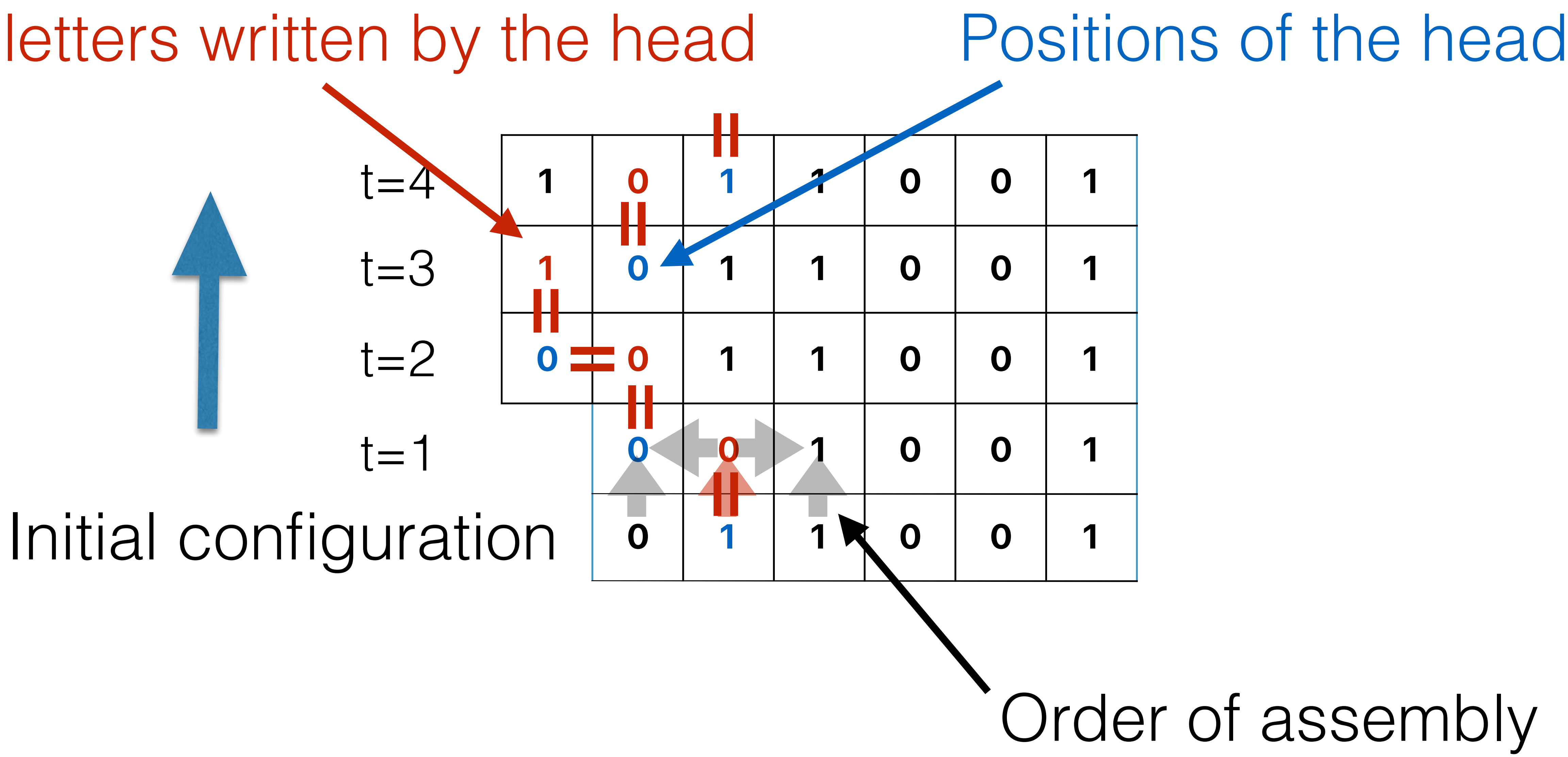
Universality in assembly models

Nicolas Schabanel
CNRS
LIP & IXXI - ÉNS de Lyon

Universality in algorithmic self-assembly

- Can we decide if an assembly terminates?
- Can we design a tile system for a given shape?
- Do I need more than one tileset?
- Do I need more than one tile?
- Do I need more than one molecule?

Algorithmic self-assembly simulates any Turing machine at T°2 in 2D



Algorithmic self-assembly simulates any Turing machine at T°2 in 2D

Position and state of the head

Current tape right end

t=4	0	0	1,Halt	1	★
t=3	0	0	0,q'''	1	★
t=2	0	0	0	0,q''	★
t=1	0	0	1,q'	★	
t=0	0	1,q	1	★	

Transitions:

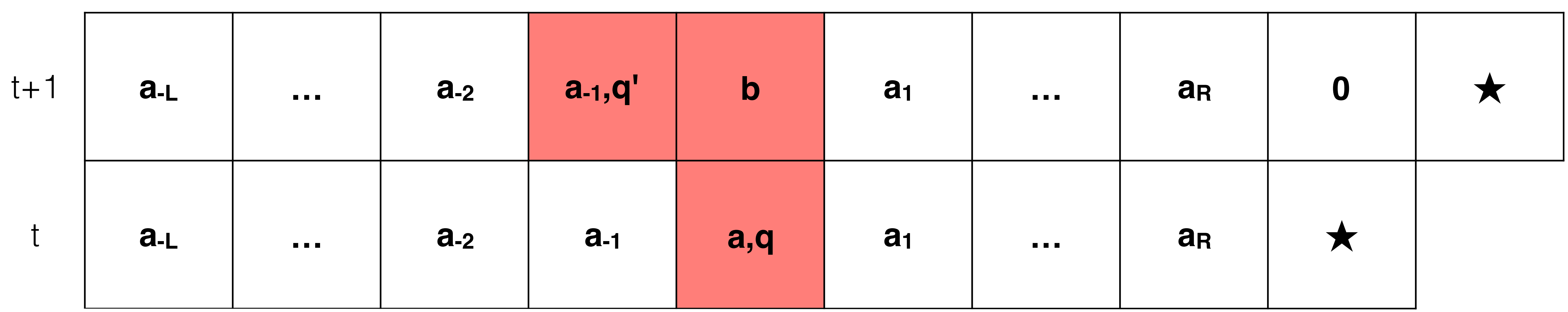
$$(0,q''') \longmapsto (1,\mathrm{Halt},\bullet)$$

$$(0,q'') \longmapsto (1,q''',\leftarrow)$$

$$(1,q') \longmapsto (0,q'',\rightarrow)$$

$$(1,q) \longmapsto (0,q',\rightarrow)$$

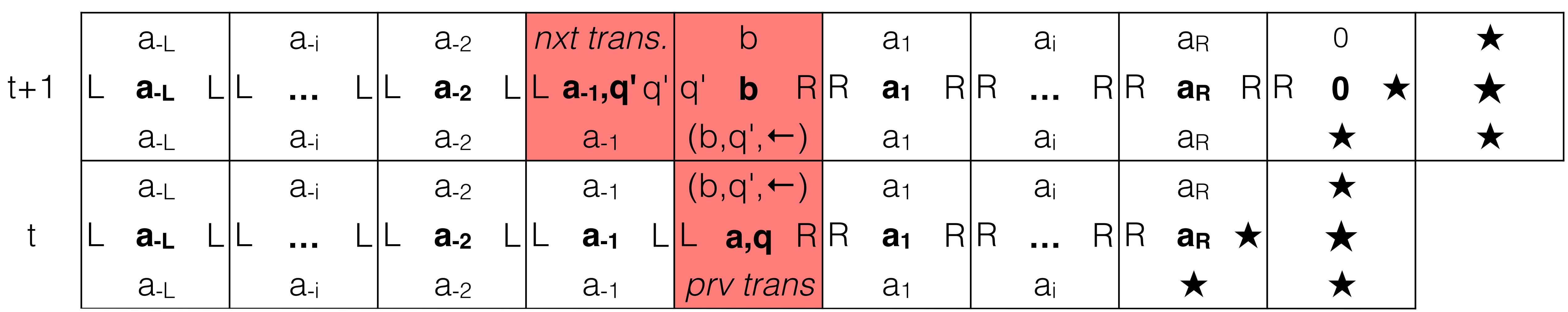
Algorithmic self-assembly simulates any Turing machine at T°2 in 2D



Tiles for (a,q) $\mapsto$ (b,q',$\leftarrow$)

Algorithmic self-assembly simulates any Turing machine at T°2 in 2D

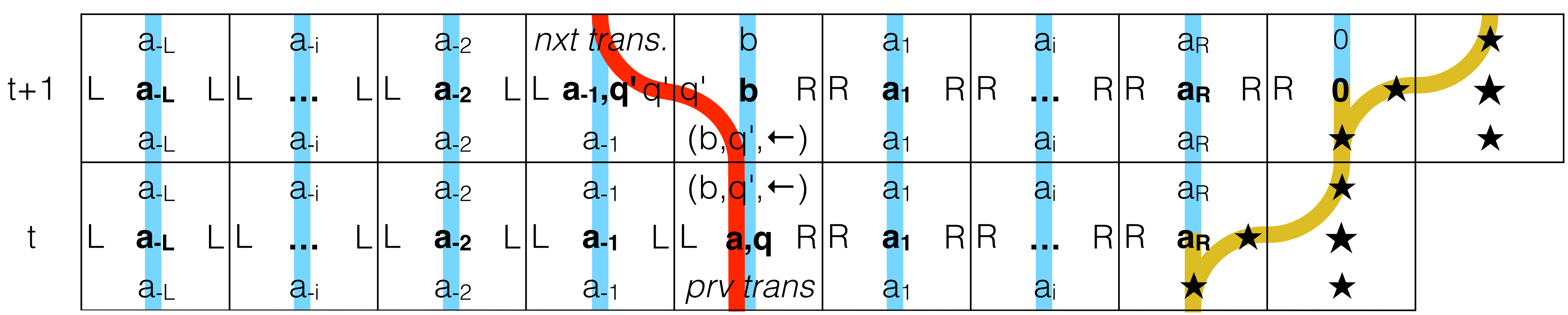
1) Organize the information flow passing through the glues on the sides



Tiles for $(a,q) \mapsto (b,q',\leftarrow)$

Algorithmic self-assembly simulates any Turing machine at T°2 in 2D

1) Organize the information flow passing through the glues on the sides

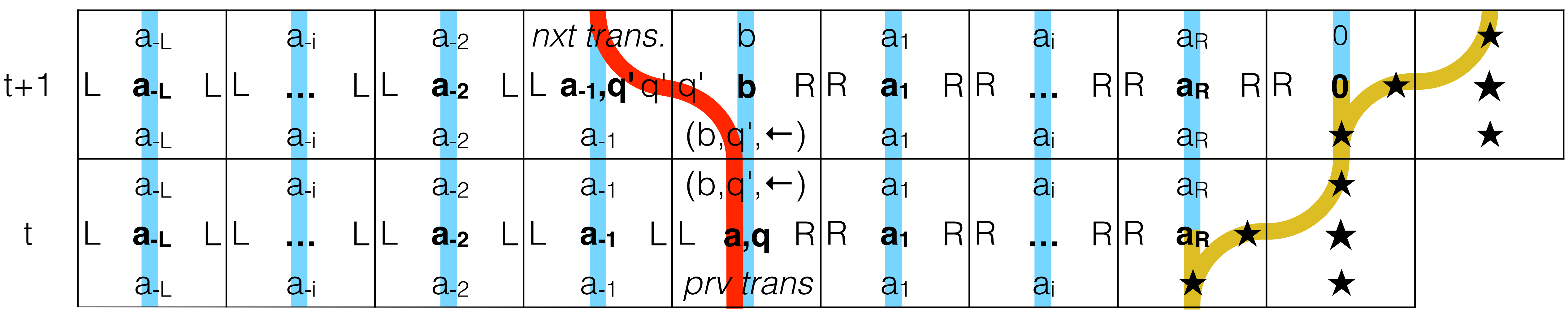


Tiles for $(a,q) \mapsto (b,q',\leftarrow)$

Algorithmic self-assembly simulates any Turing machine at T°2 in 2D

1) Organize the information flow passing through the glues on the sides

2) Identify each type of tiles



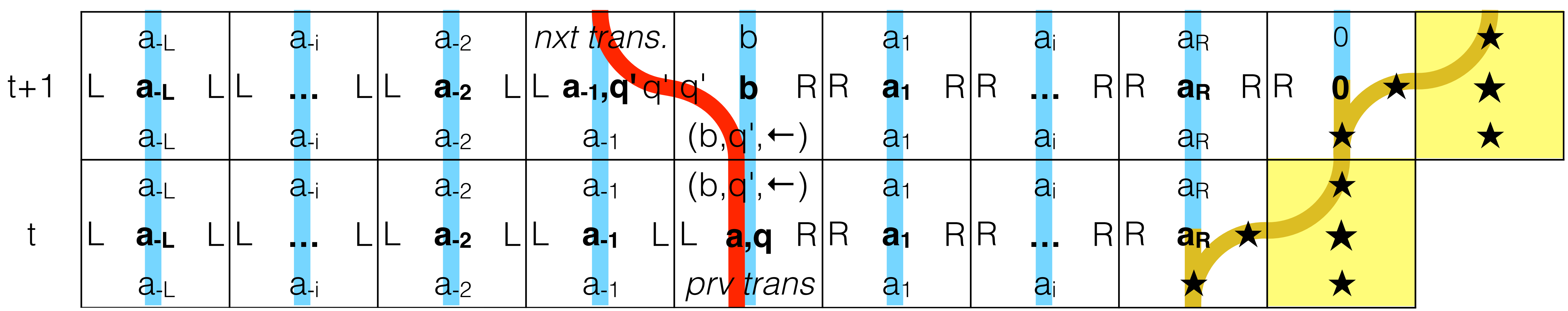
Tiles for (a,q) ⟼ (b,q',←)

Algorithmic self-assembly simulates any Turing machine at T°2 in 2D

1) Organize the information flow passing through the glues on the sides

2) Identify each type of tiles

The skeleton



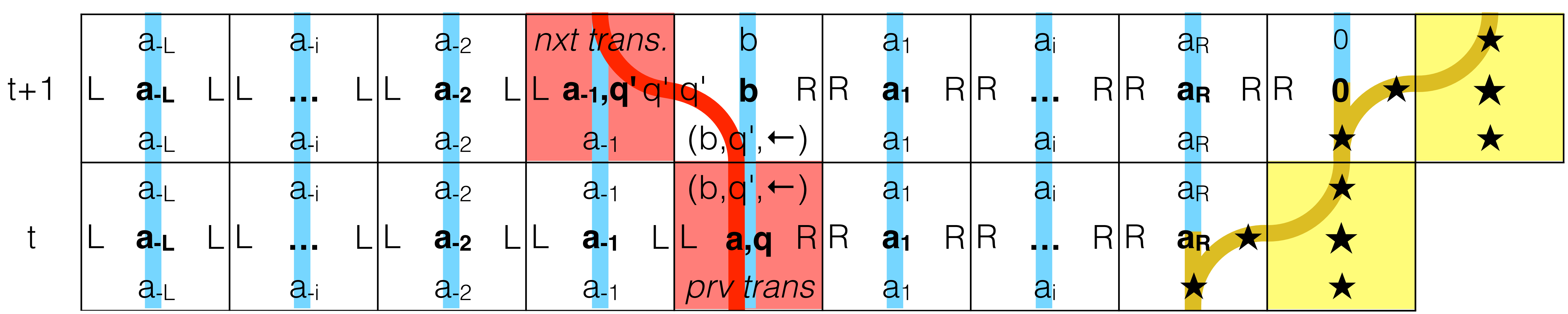
Tiles for $(a,q) \mapsto (b,q',\leftarrow)$

Algorithmic self-assembly simulates any Turing machine at T°2 in 2D

1) Organize the information flow passing through the glues on the sides

2) Identify each type of tiles

The skeleton



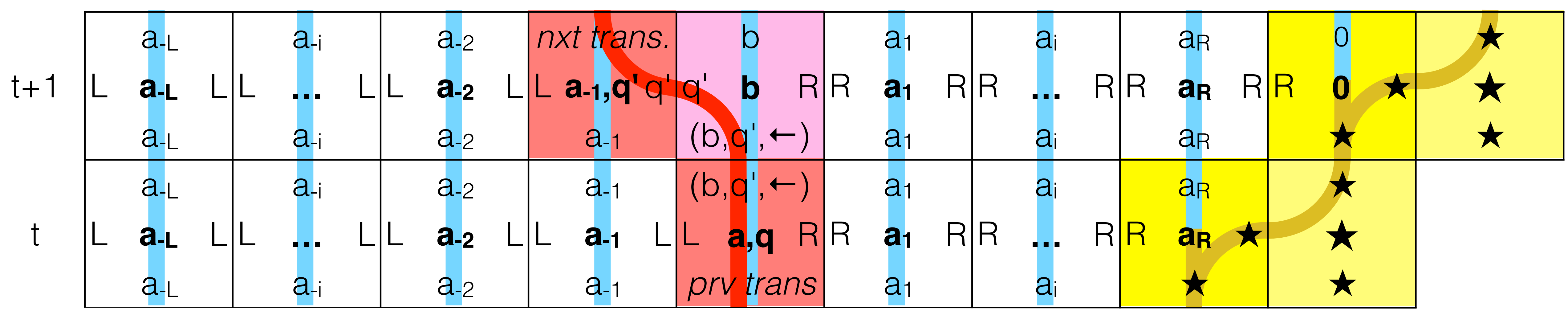
Tiles for (a,q) ⟼ (b,q',←)

Algorithmic self-assembly simulates any Turing machine at T°2 in 2D

1) Organize the information flow passing through the glues on the sides

2) Identify each type of tiles

The carriers



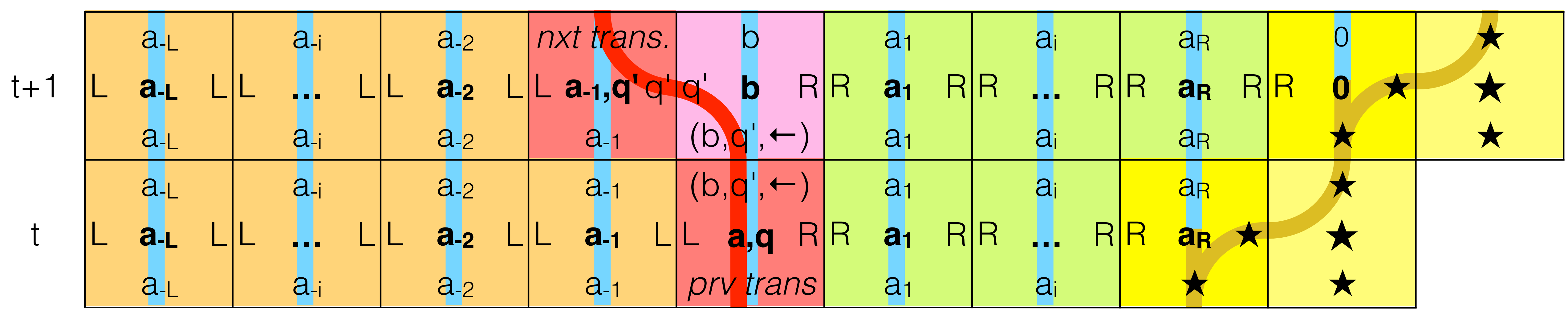
Tiles for $(a,q) \mapsto (b,q',\leftarrow)$

Algorithmic self-assembly simulates any Turing machine at T°2 in 2D

1) Organize the information flow passing through the glues on the sides

2) Identify each type of tiles

The filling tiles



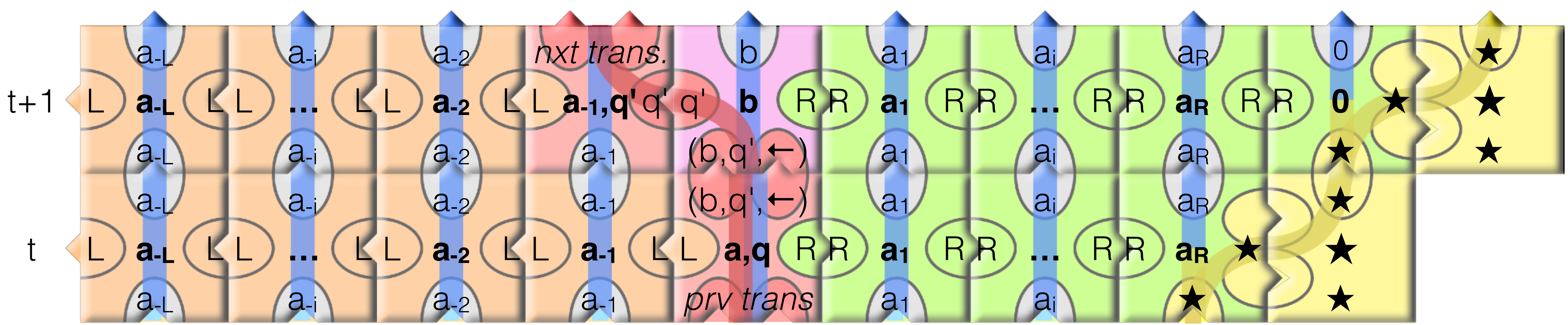
Tiles for (a,q) ⟼ (b,q',←)

Algorithmic self-assembly simulates any Turing machine at T°2 in 2D

1) Organize the information flow passing through the glues on the sides

2) Identify each type of tiles

3) Set the order, glue strength & colors to match the flow

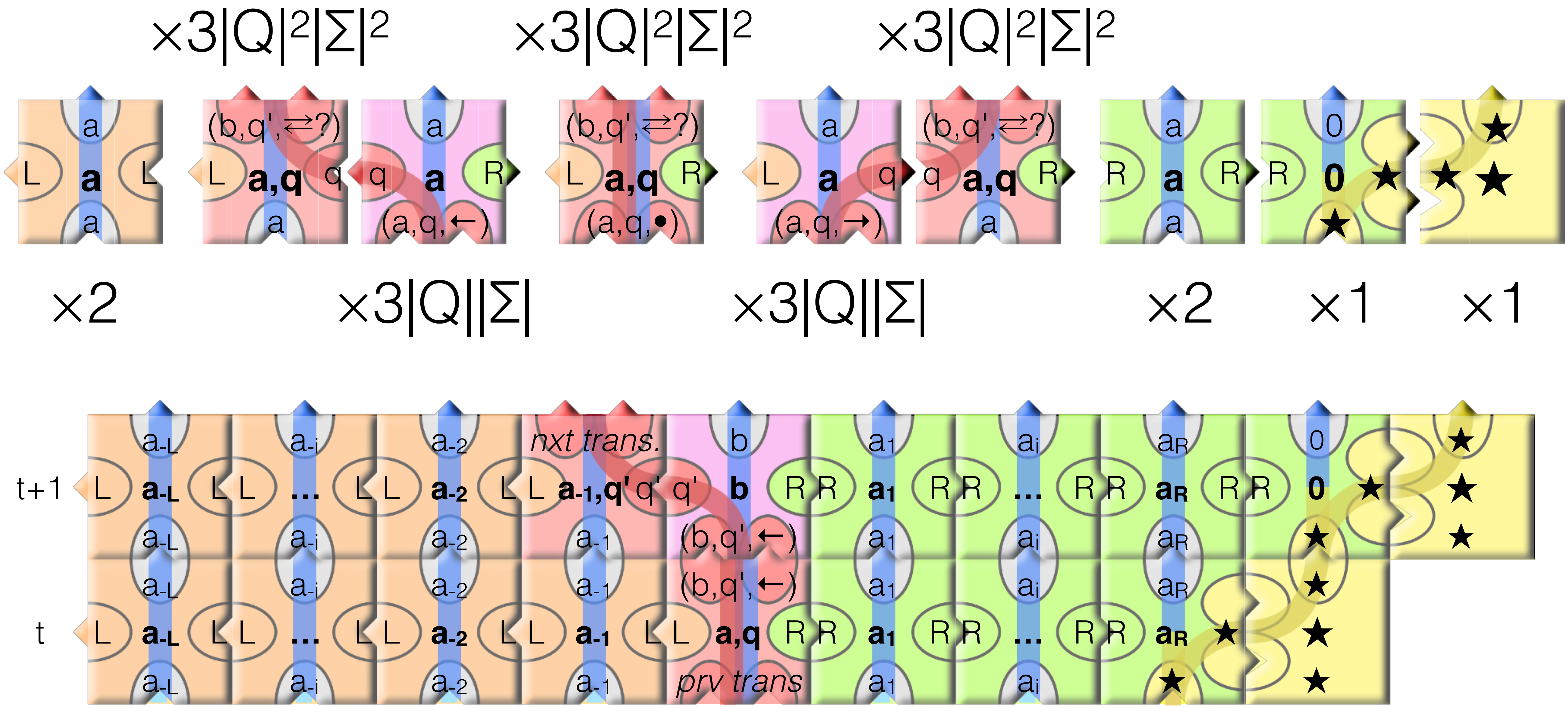


Tiles for (a,q) ↦ (b,q',←)

The tileset

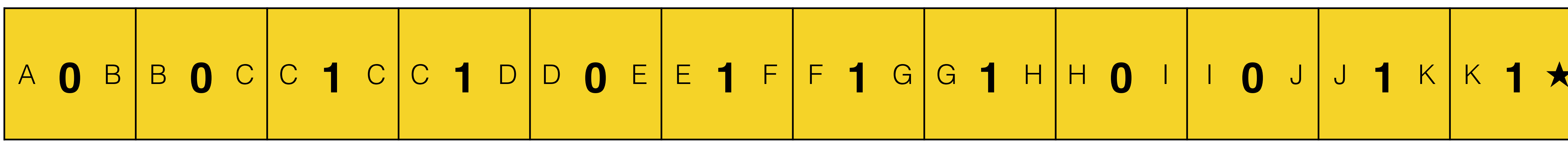
with size $O(|Q|^2|\Sigma|^2)$

The tiles for each transition $(a,q) \mapsto (b,q',\rightleftarrows?)$



Encoding the input as a seed

- **Solution 1: hardcoding**
  Uses **n** tiles for n bits



- Each tile requires **4 log n bits** to encode its glues, thus this encoding uses **4n log n bits** in total
- Can we do better, with less tiles ?

Kolmogorov complexity

- Given a universal Turing machine U:

  $$K_U(x) = \min \{ |p| : U(p,\varepsilon) = x \}$$

  is the size of the smallest program in U that outputs x

- **Fact.** $\forall U, \exists A$ s.t. $\forall x$, $K_U(x) \leq |x| + A$

  Proof. A is the size of the program "print" in U.

- **Theorem. $K_U(x)$ is independent of U**, indeed:

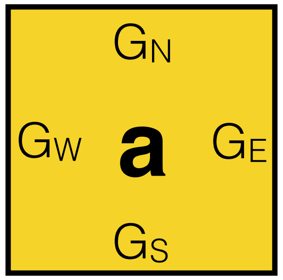
  $$\forall U, U' \; \exists A, B \text{ s.t. } \forall x,\; K_{U'}(x) - A \leq K_U(x) \leq K_{U'}(x) + B$$

  Proof. A and B are the sizes of the programs that execute U and U' respectively in U' and U.

Lower bounding the required number of tiles to encode the seed

- The Kolmogorov complexity **K(x)** is the size of the smallest program that outputs **x**

- Bit size of the encoding with **T** tiles is $\boldsymbol{\leq 4\ T \log_2 T}$

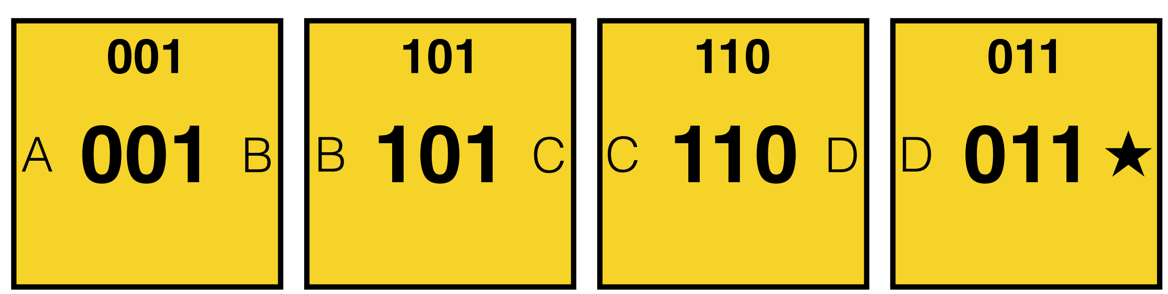


- A tileset that self-assembles x is a program that outputs x,

Thus: $\boldsymbol{4T \log T \geq K(x)}$, I.e. $\boldsymbol{T \geq K(x) / 4 \log K(x)}$

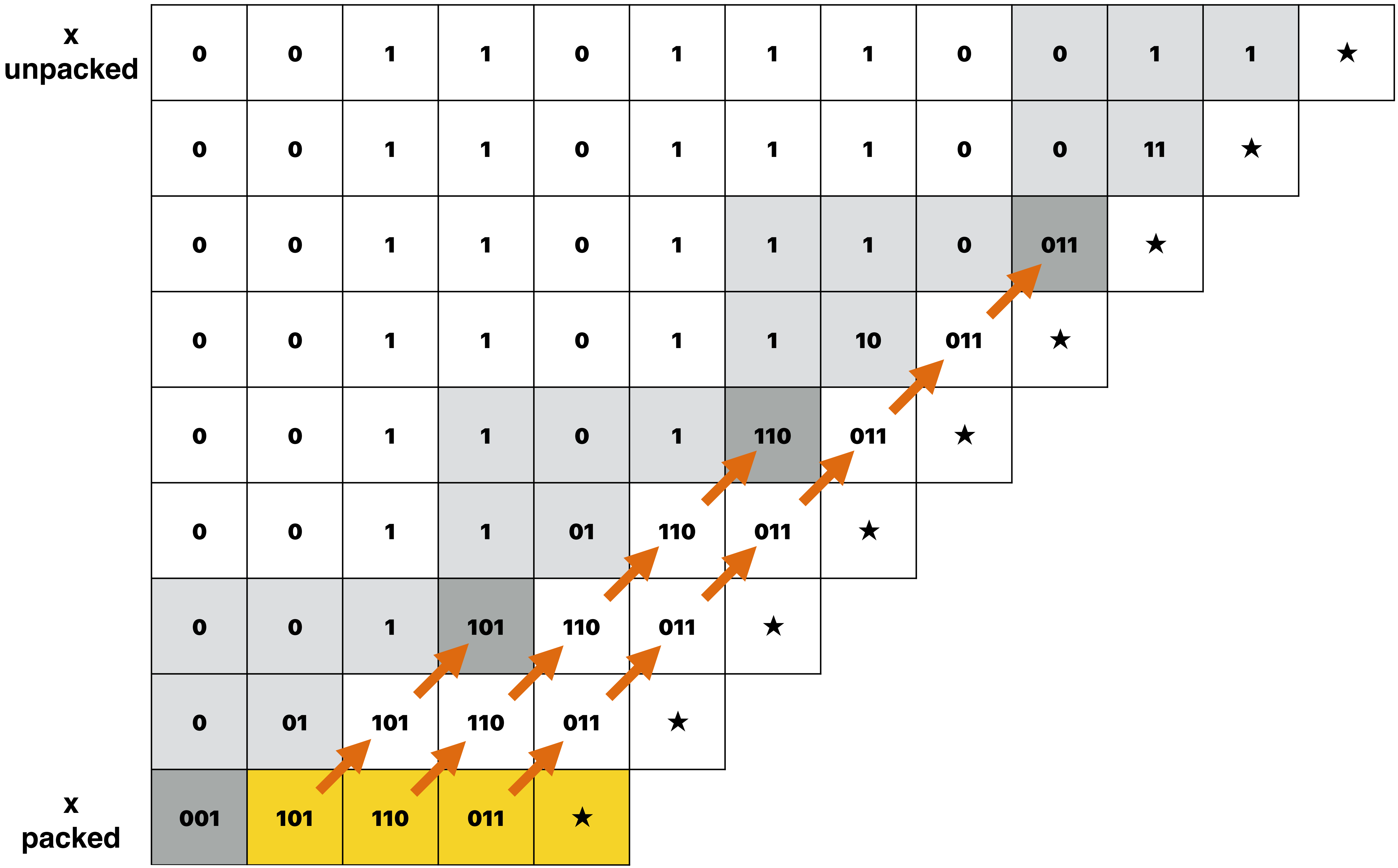
Unpacking a binary string

- Cut string **x** in **n / b** chunks of **b** bits and uncompress it
- Example: **x = 001 101 110 011** and **b = 3**

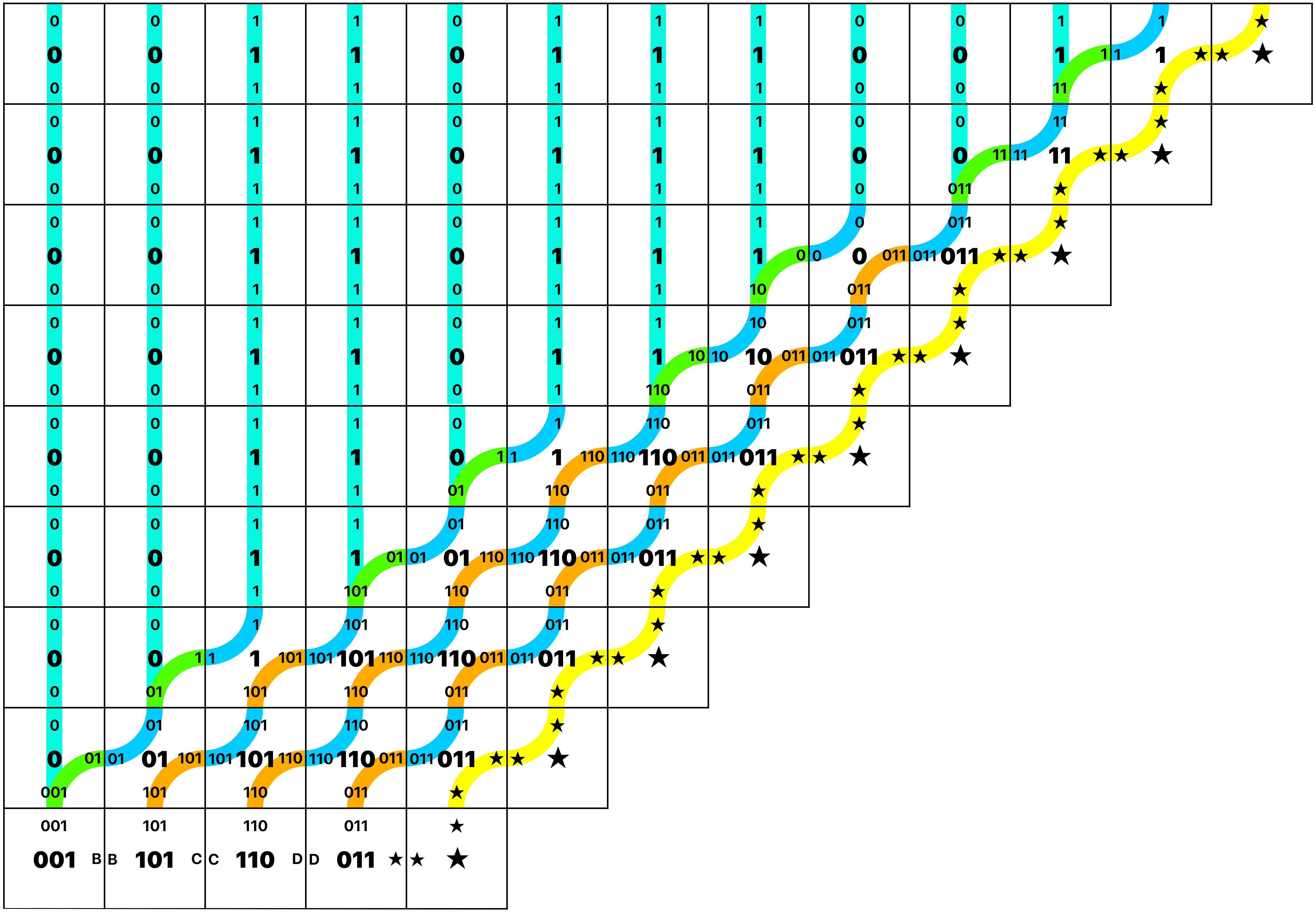


n / b tiles

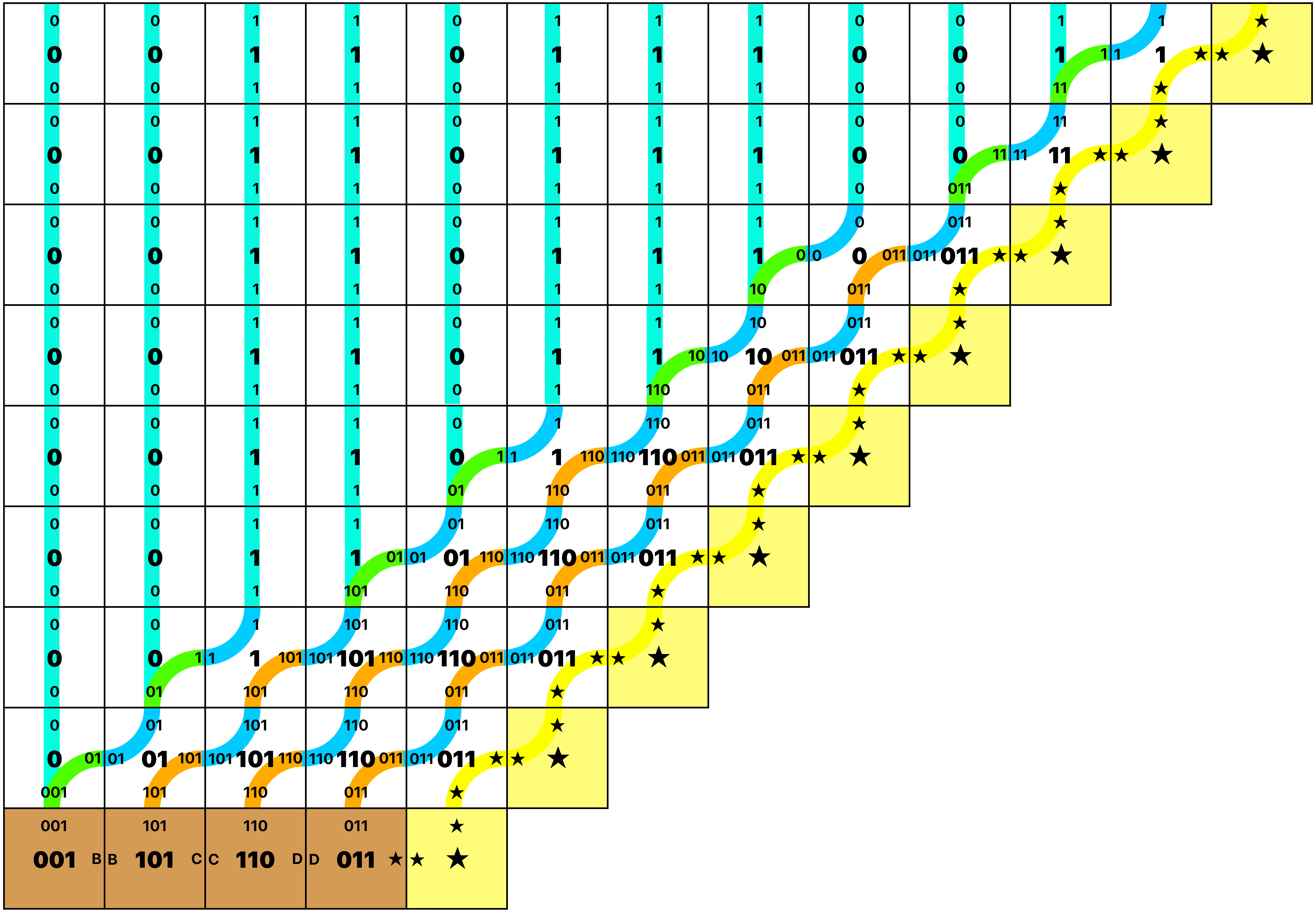
Unpacking a binary string



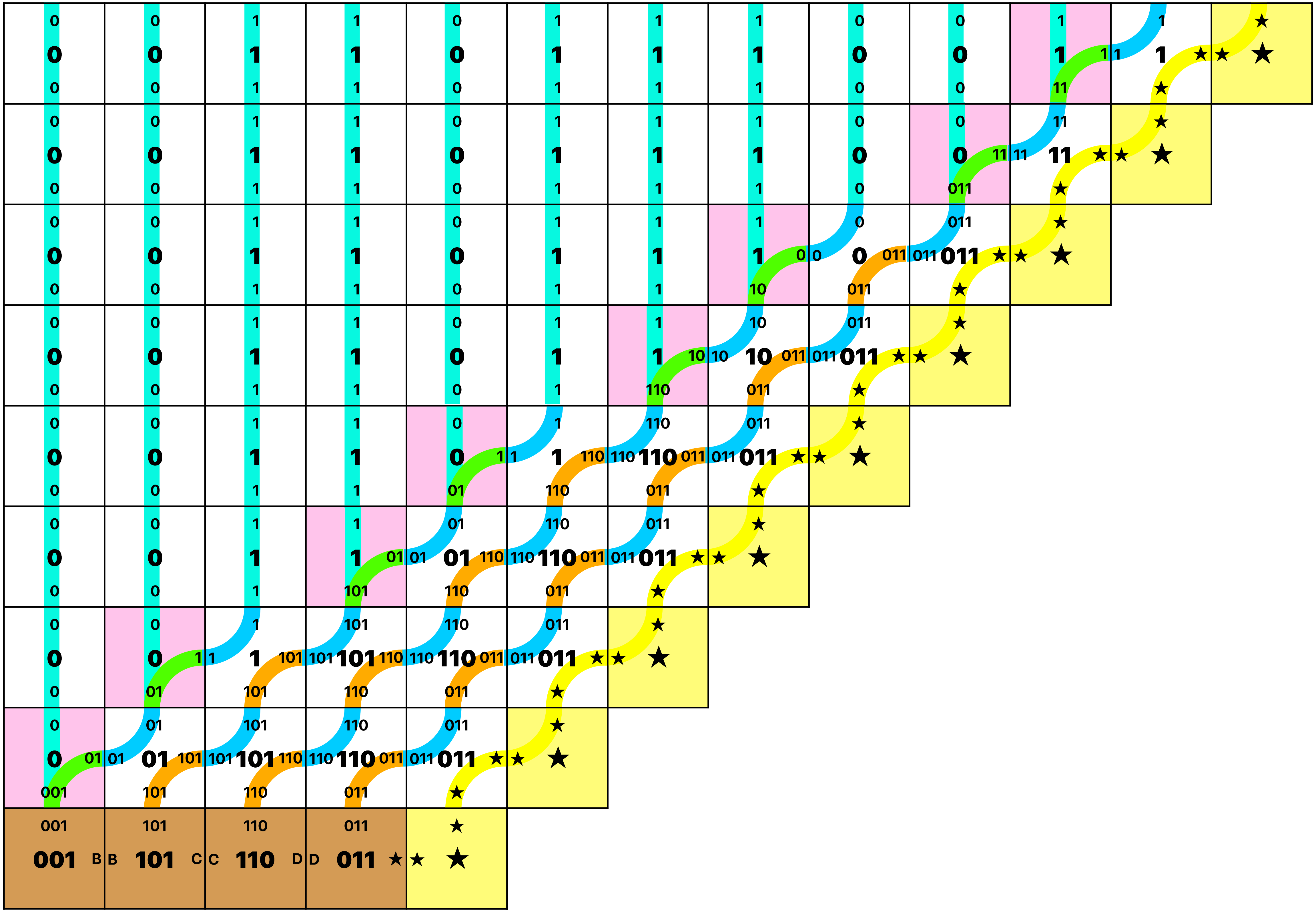
Determine the flow of information



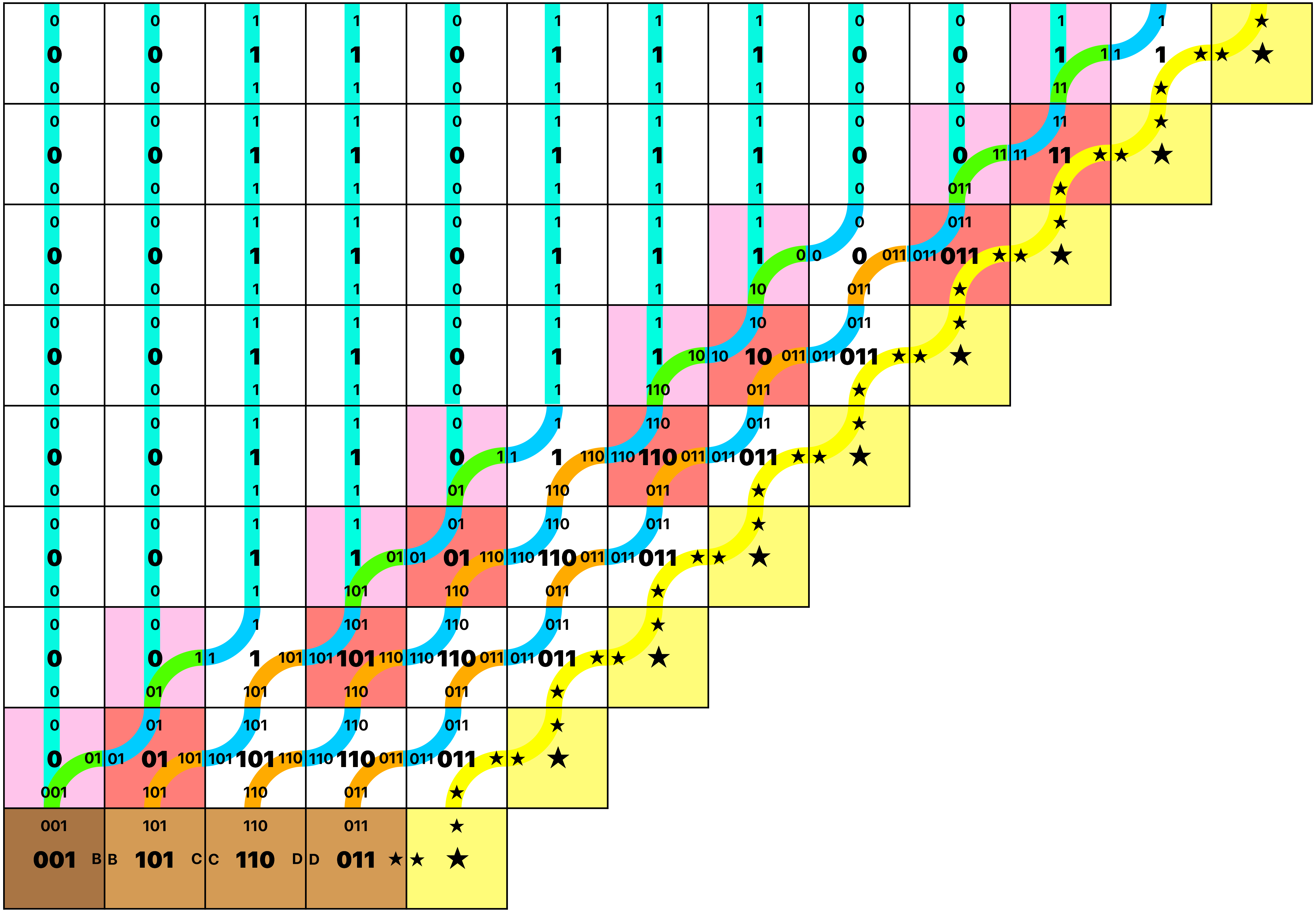
Determine the various tile types



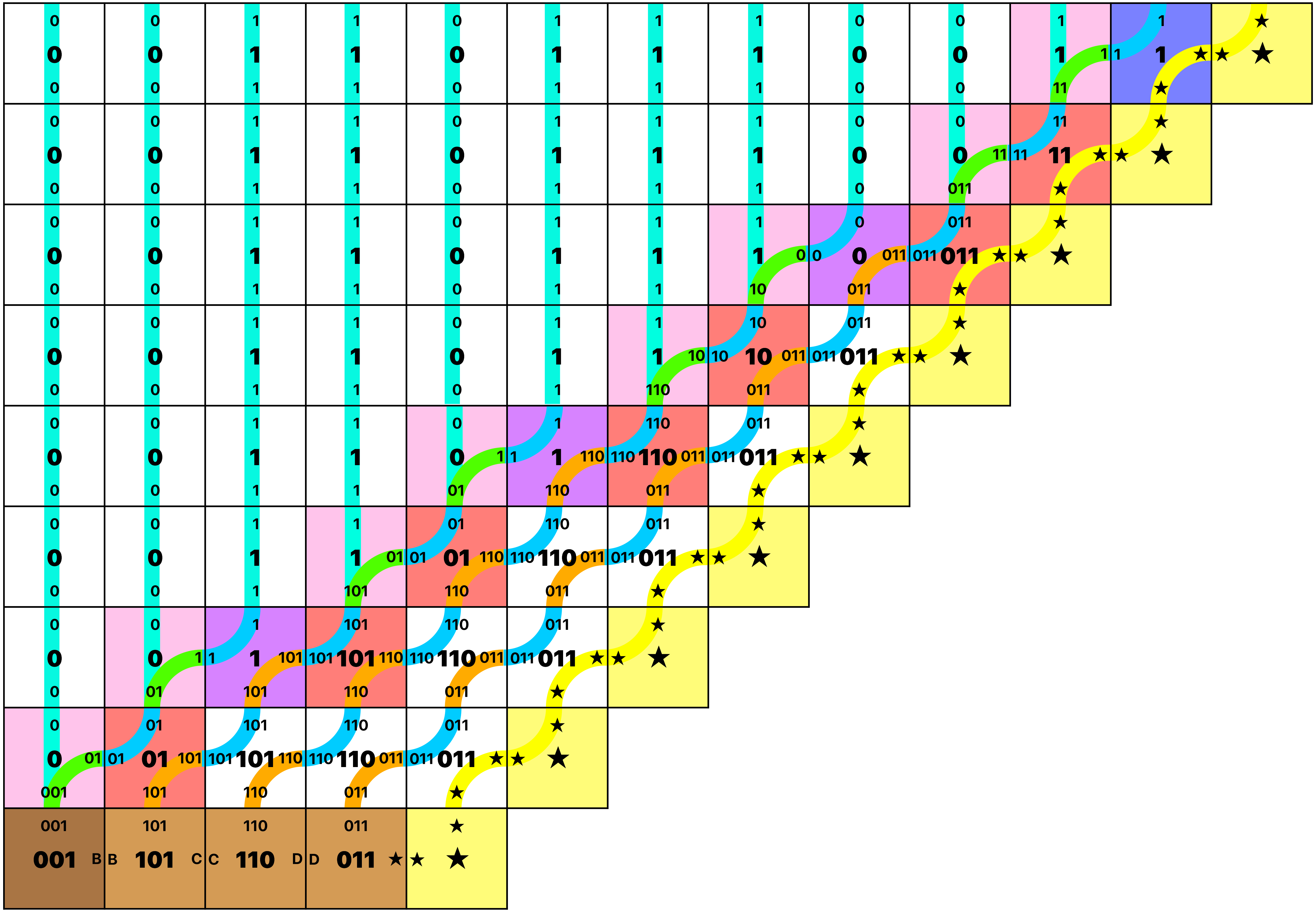
Determine the various tile types



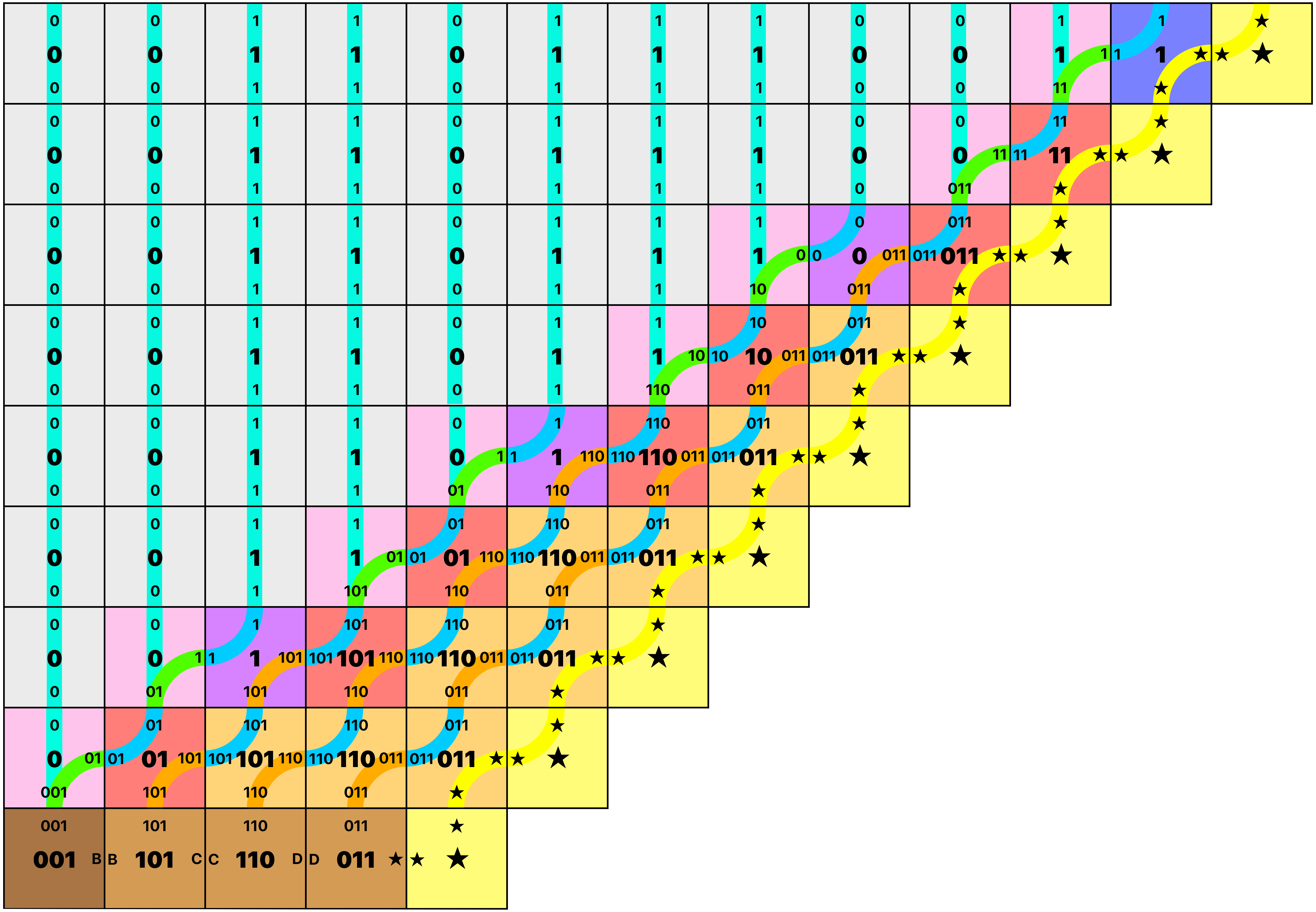
Determine the various tile types

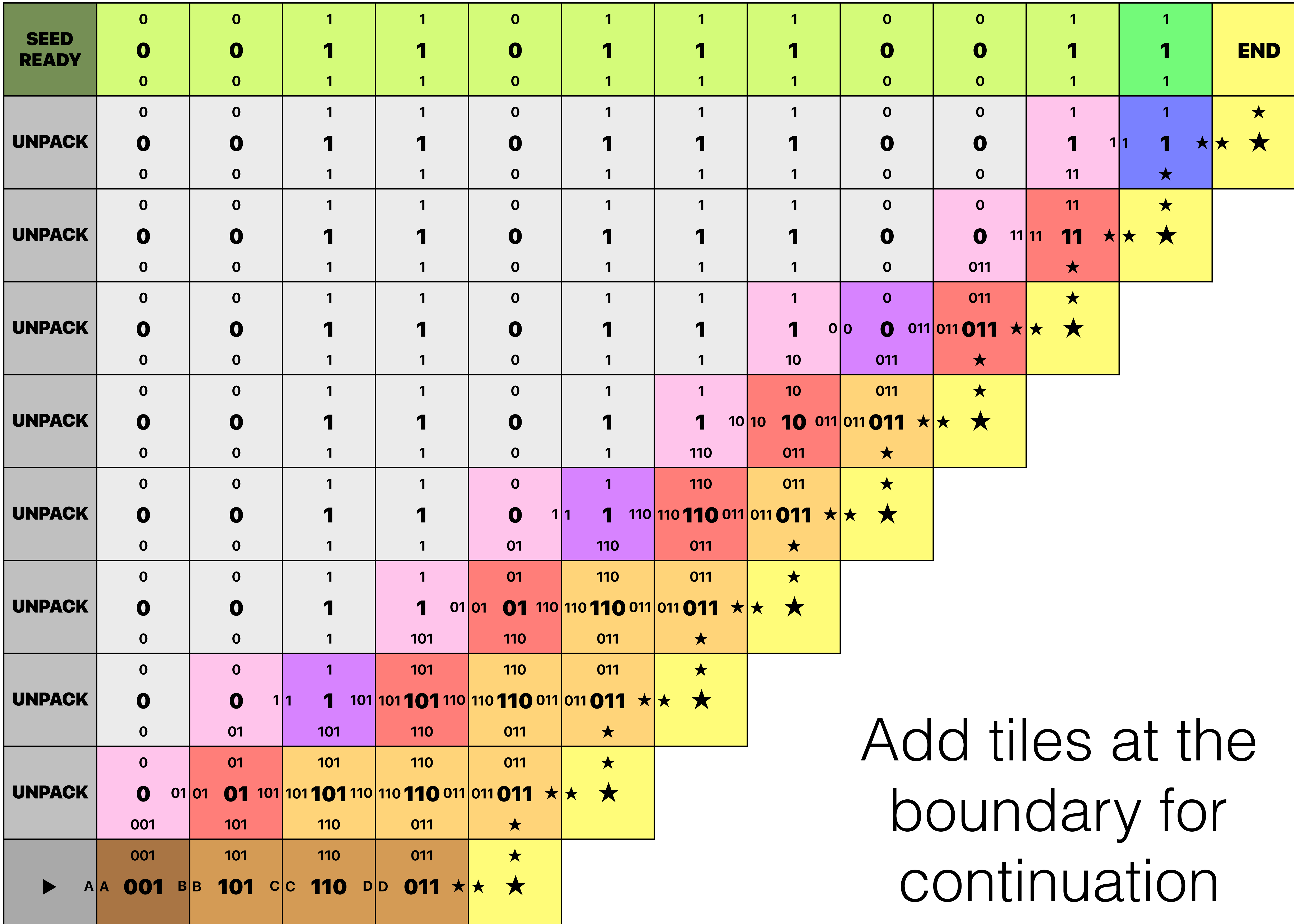


Determine the various tile types

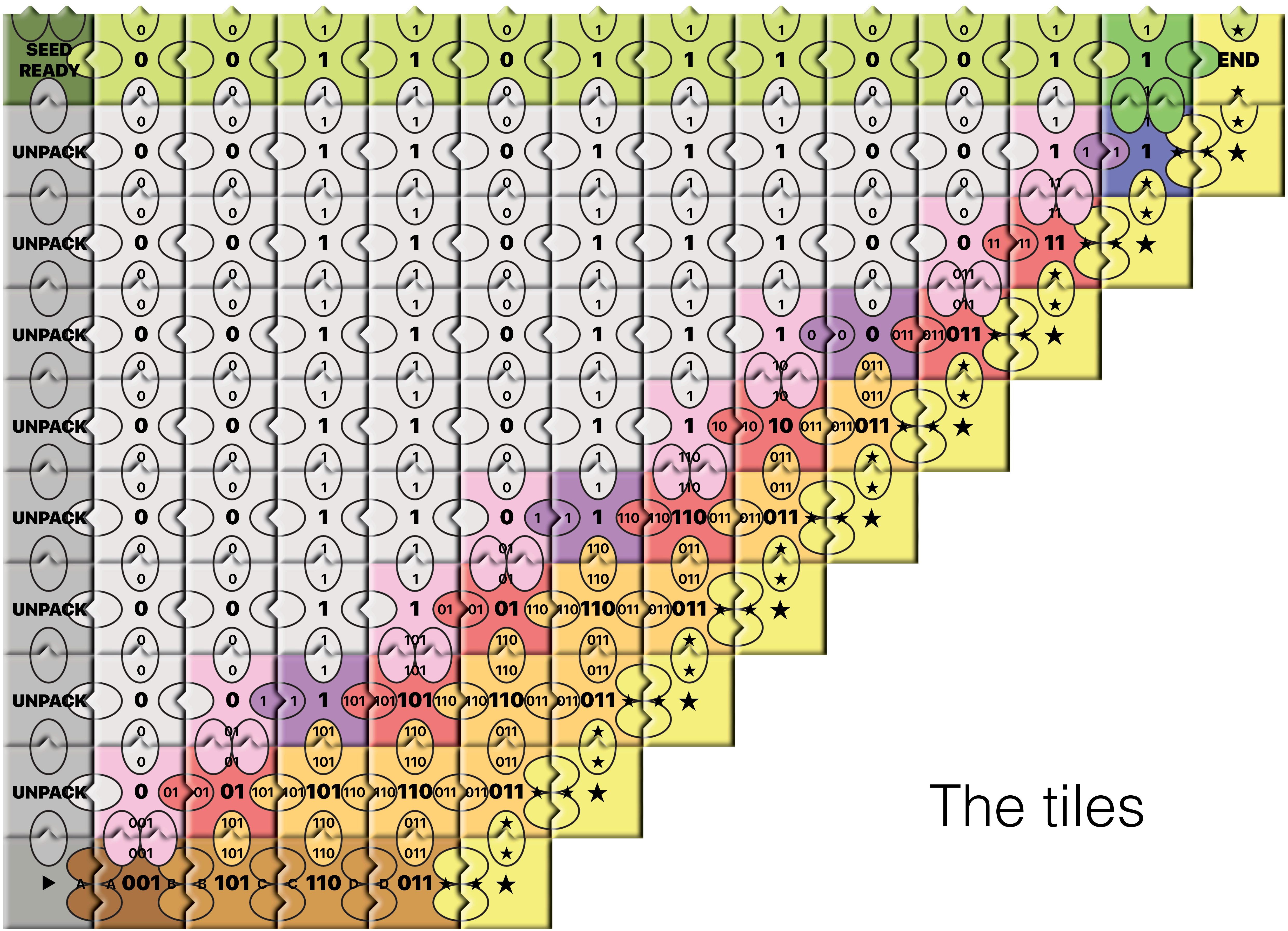


Determine the various tile types

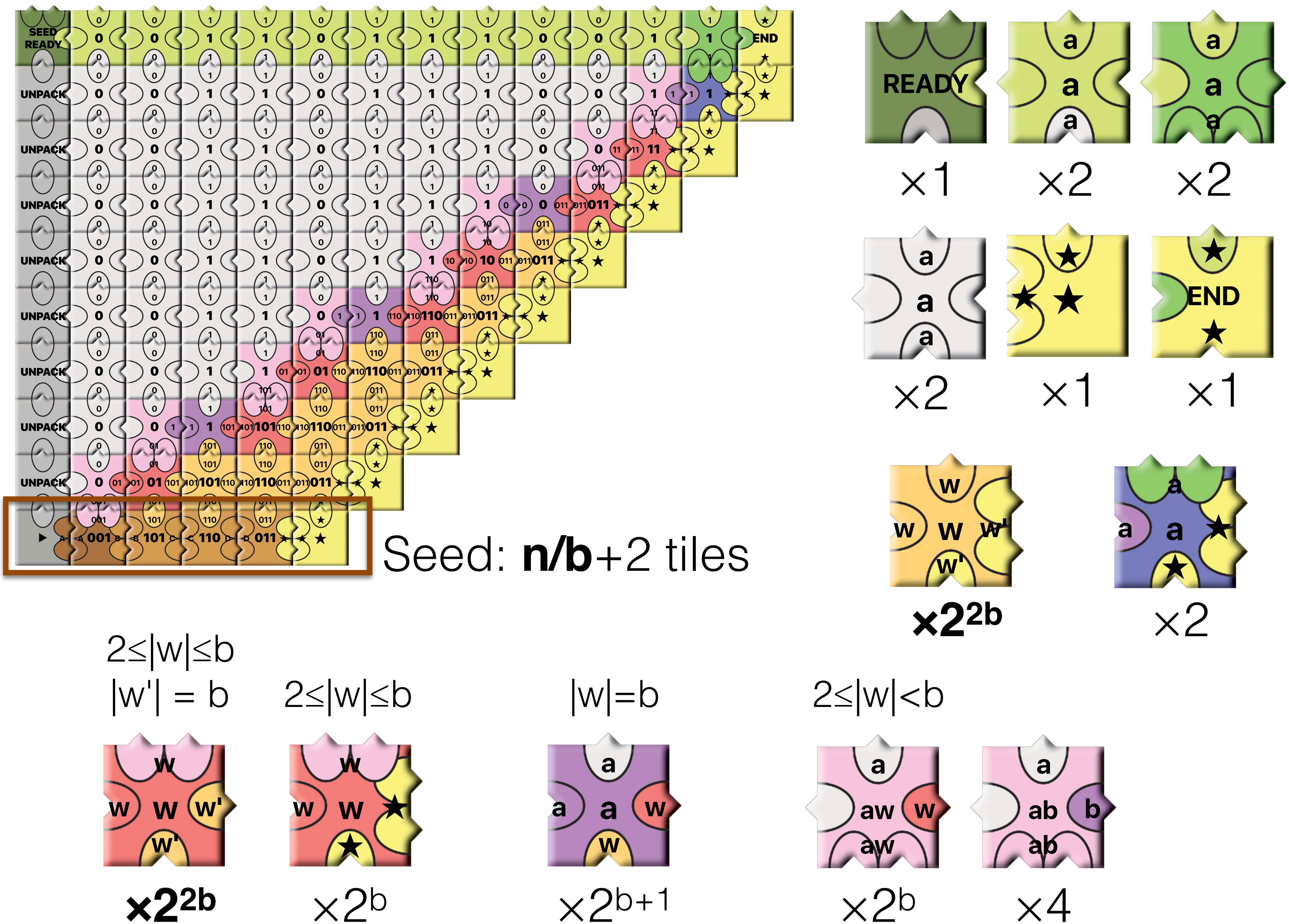




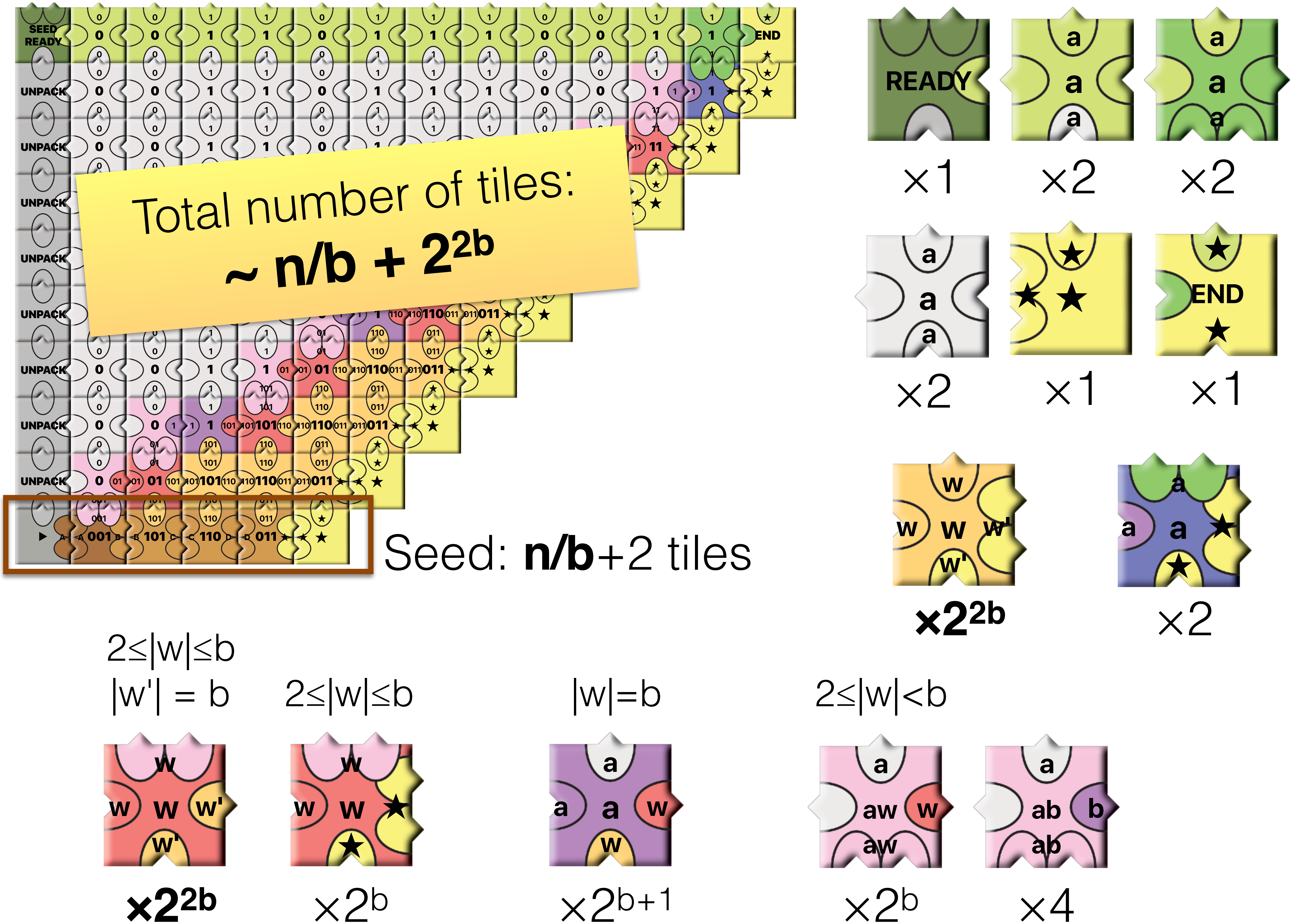
Add tiles at the boundary for continuation

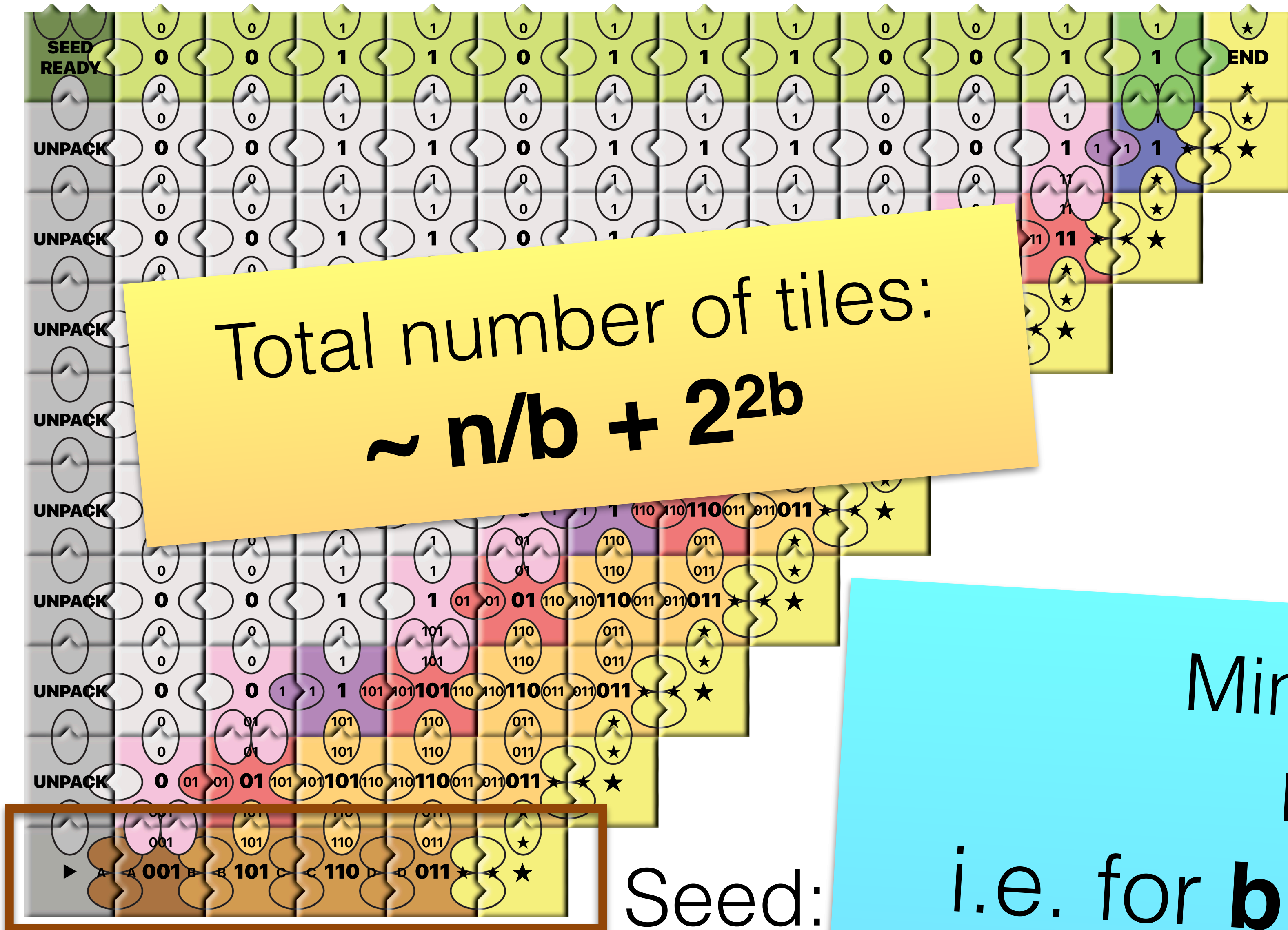


The tiles

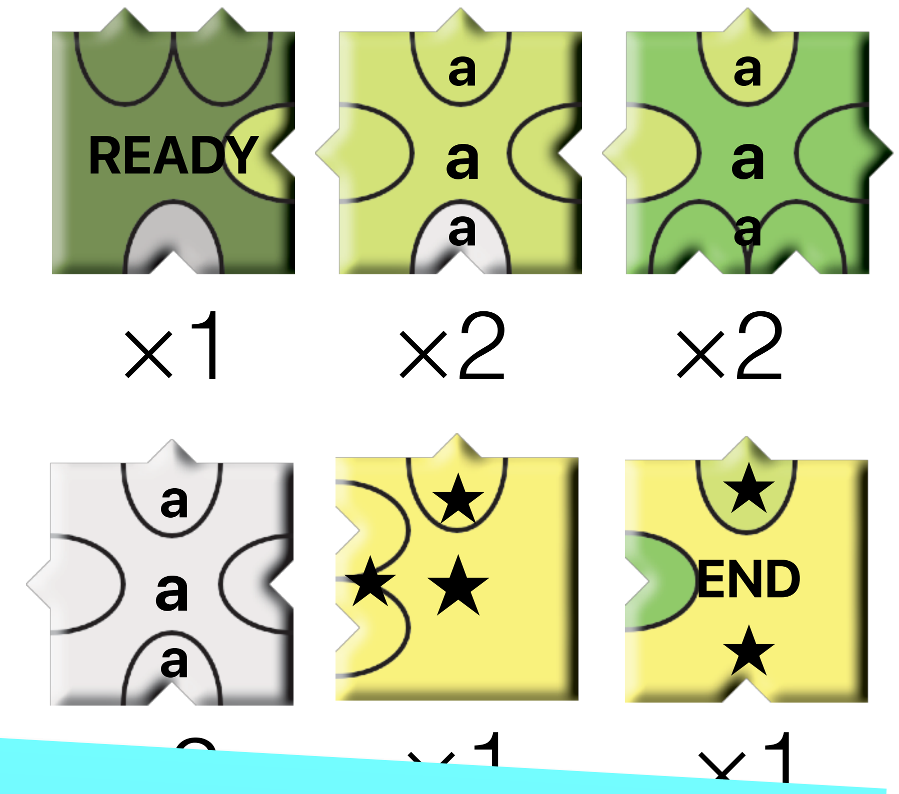


SEED READY
UNPACK
END
READY
×1
×2
×2
×2
×1
×1
Seed: **n/b**+2 tiles
×2^{2b}
×2
2≤|w|≤b
|w'| = b
2≤|w|≤b
|w|=b
2≤|w|<b
×2^{2b}
×2^b
×2^{b+1}
×2^b
×4

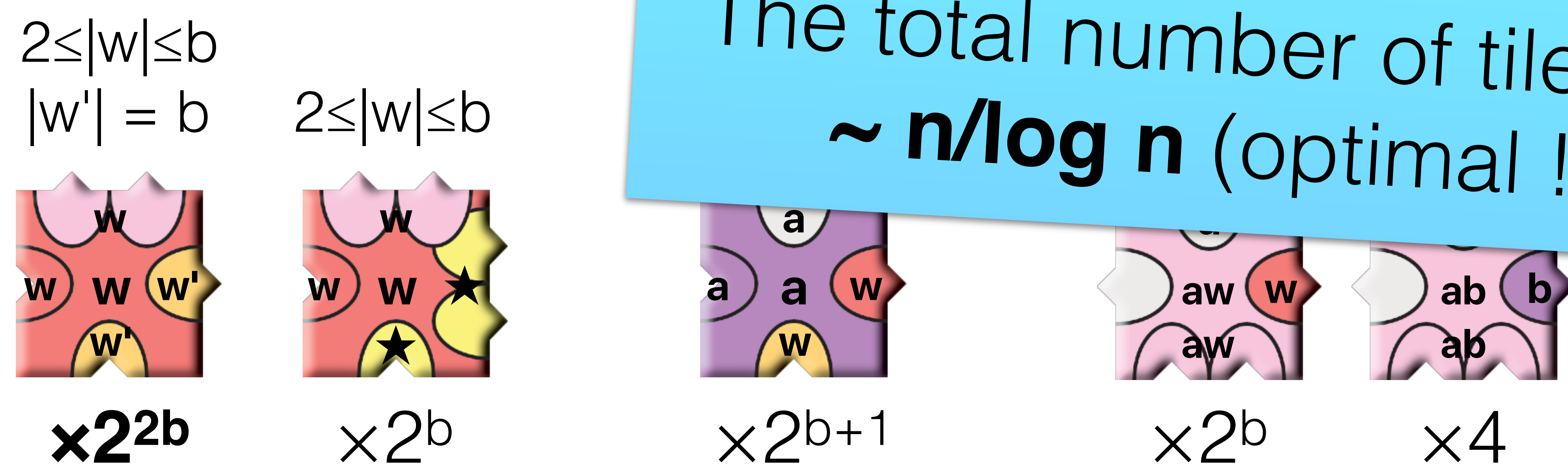

Total number of tiles:
~ n/b + 2^2b
Seed: n/b+2 tiles
SEED READY
UNPACK
END
READY
×1
×2
×2
×2
×1
×1
×2^2b
×2
2≤|w|≤b
|w'| = b
2≤|w|≤b
|w|=b
2≤|w|<b
×2^2b
×2^b
×2^b+1
×2^b
×4



Total number of tiles:
~ n/b + 2^{2b}



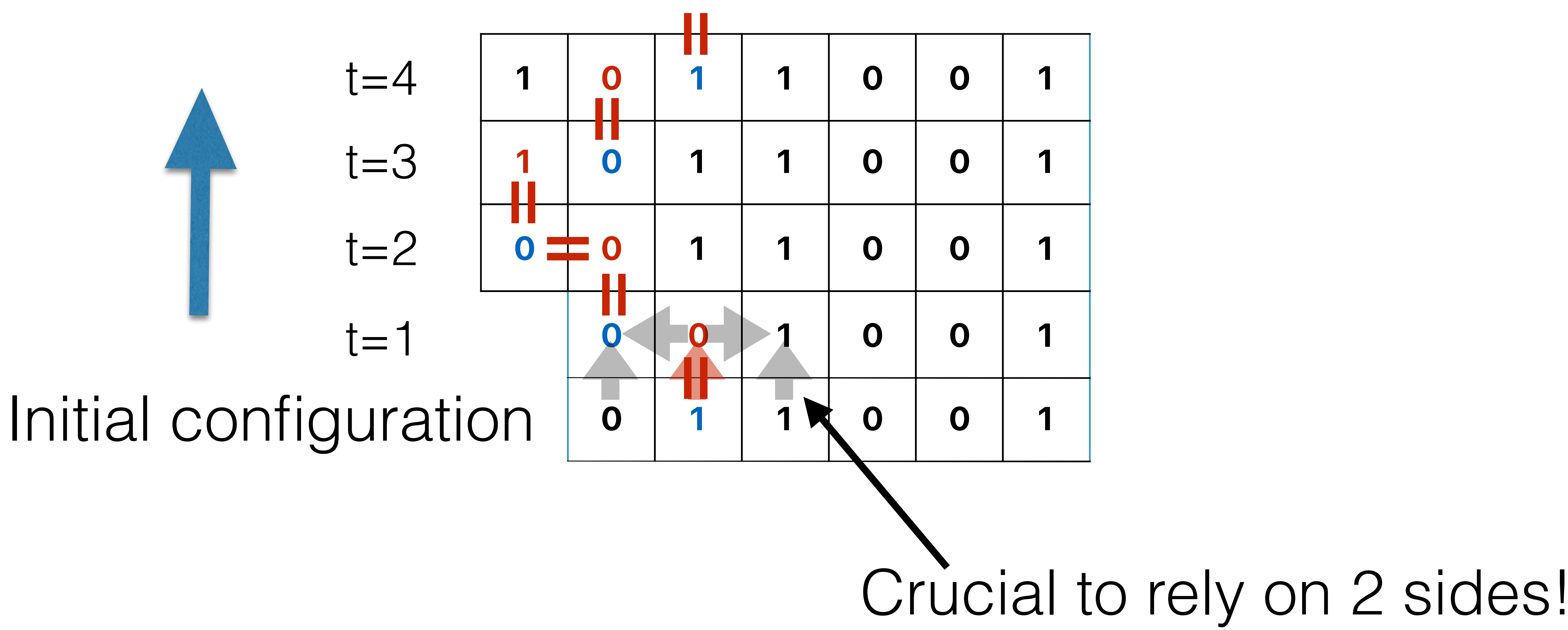
Minimized when
$n/b = 2^{2b}$
i.e. for **b ~ log(n/log n)/2**
The total number of tiles is:
~ n/log n (optimal !)



What about $T^{\circ}=1$?

Does algorithmic self-assembly simulate Turing machine at **T°1** in 2D?

- How to read and propagate the position of the head?



Does algorithmic self-assembly simulate Turing machine at **T°1** in 2D?

- **Theorem.** [Meunier, Regnault, 2015]
  Any deterministic T°1 tile set can be pumped outside a fixed radius in 2D
- **Corollary.** No T°1 tilesystem is Turing complete in 2D

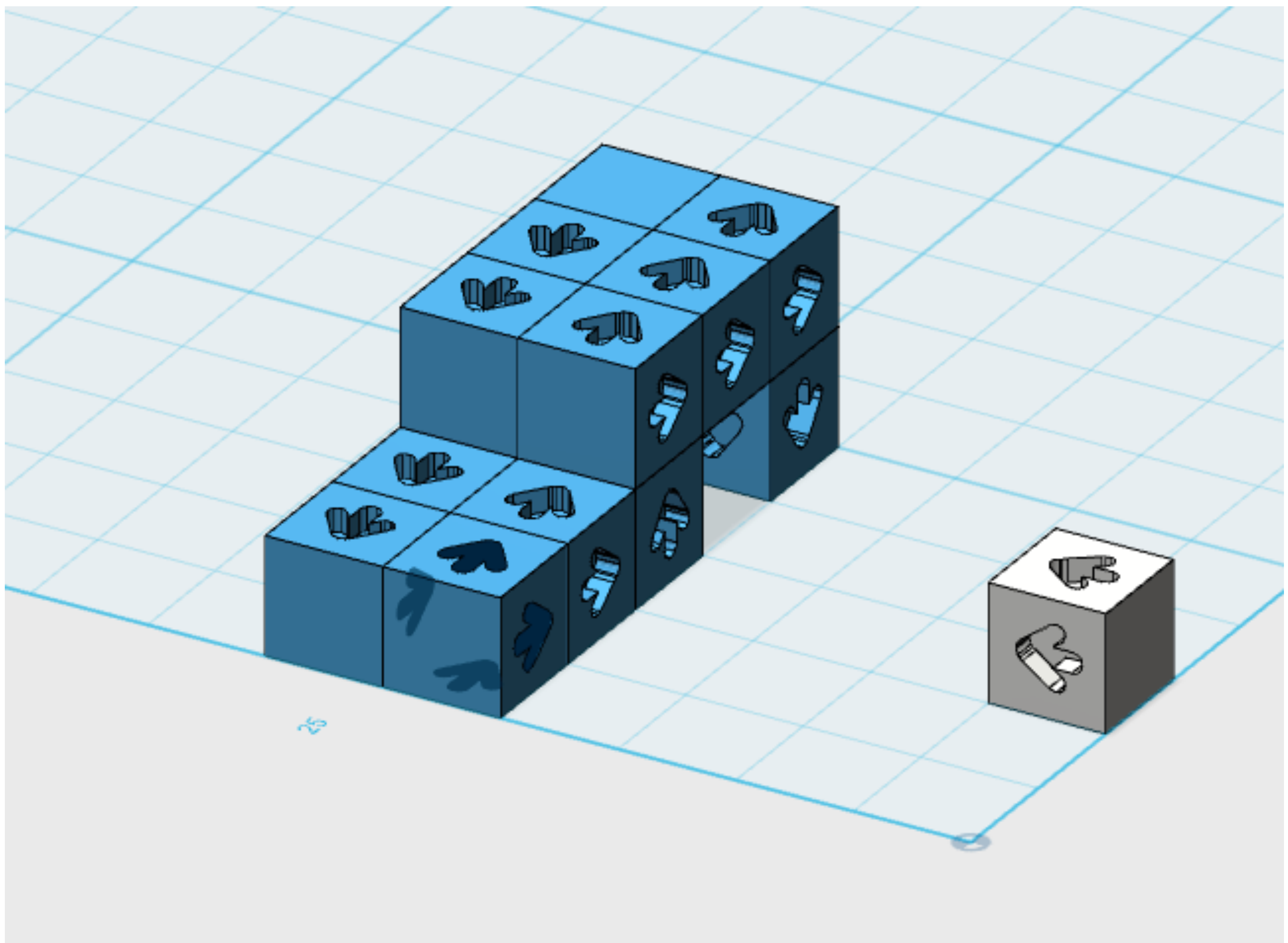
Algorithmic self-assembly simulates any Turing machine at T°1 in 3D

- **Theorem.** *[Cook, Fu, Schweller, 2011]*
  There is a T°1 tile set that simulates any Turing machine with 2-layers in 3D.

- *Key beautiful idea:* Blocking probe crystal

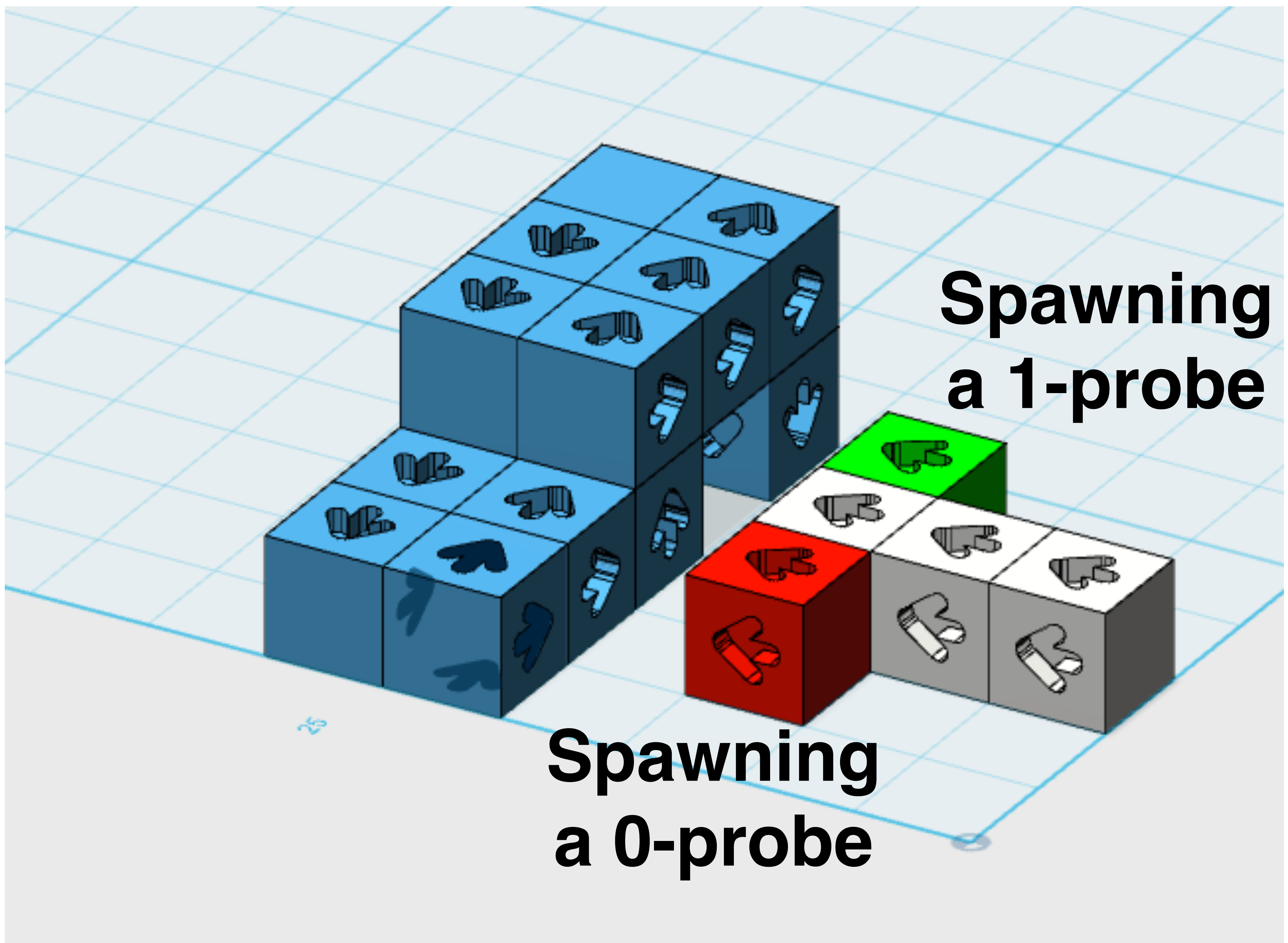
Algorithmic self-assembly simulates any Turing machine at T°1 in 3D

- *Key beautiful idea:* Blocking probe crystal



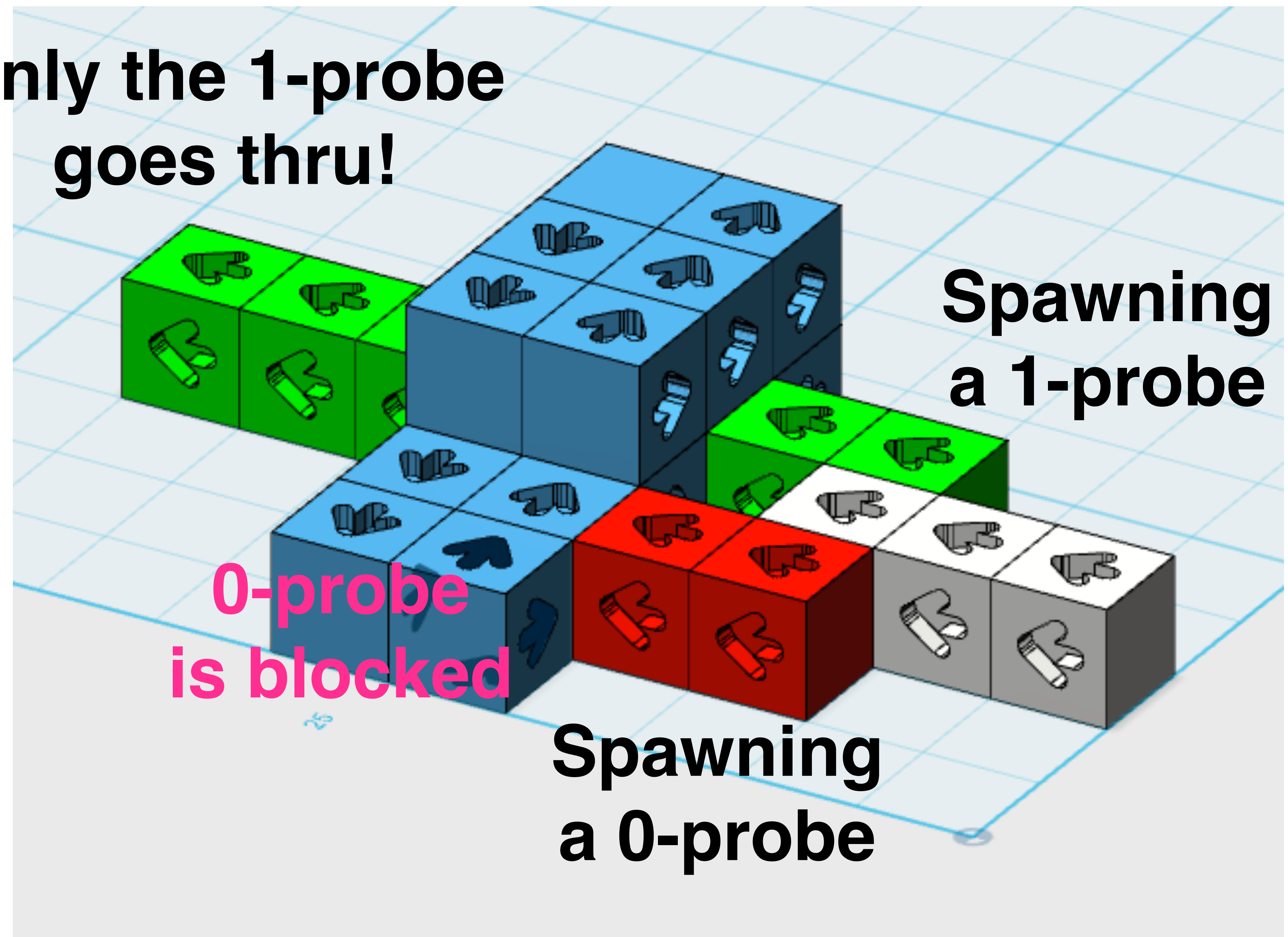
Algorithmic self-assembly simulates any Turing machine at T°1 in 3D

- *Key beautiful idea:* Blocking probe crystal



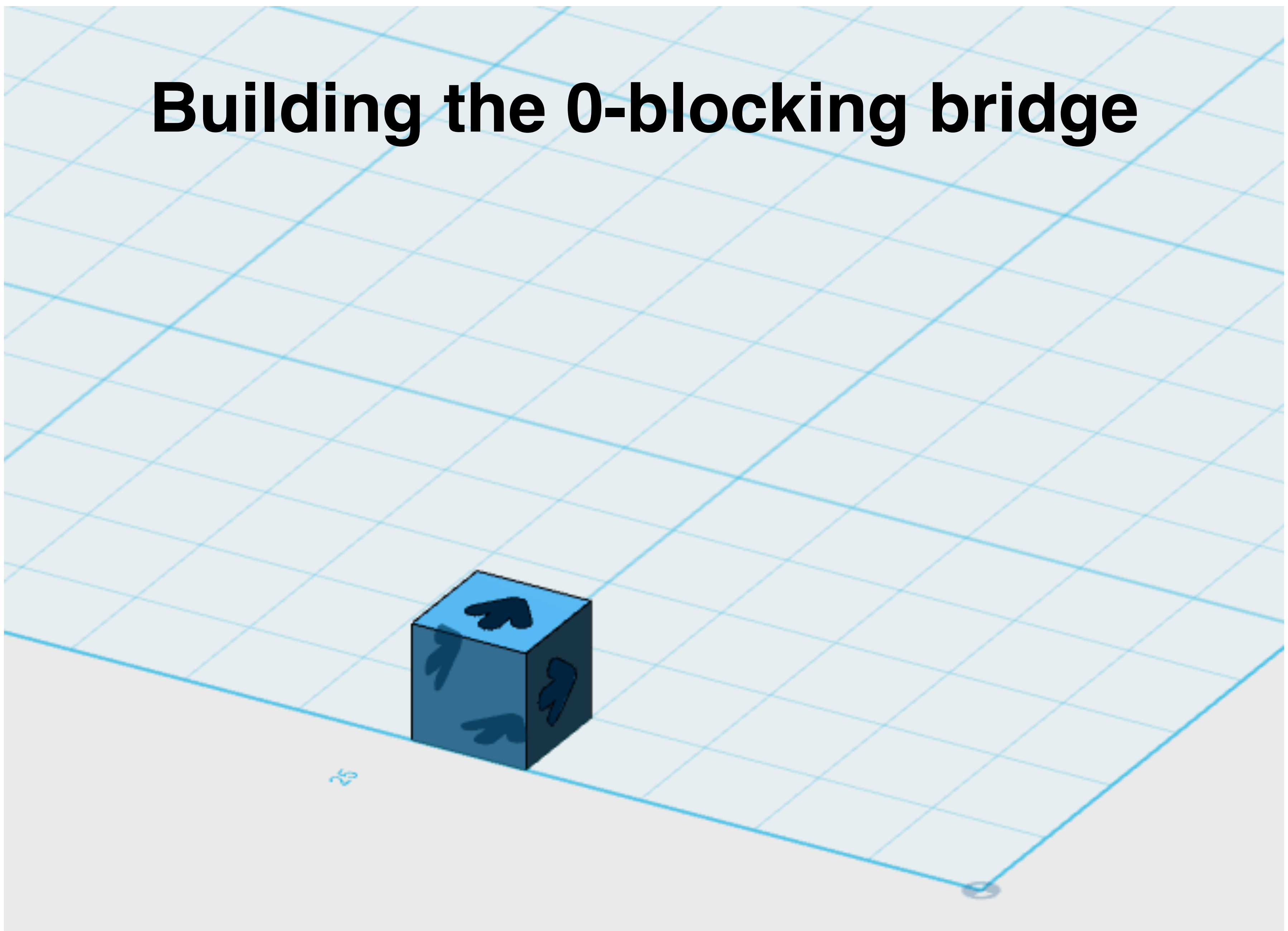
Algorithmic self-assembly simulates any Turing machine at T°1 in 3D

- *Key beautiful idea:* Blocking probe crystal



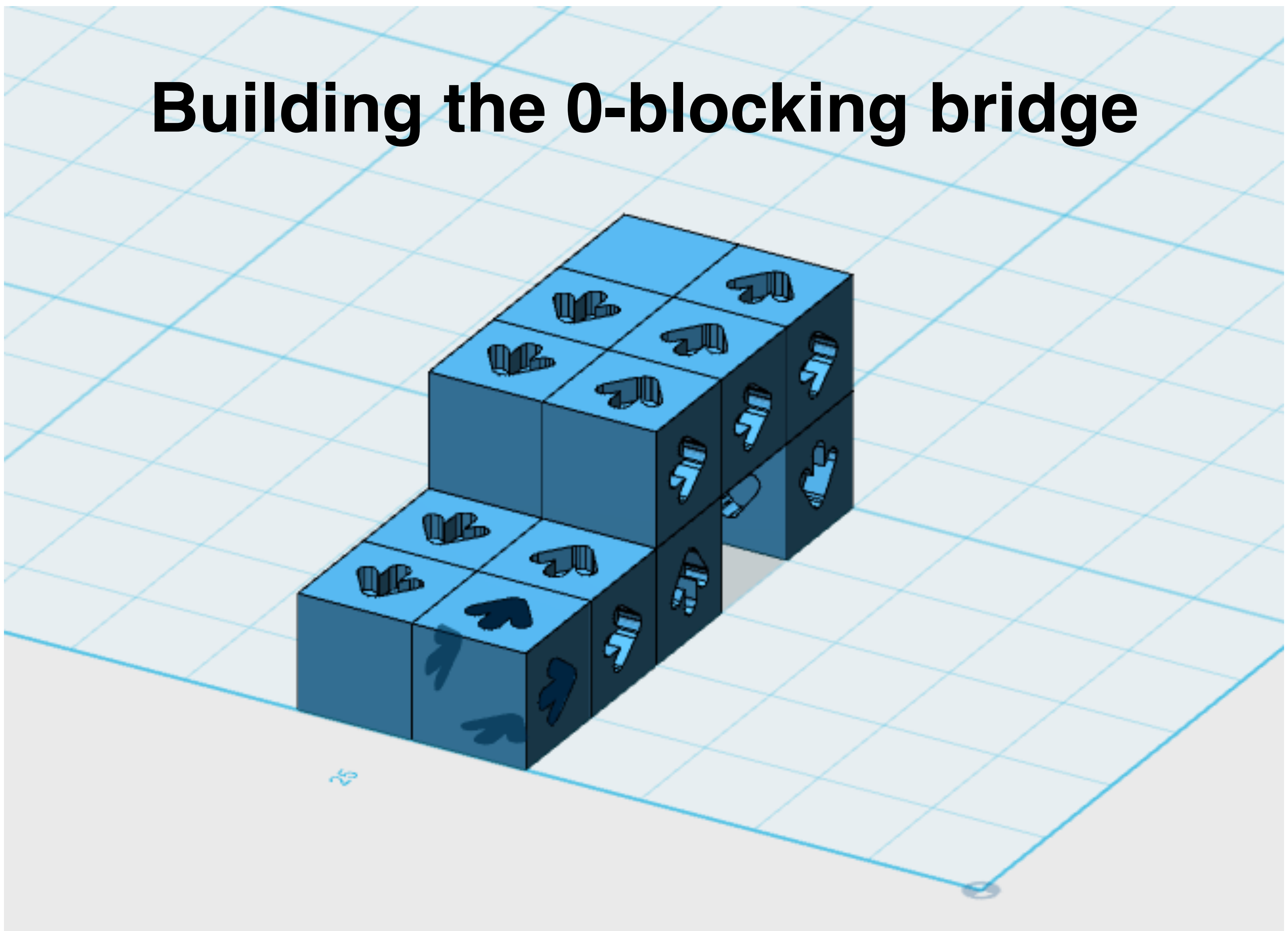
Algorithmic self-assembly simulates any Turing machine at T°1 in 3D

- *Key beautiful idea:* Blocking probe crystal



Algorithmic self-assembly simulates any Turing machine at T°1 in 3D

- *Key beautiful idea:* Blocking probe crystal



Algorithmic self-assembly simulates any Turing machine at T°1 in 3D

- *Key beautiful idea:* Blocking probe crystal

We can thus read and write 0 and 1 on the "next tape" !

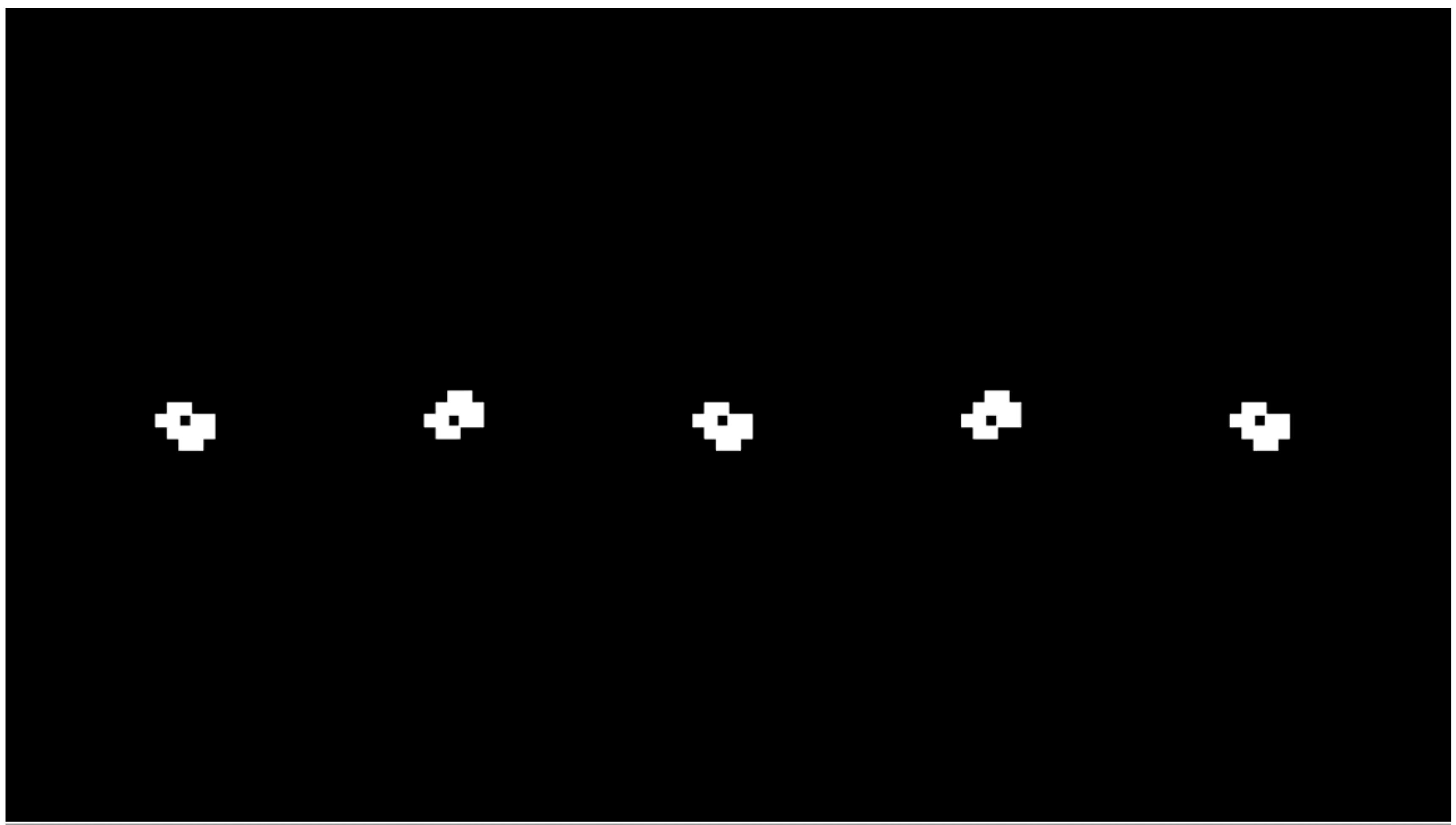
Crystals are Turing complete !!!

Universal Tileset at T°2

A universal tile set to build any (assemblable) shape

Is there a universal tileset at T°=2 ?

- Rescaling : *intrinsic simulation*

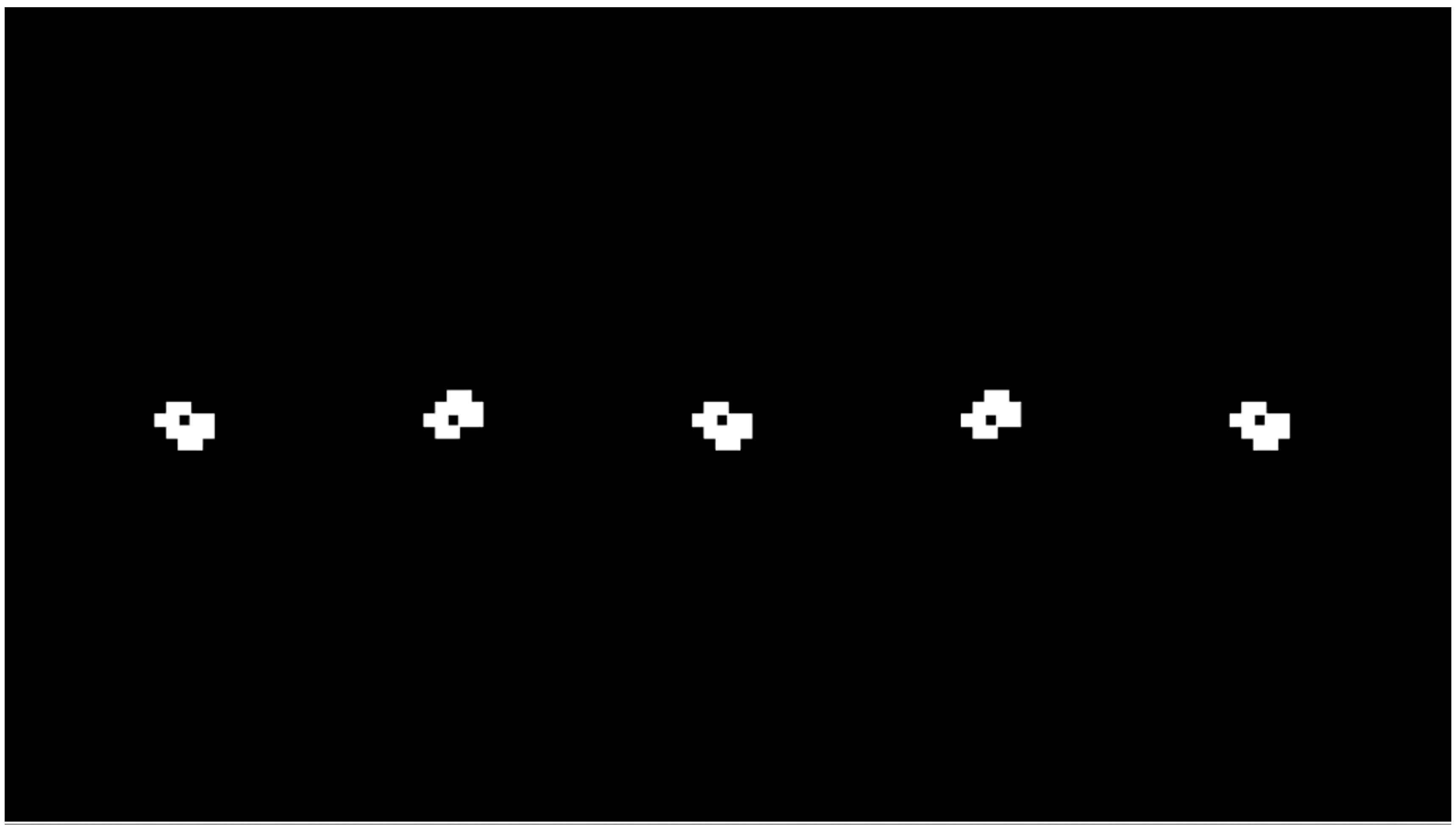


Brice Due 2006

The Game of Life self-simulating itself intrinsically:
Smaller cells simulate macro-cells

Is there a universal tileset at T°=2 ?

- Rescaling : *intrinsic simulation*

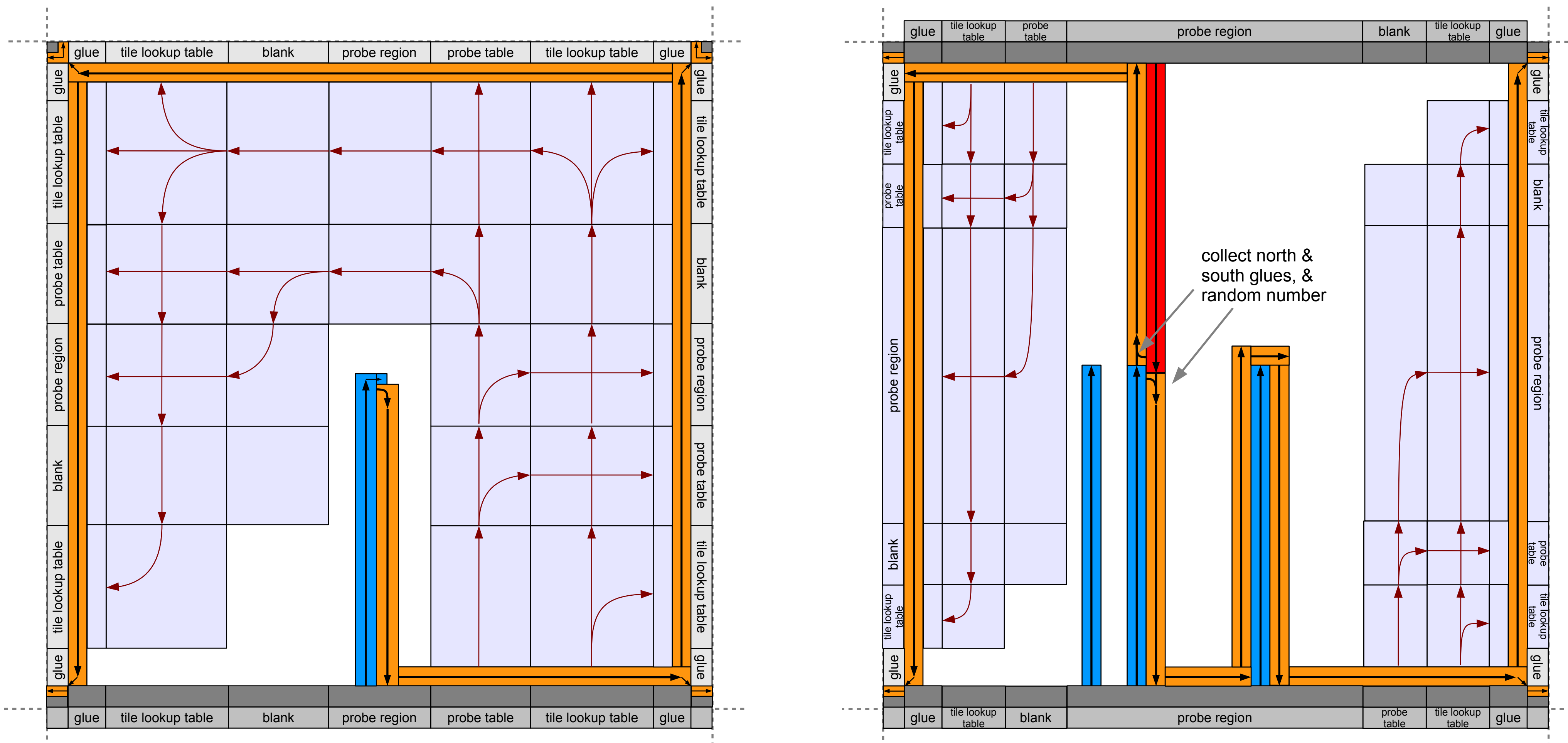


Brice Due 2006

The **Game of Life** self-simulating itself intrinsically:
Smaller cells simulate macro-cells

Is there a universal tileset at T°=2 ?

- Rescaling : *intrinsic simulation*

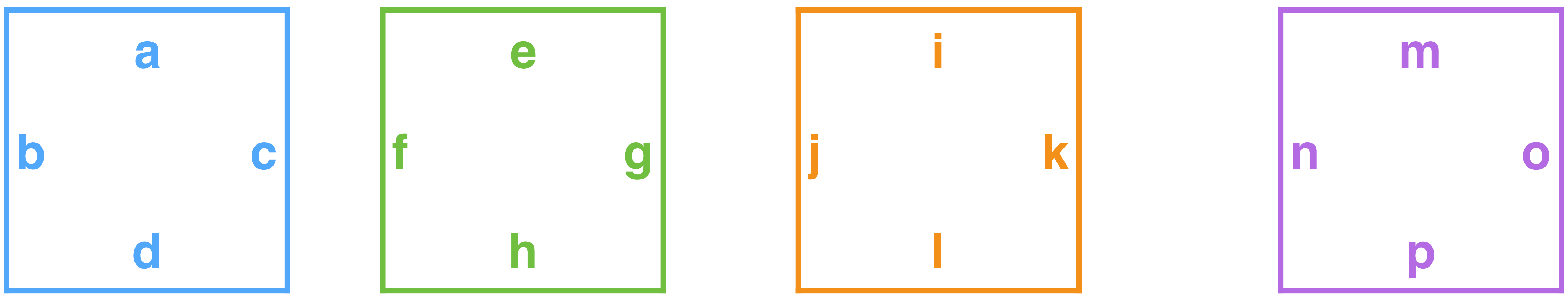


Doty Lutz Patitz Schweller Summers Woods 2012

Examples of macro-tiles

How many tiles do we need ?

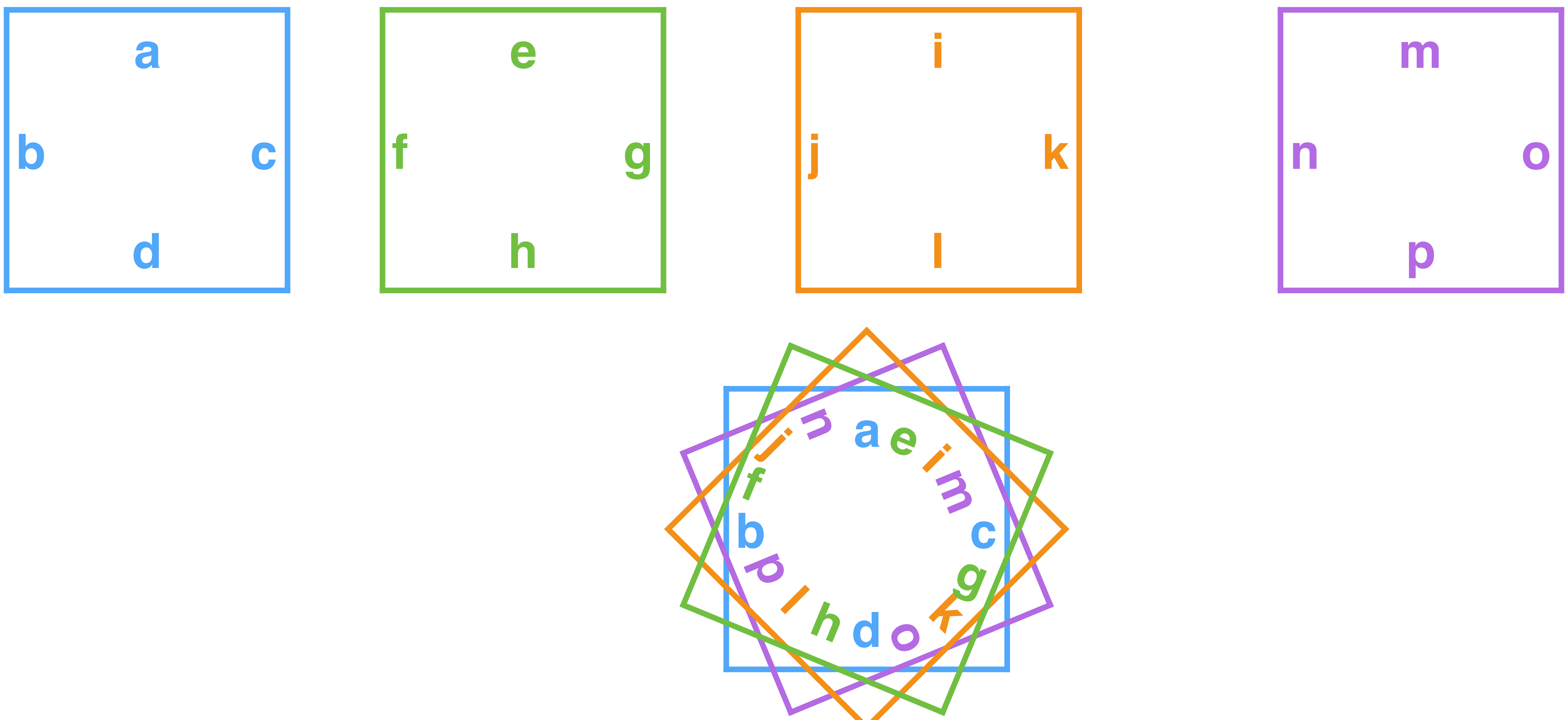
- Just ONE with rotation!… What?!?… But a *polygonal* one



Demaine Demaine Fekete Patitz Schweller Winslow Woods 2012

How many tiles do we need ?

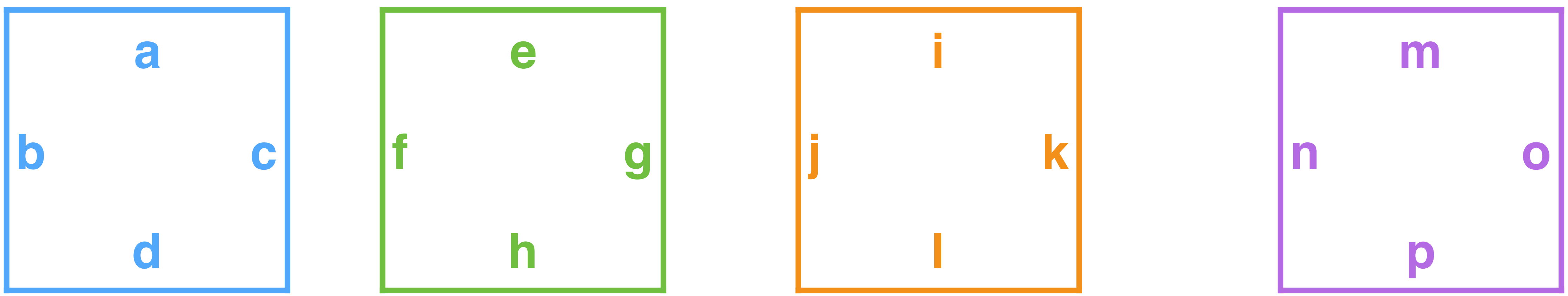
- Just ONE with rotation!… What?!?… But a *polygonal* one

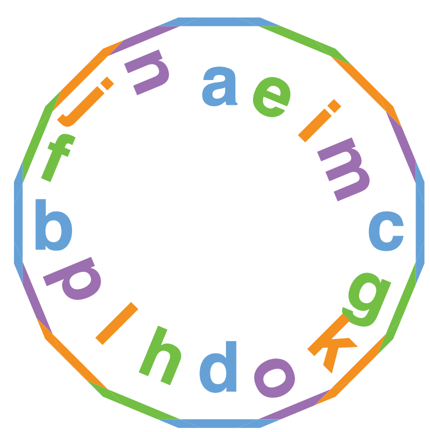


Demaine Demaine Fekete Patitz Schweller Winslow Woods 2012

How many tiles do we need ?

- Just ONE with rotation!… What?!?… But a *polygonal* one



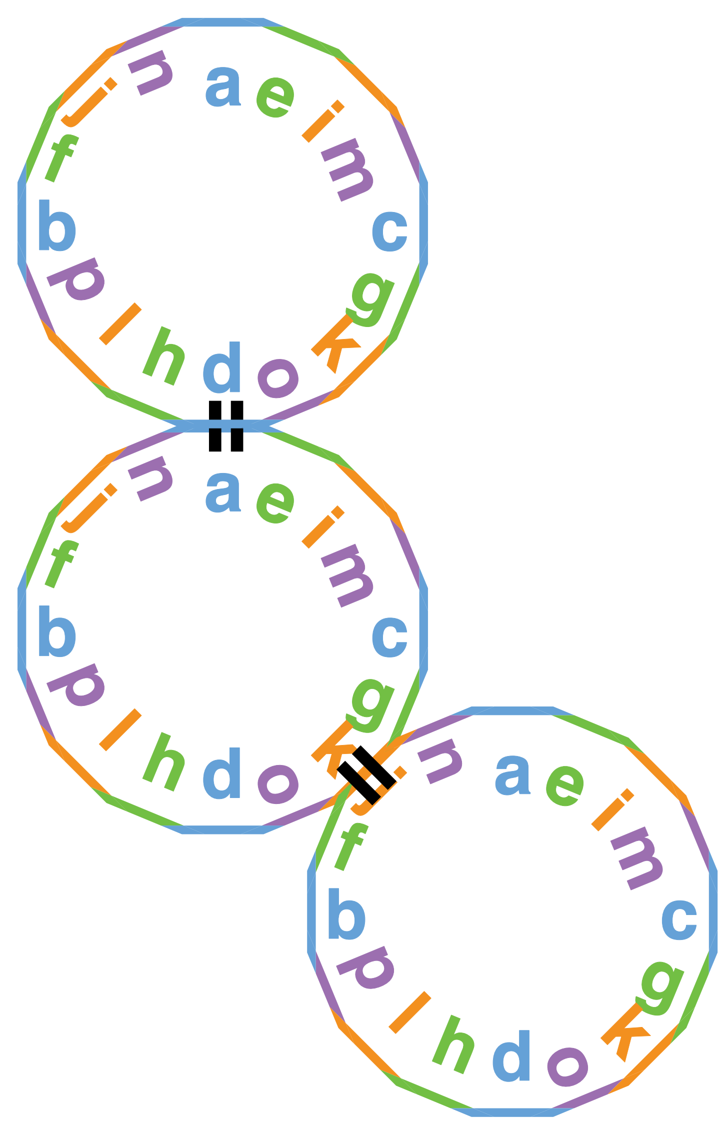


Demaine Demaine Fekete Patitz Schweller Winslow Woods 2012

How many tiles do we need ?

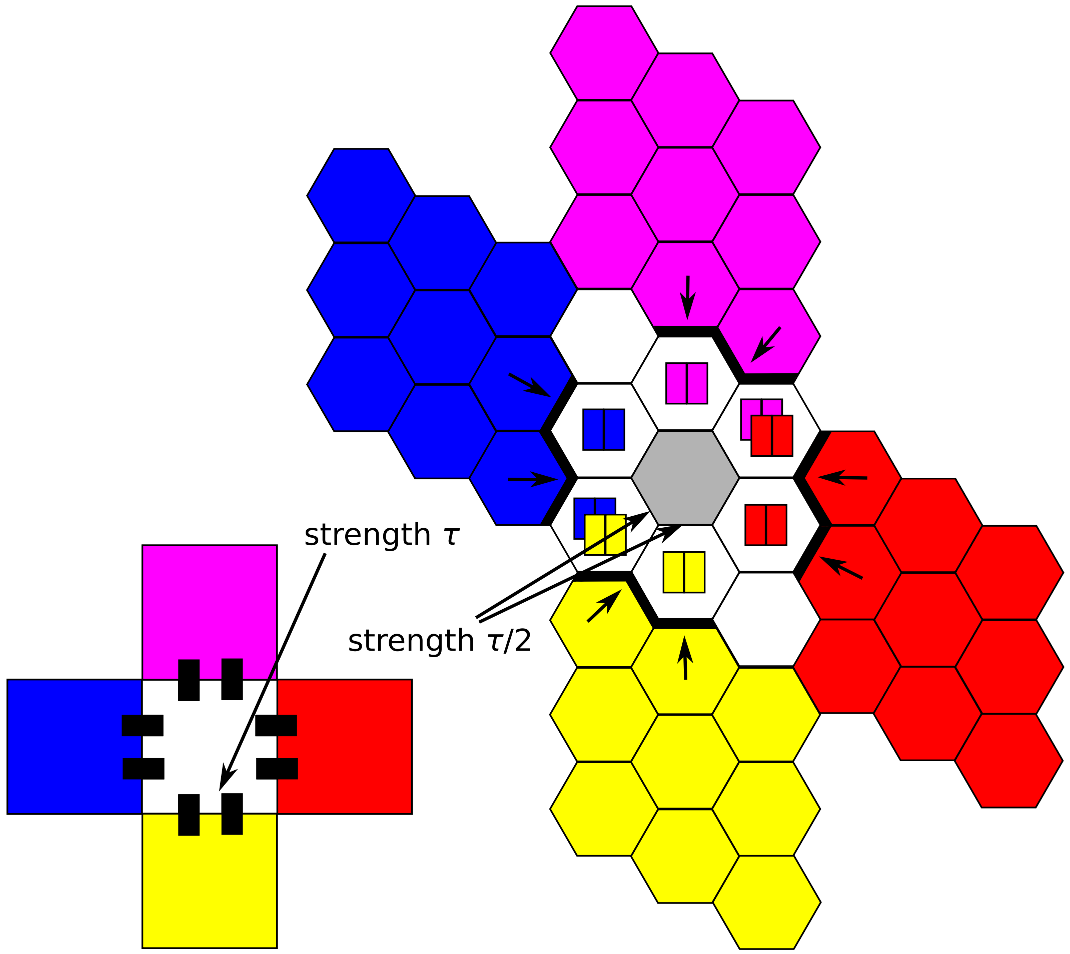
- Just ONE with rotation!… What?!?… But a *polygonal* one

Problem with glue of strength 2 !!!

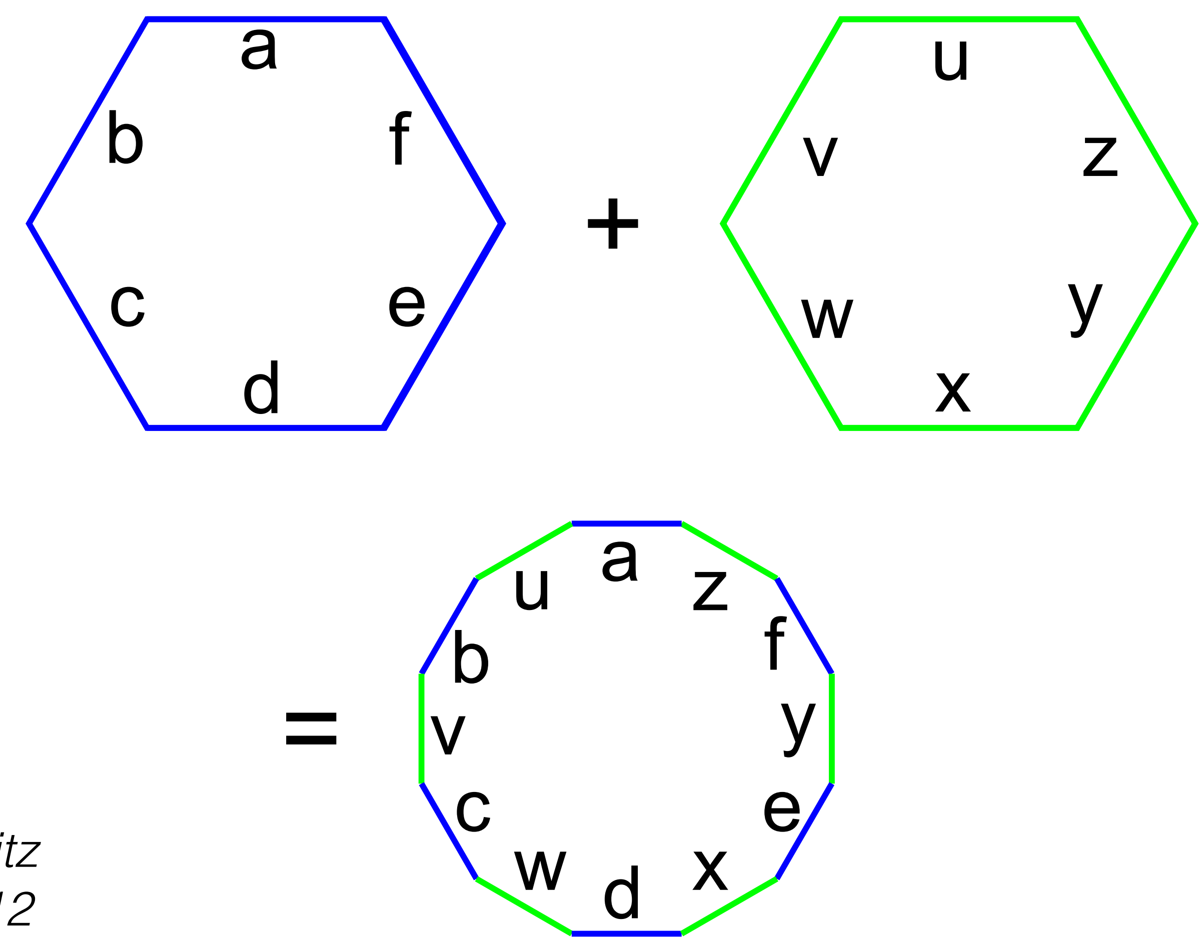


Demaine Demaine Fekete Patitz Schweller Winslow Woods 2012

Encoding strength 2 glues into strength 1 glue in hexagonal tiles

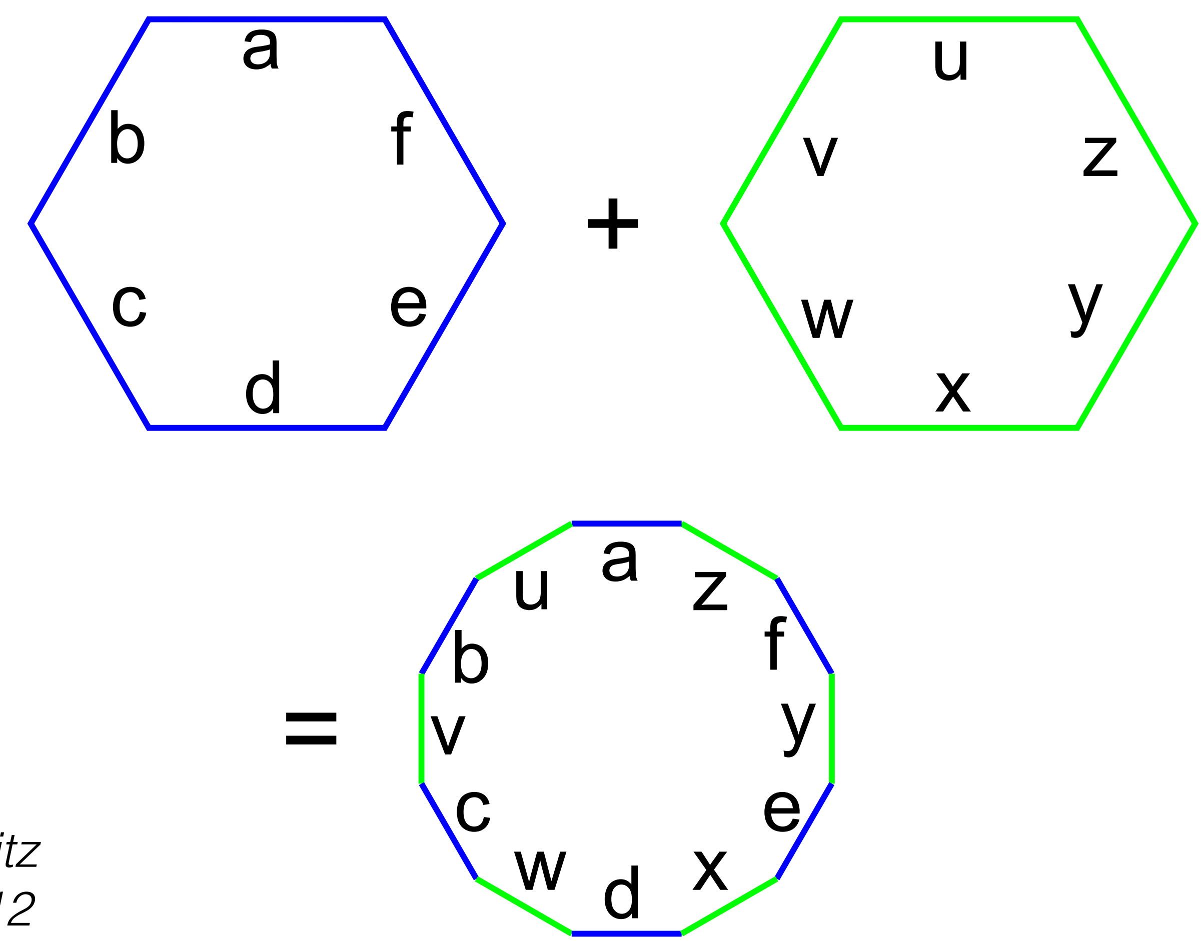


A **single** (polygonal) tile is enough !



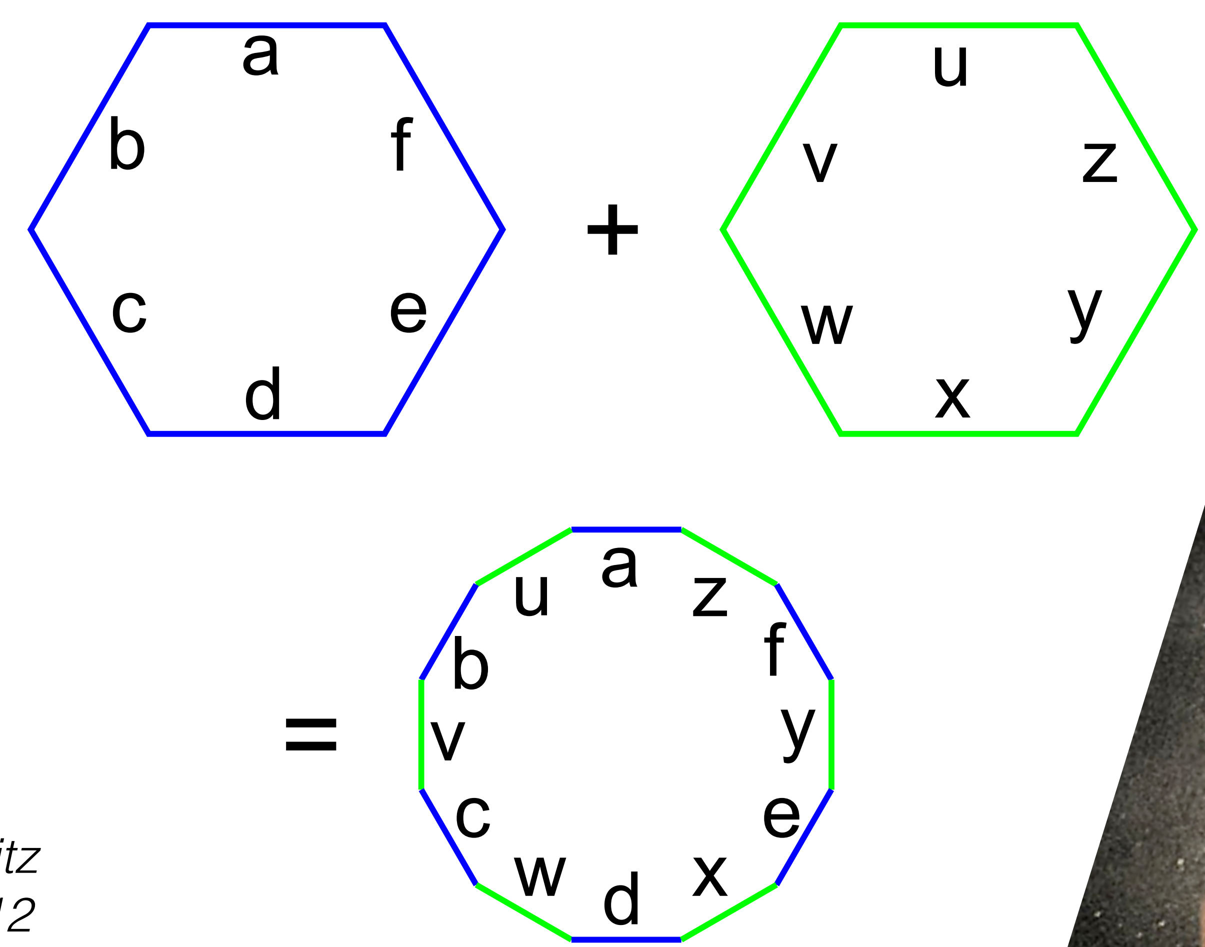
Demaine Demaine Fekete Patitz Schweller Winslow Woods 2012

A **single** (polygonal) tile is enough !



Demaine Demaine Fekete Patitz Schweller Winslow Woods 2012

A **single** (polygonal) tile is enough !



Demaine Demaine Fekete Patitz Schweller Winslow Woods 2012

The magic powder can assemble anything!

One molecule is enough !
(Next week)

Co-transcriptional folding

Joint work with Cody Geary Pierre-Étienne Meunier and Shinnosuke Seki