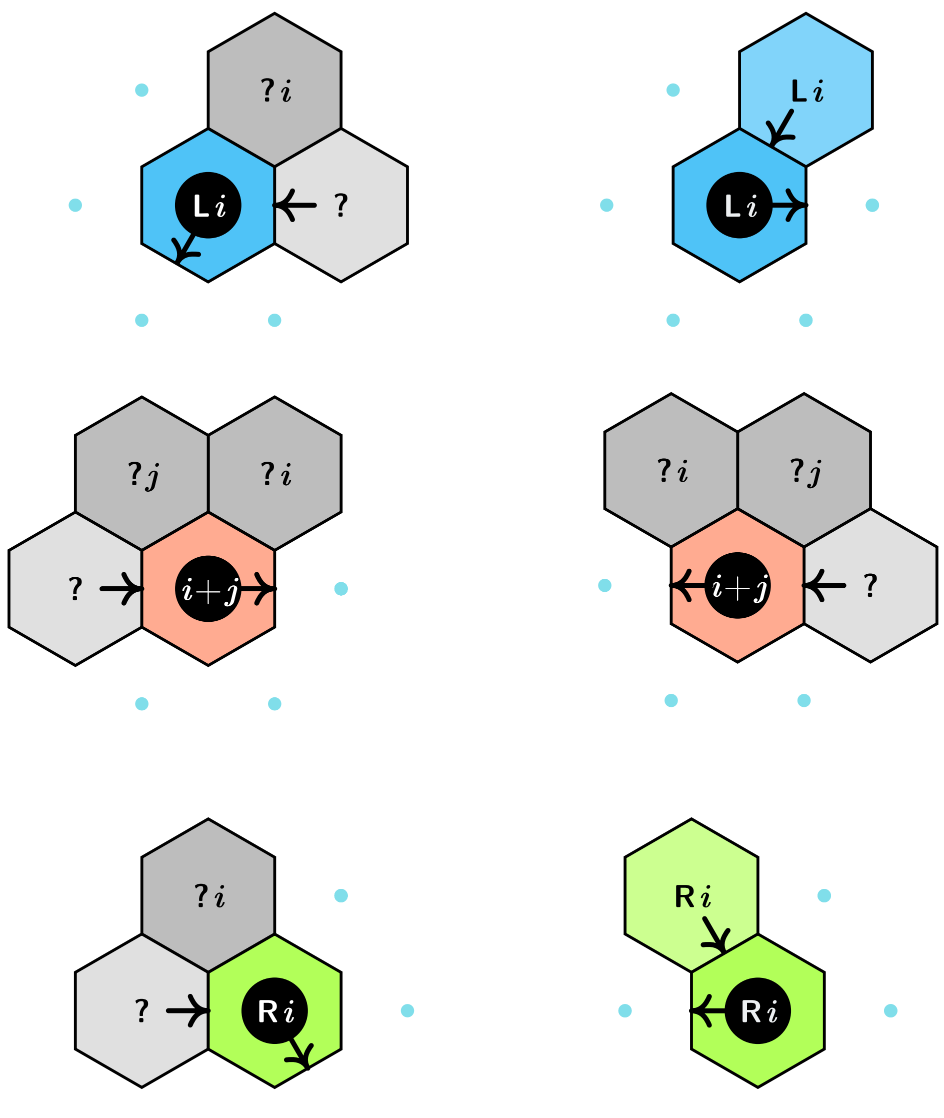


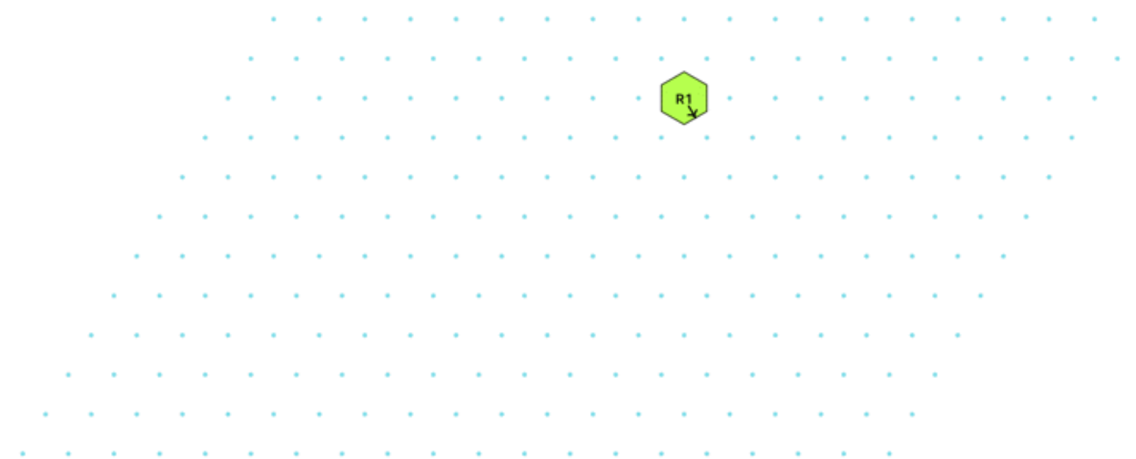


Turmites implement CA

Left/Right Swiping

The rule:

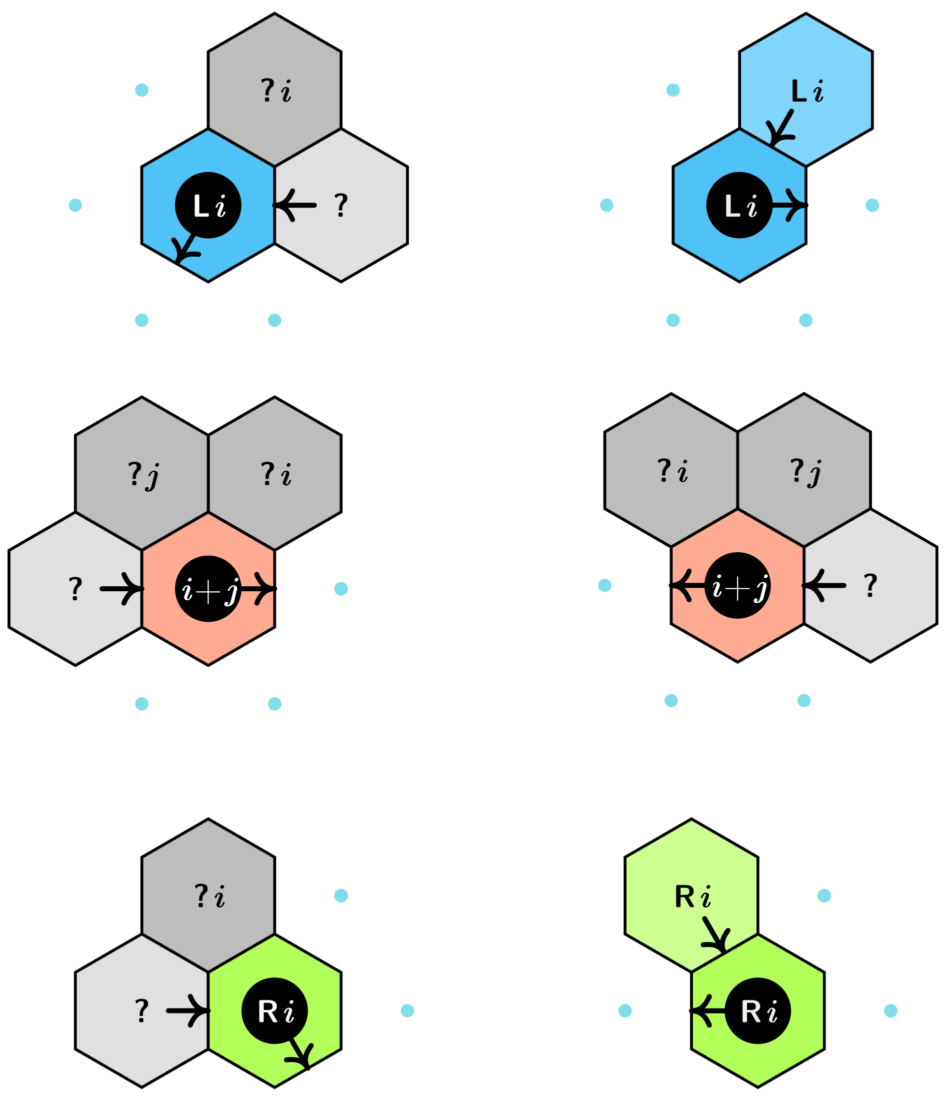


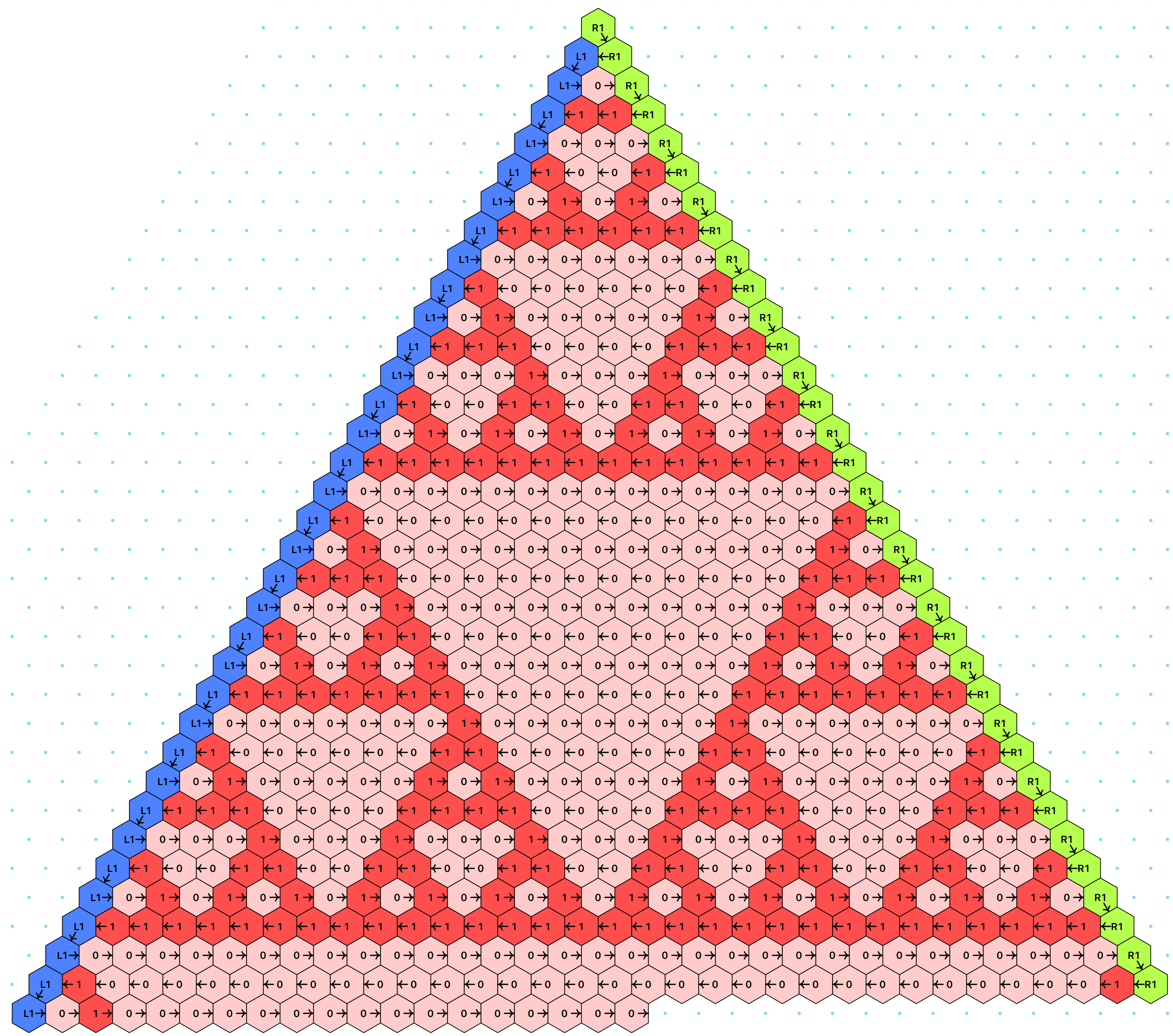


Turmites implement CA

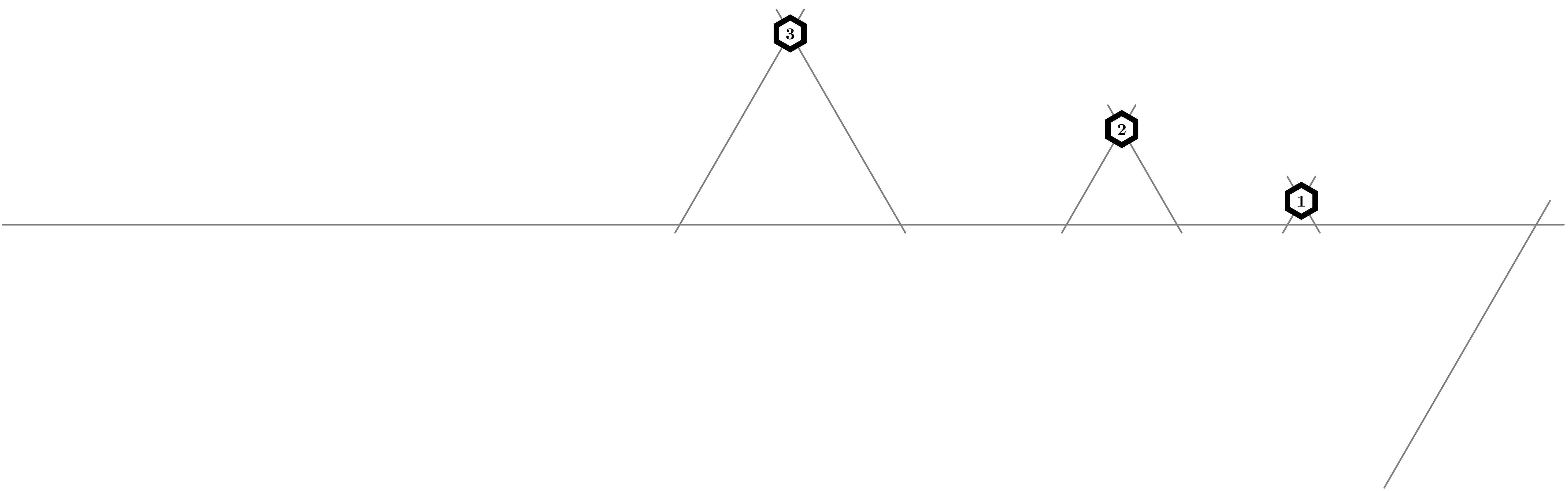
Left/Right Swiping

The rule:

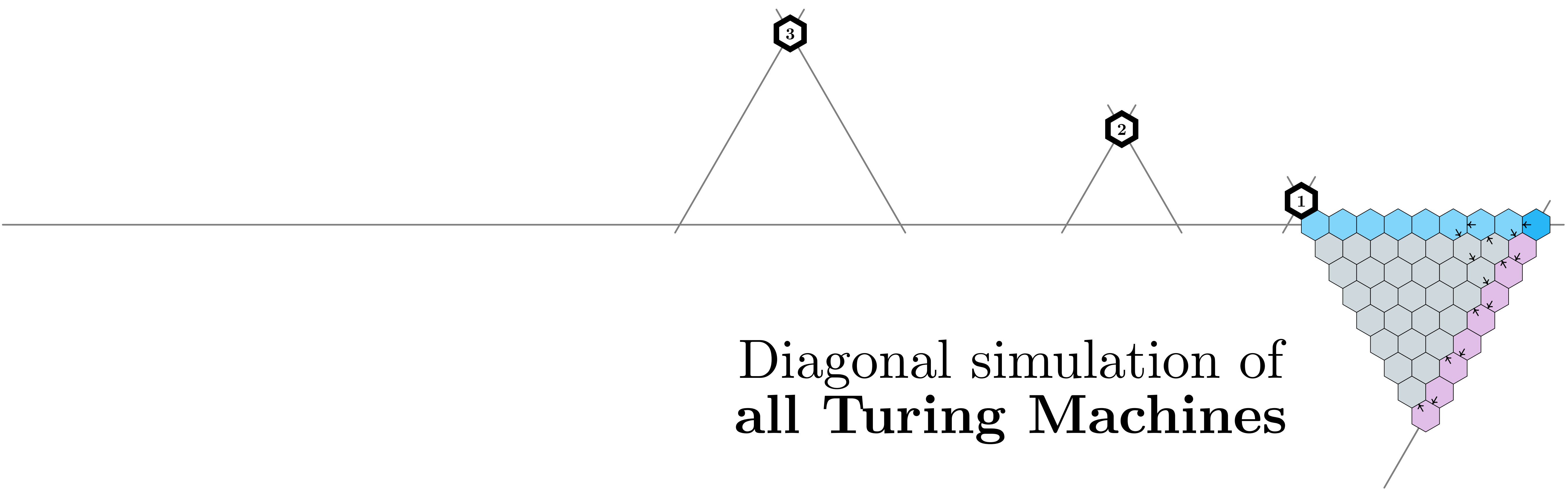




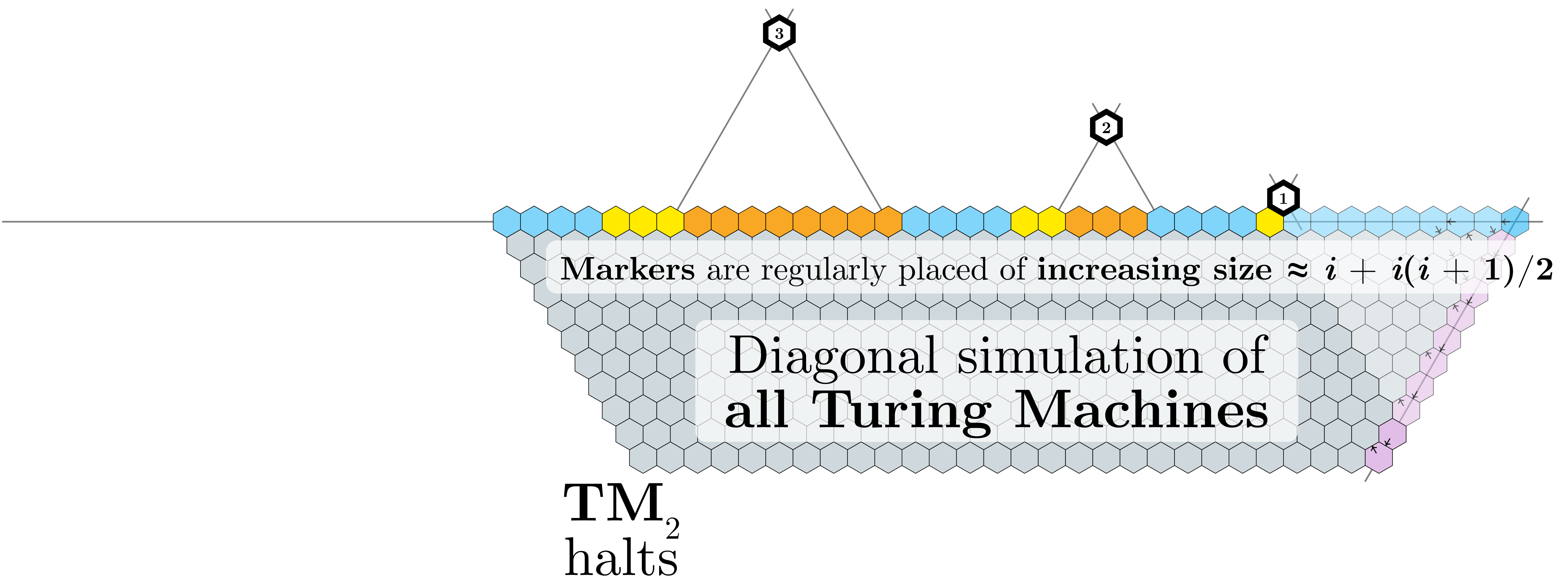
Turmites doodle uncomputably



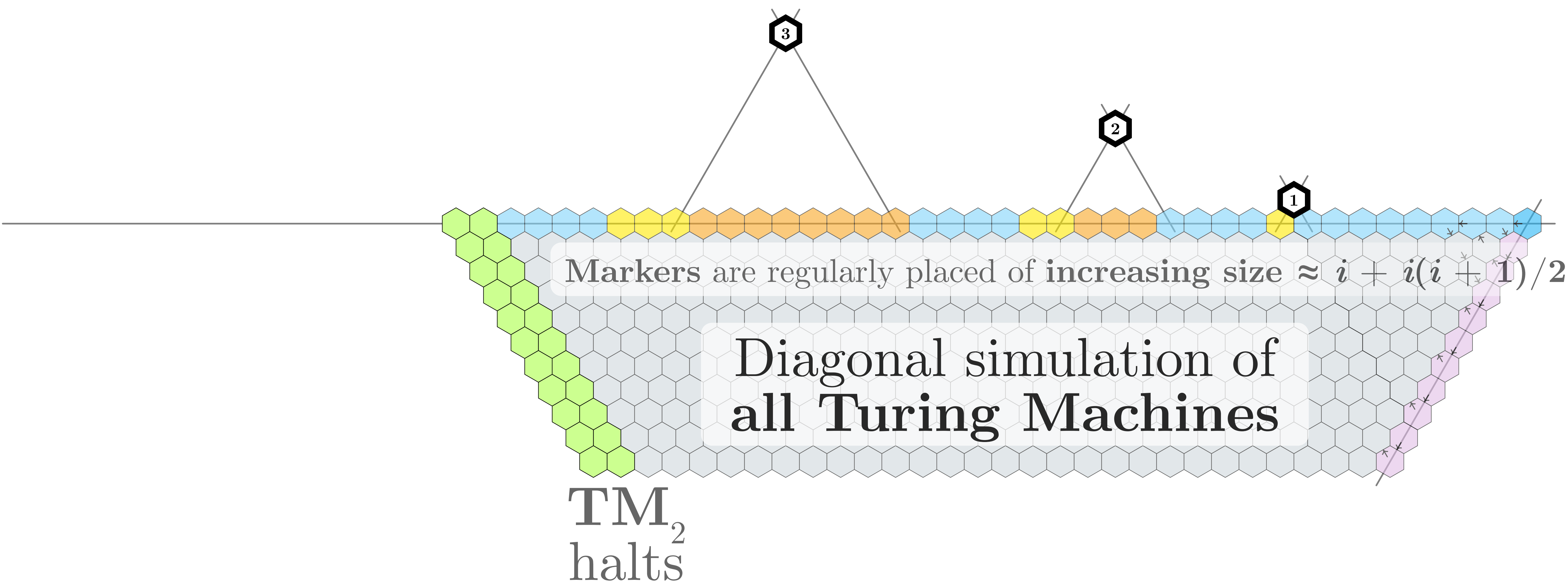
Turmites doodle uncomputably



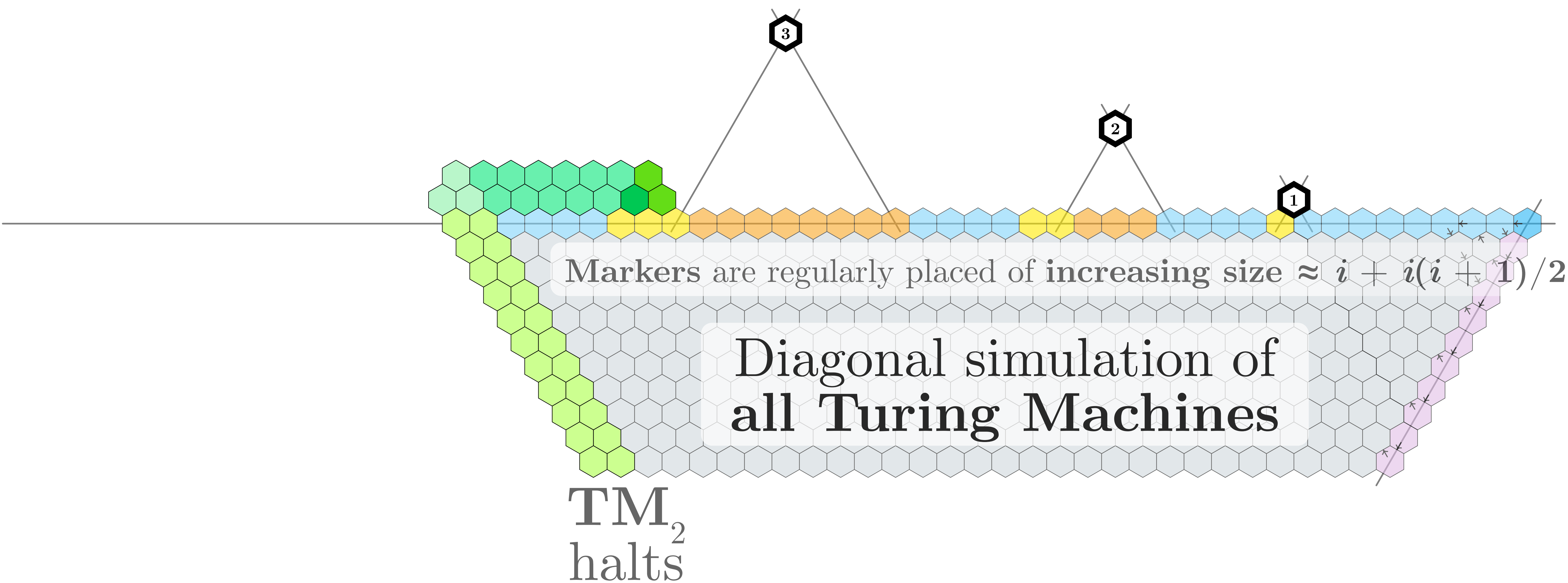
Turmites doodle uncomputably



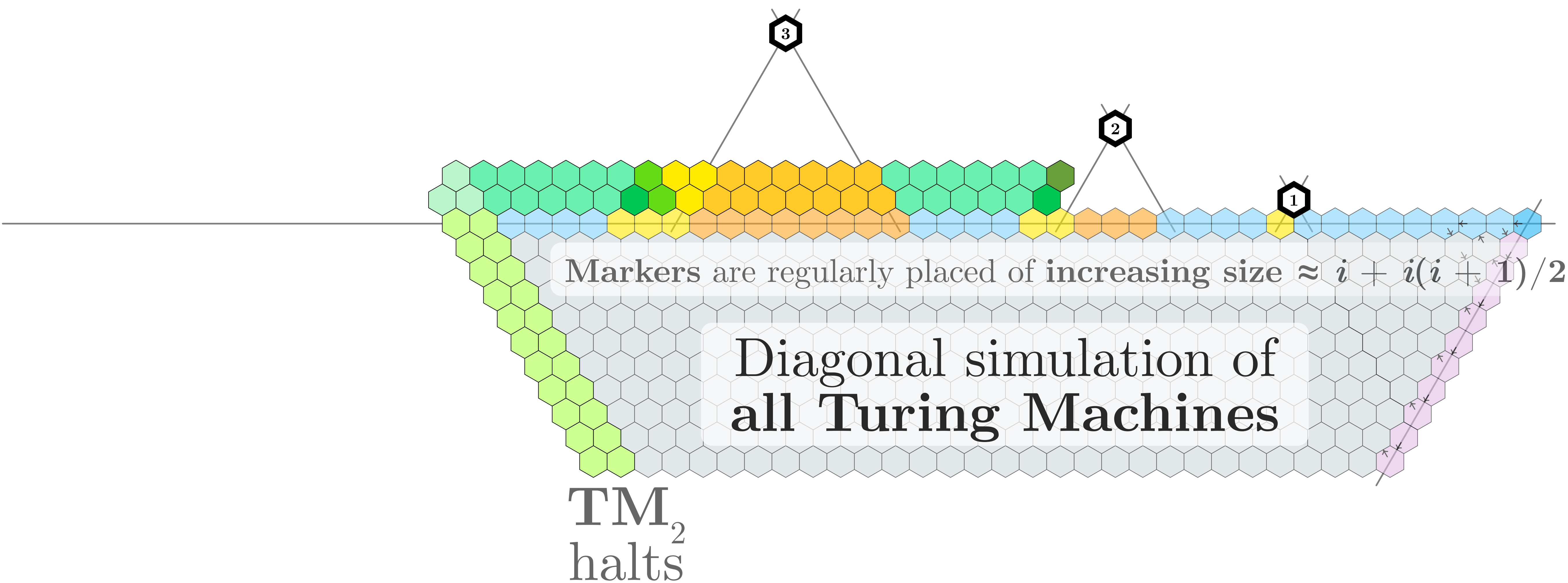
Turmites doodle uncomputably



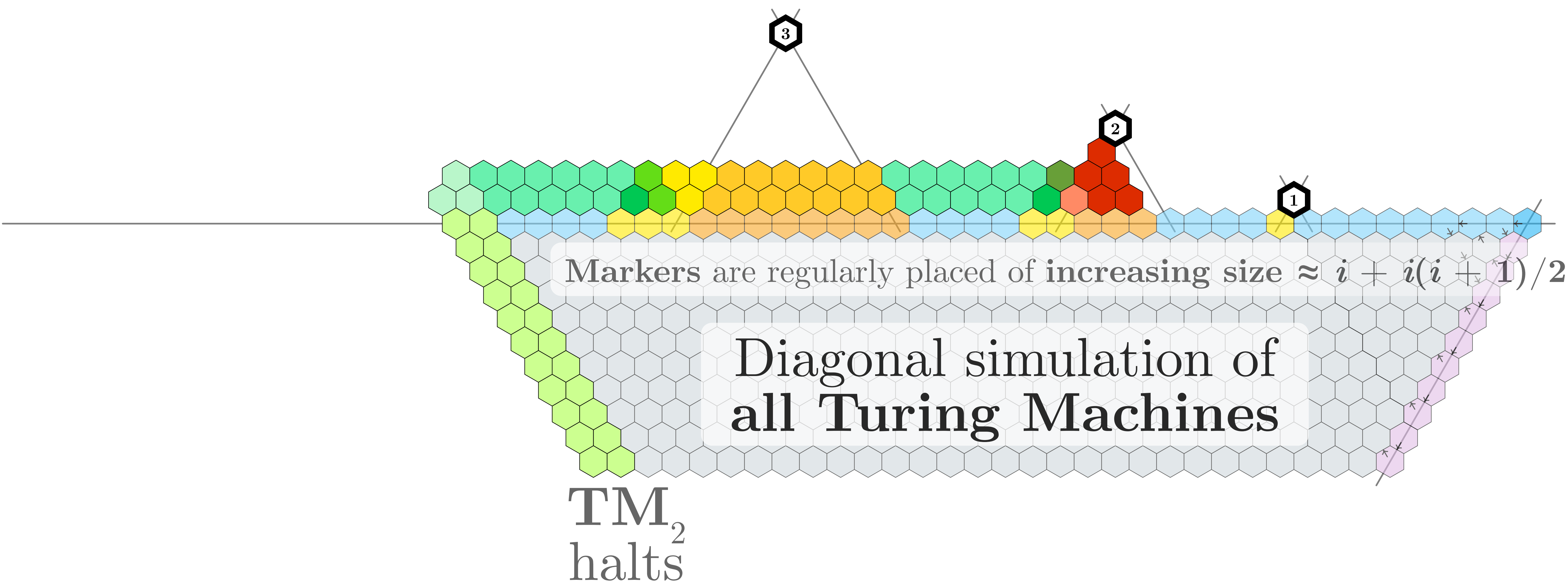
Turmites doodle uncomputably



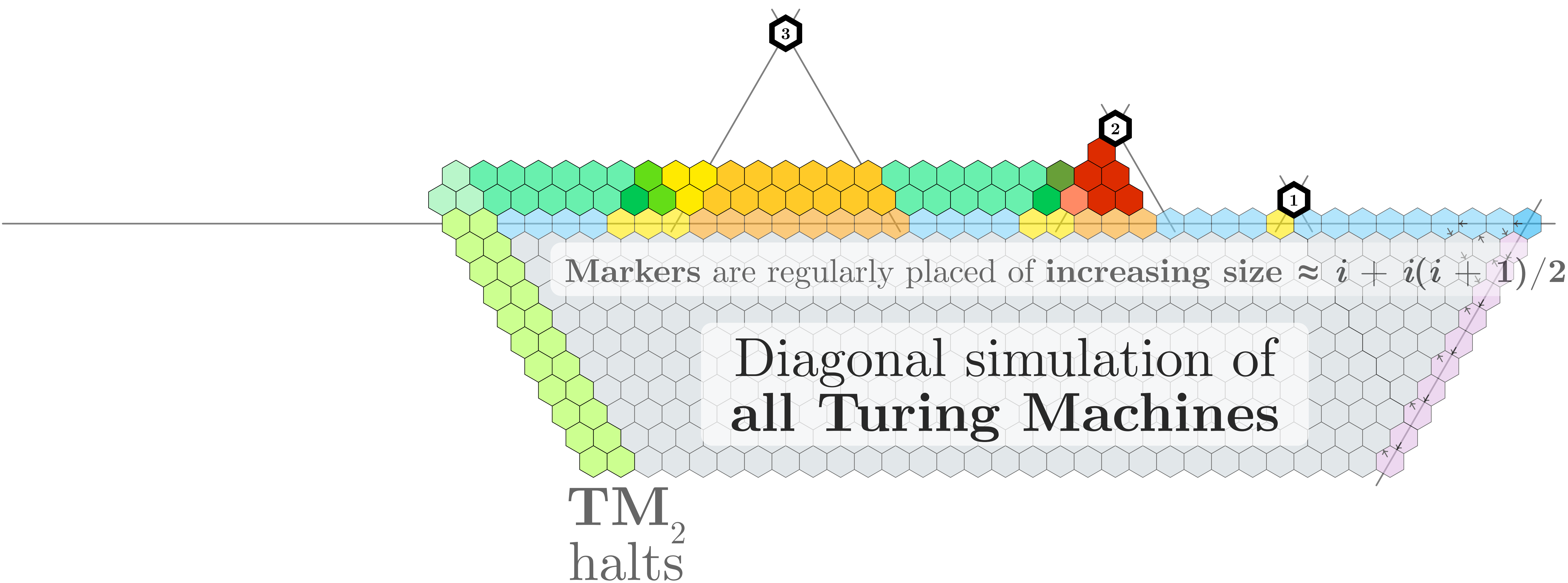
Turmites doodle uncomputably



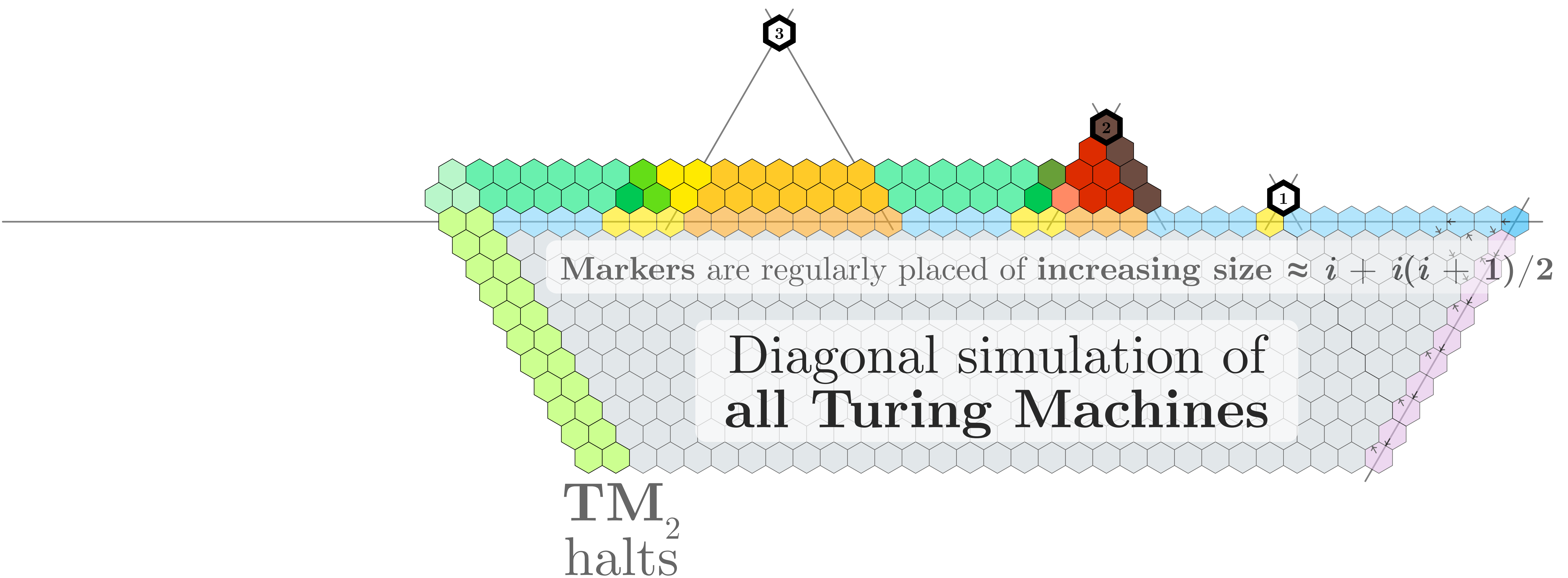
Turmites doodle uncomputably



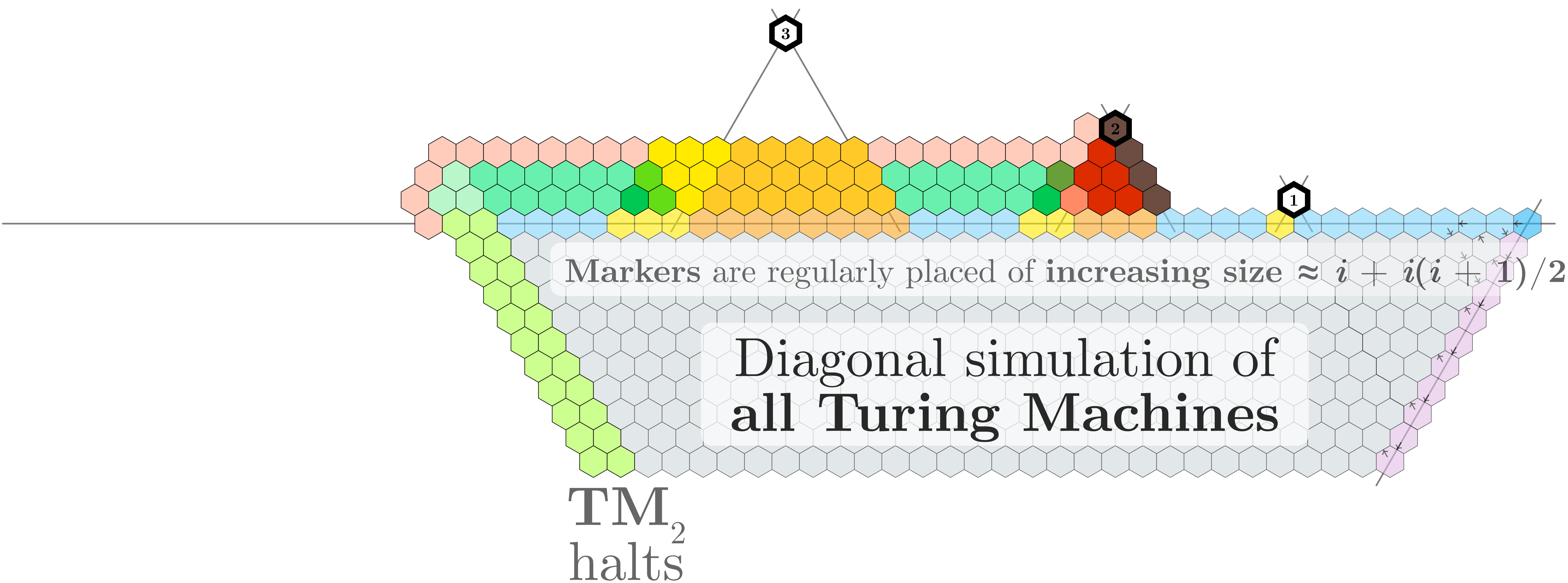
Turmites doodle uncomputably



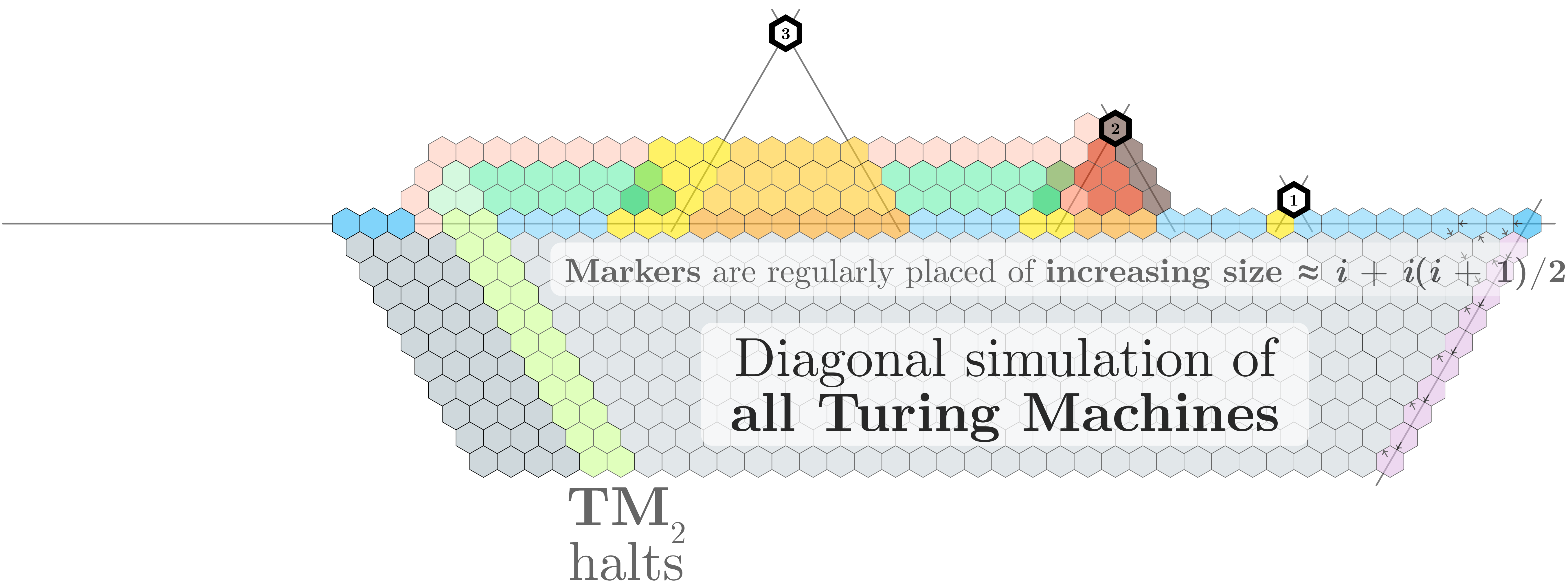
Turmites doodle uncomputably



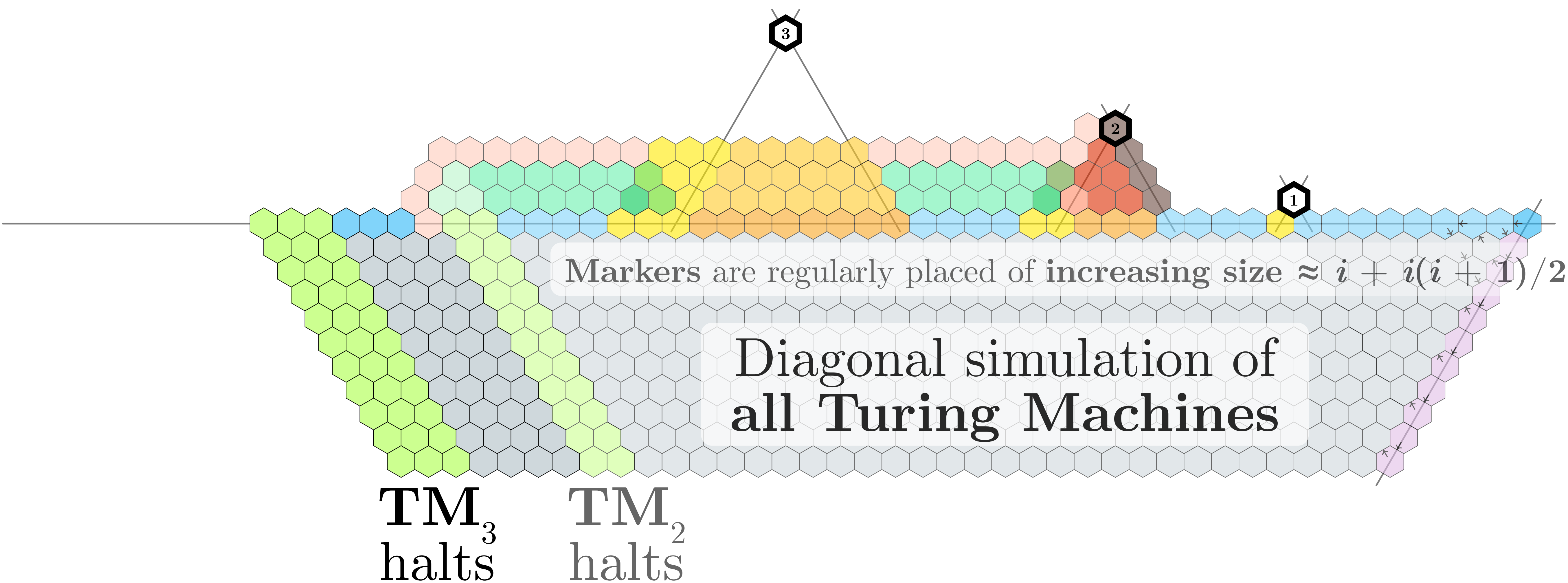
Turmites doodle uncomputably



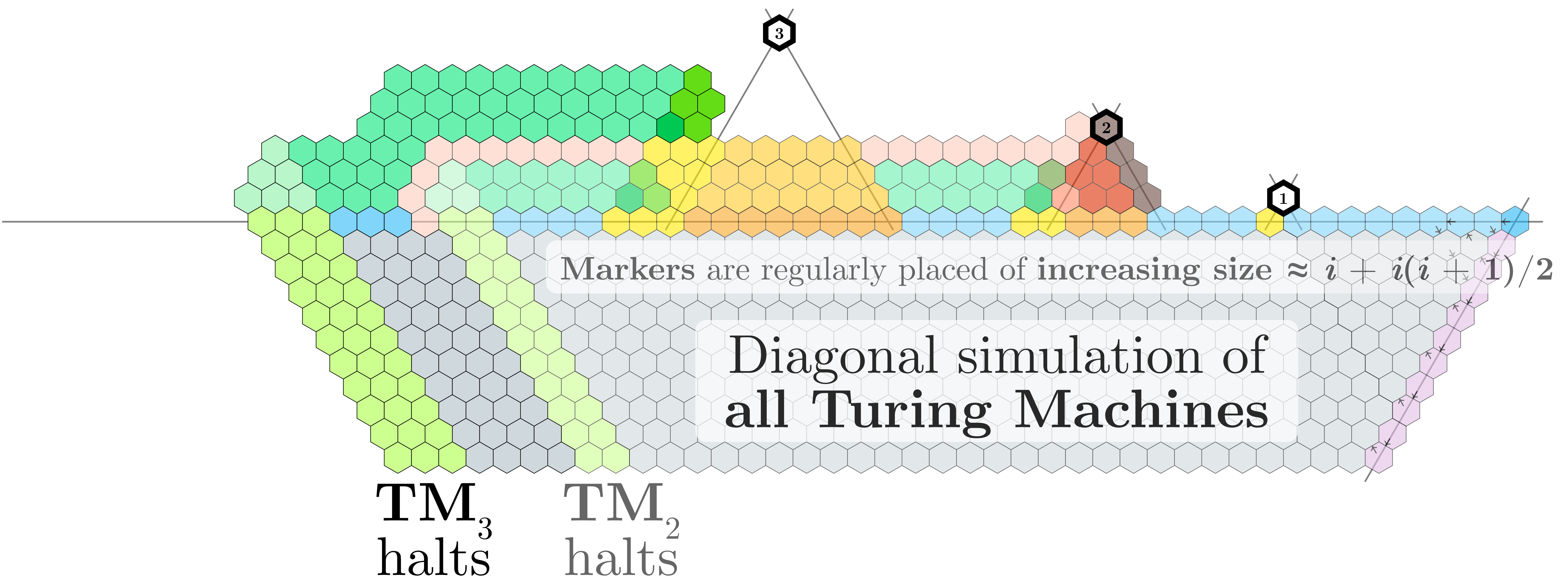
Turmites doodle uncomputably



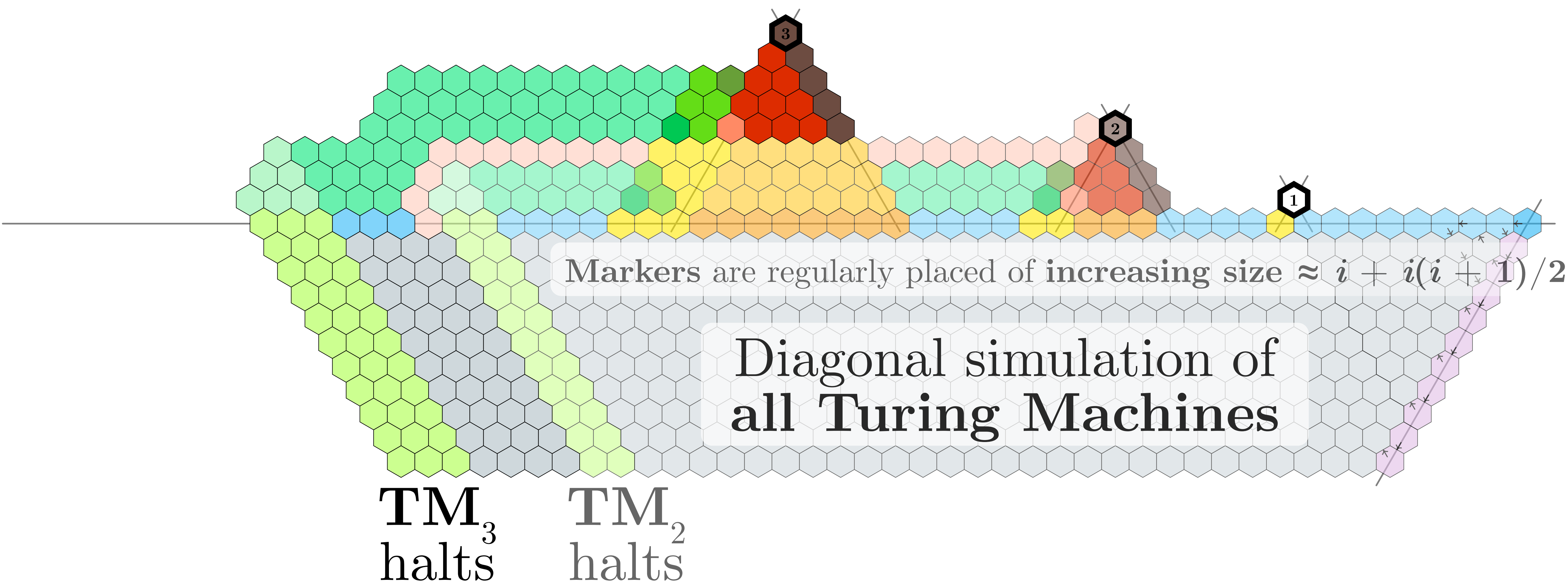
Turmites doodle uncomputably



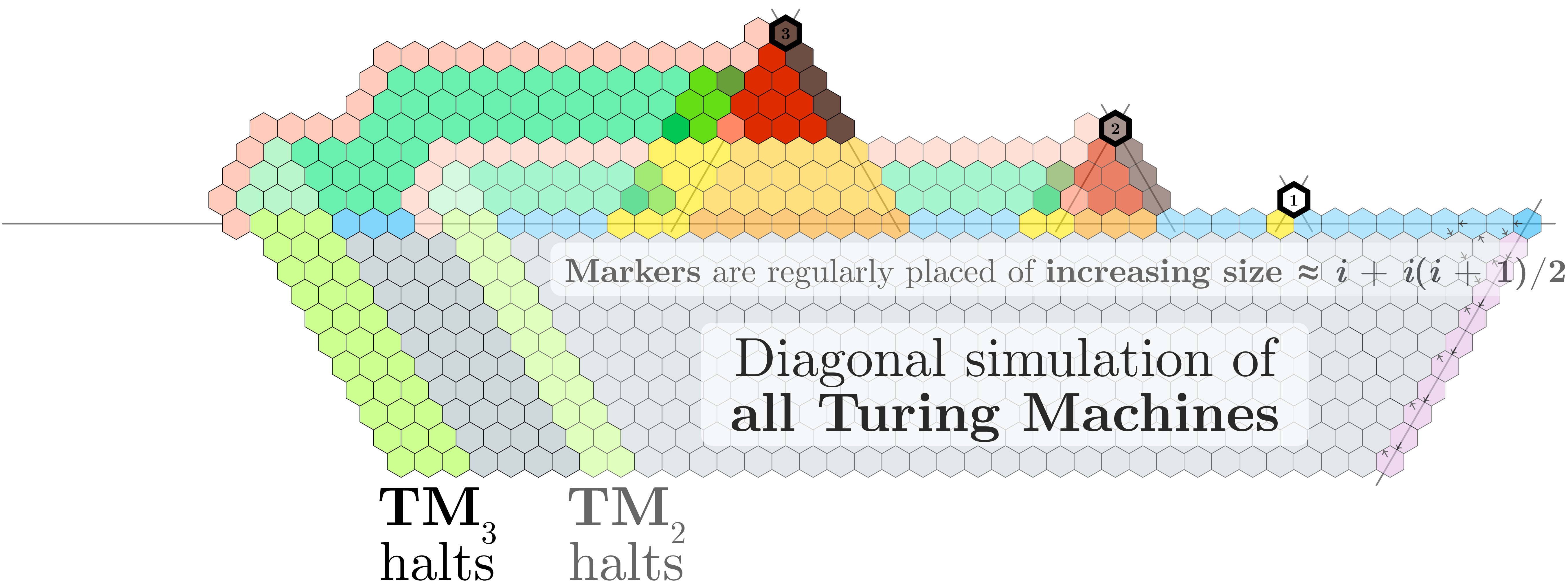
Turmites doodle uncomputably



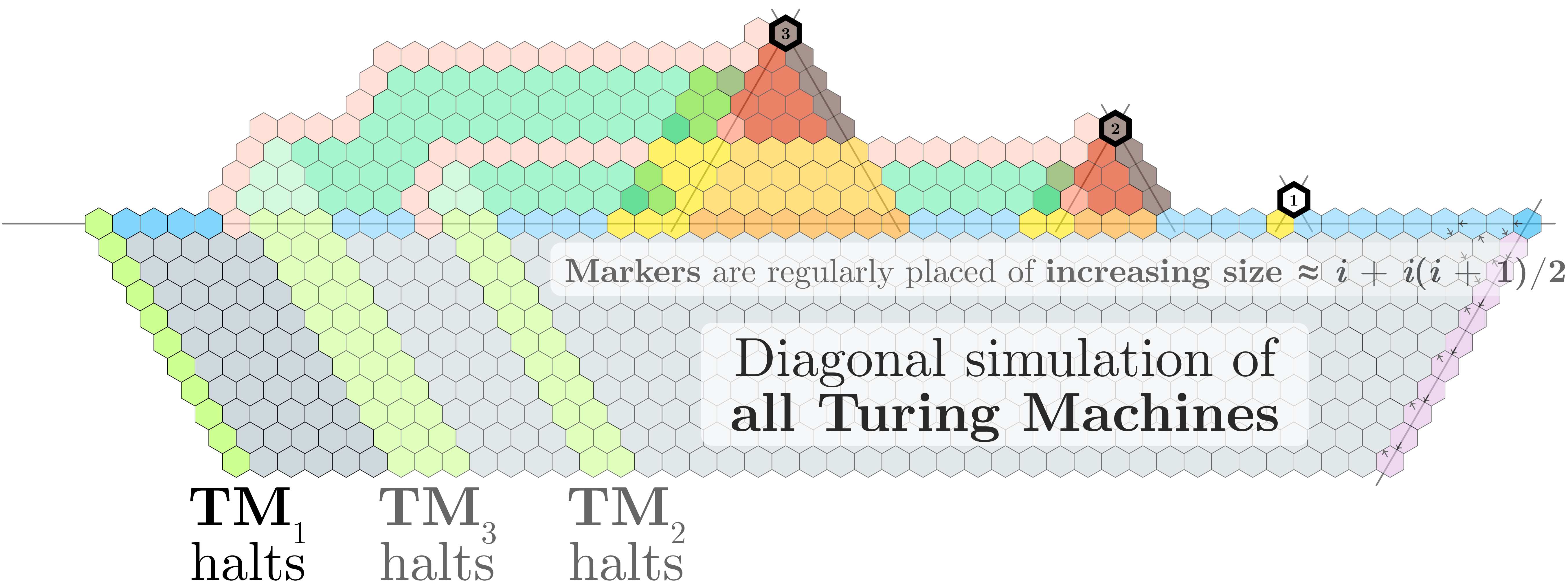
Turmites doodle uncomputably



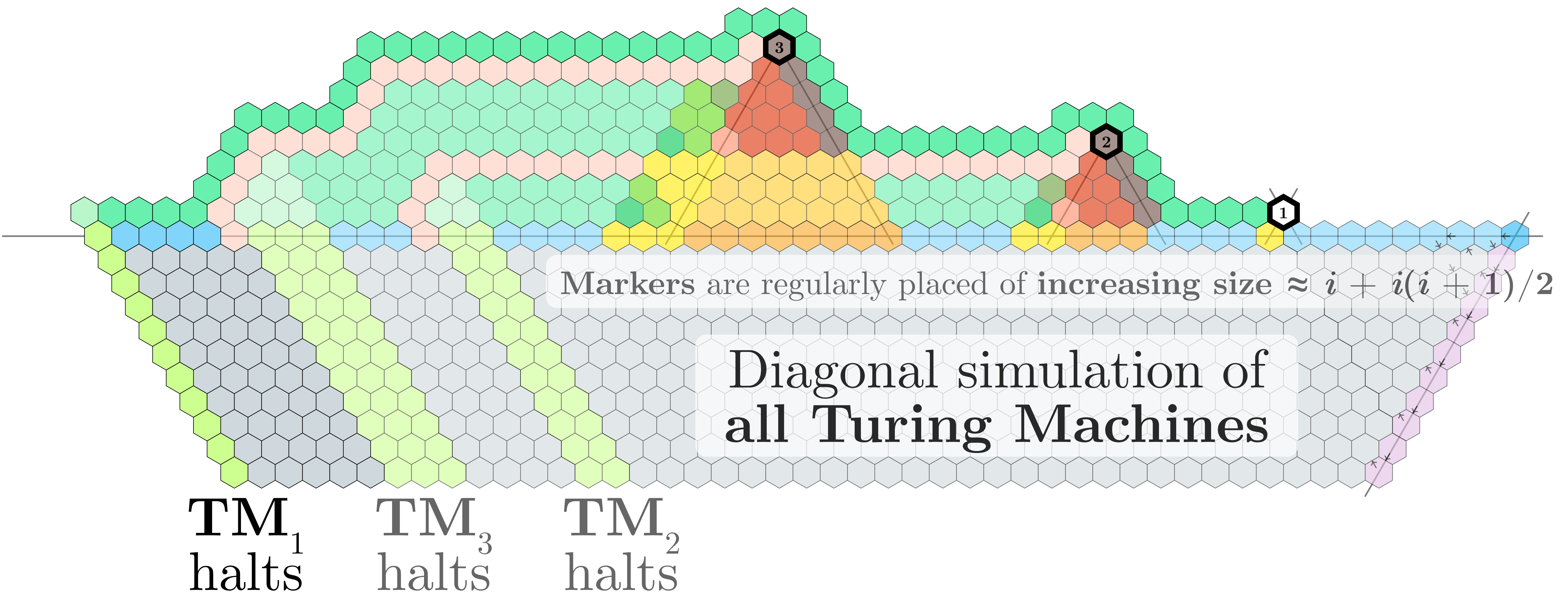
Turmites doodle uncomputably



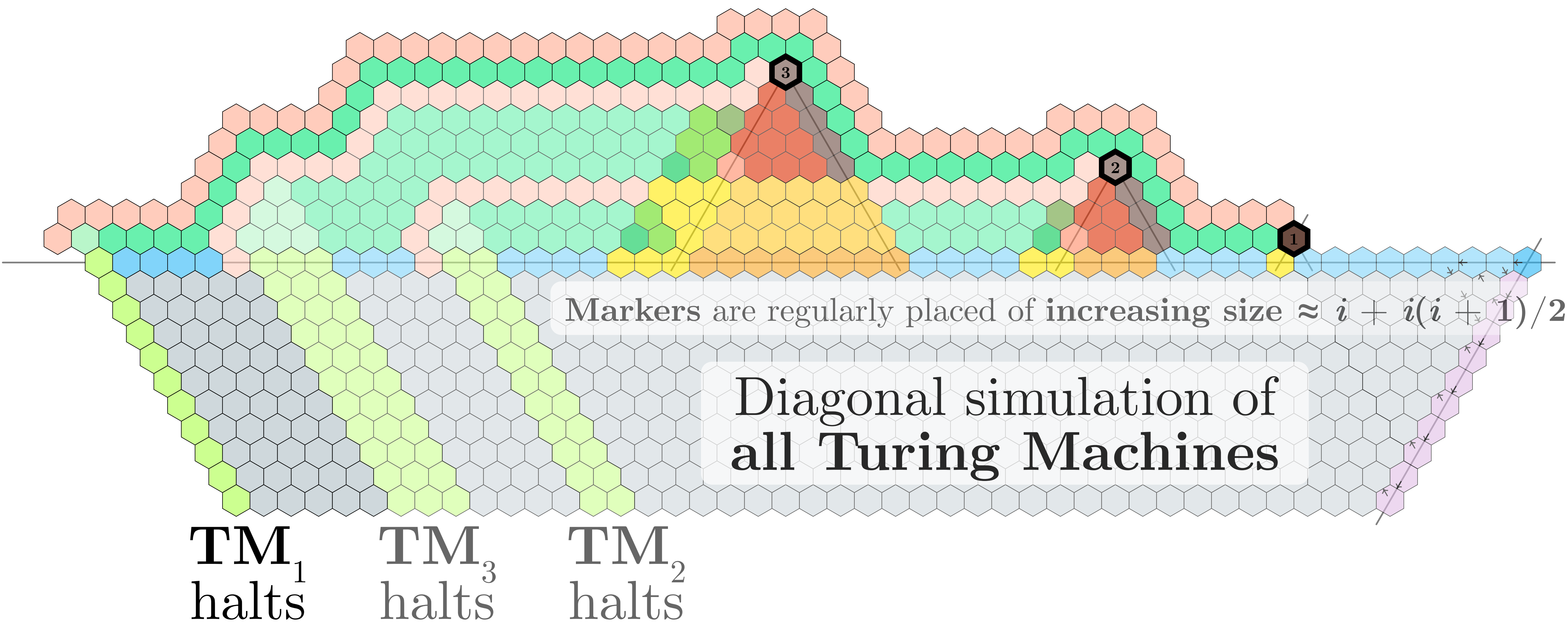
Turmites doodle uncomputably



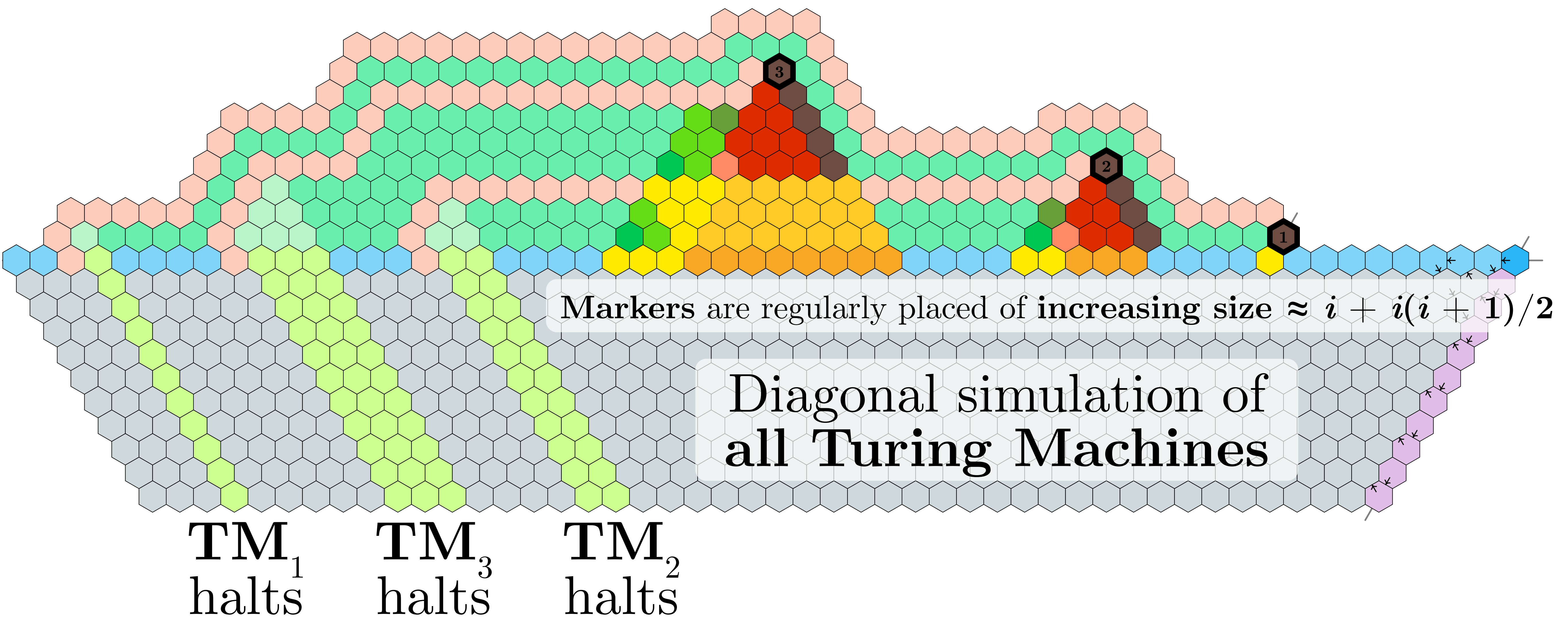
Turmites doodle uncomputably



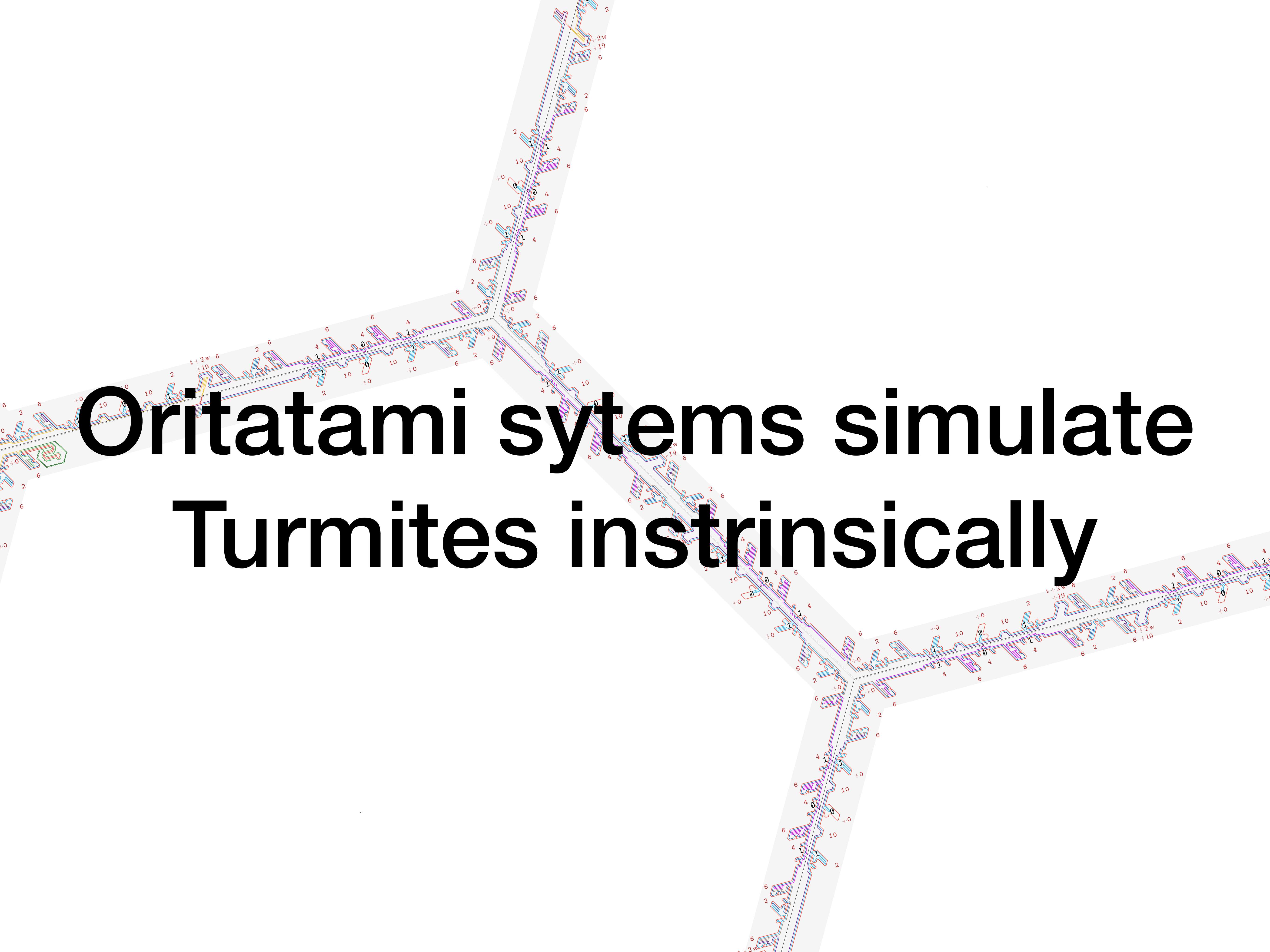
Turmites doodle uncomputably



Théorème. **Turmites doodle uncomputably**

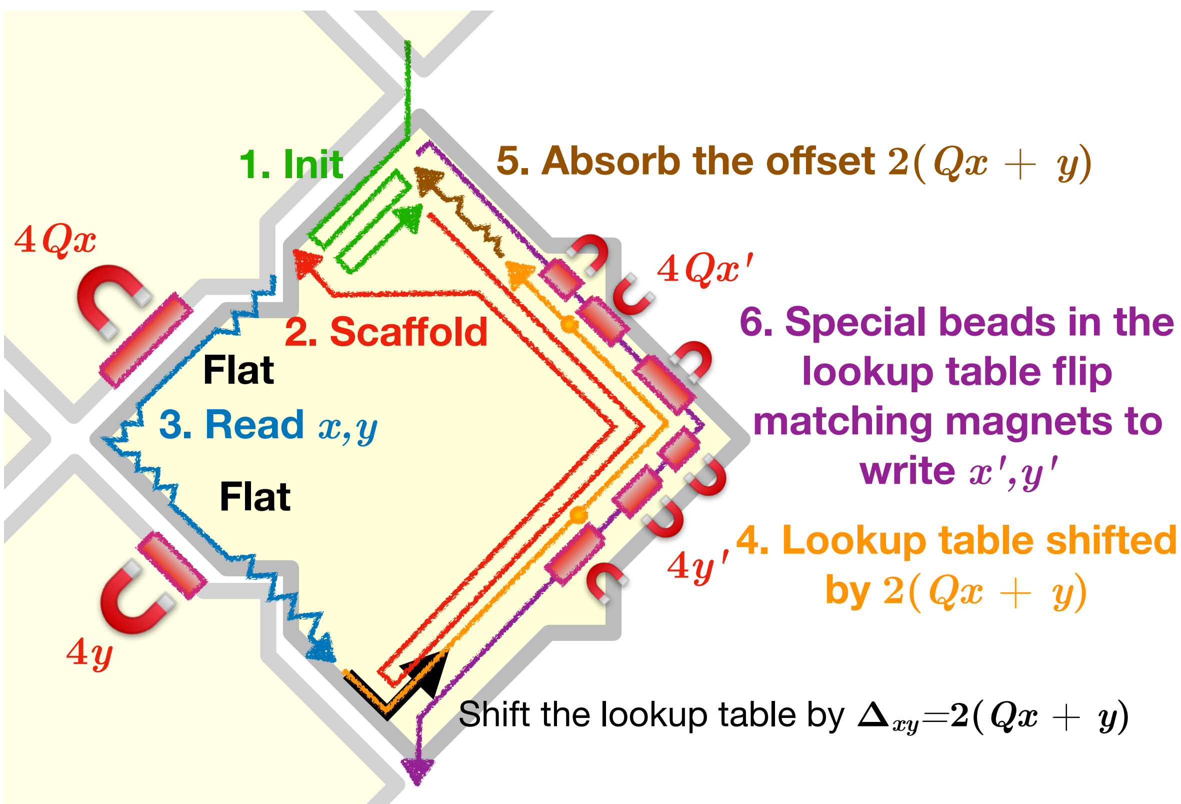


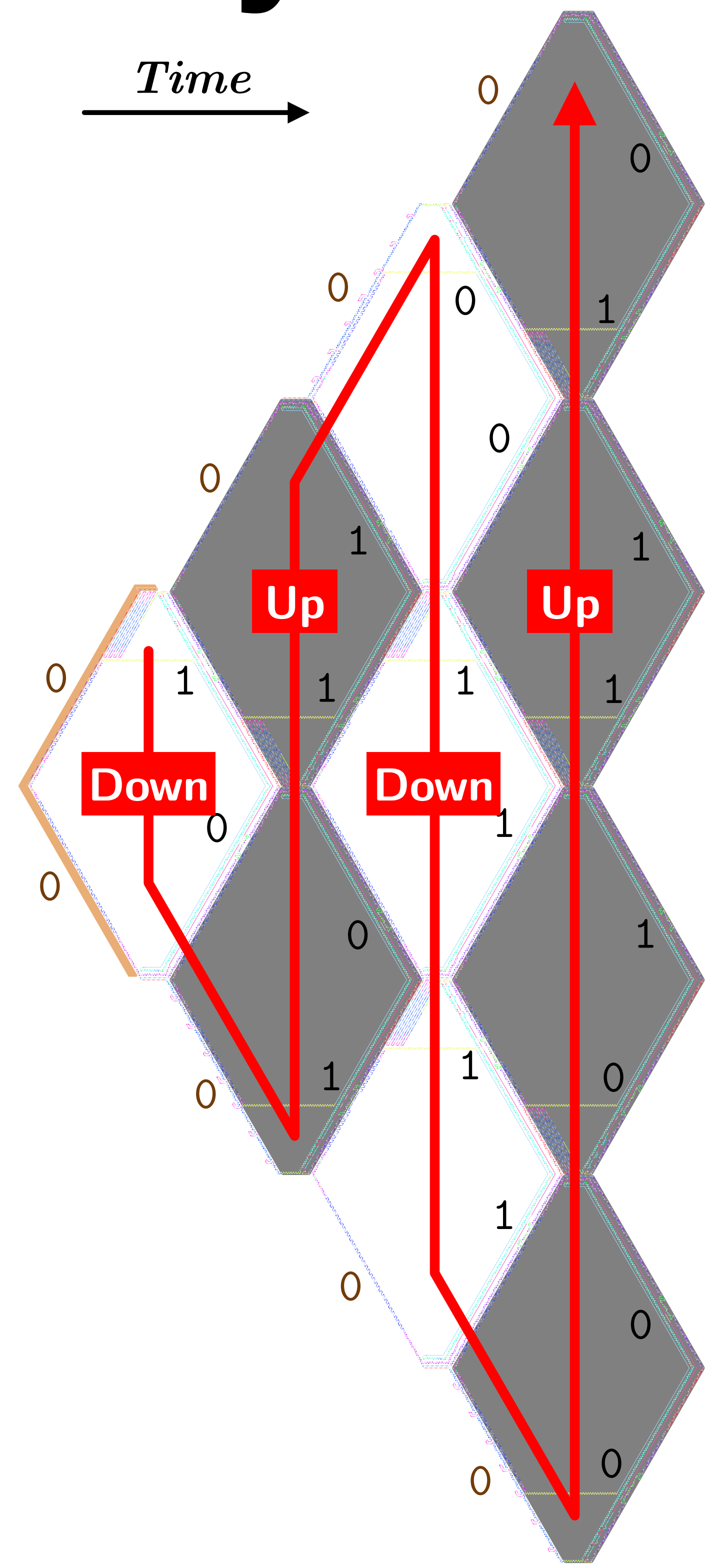
Oritatami sytems simulate Turmites instrinsically



Oritatami sytems simulate 1D CA instrinsically

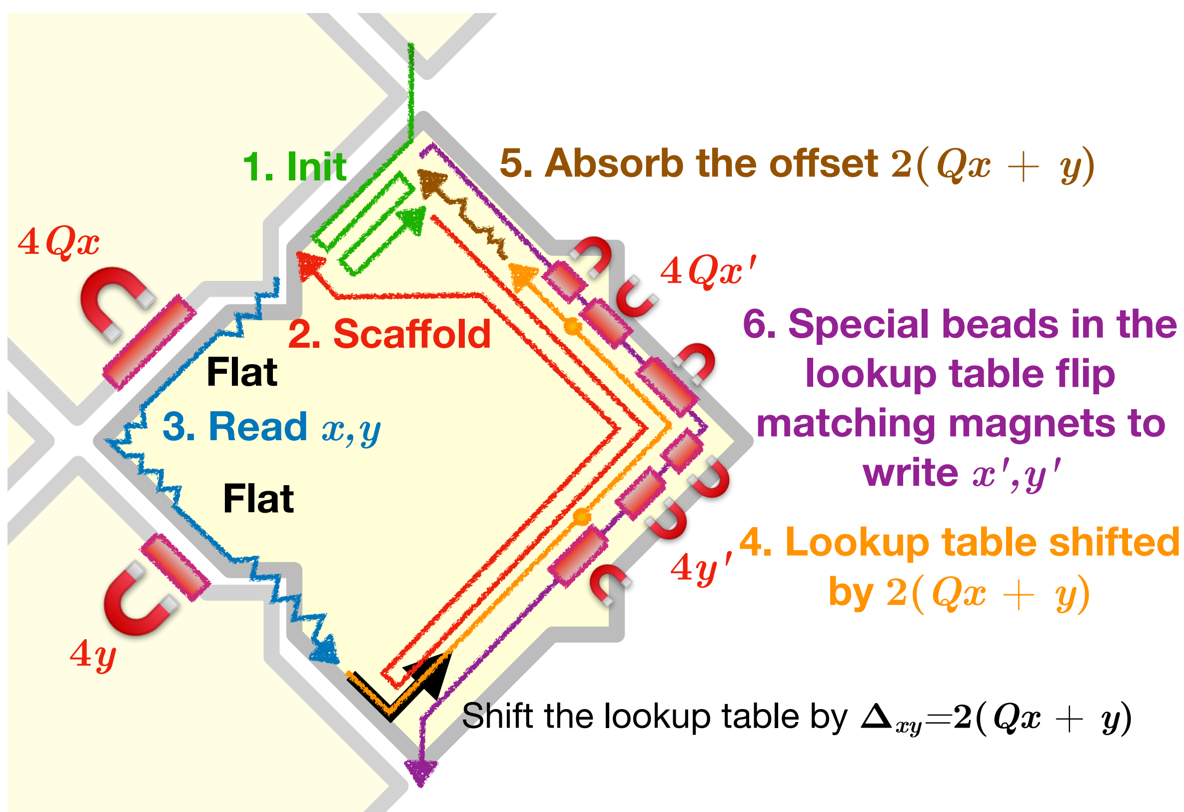
- **Previous work.** [PSSU, 2020]
  1D Cellular automata intrinsic simulation

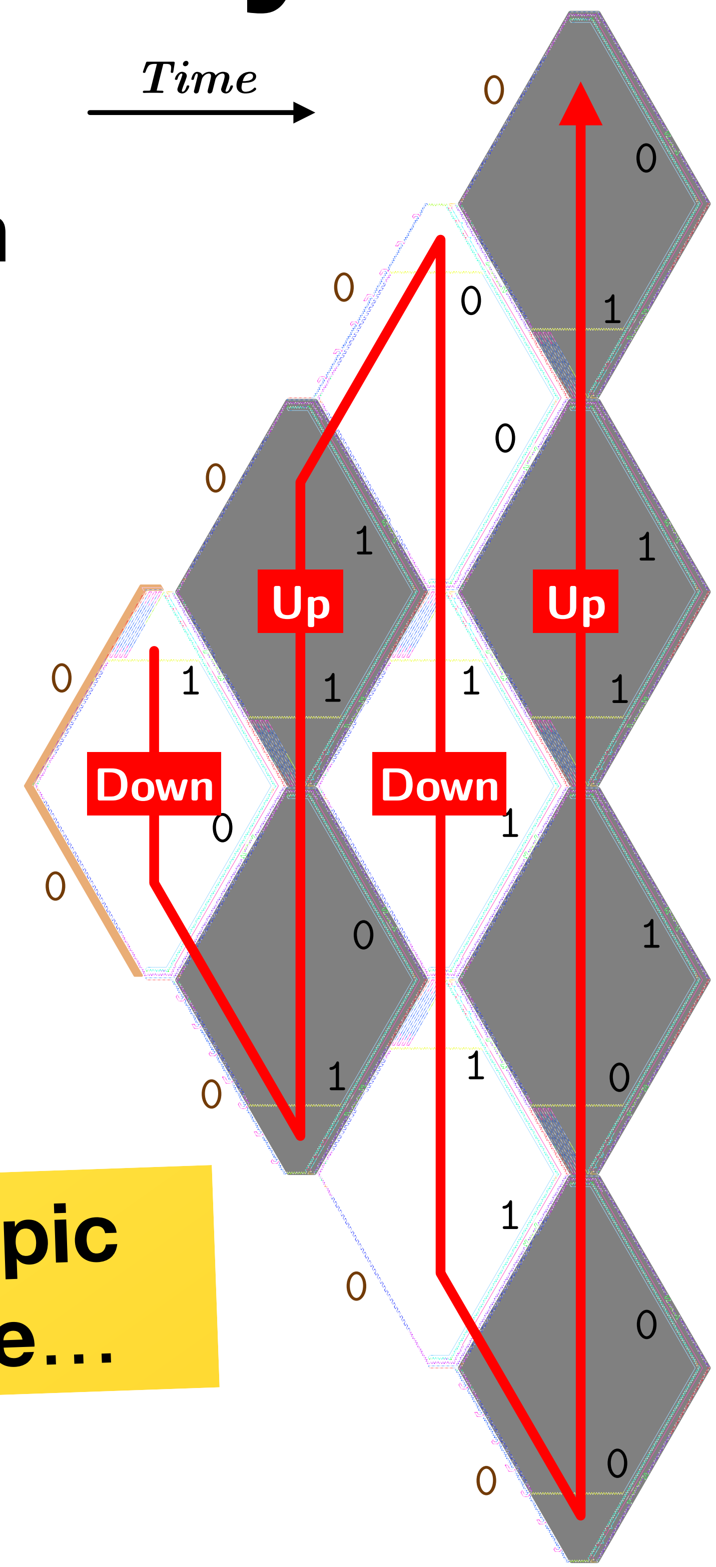




Oritatami sytems simulate 1D CA instrinsically

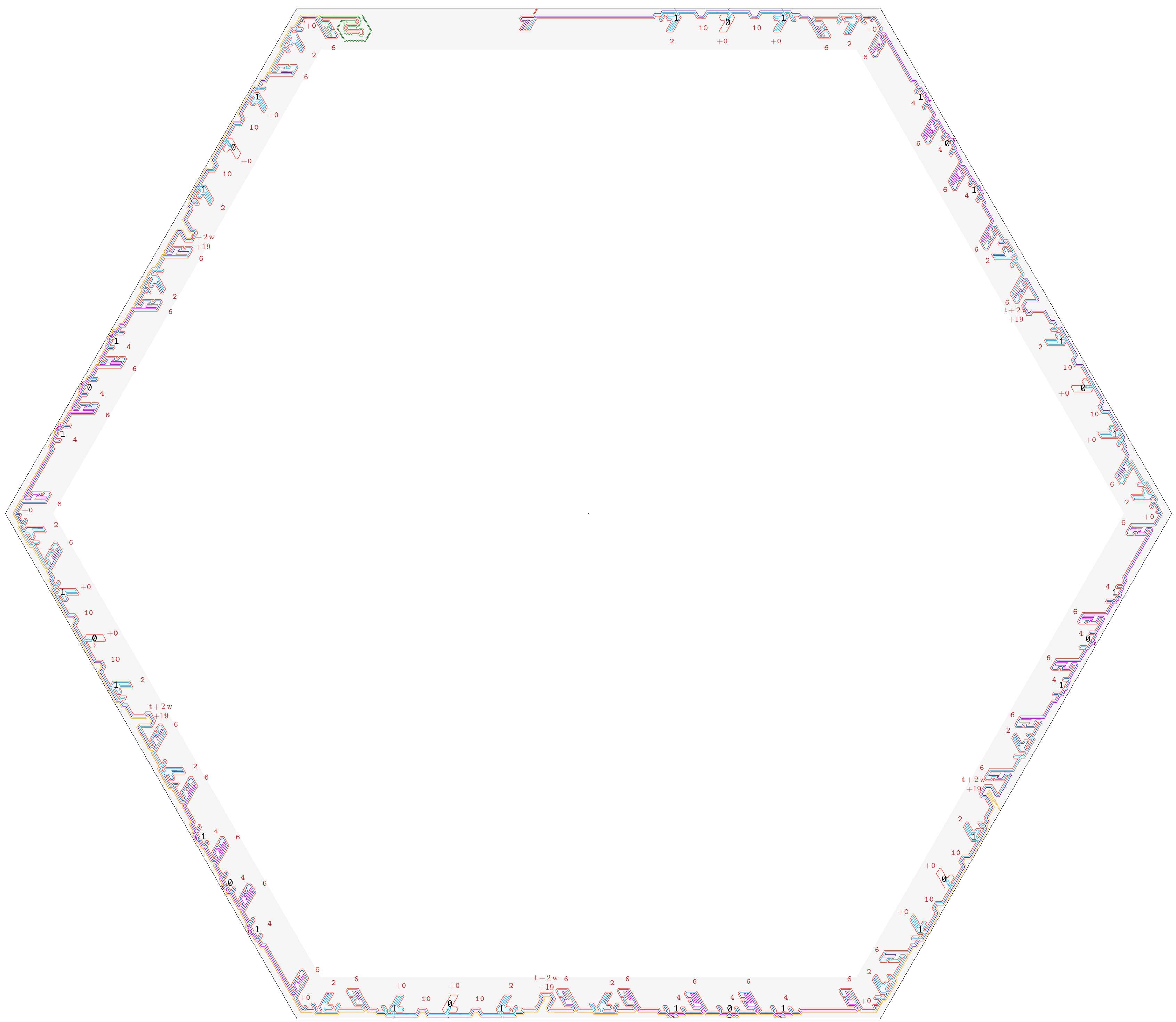
- **Previous work.** [PSSU, 2020]
  1D Cellular automata intrinsic simulation



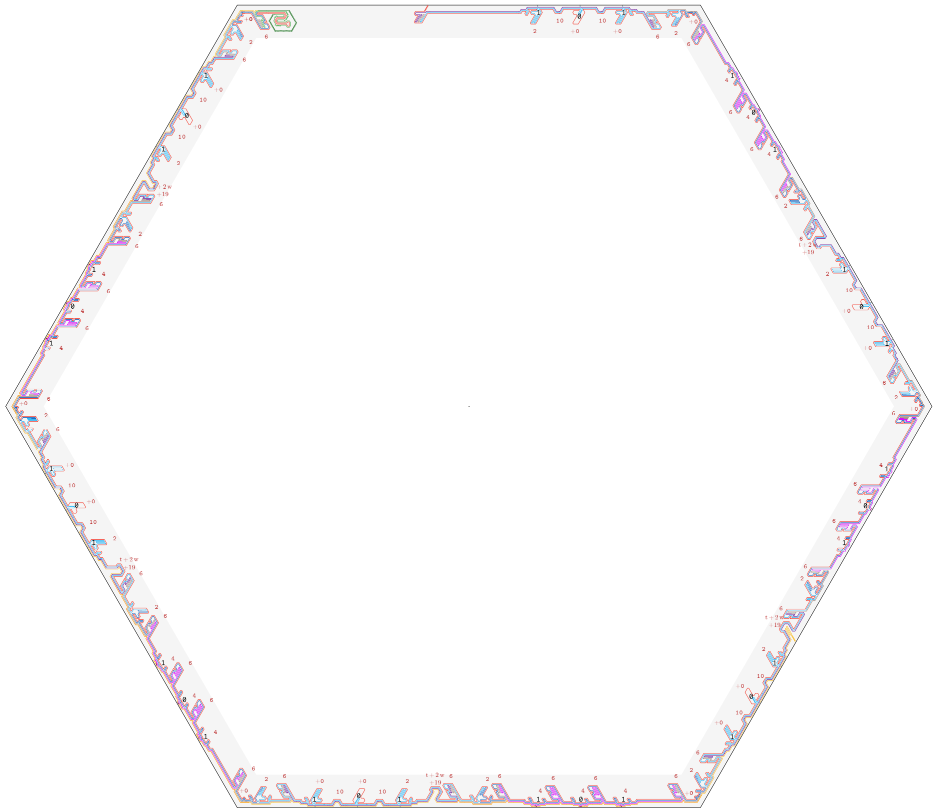


2 Problems. Supercells must be **isotropic**
We need to **exit from an arbitrary side…**

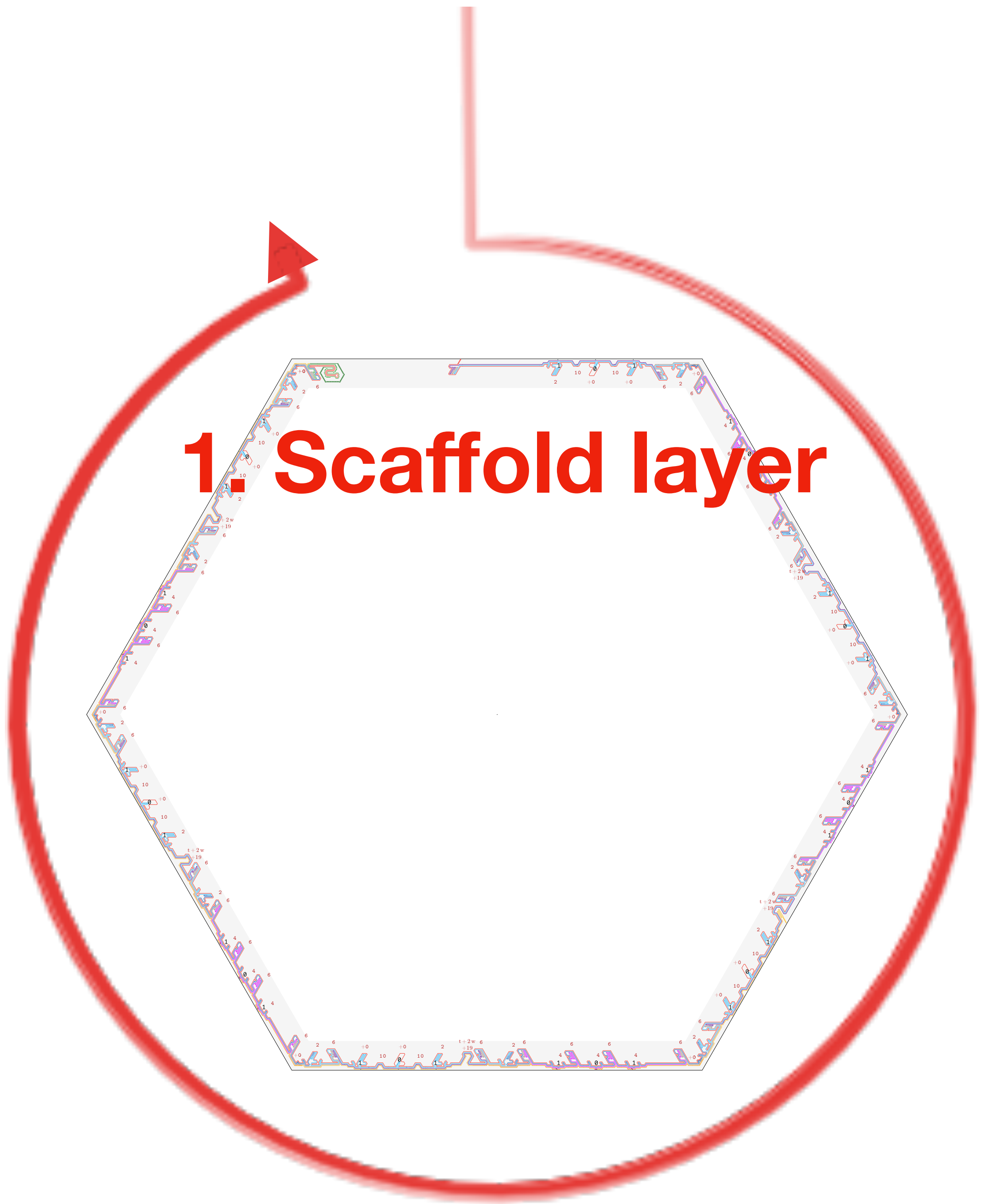
the Supercell



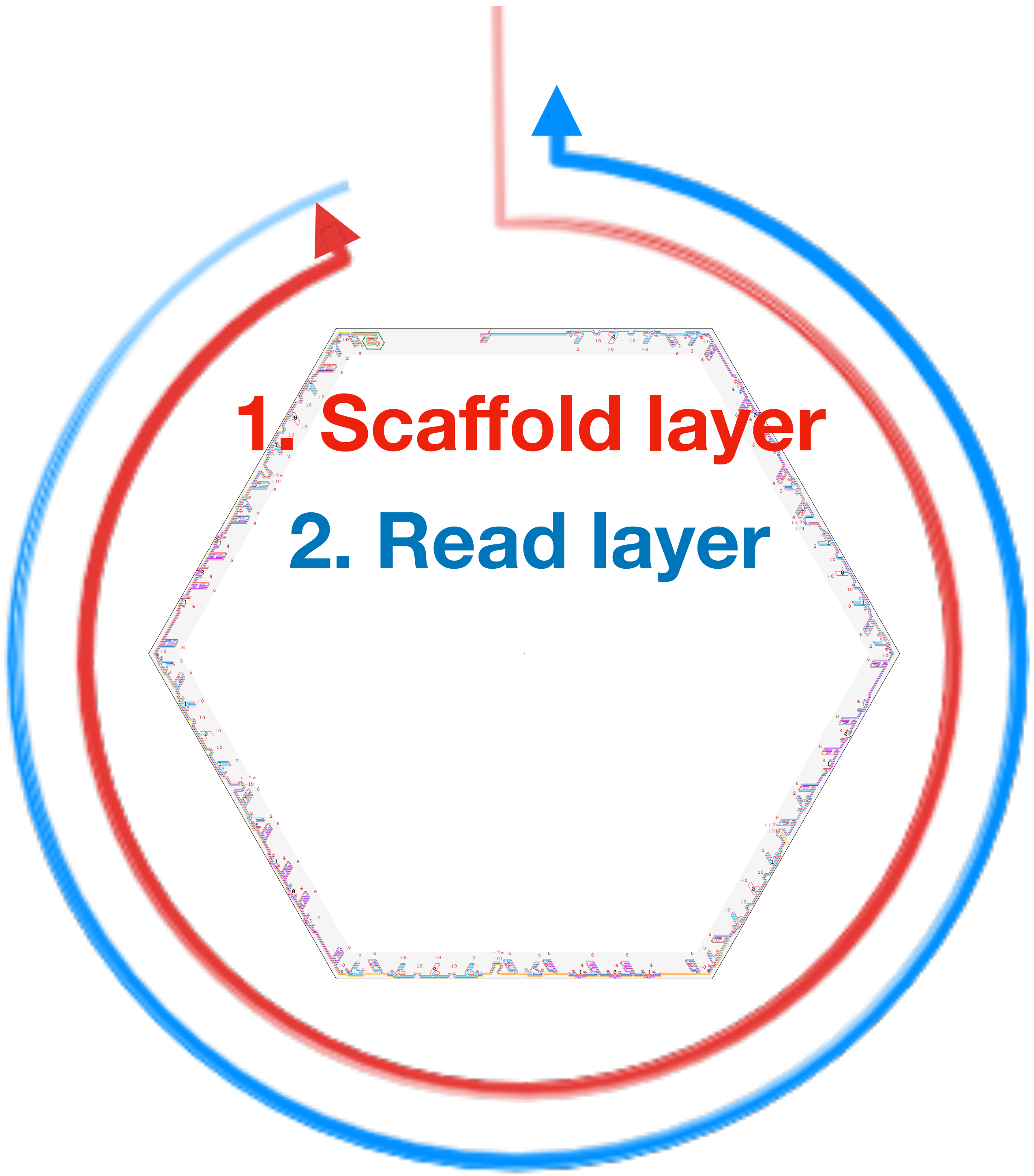
the Supercell



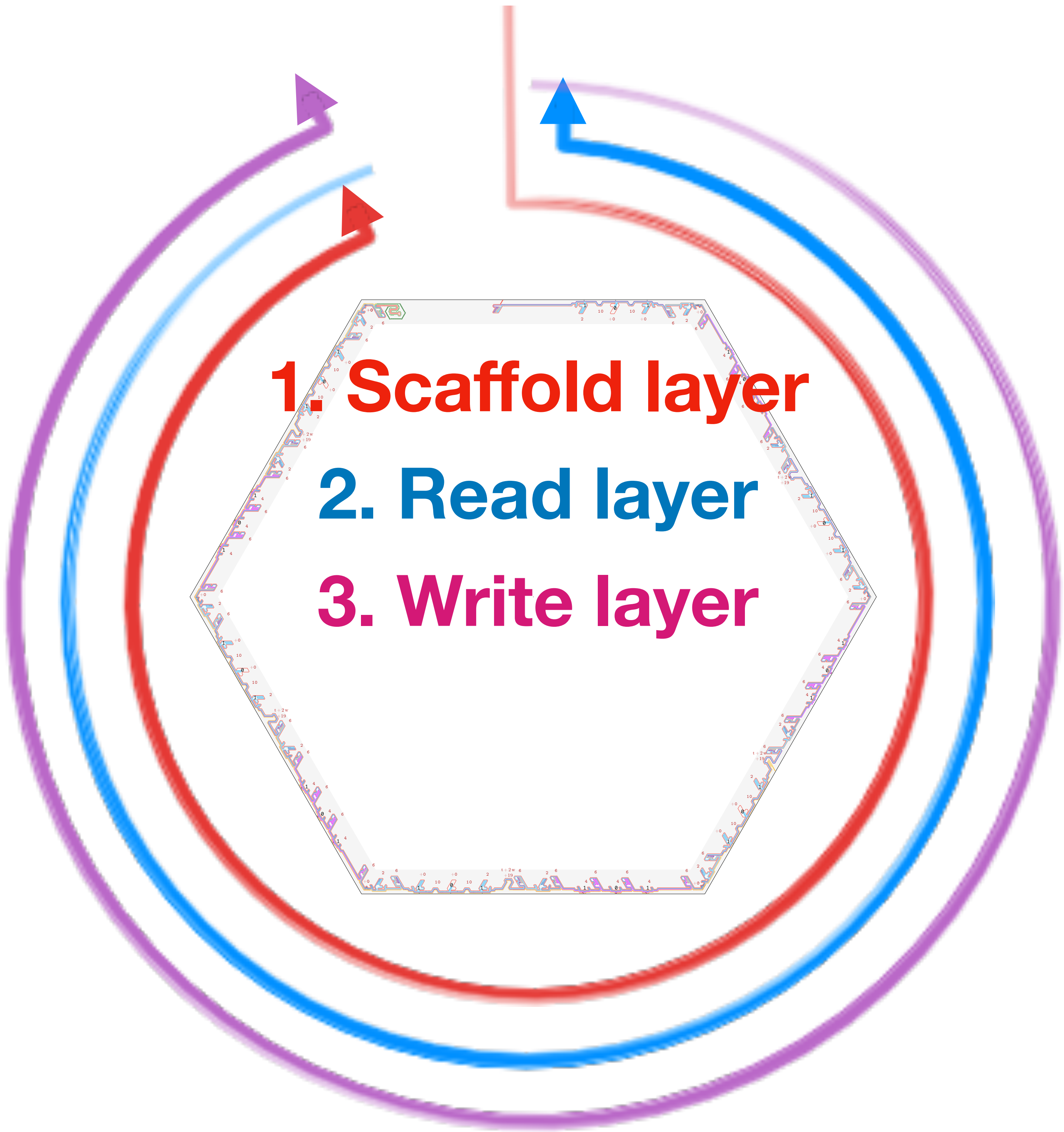
the Supercell



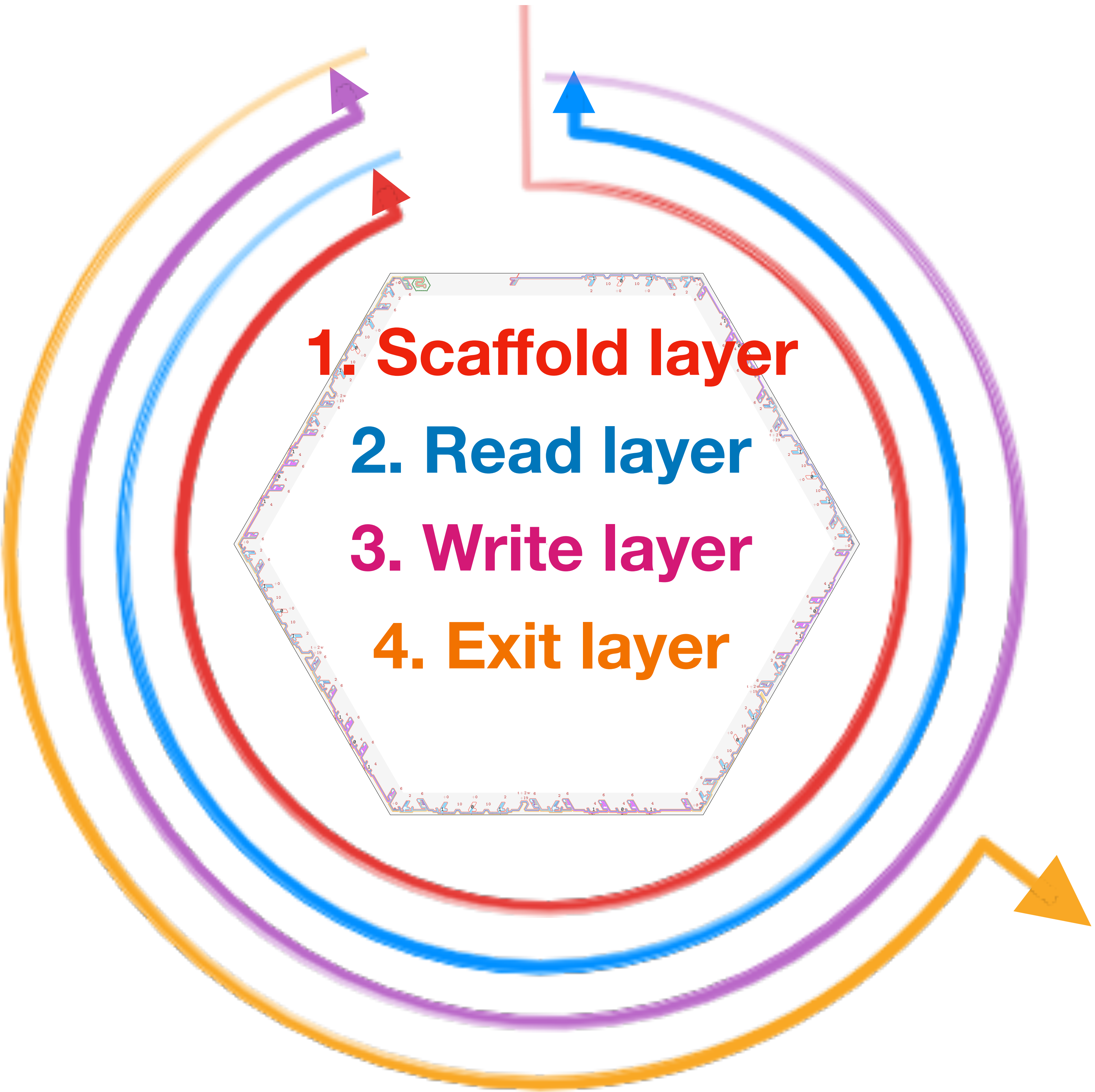
the Supercell



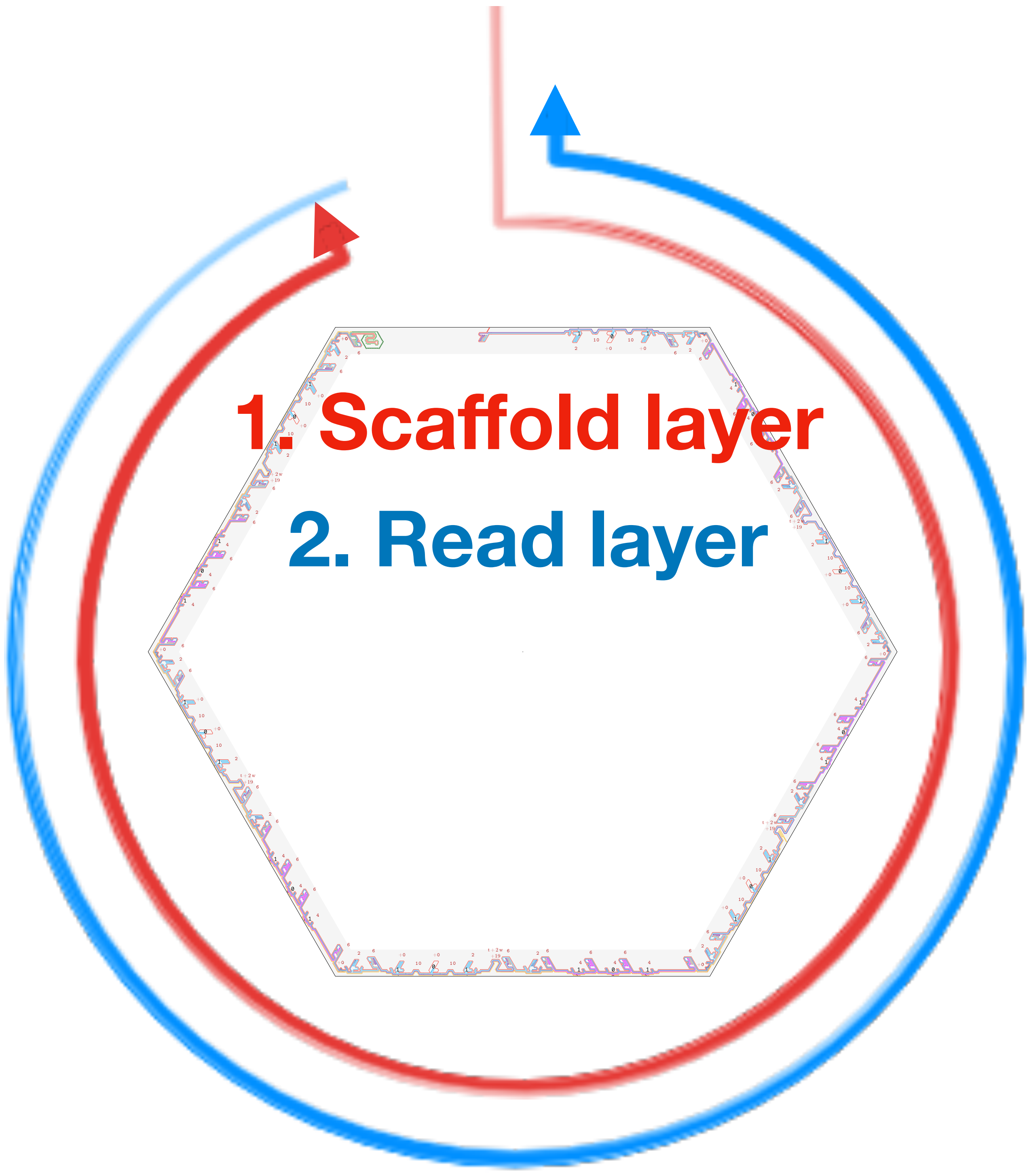
the Supercell



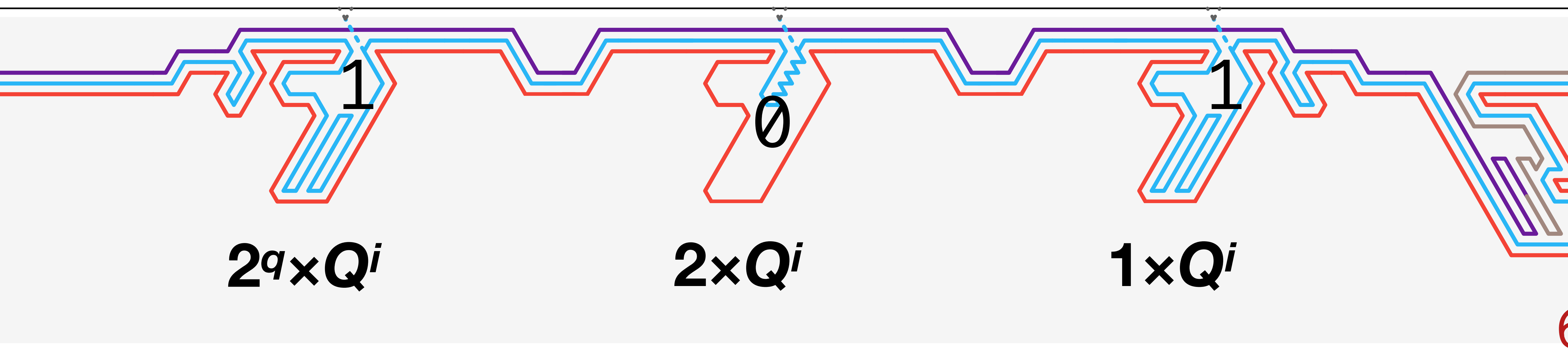
the Supercell



the Supercell



Reading. Reading boxes

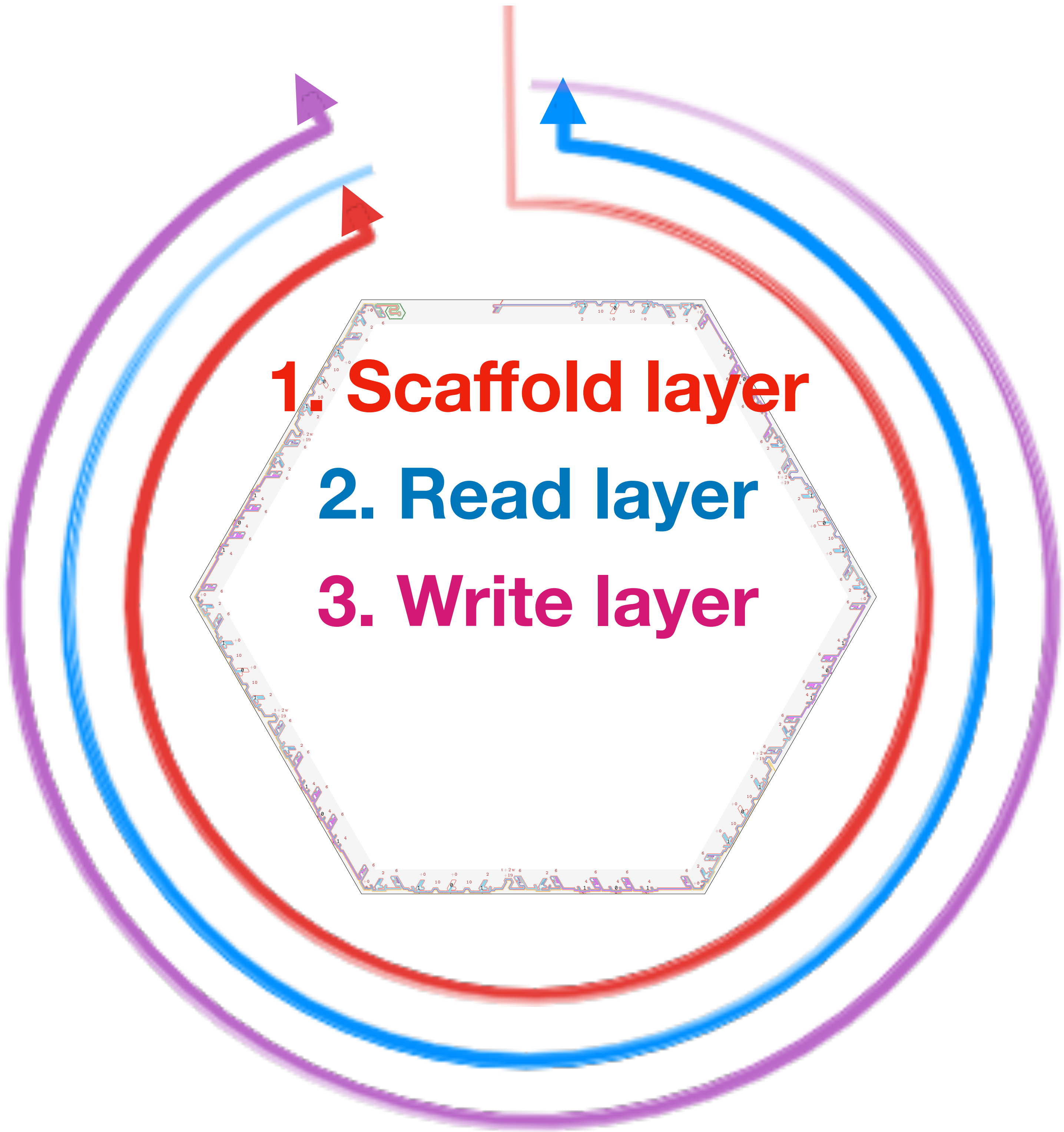


If j–th bit = 1, then Offset += $2^j Q^i$

⇒ Offset on i–th side = state(i)×Q^i

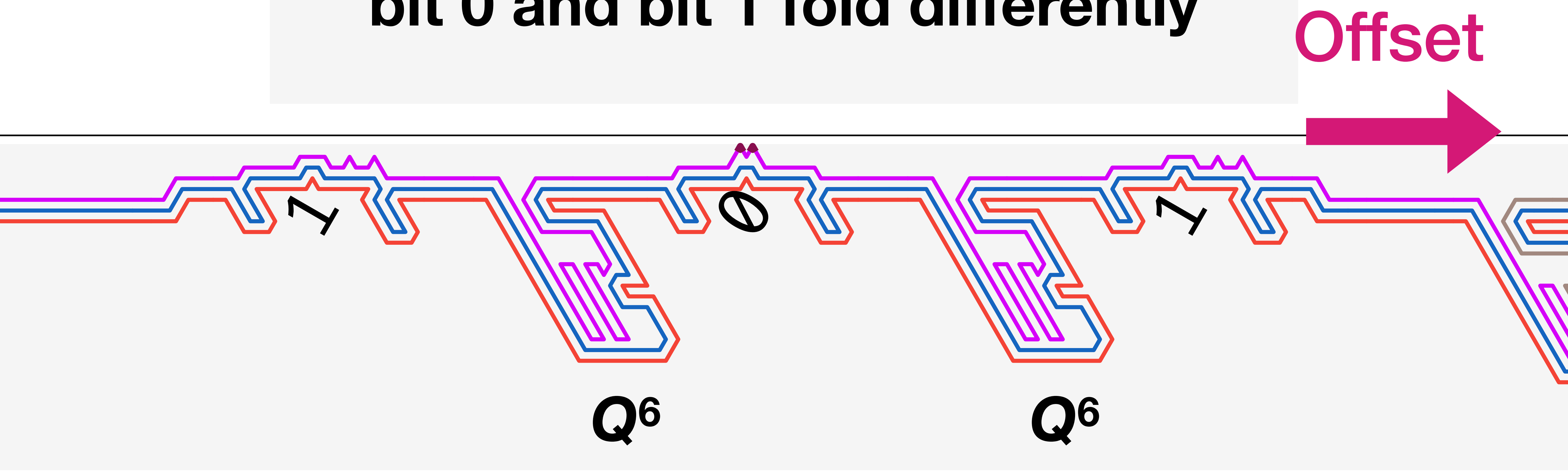
⇒ Total Offset on all side = <states> ∈ 0..Q^6–1

the Supercell



Writing. Offset pulls the transition table to the right

bit 0 and bit 1 fold differently

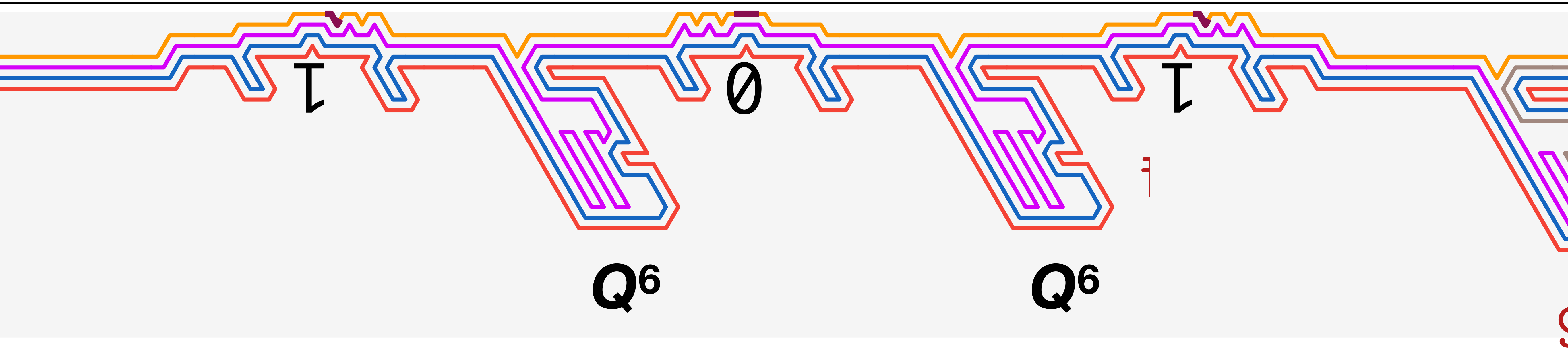


The boxes hide the Q^6 unused entries in the transition table

Writing. Offset pulls the transition table to the right

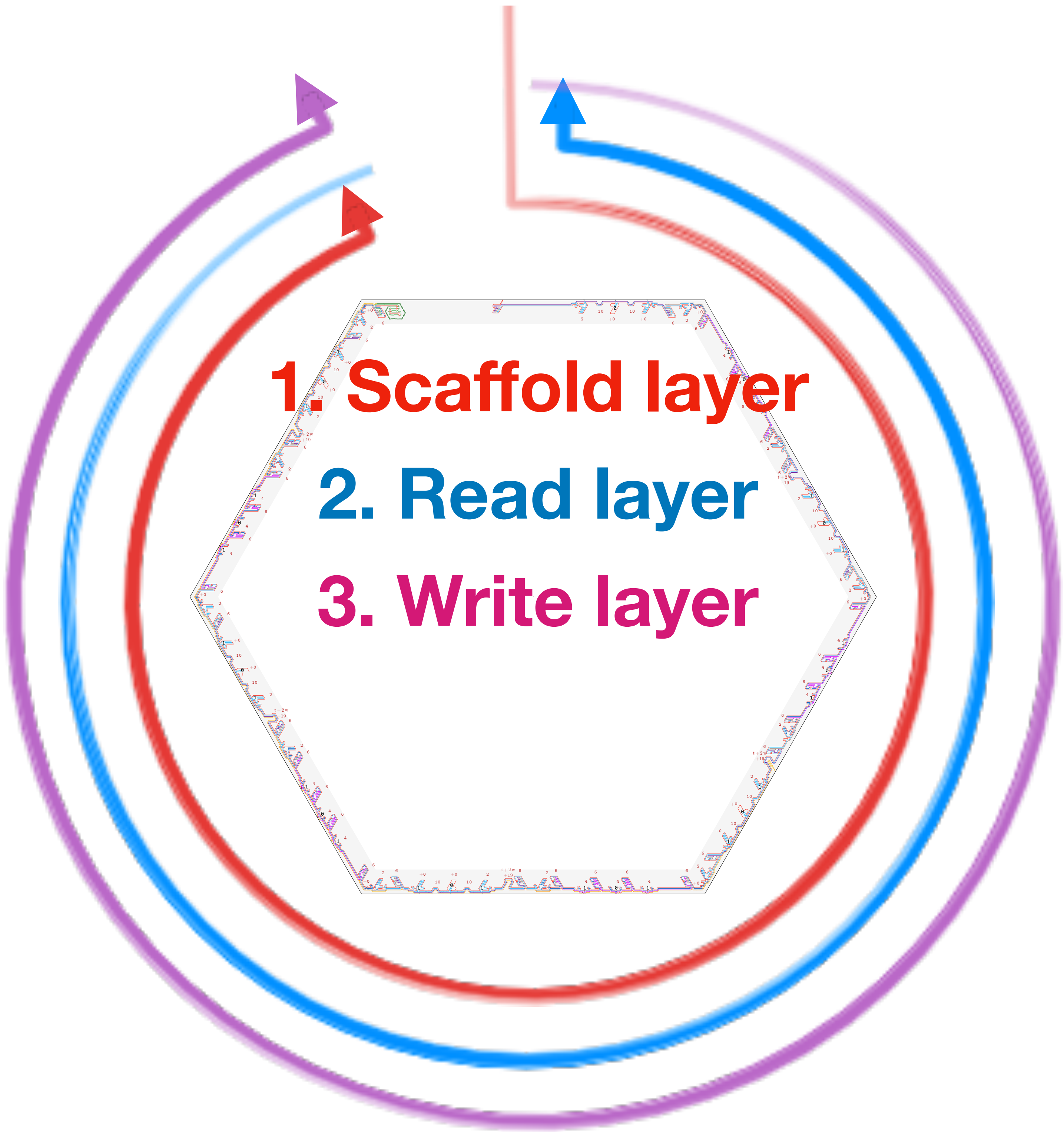
bit 0 and bit 1 fold differently

⇒ the exit layer shows or hide the special beads

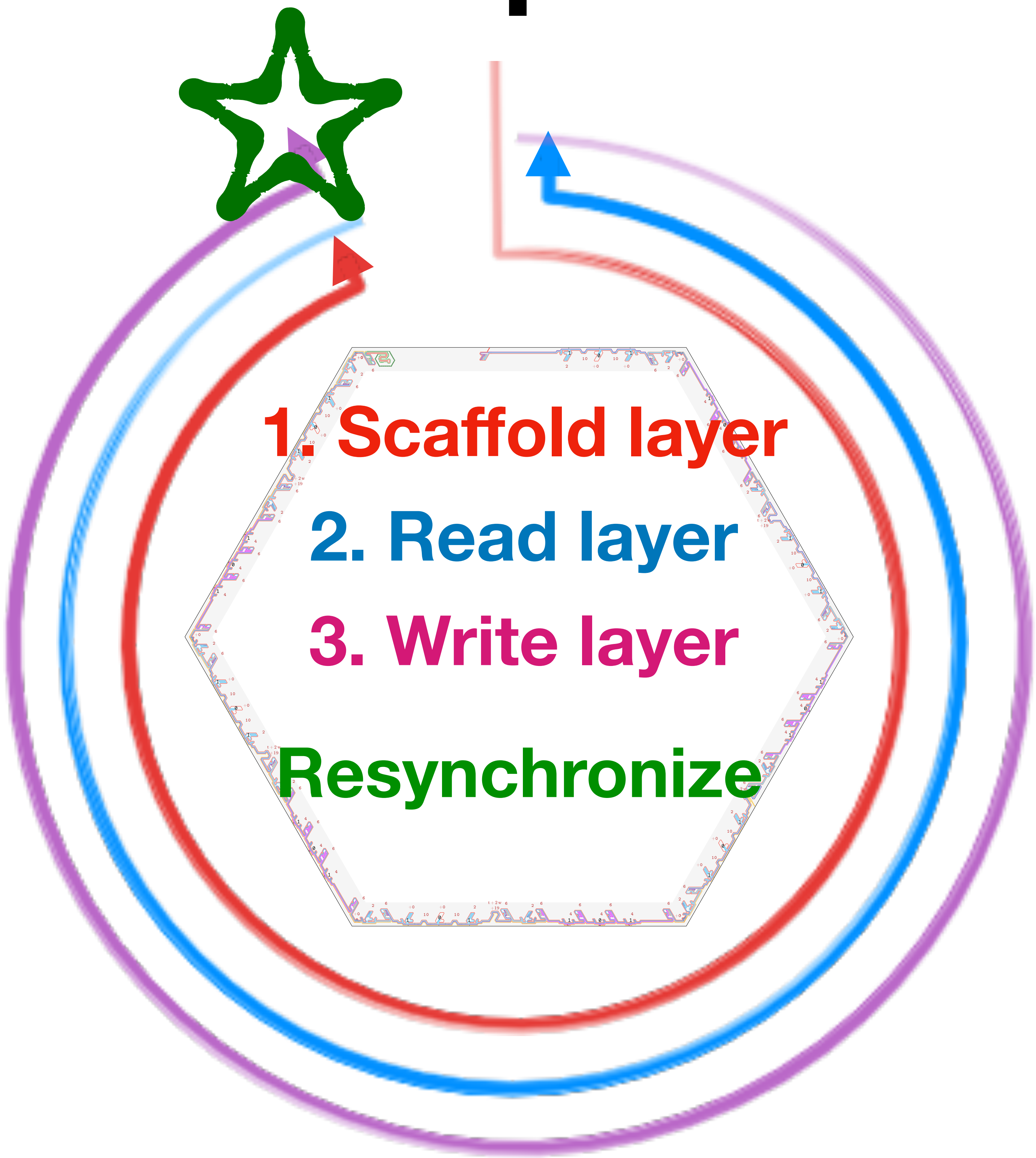


The boxes hide the Q^6 unused entries in the transition table

the Supercell

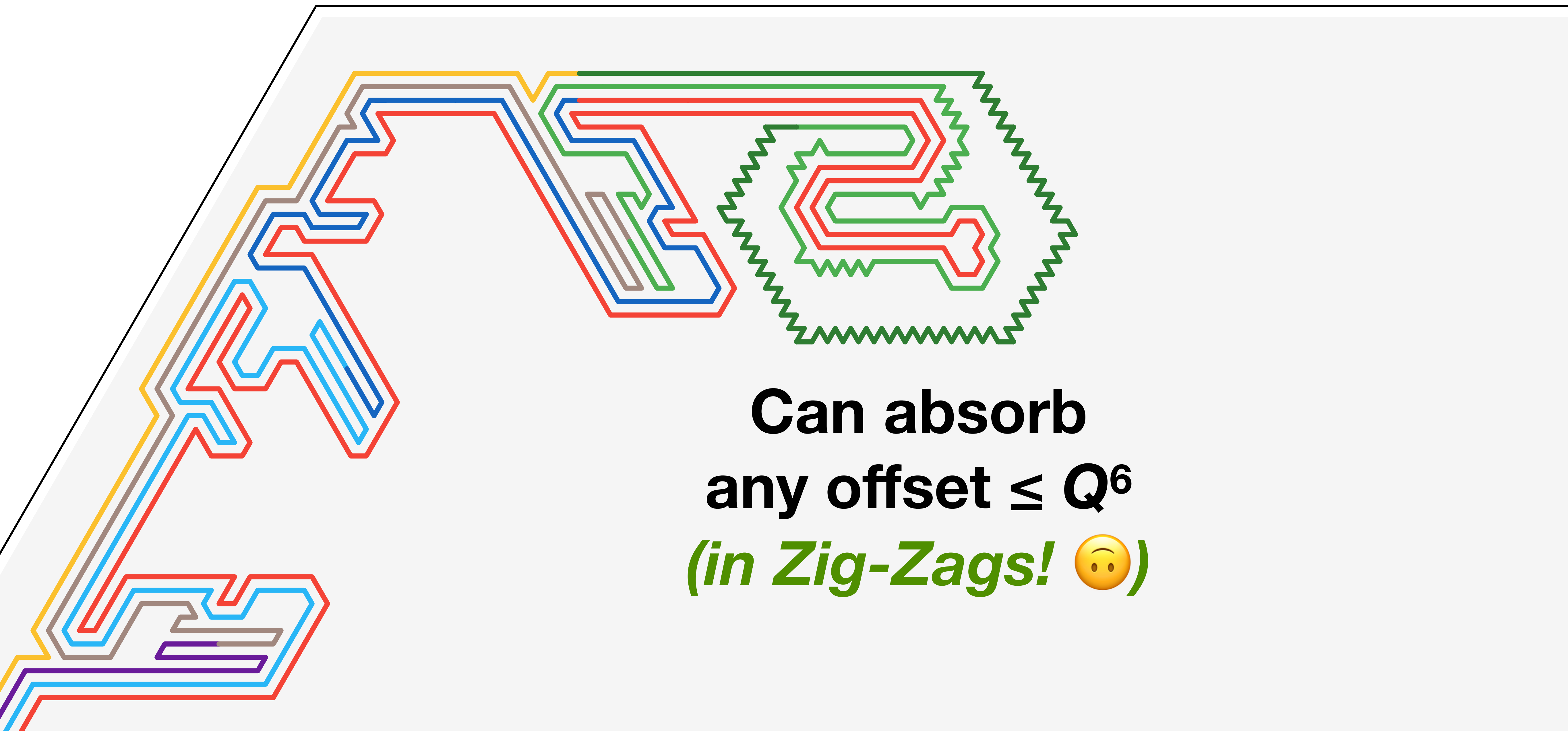


the Supercell



Resynchronizing.

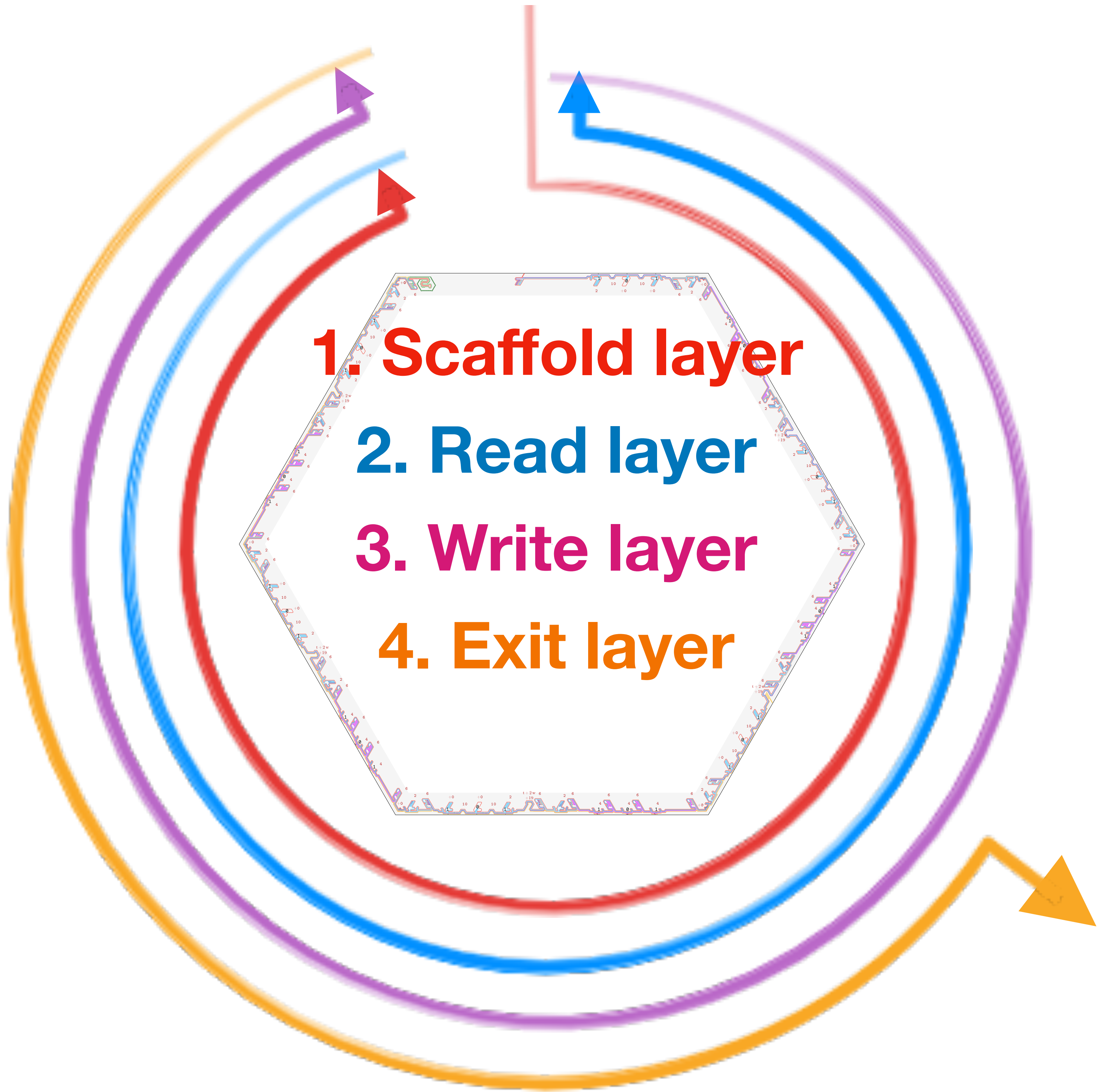
Speedbumps [PSSU2020]



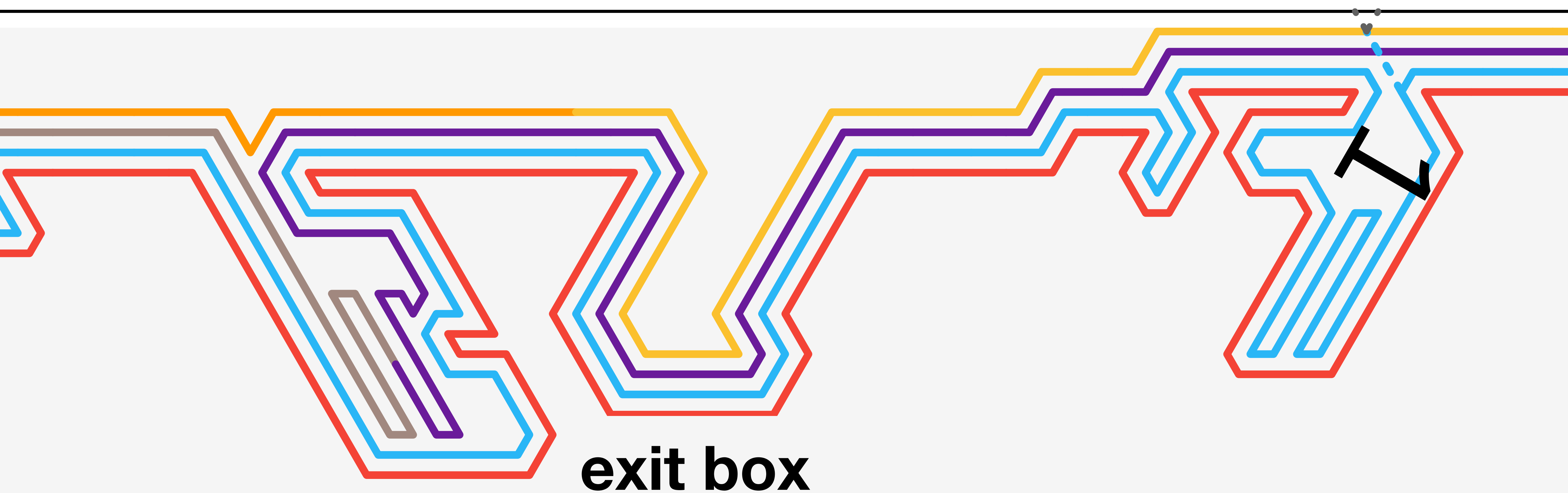
Can absorb any offset ≤ Q^6

(in Zig-Zags! 🙃)

the Supercell

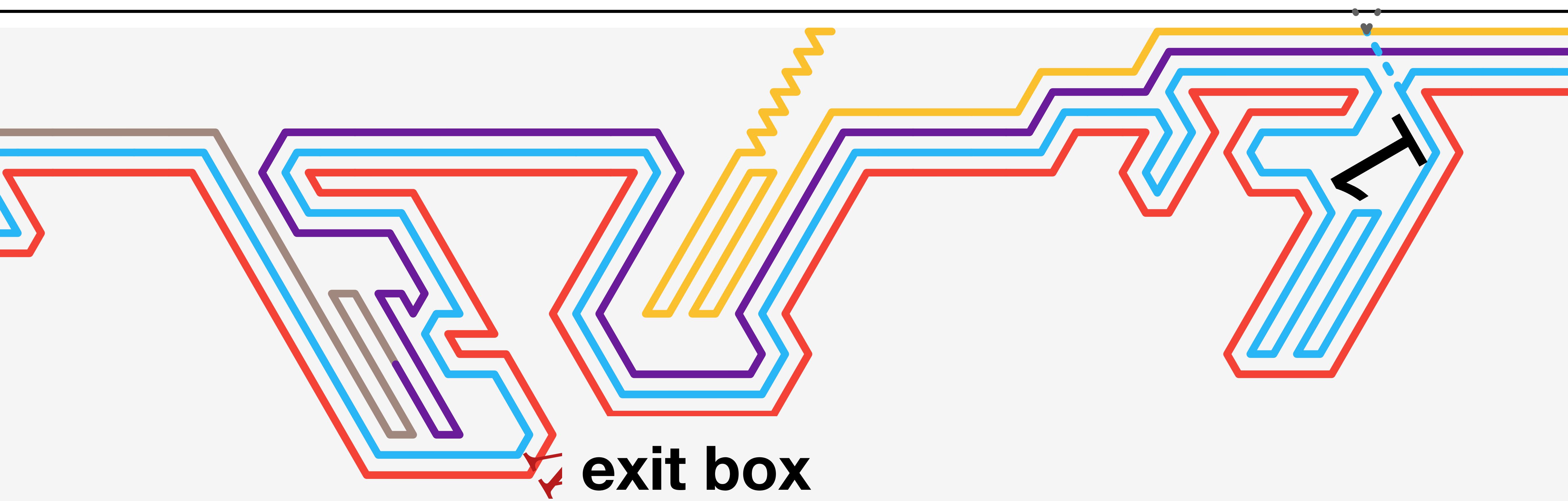


Exiting… or not



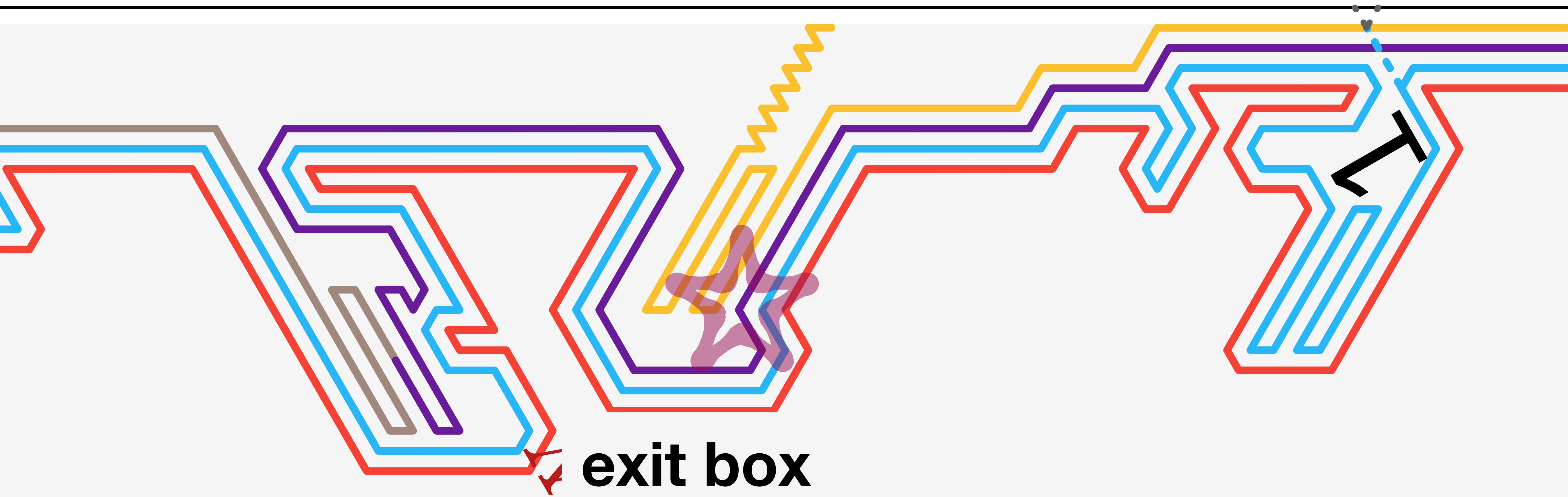
By default, exit layer follows the border of the exit box

Exiting... or not



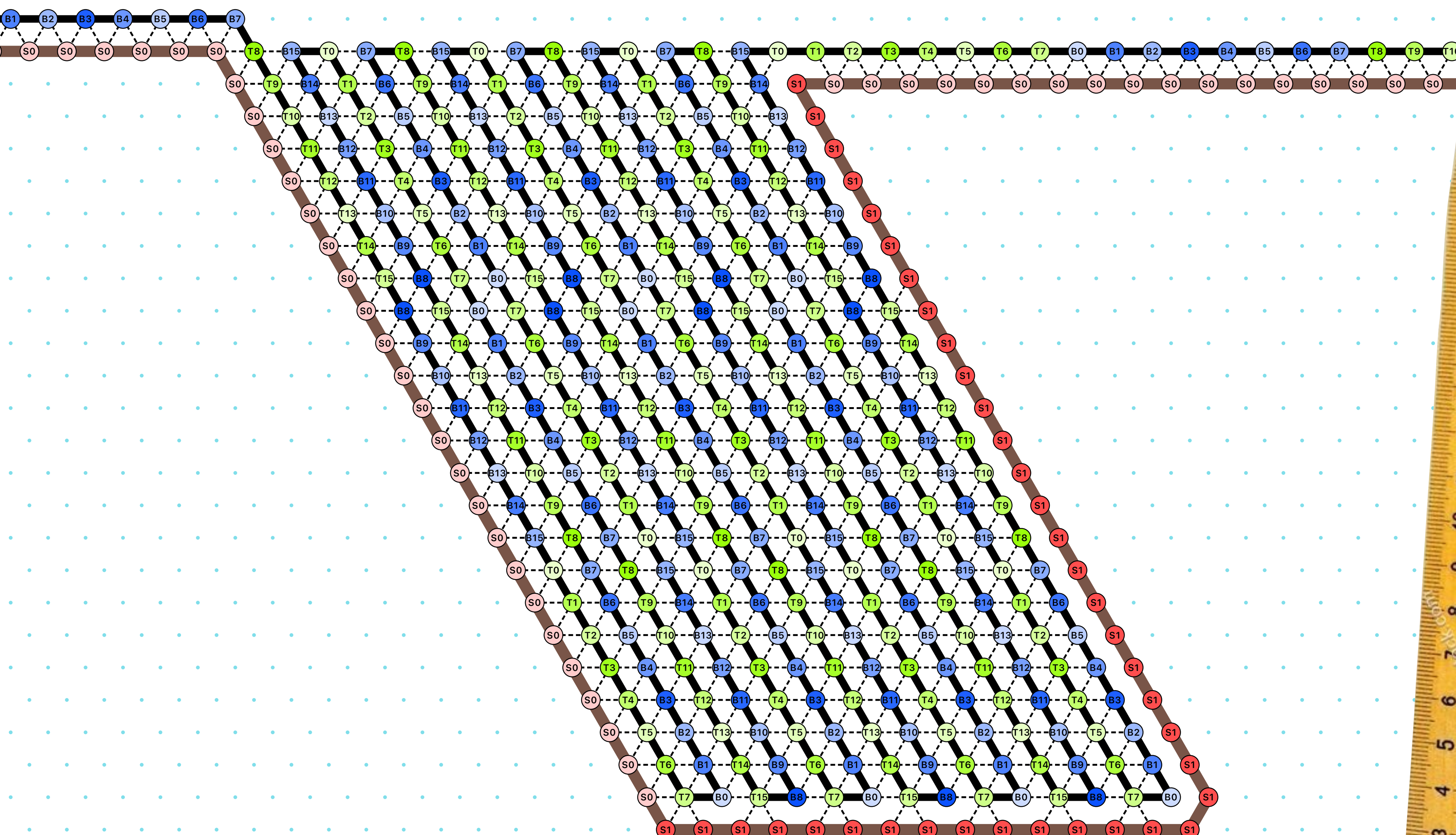
With the proper signal (offset!), exit layer folds upon itself and... exit!

Exiting… or not



With the proper signal (offset!), exit layer folds upon itself and… exit!

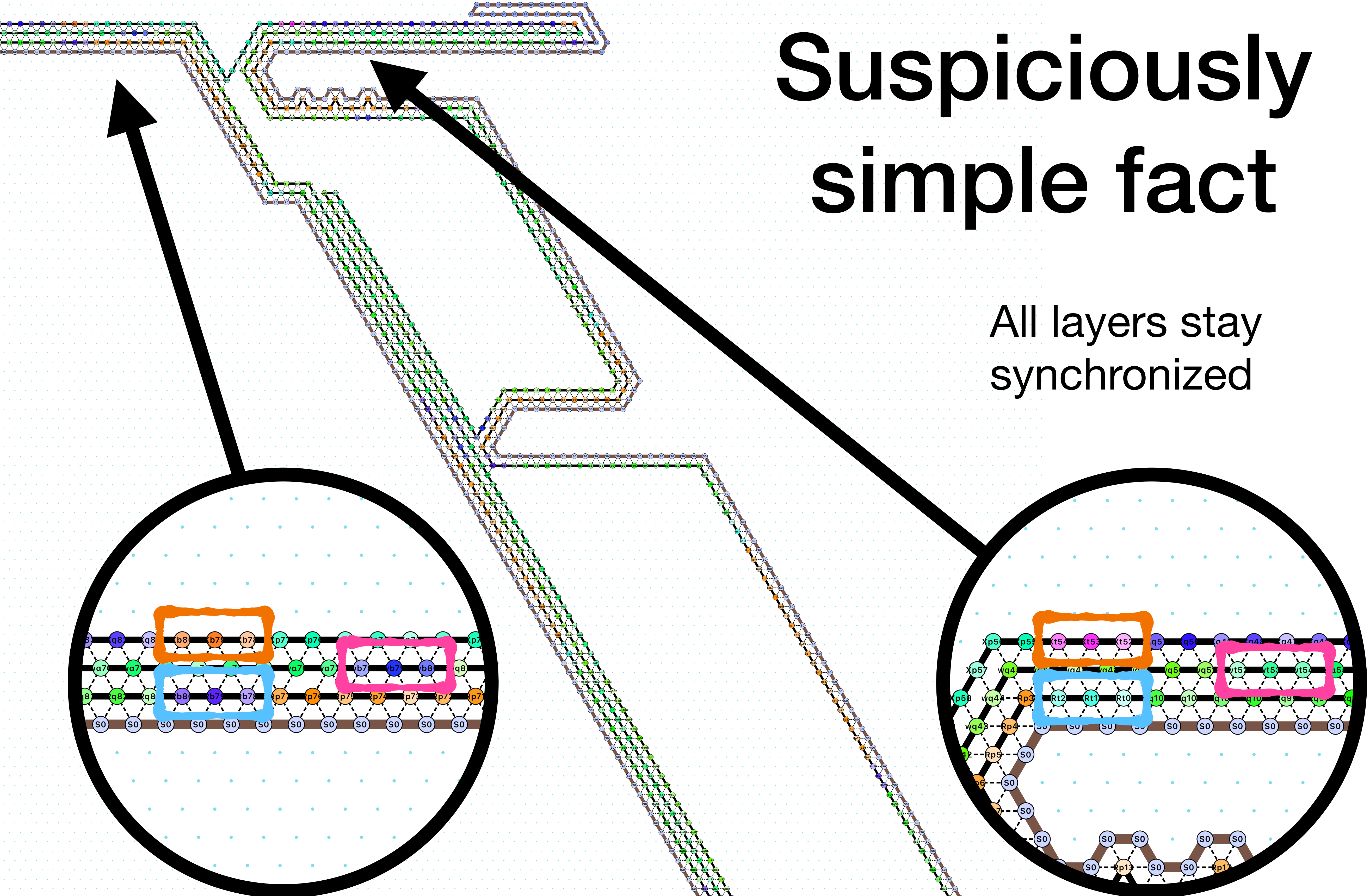
Key new tool: Folding meter



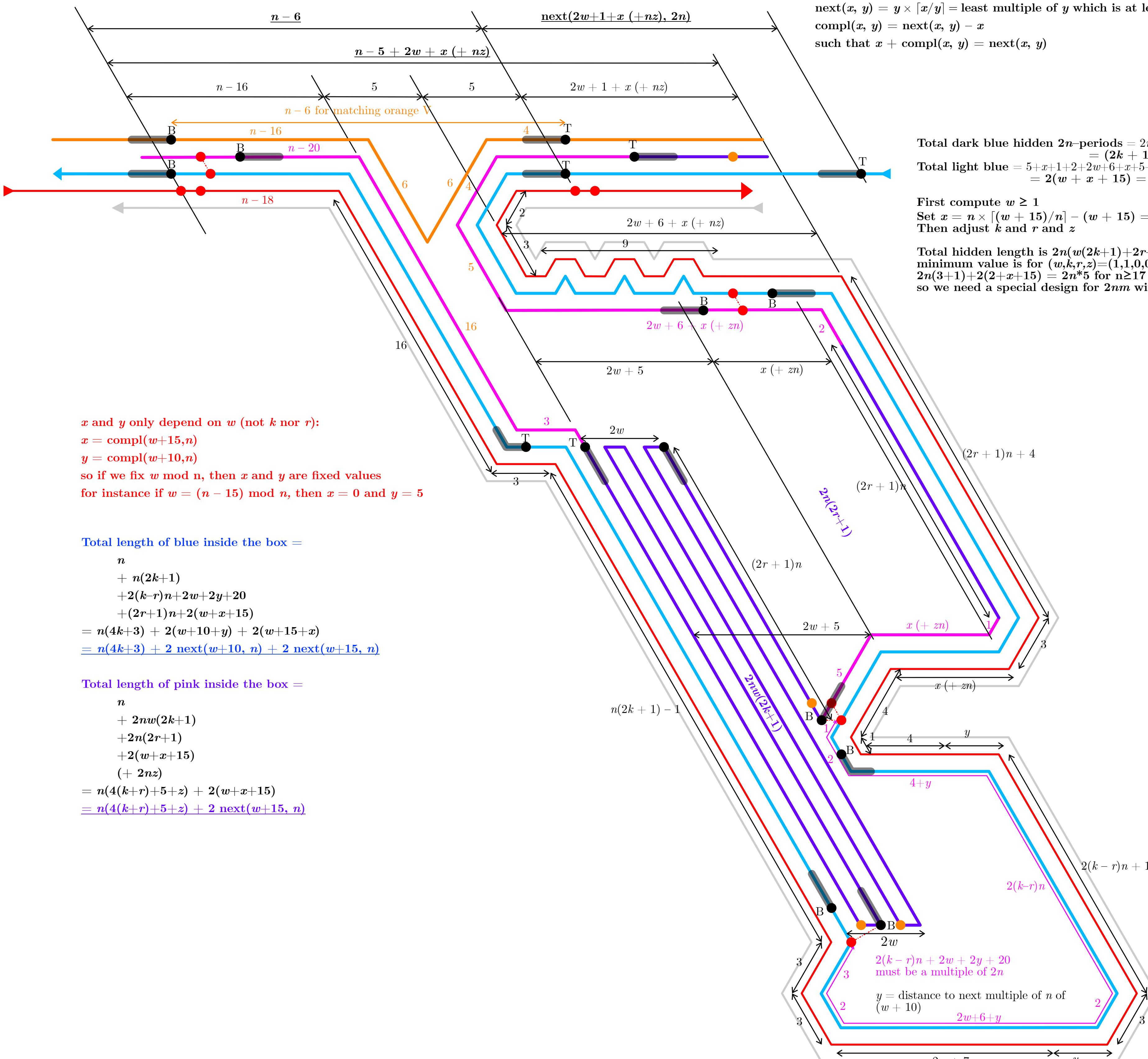
it folds upon itself into boxes

Suspiciously simple fact

All layers stay synchronized



Almost there



next(x, y) $= y \times \lceil x/y \rceil$ = least multiple of y which is at least x

compl(x, y) = next$(x, y) - x$

such that x + compl(x, y) = next(x, y)

Total dark blue hidden $2n$–periods $= 2r + 1 + w(2k + 1)$
$= \mathbf{(2k + 1) + 2r + 1}$

Total light blue $= 5{+}x{+}1{+}2{+}2w{+}6{+}x{+}5{+}4{+}7$
$= \mathbf{2(w + x + 15)}$ **= a multiple of $2n$**

First compute $w \geq 1$
Set $x = n \times \lceil (w + 15)/n \rceil - (w + 15)$ = compl(w+15, n)
Then adjust k and r and z

Total hidden length is $2n(w(2k{+}1){+}2r{+}1{+}z){+}2(w{+}x{+}15)$
minimum value is for $(w,k,r,z)=(1,1,0,0)$
$2n(3{+}1){+}2(2{+}x{+}15) = 2n{*}5$ for n≥17
so we need a special design for $2nm$ with $m \leq 4$

x and y only depend on w (not k nor r):
x = compl(w+15,n)
y = compl(w+10,n)
so if we fix w mod n, then x and y are fixed values
for instance if $w = (n - 15)$ mod n, then $x = 0$ and $y = 5$

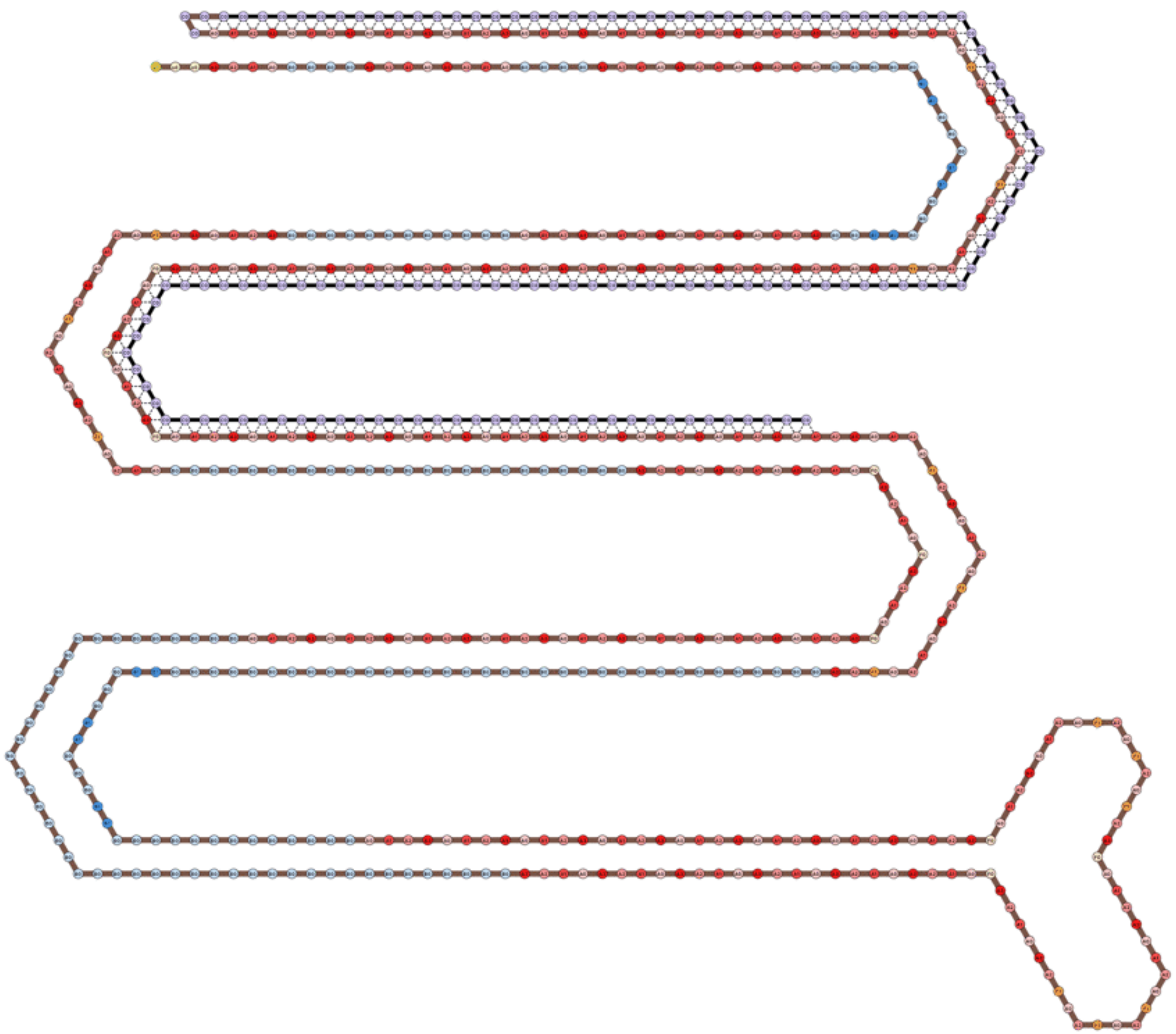
Total length of blue inside the box =

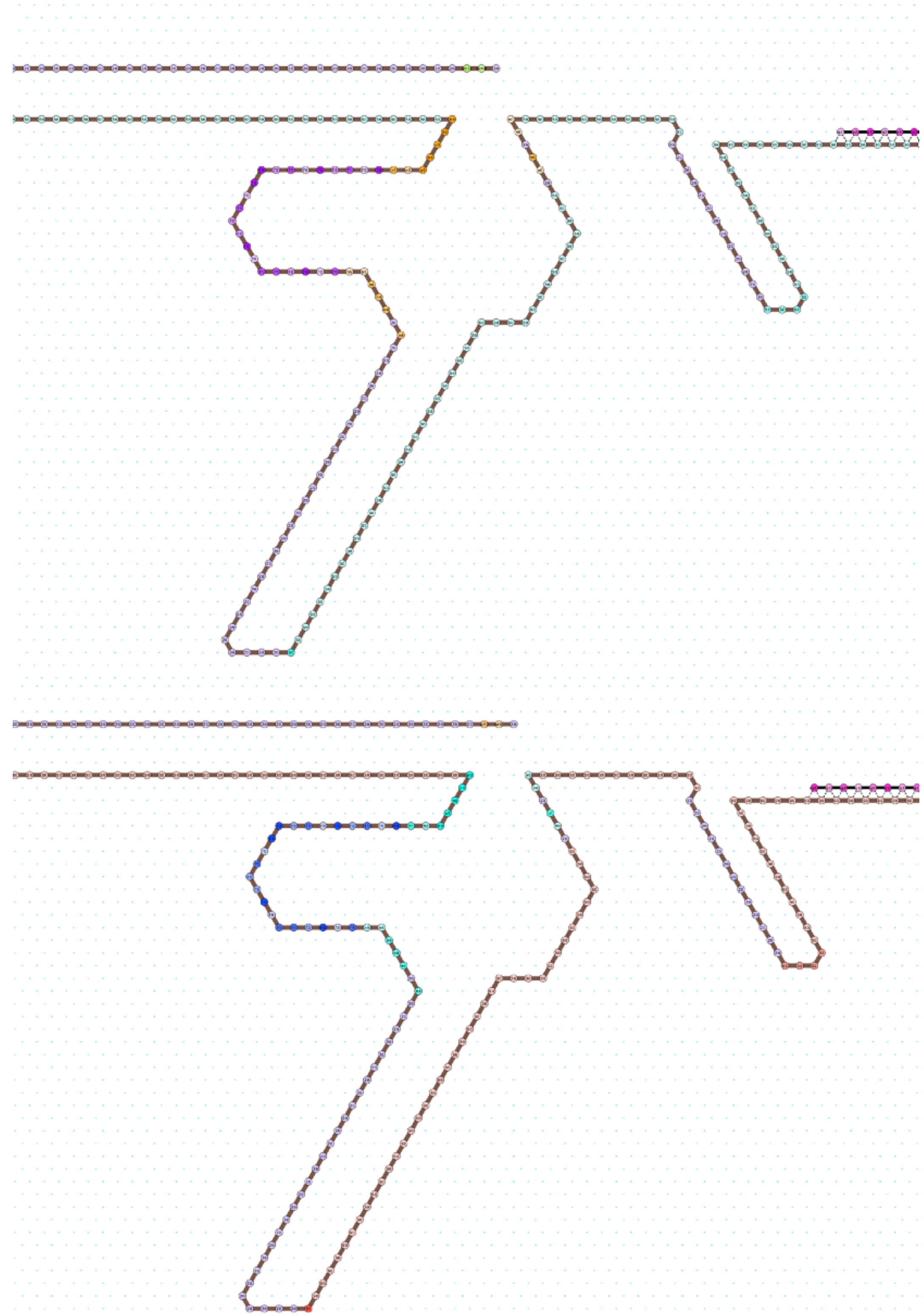
n
$+\ n(2k{+}1)$
$+2(k{-}r)n{+}2w{+}2y{+}20$
$+(2r{+}1)n{+}2(w{+}x{+}15)$
$= n(4k{+}3) + 2(w{+}10{+}y) + 2(w{+}15{+}x)$
$\underline{= n(4k{+}3) + 2\ \text{next}(w{+}10,\ n) + 2\ \text{next}(w{+}15,\ n)}$

Total length of pink inside the box =

n
$+\ 2nw(2k{+}1)$
$+2n(2r{+}1)$
$+2(w{+}x{+}15)$
$(+\ 2nz)$
$= n(4(k{+}r){+}5{+}z) + 2(w{+}x{+}15)$
$\underline{= n(4(k{+}r){+}5{+}z) + 2\ \text{next}(w{+}15,\ n)}$

Almost there





Conclusion

- Furthermore, Oritatami & Turmite doodles can have **any density expressible in… Π2** (?)
- **No need for parallelism**
- **No need for 3D**
- Lines just don't cross!
- Some more work and we'll have an running implementation!
- New tools for Oritatami: **Folding meter, oubliettes, distant sensor & crazy speedbumps**!