FRACTALS, REACHABILITY, AND COMPUTATION IN MODELS OF DNA SELF-ASSEMBLY AND CHEMICAL REACTION NETWORKS

by: Ryan Knobel

Advisors

- Dr. Tim Wylie (Chair)
- Dr. Robert Schweller
- Dr. Bin Fu
- Dr. Austin Luchsinger

Coauthors

- Michael Alanis
- Divya Bajaj
- Josh Brunner
- Jose-Luis Castellanos
- Michael Coulombe
- Erik Demaine
- Jenny Diomidova
- Tim Gomez
- Elise Grizzel
- Asher Haun
- Markus Hecher
- Jayson Lynch
- Aiden Massie
- Juan Manuel Perez
- Tom Peters
- Rene Reyes
- Andrew Rodriguez
- Marco Rodriguez
- Adrian Salinas
- Pablo Santos
- Ramiro Santos

Overview

1. Building Fractals in Seeded Tile Automata
2. Reachability in Chemical Reaction Networks
3. Game Complexity

Fractals in Seeded Tile Automata

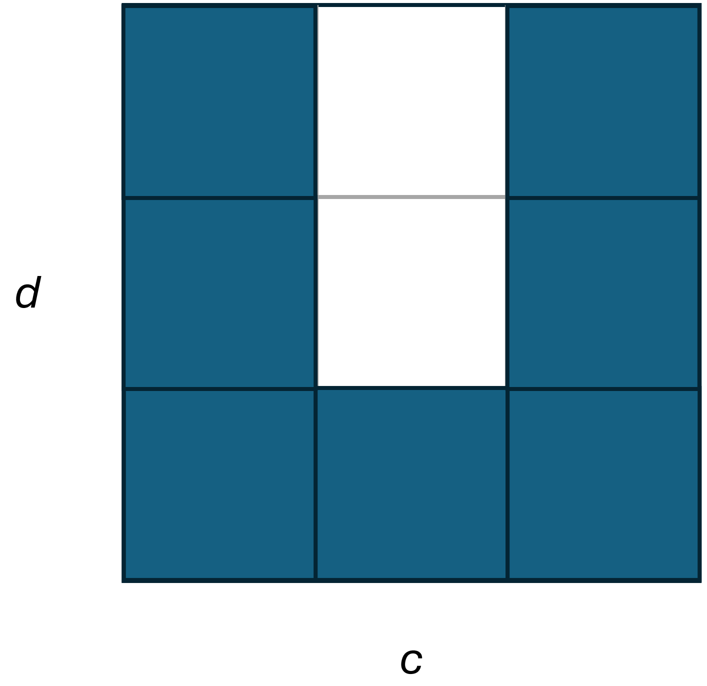
Fractals in Seeded TA

Authors: Asher Haun, **Ryan Knobel,** Adrian Salinas, Ramiro Santos, Robert Schweller, Timothy Wylie

Discrete Self-Similar Fractals

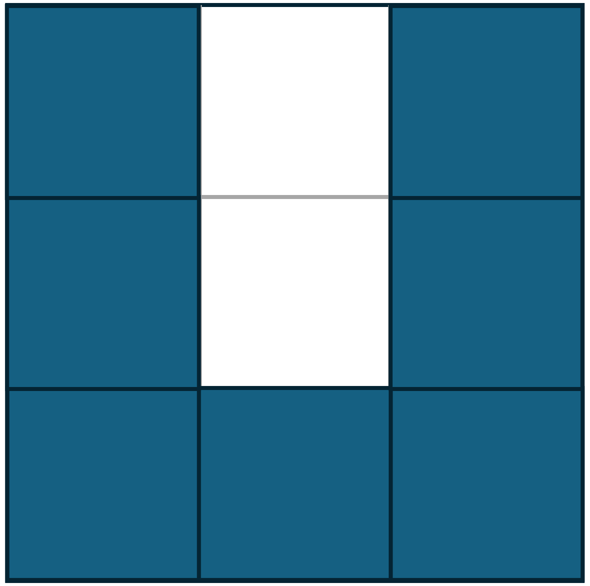
What are they?

Generator



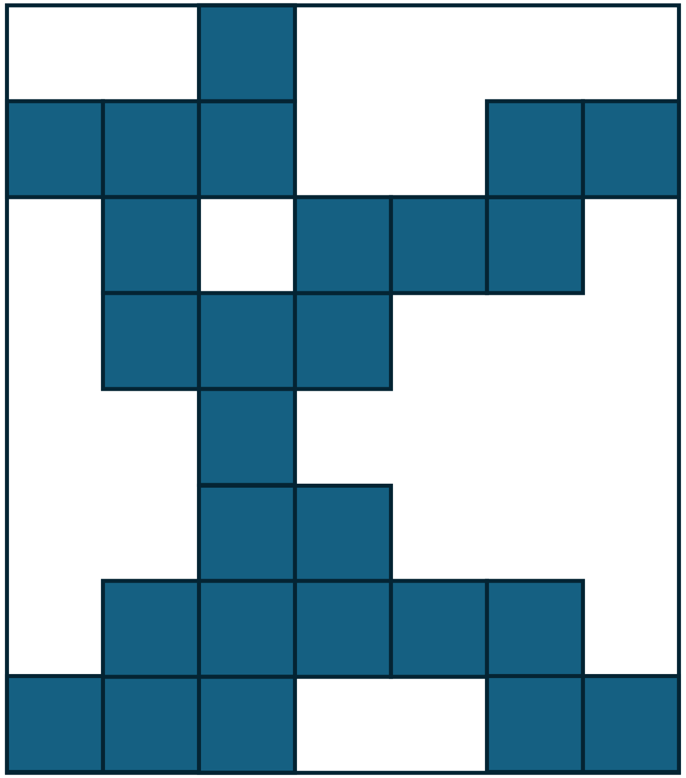
Generator

1. Connected



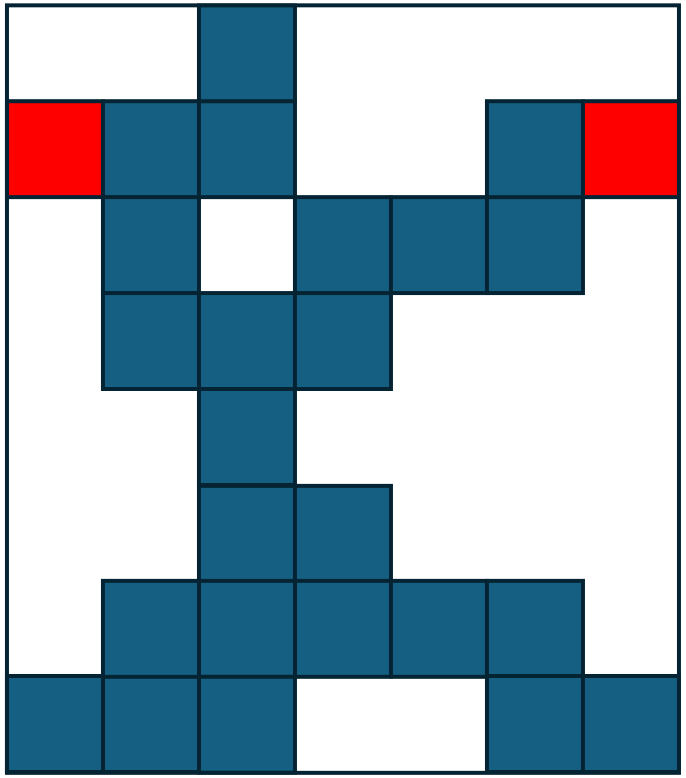
Generator

1. Connected
2. Feasible



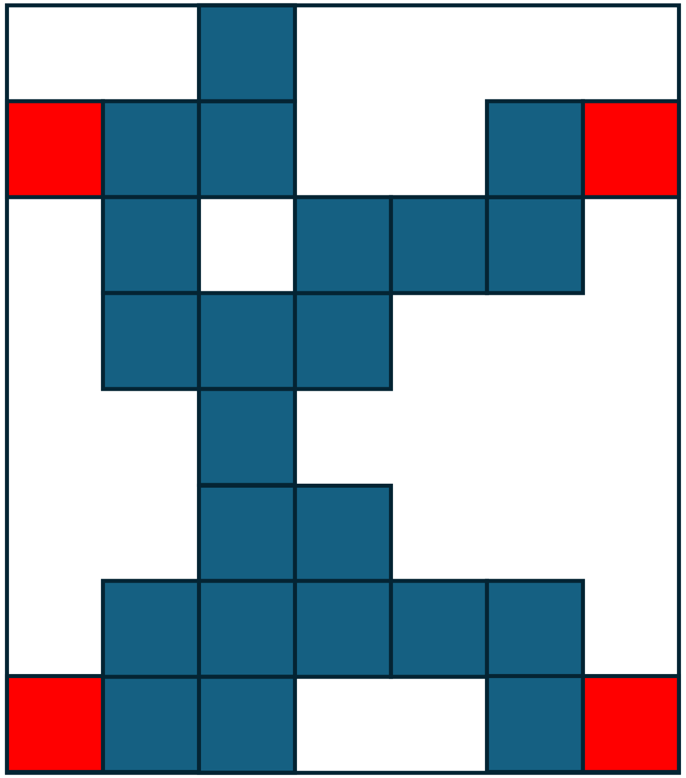
Generator

1. Connected
2. Feasible



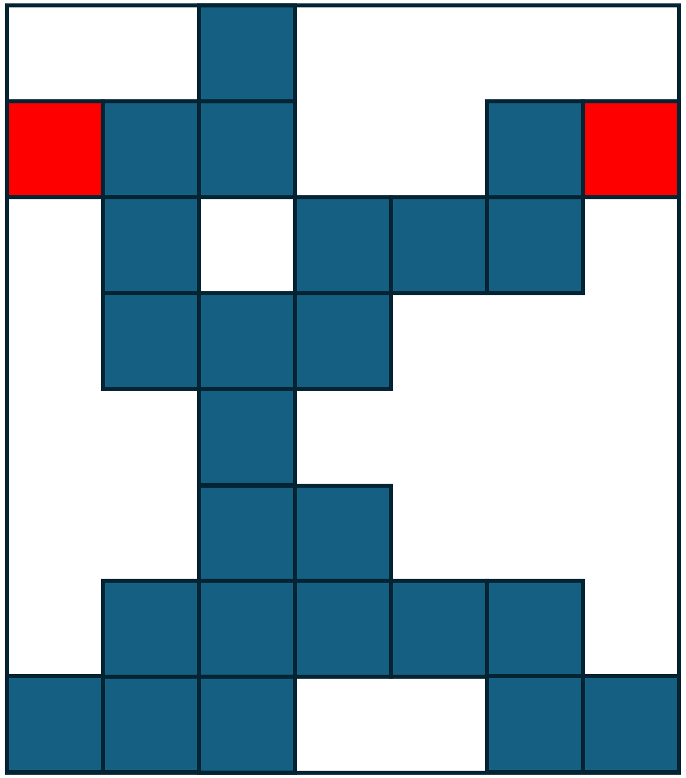
Generator

1. Connected
2. Feasible



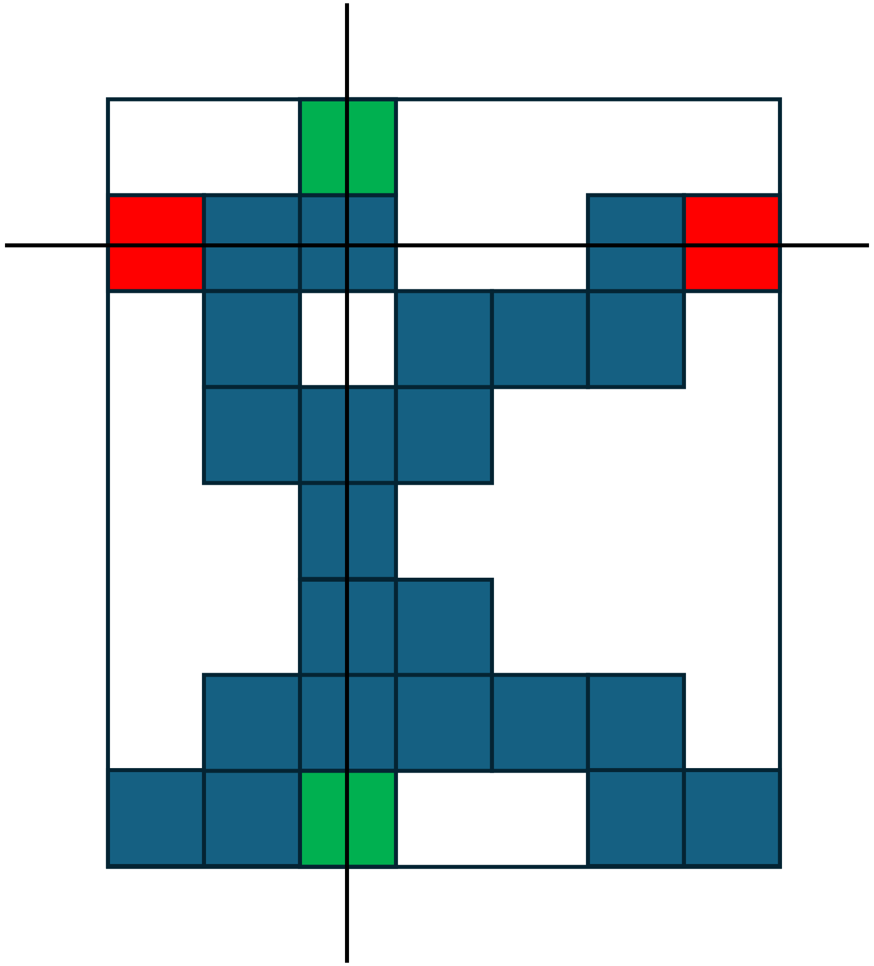
Generator

1. Connected
2. Feasible



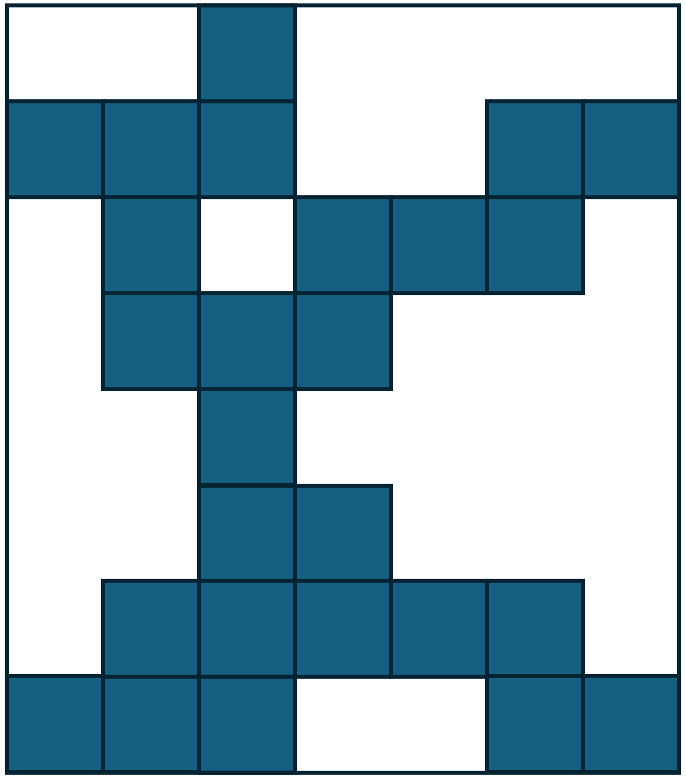
Generator

1. Connected
2. Feasible



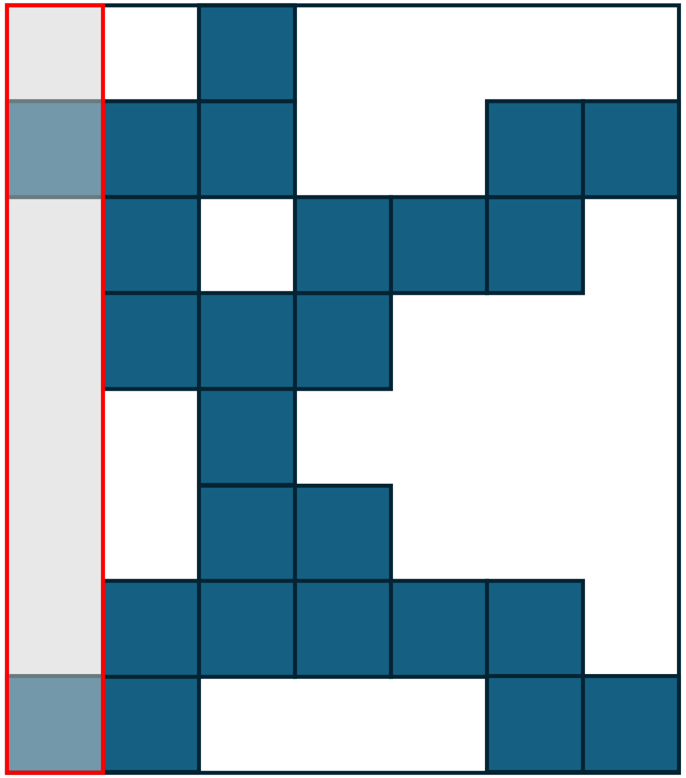
Generator

1. Connected
2. Feasible



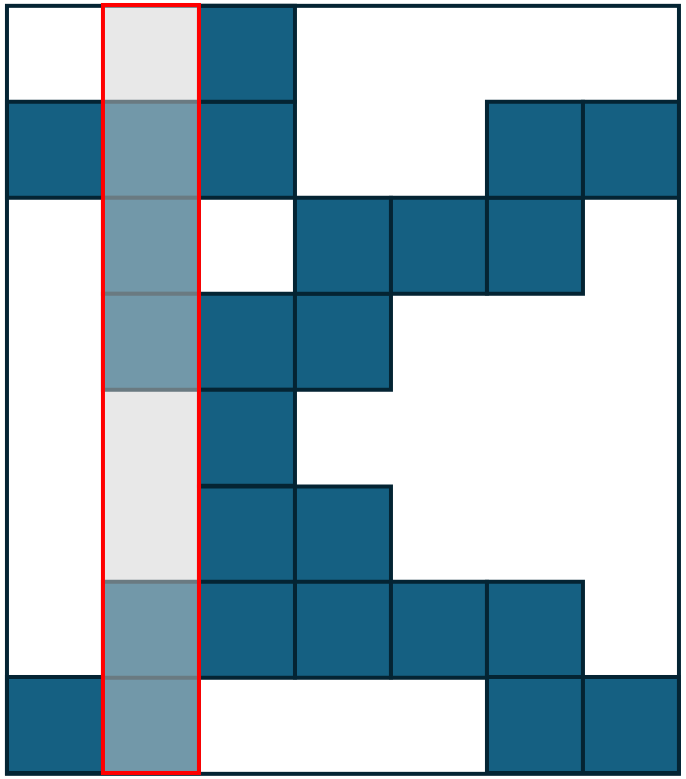
Generator

1. Connected
2. Feasible



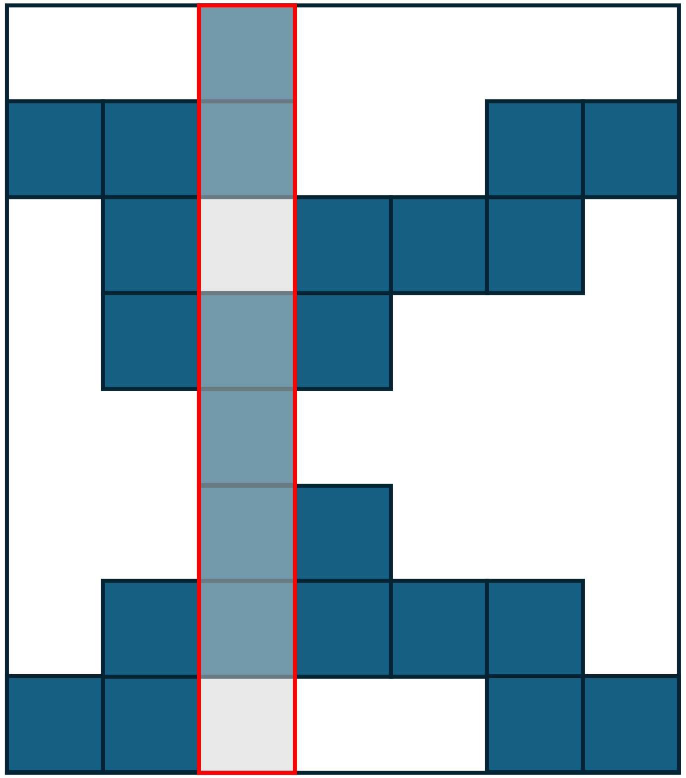
Generator

1. Connected
2. Feasible



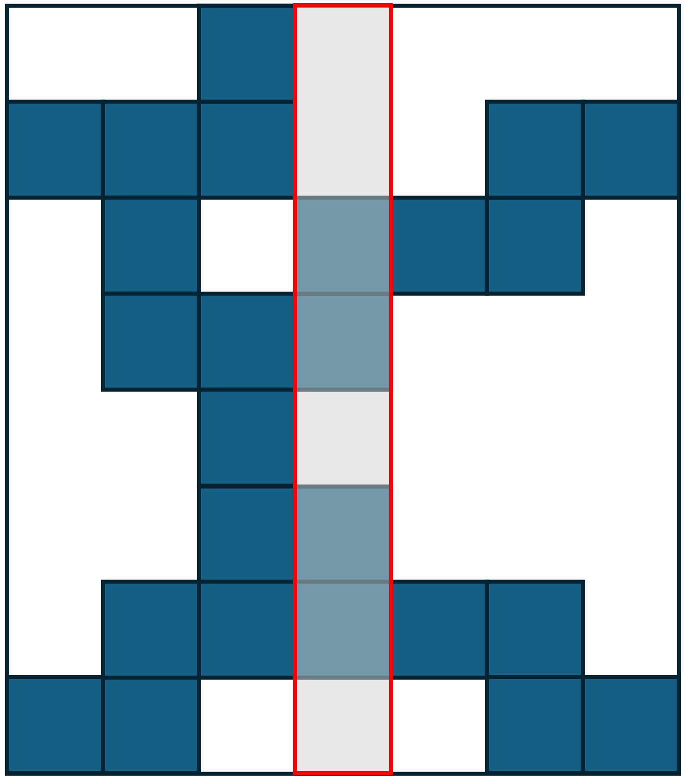
Generator

1. Connected
2. Feasible



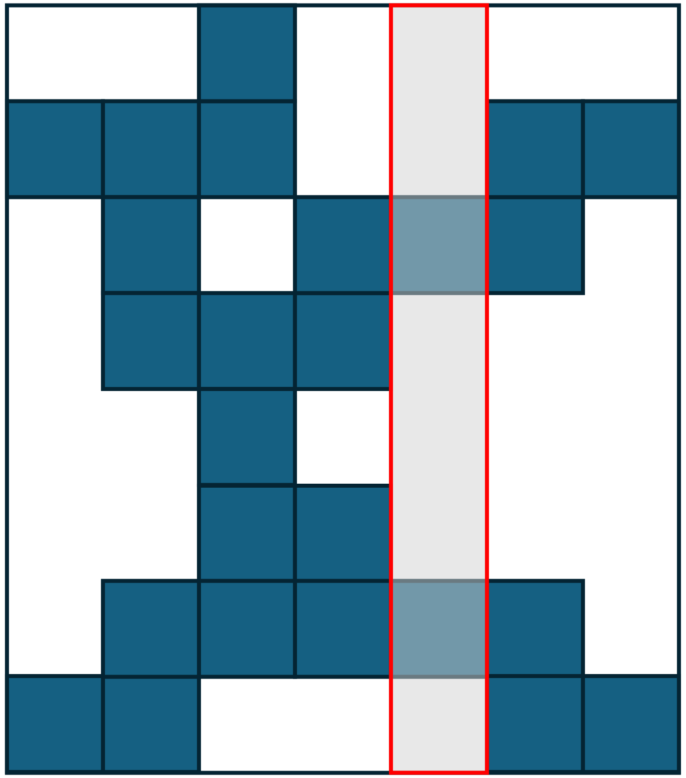
Generator

1. Connected
2. Feasible



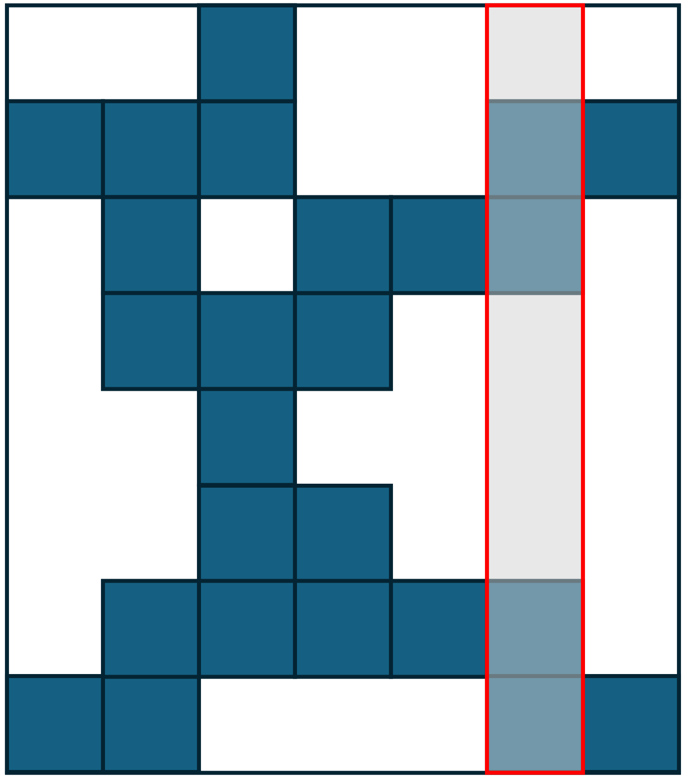
Generator

1. Connected
2. Feasible



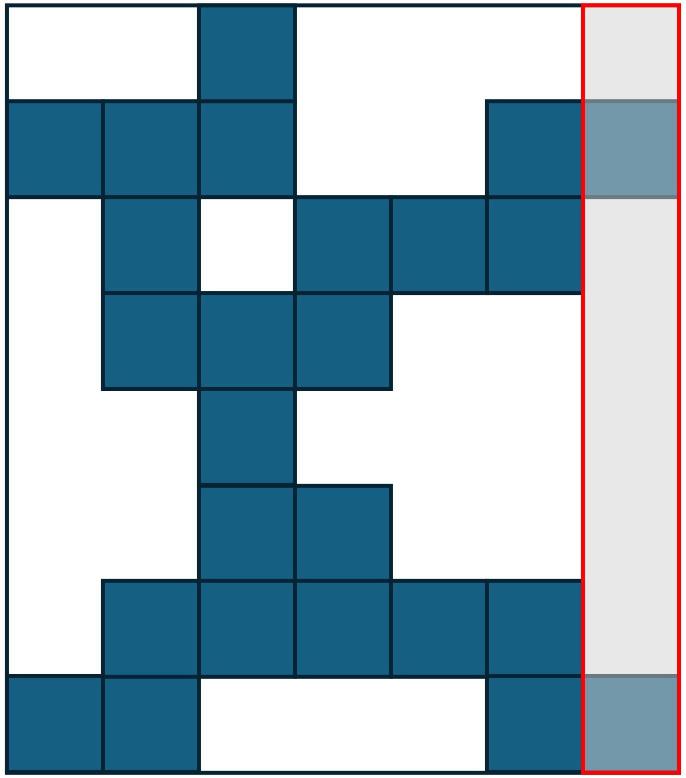
Generator

1. Connected
2. Feasible



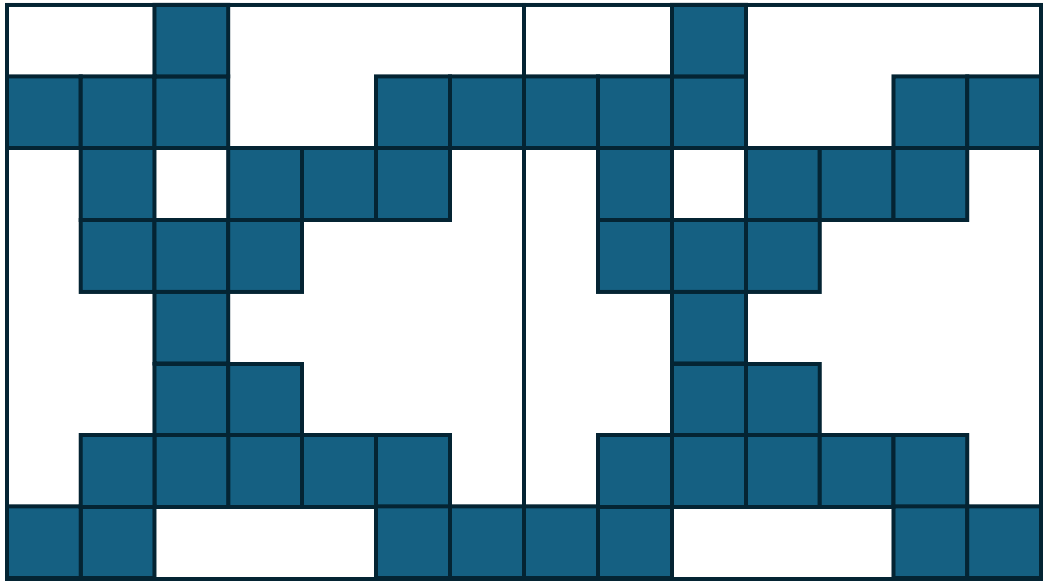
Generator

1. Connected
2. Feasible



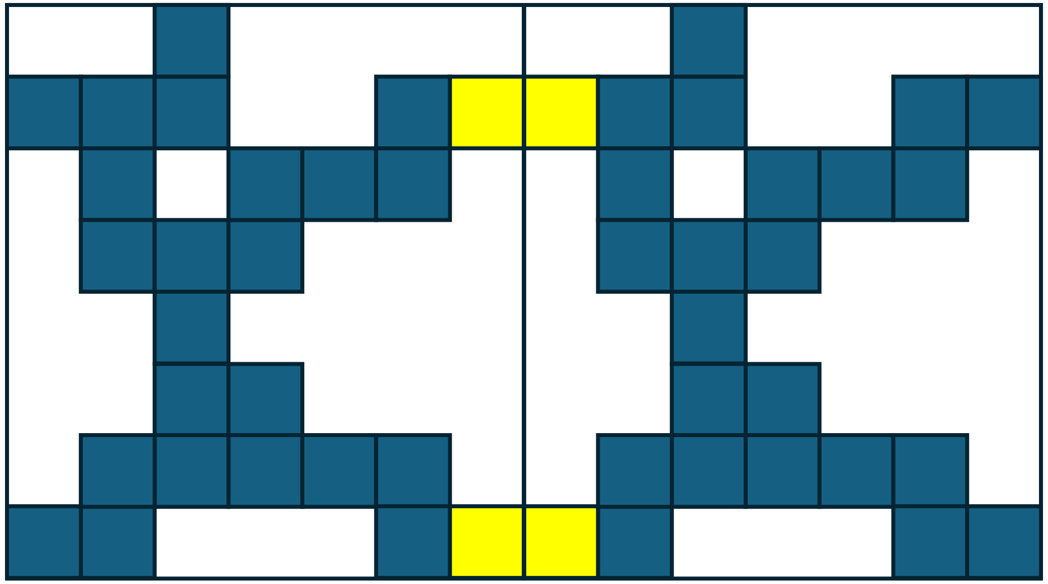
Generator

1. Connected
2. Feasible



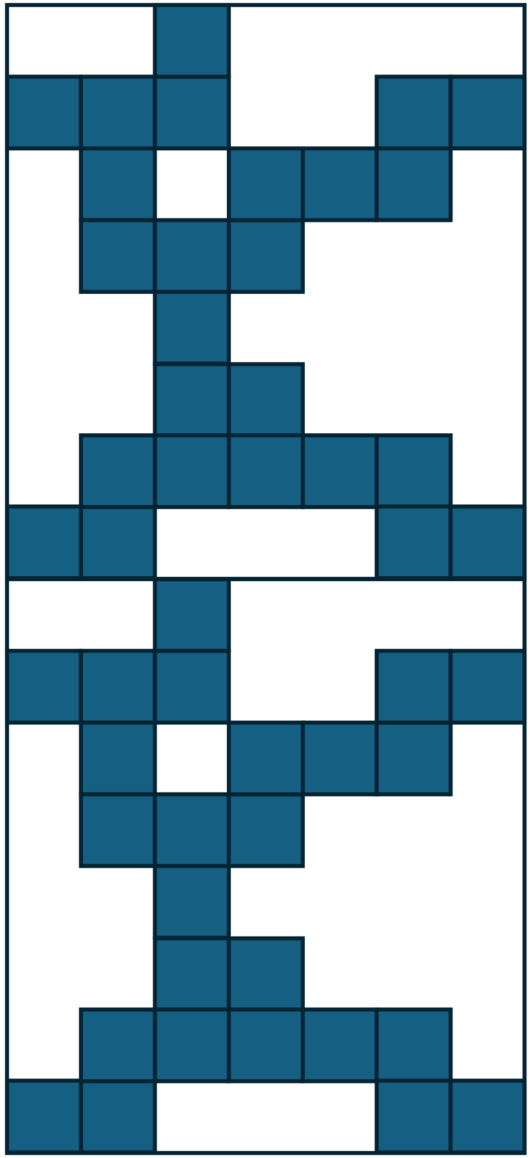
Generator

1. Connected
2. Feasible

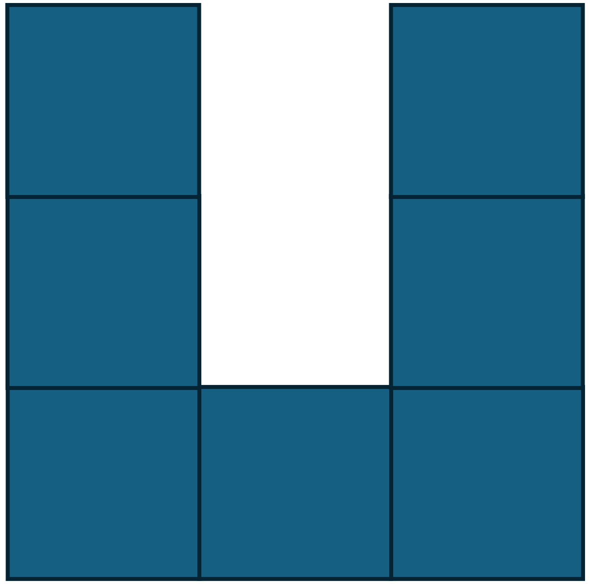


Generator

1. Connected
2. Feasible

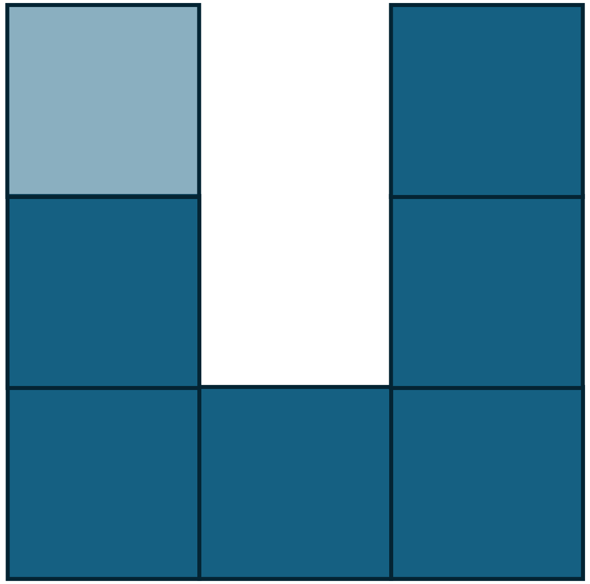


Discrete Self-Similar Fractal

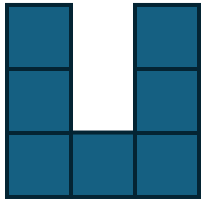


Stage 1

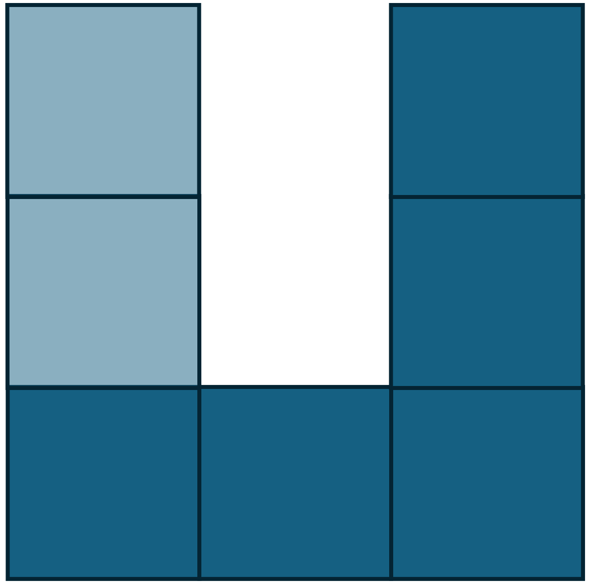
Discrete Self-Similar Fractal



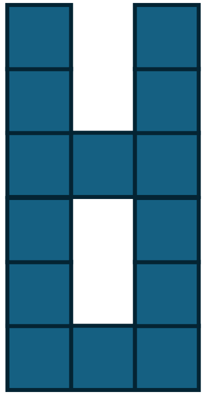
Stage 1



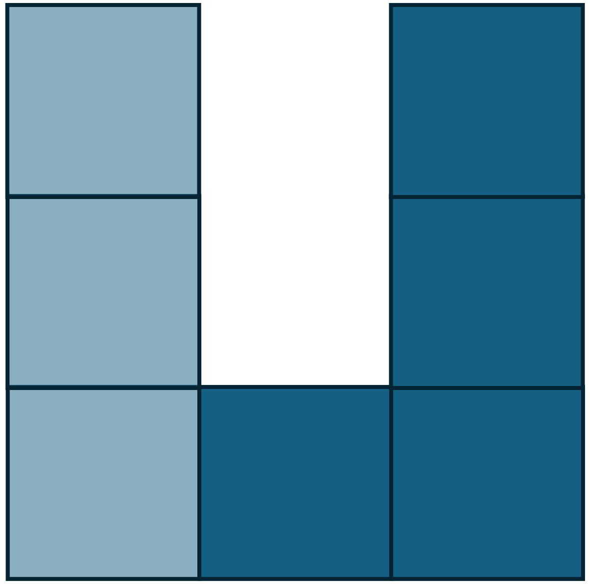
Discrete Self-Similar Fractal



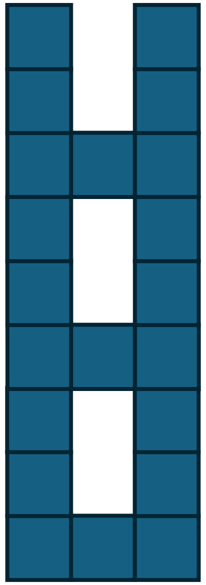
Stage 1



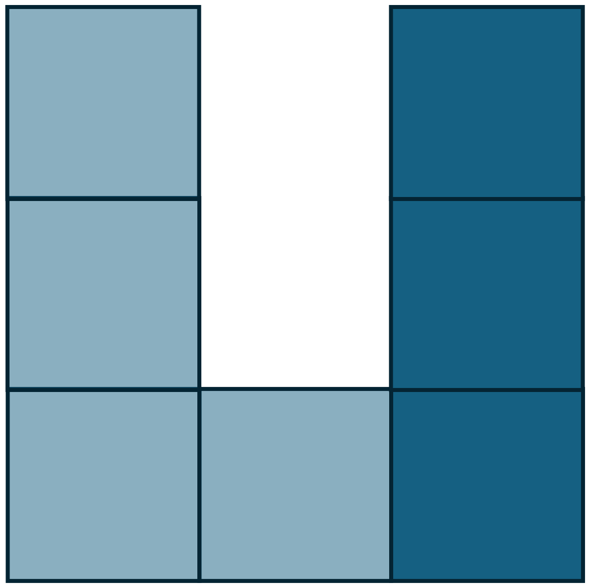
Discrete Self-Similar Fractal



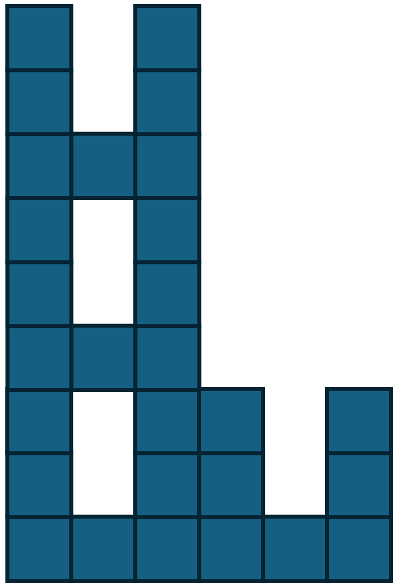
Stage 1



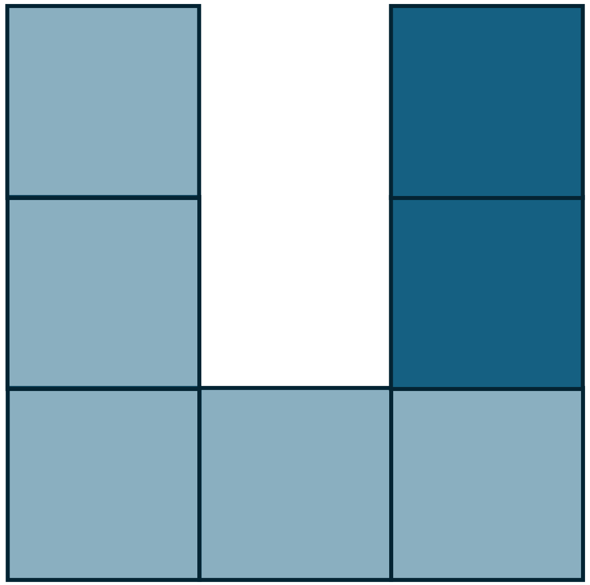
Discrete Self-Similar Fractal



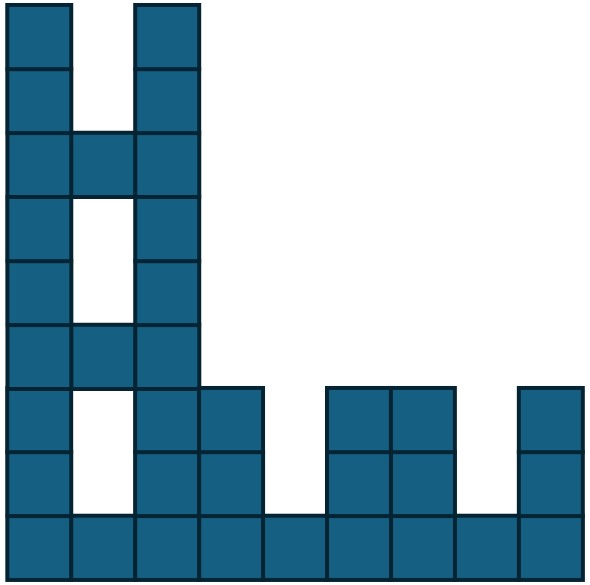
Stage 1



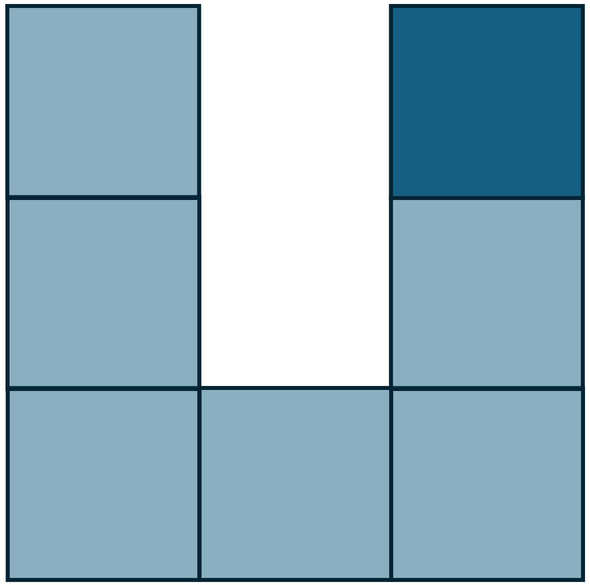
Discrete Self-Similar Fractal



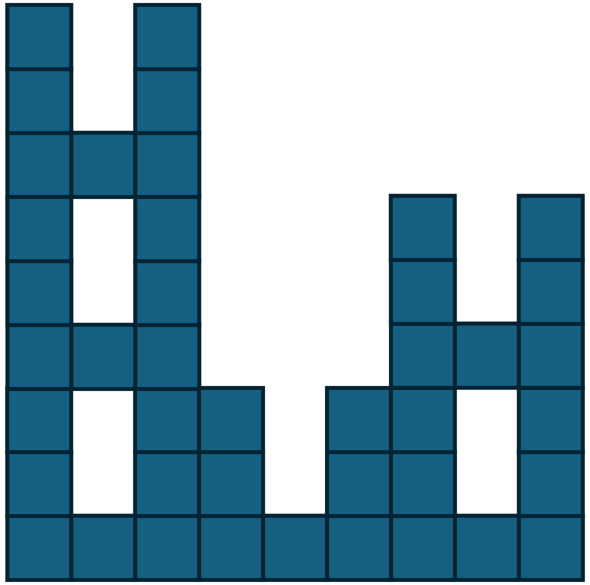
Stage 1



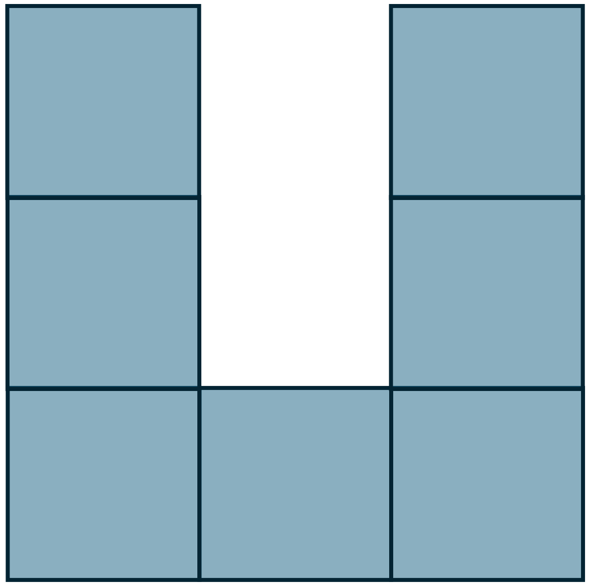
Discrete Self-Similar Fractal



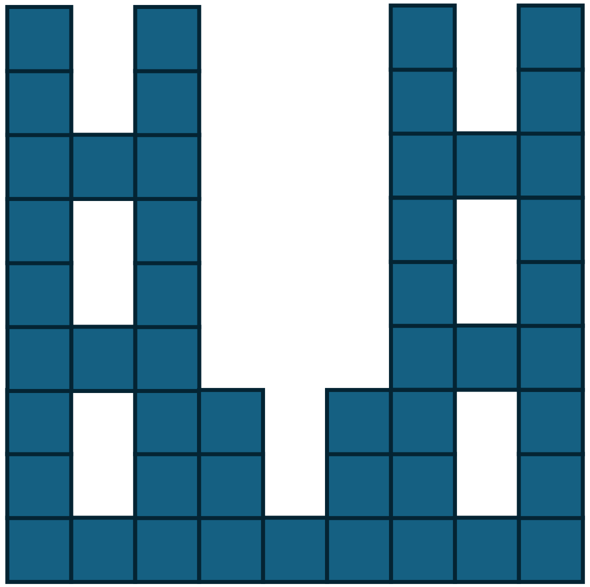
Stage 1



Discrete Self-Similar Fractal

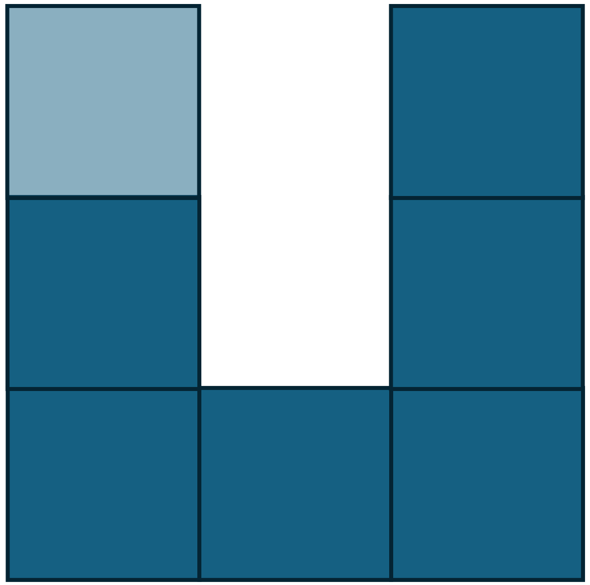


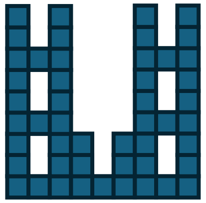
Stage 1



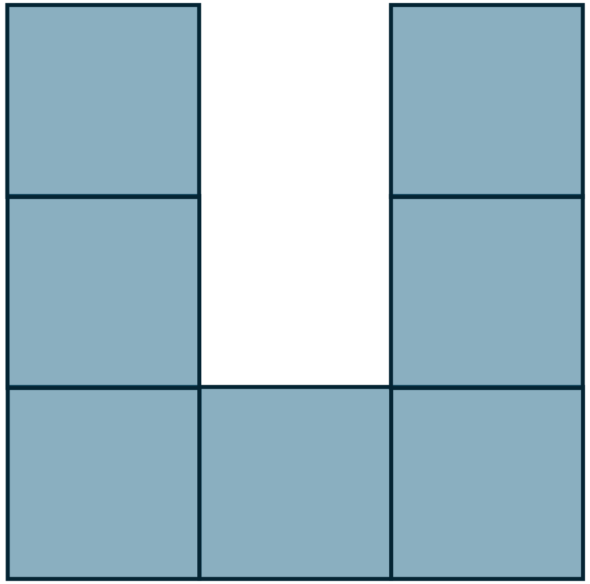
Stage 2

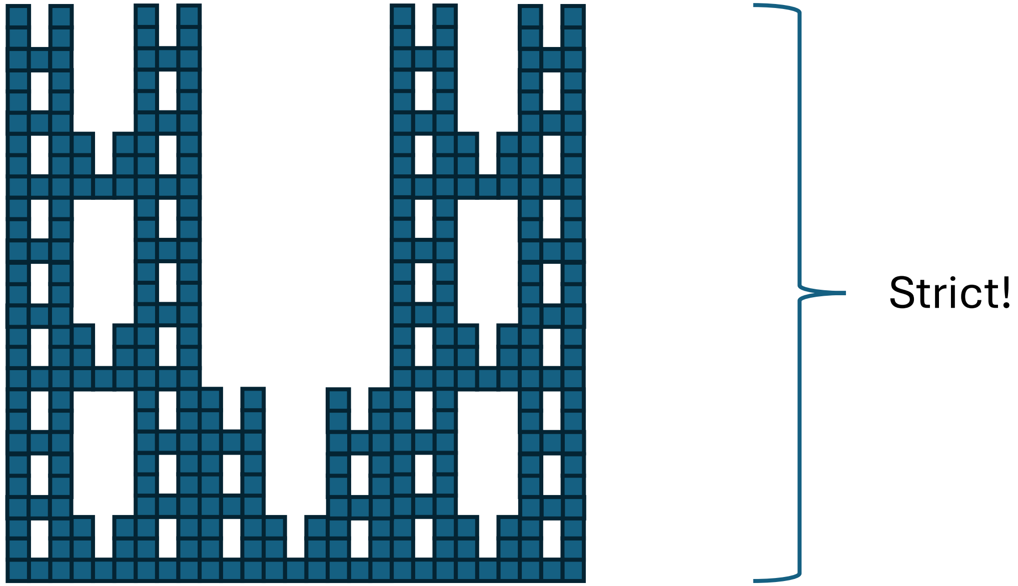
Discrete Self-Similar Fractal





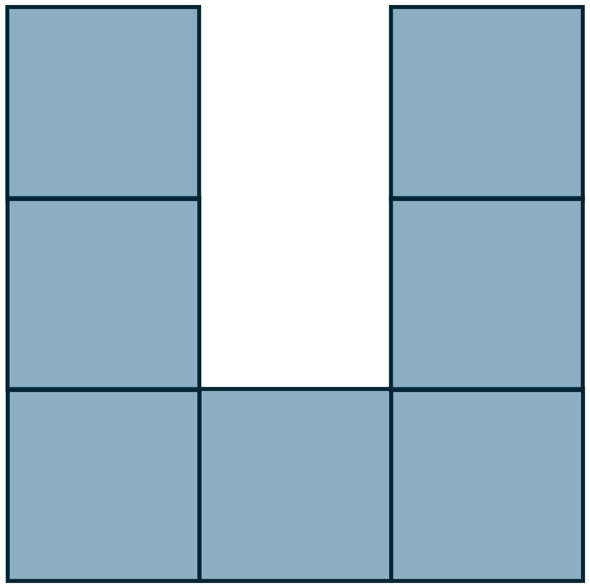
Discrete Self-Similar Fractal

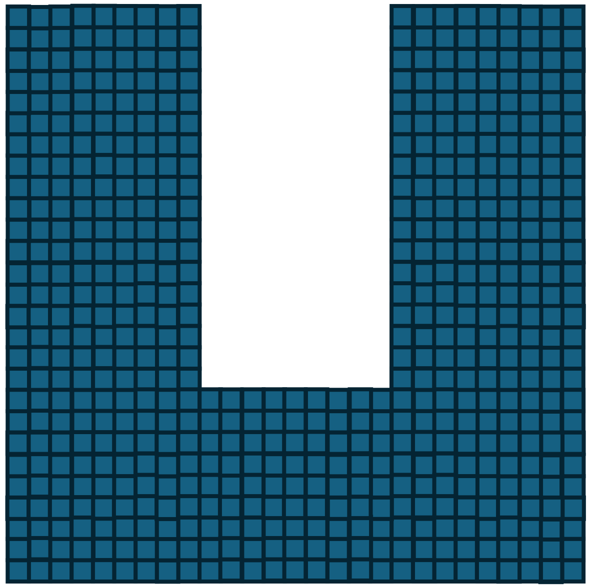




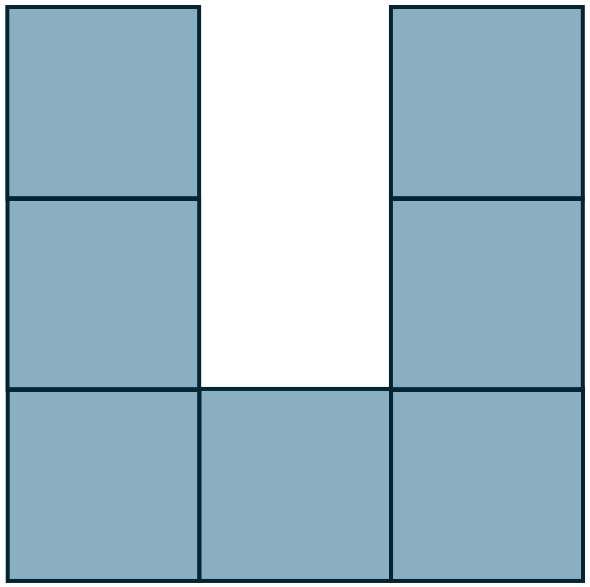
Stage 3

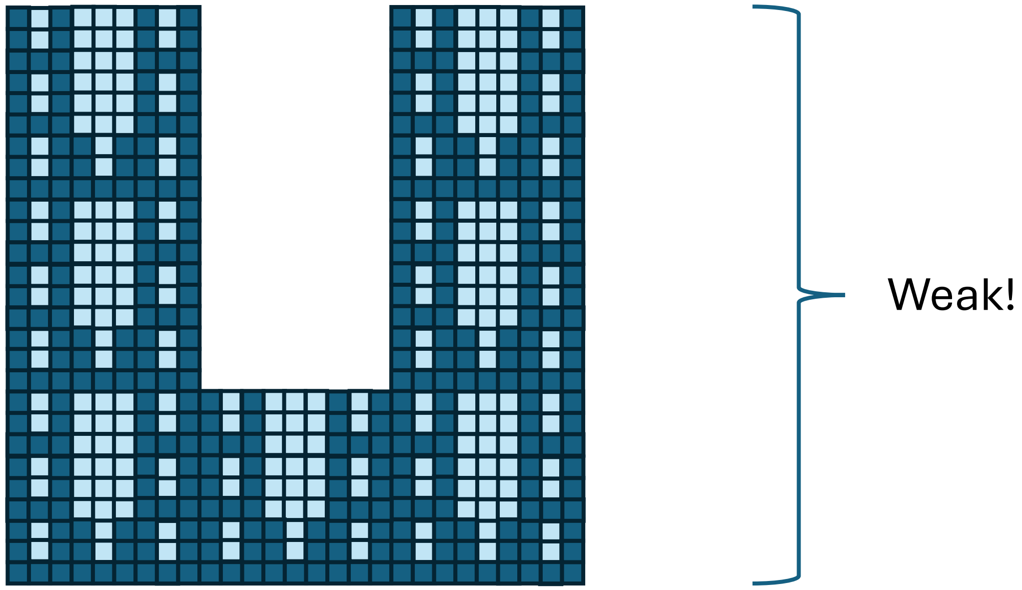
Discrete Self-Similar Fractal





Discrete Self-Similar Fractal



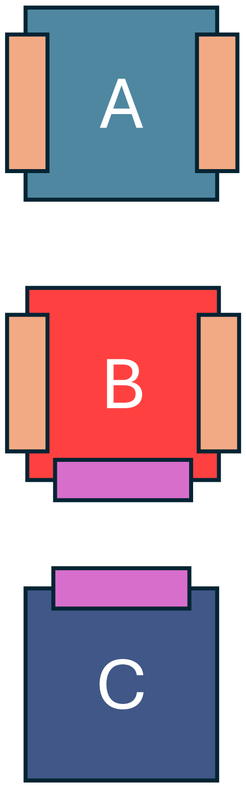


Previous Work

Why is this important?

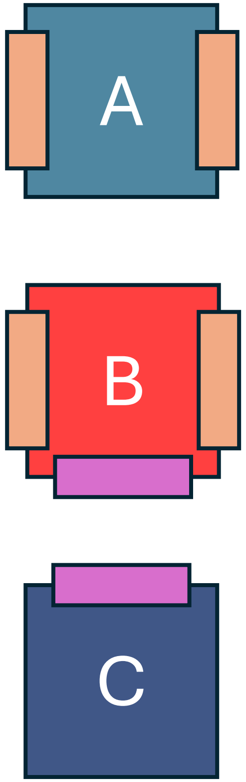
Tile Self-Assembly

Tiles:



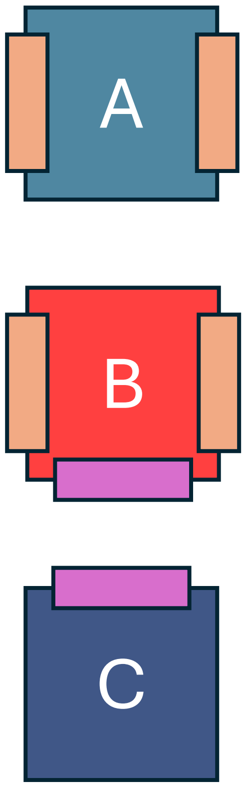
Tile Self-Assembly

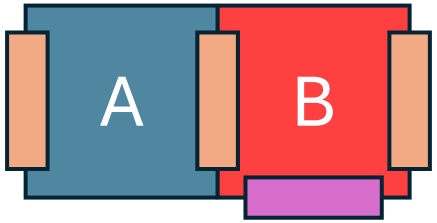
Tiles:



Tile Self-Assembly

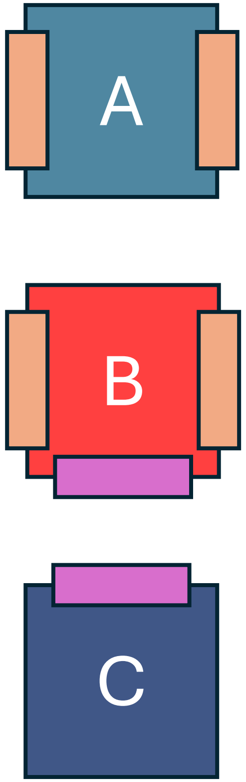
Tiles:

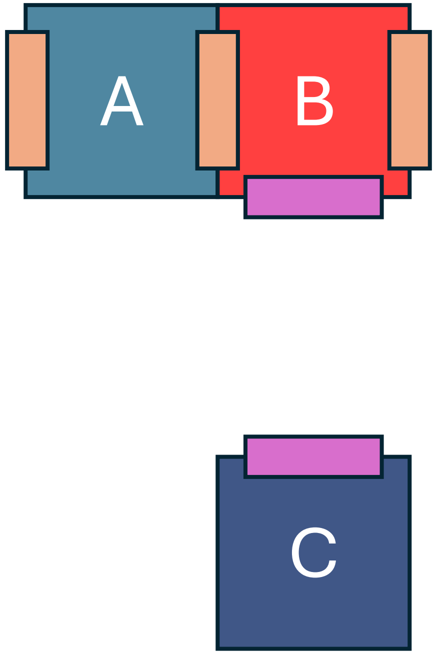




Tile Self-Assembly

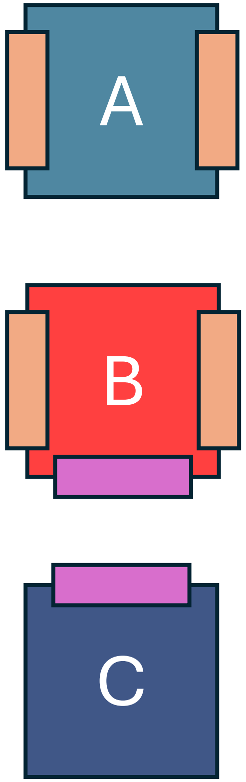
Tiles:

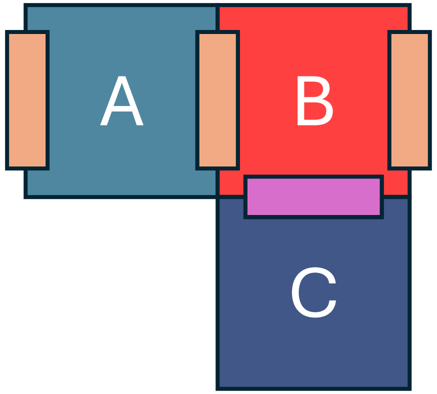




Tile Self-Assembly

Tiles:

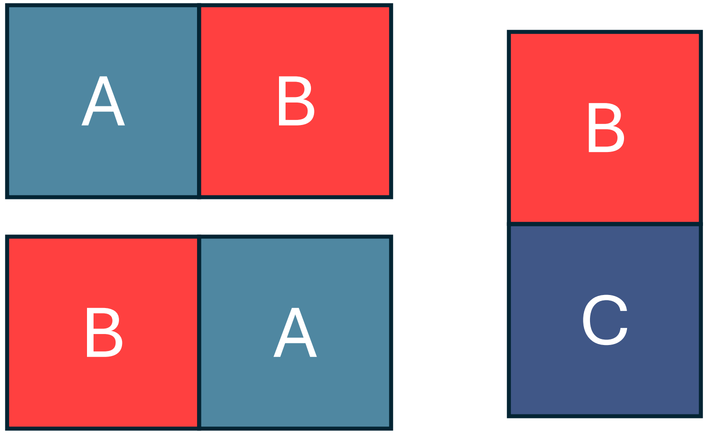




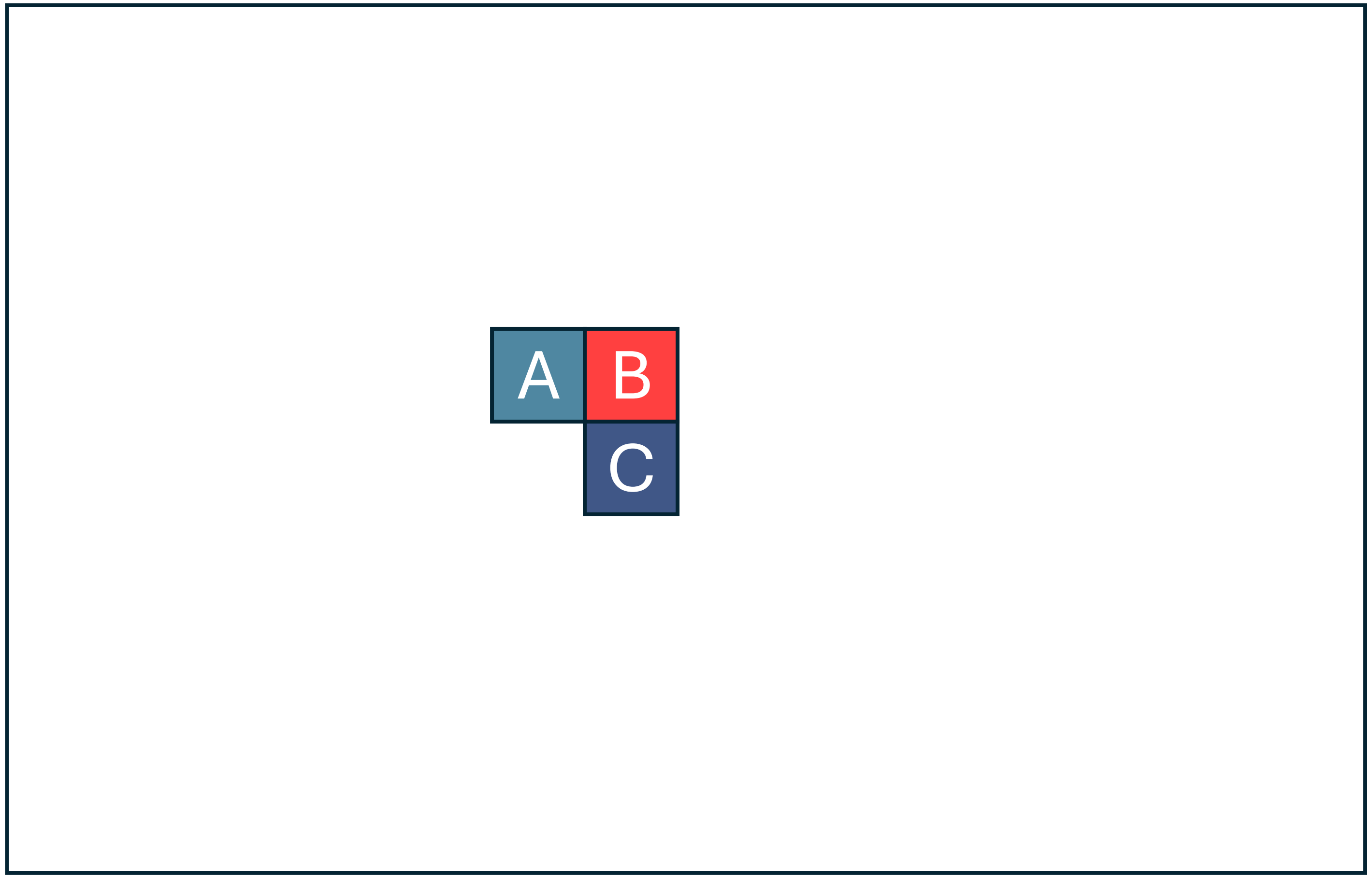
Tile Self-Assembly

Tiles:

Affinities:



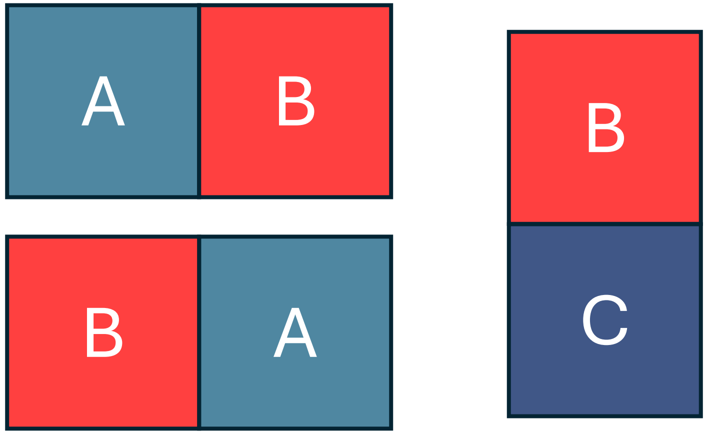
System:



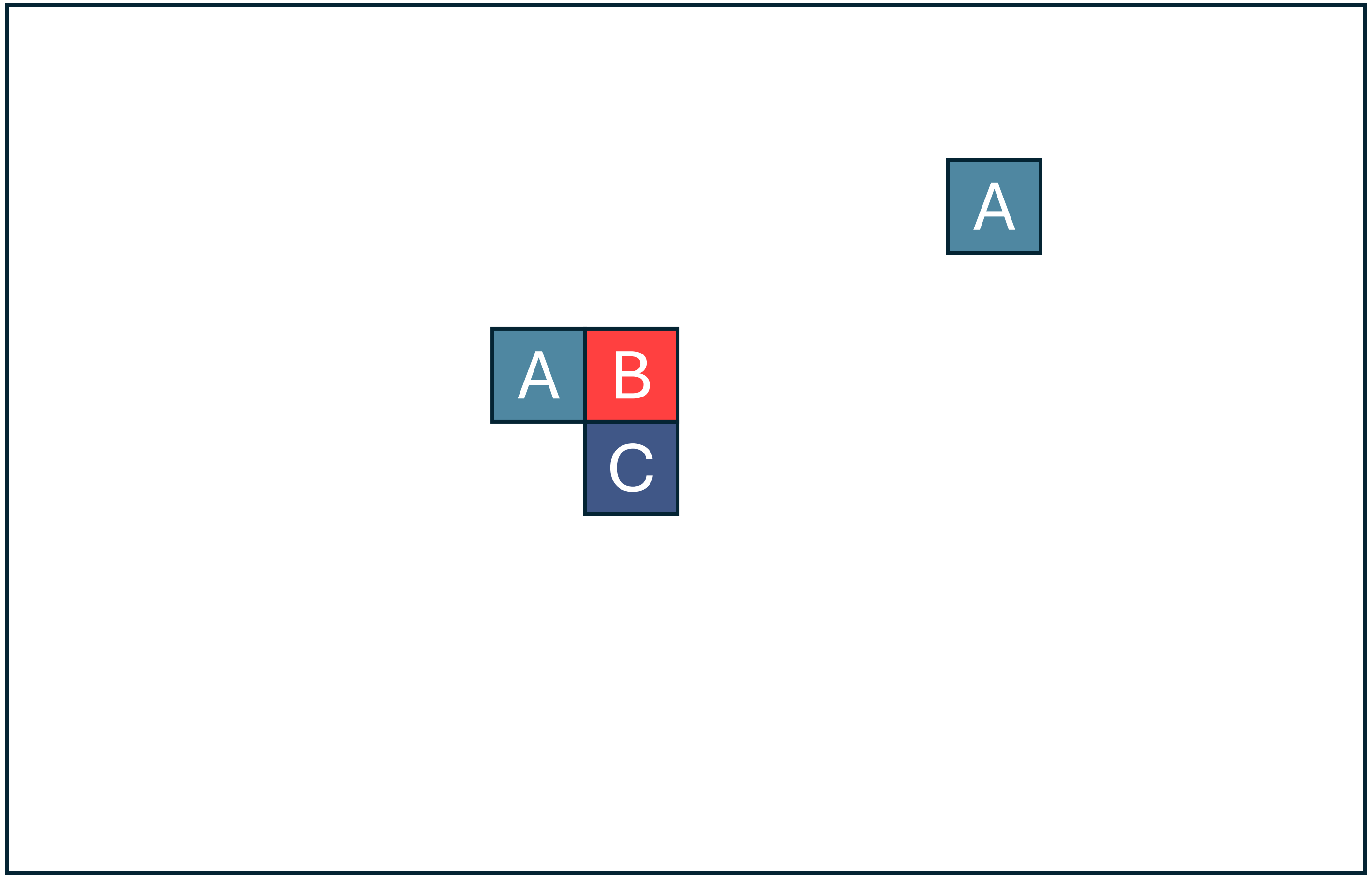
Tile Self-Assembly

Tiles:

Affinities:



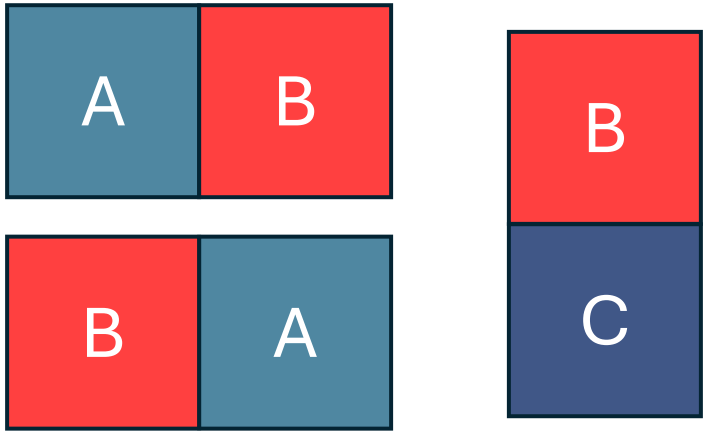
System:



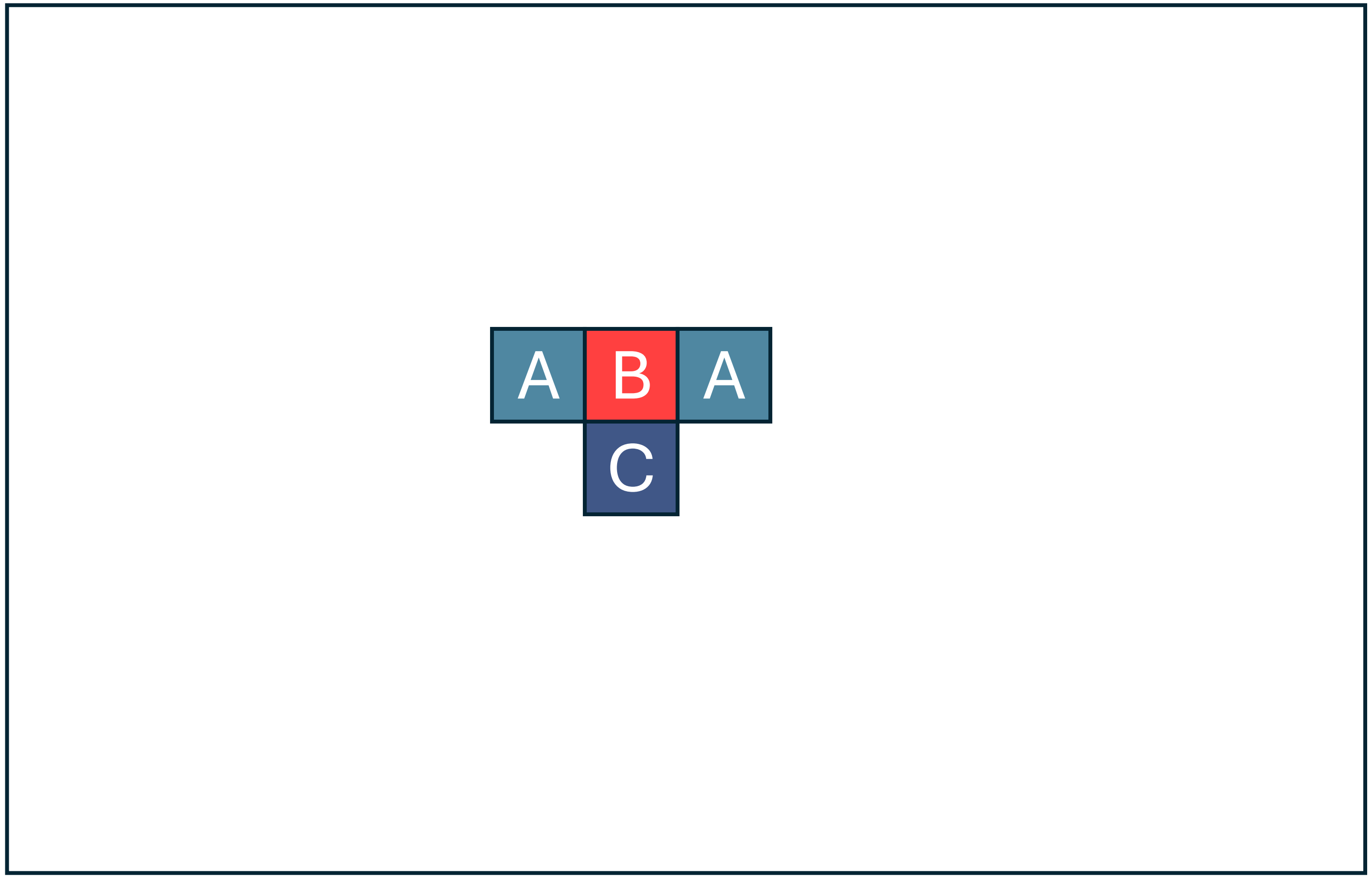
Tile Self-Assembly

Tiles:

Affinities:



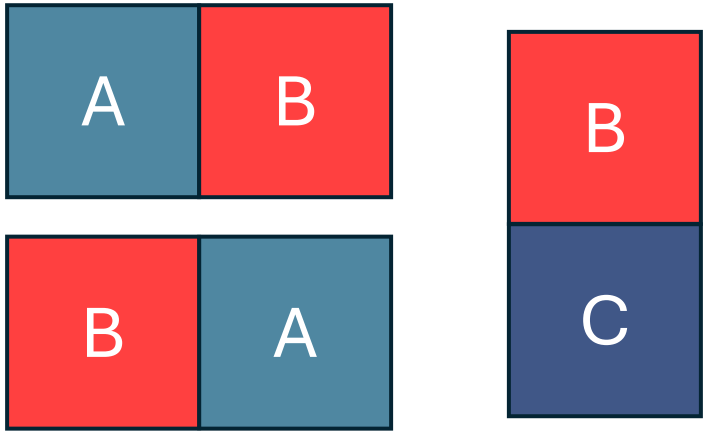
System:



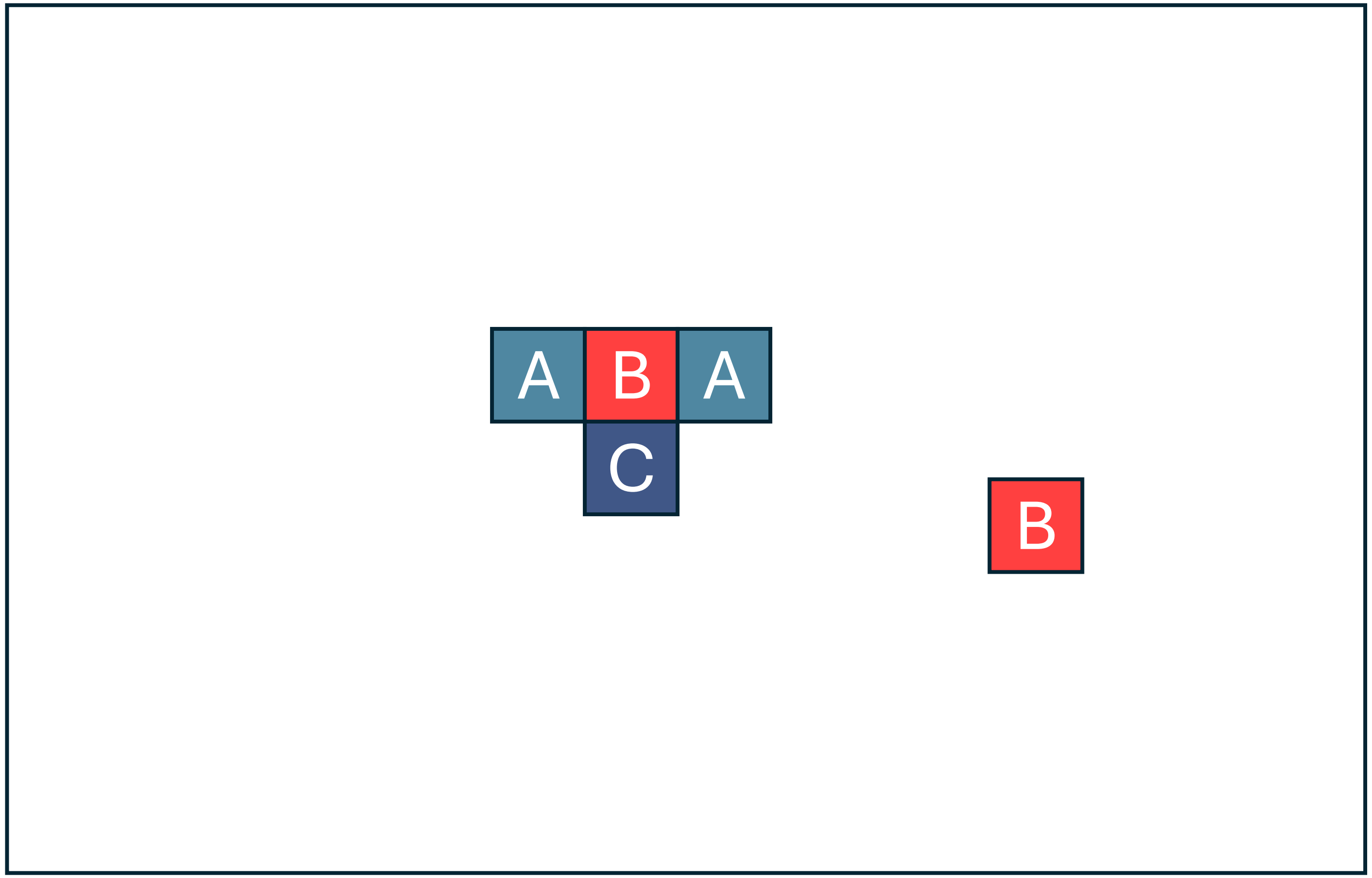
Tile Self-Assembly

Tiles:

Affinities:



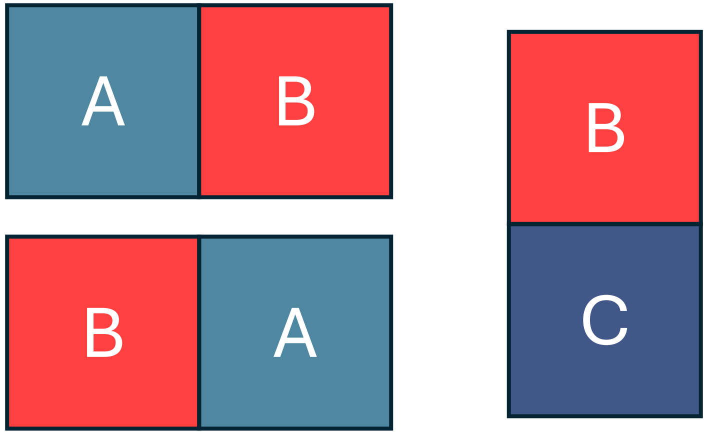
System:



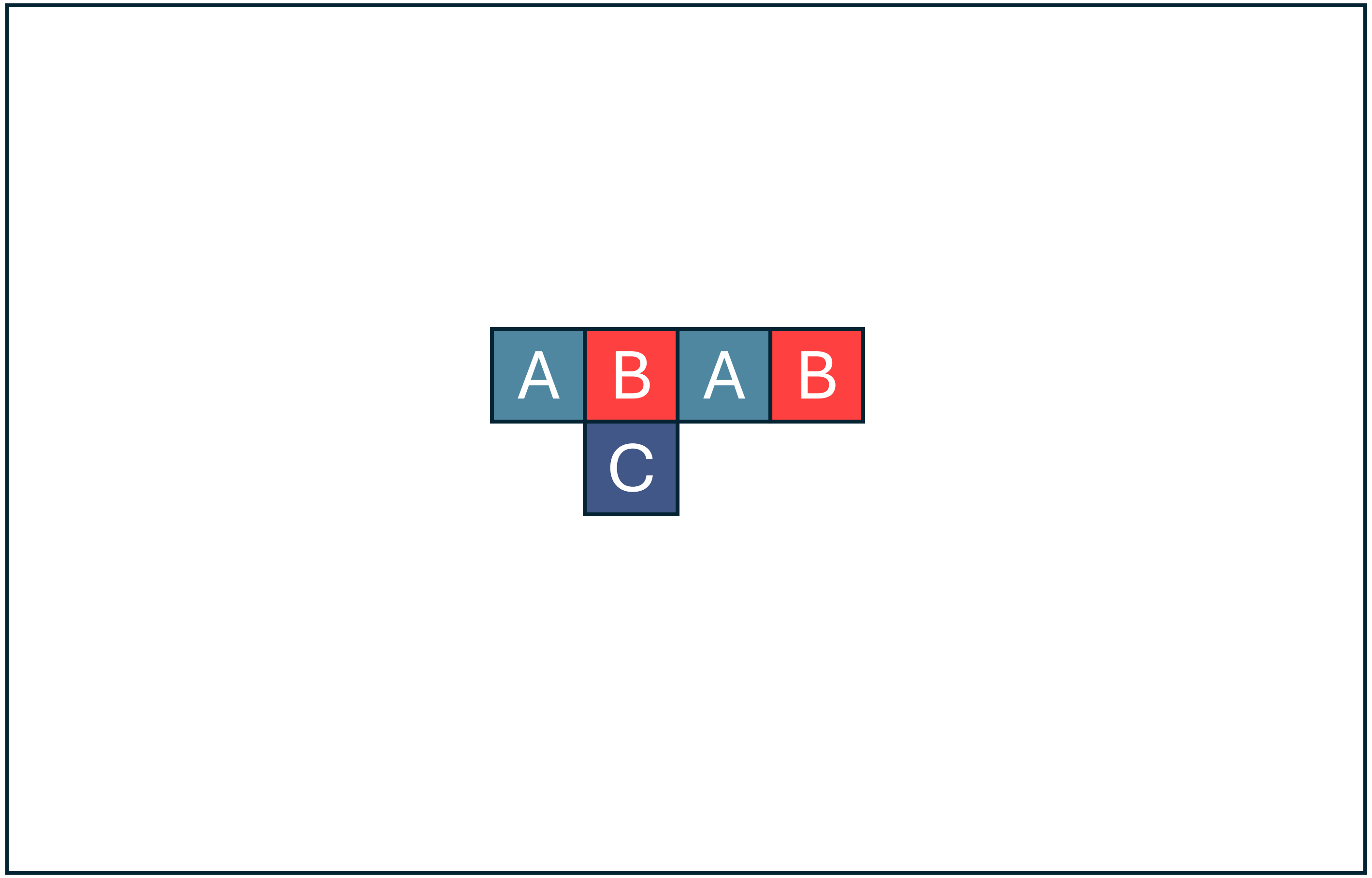
Tile Self-Assembly

Tiles:

Affinities:



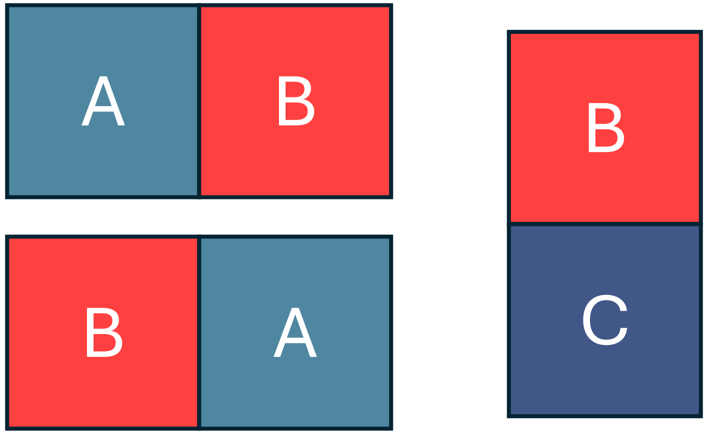
System:



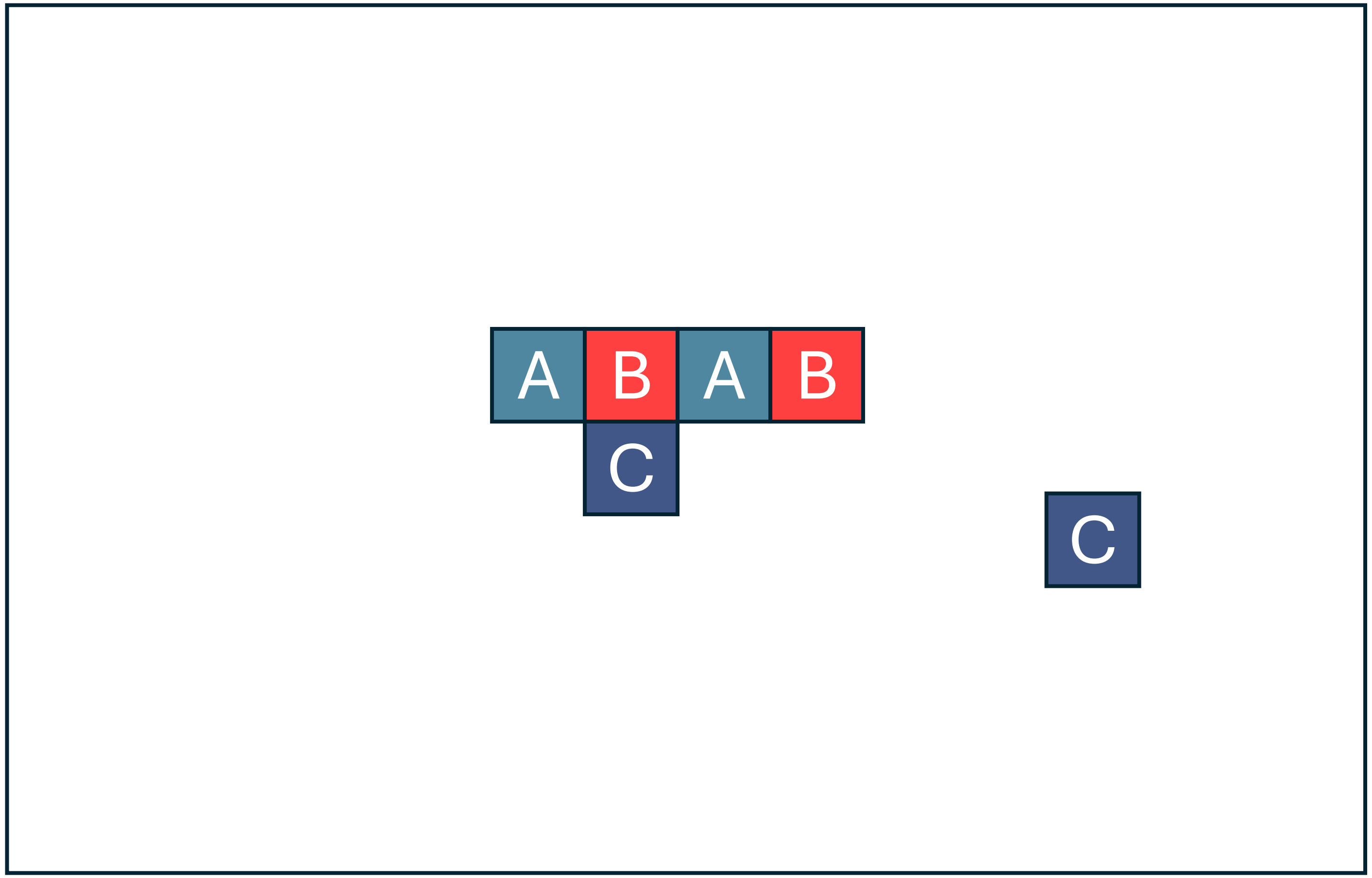
Tile Self-Assembly

Tiles:

Affinities:



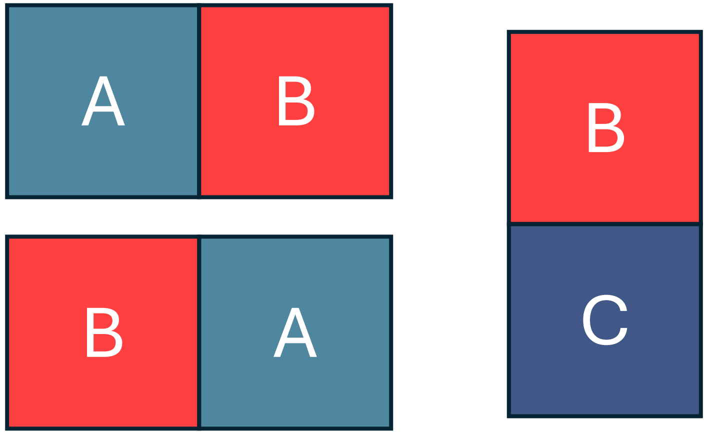
System:



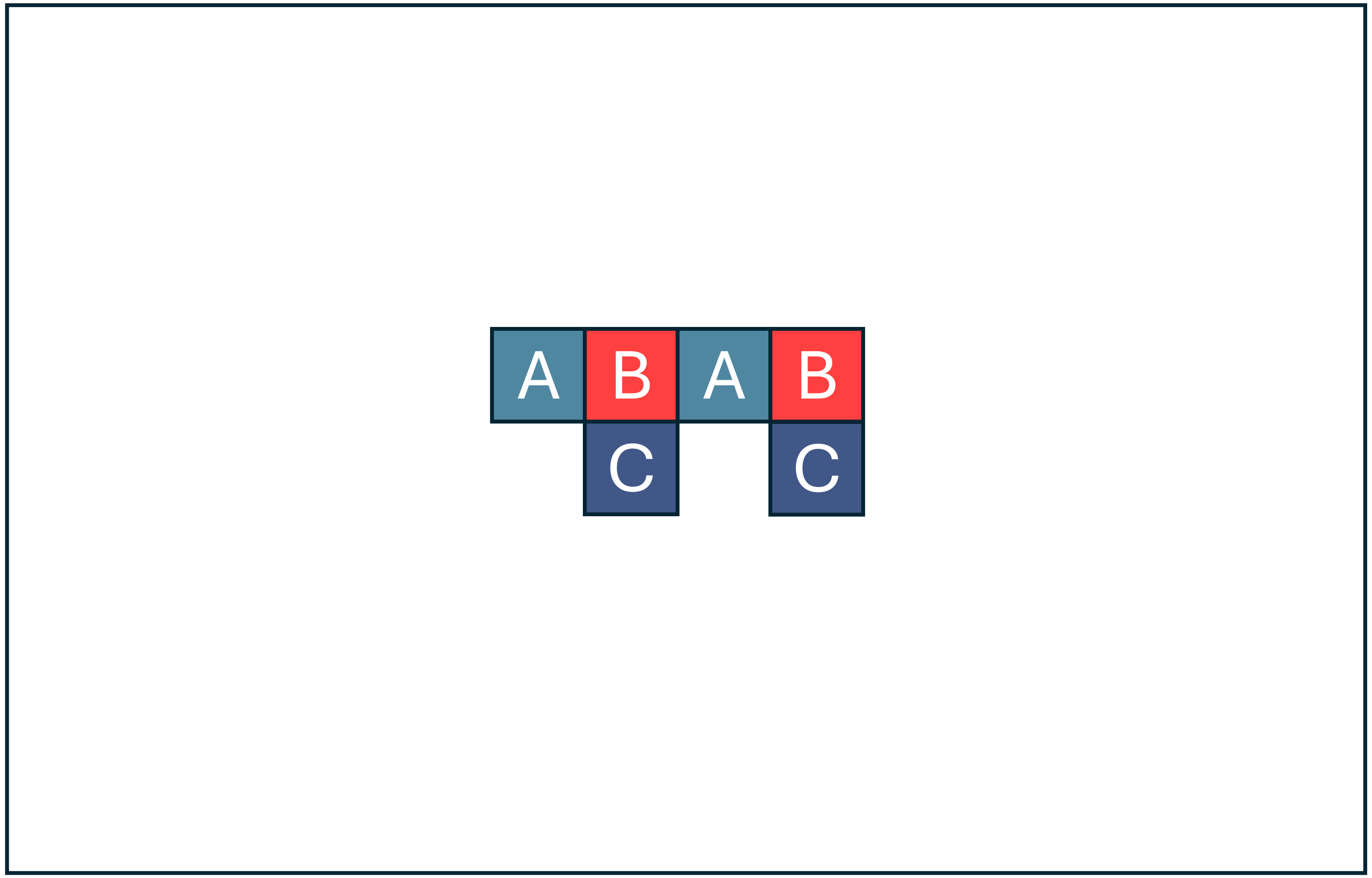
Tile Self-Assembly

Tiles:

Affinities:



System:

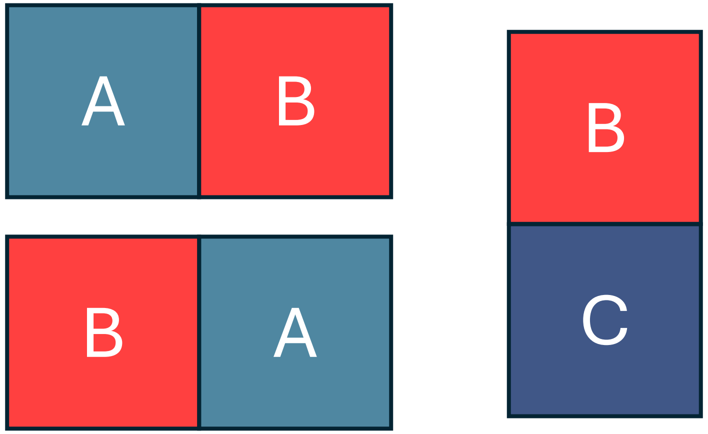


The aTAM (abstract Tile Assembly Model)

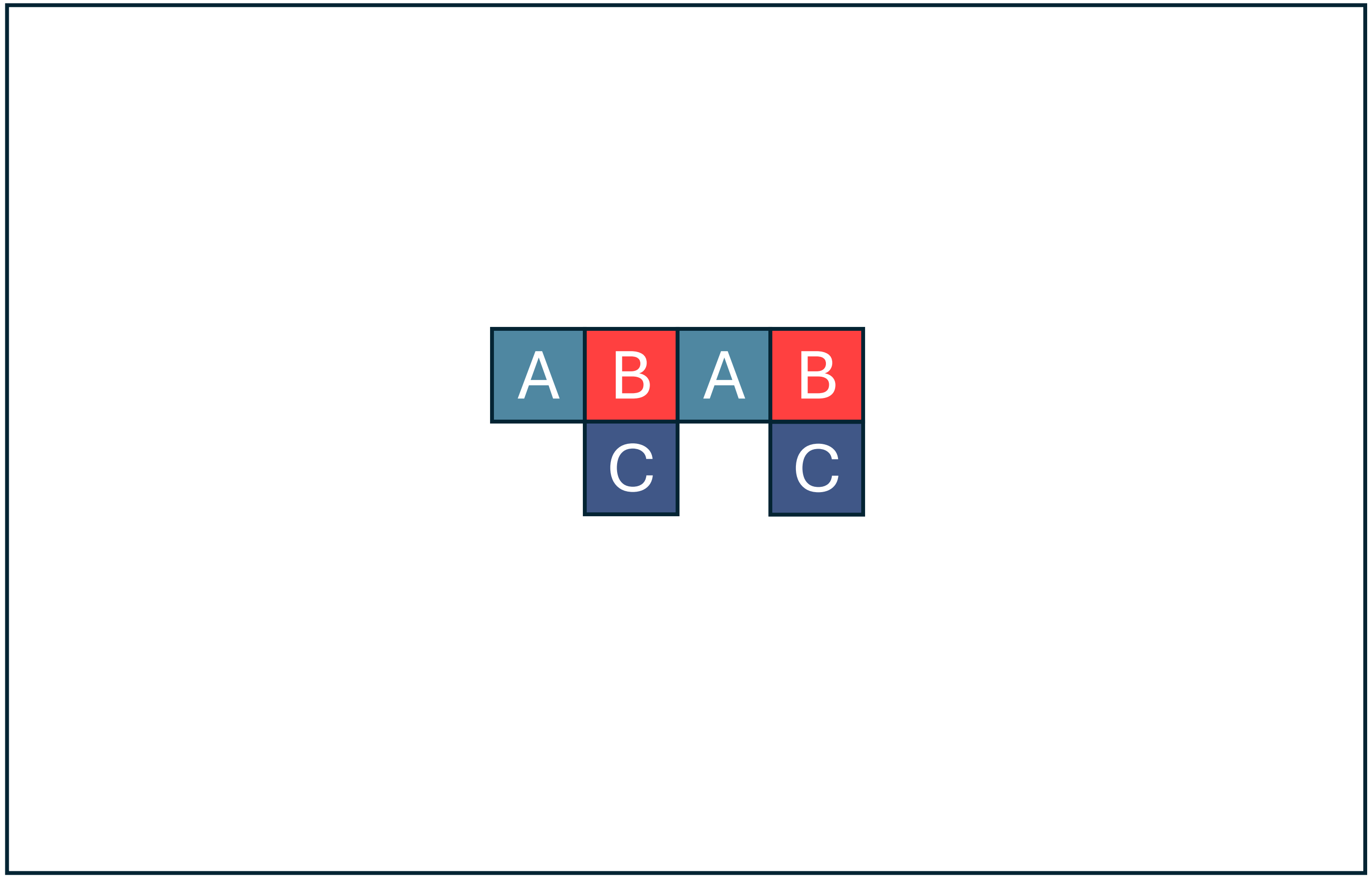
Tiles:

Affinities:

Temperature: τ



System:

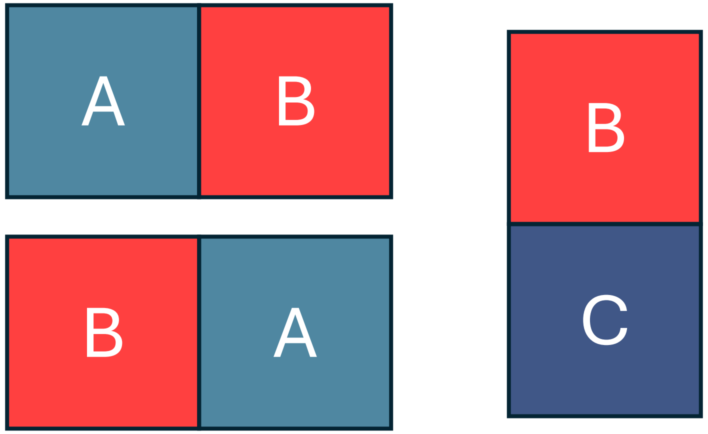


The aTAM (abstract Tile Assembly Model)

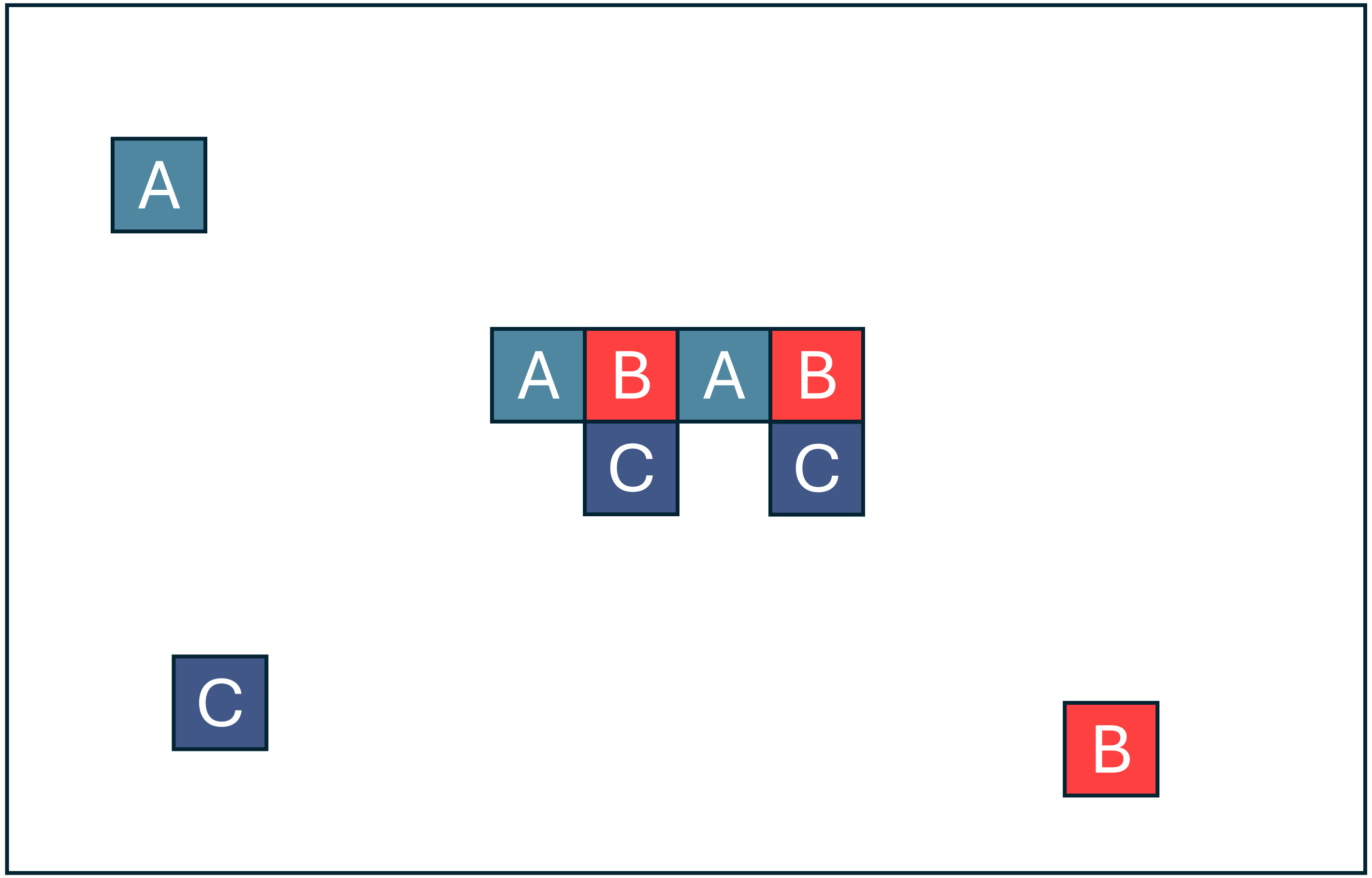
Tiles:

Affinities:

Temperature: τ



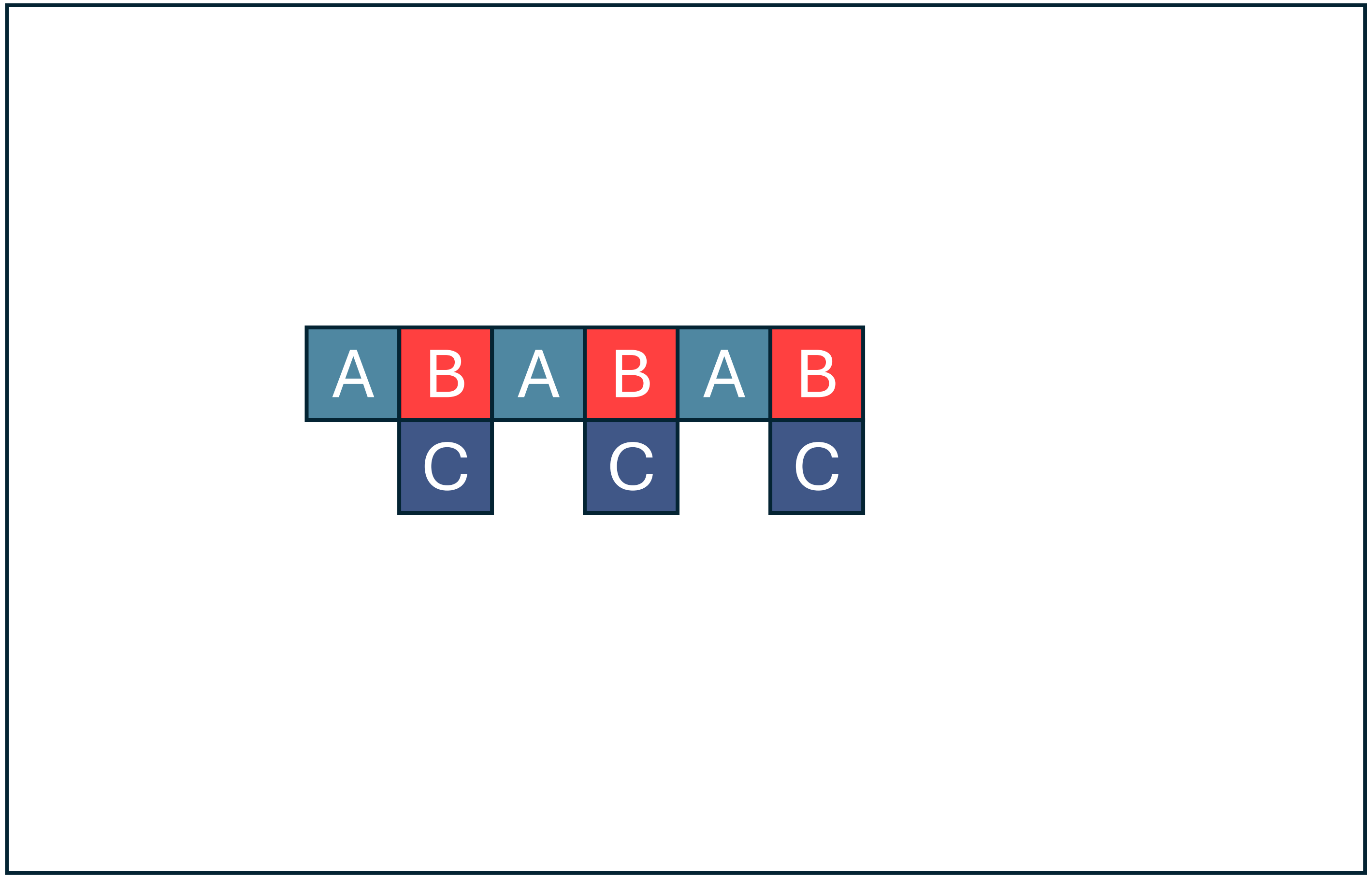
System:



The aTAM (abstract Tile Assembly Model)

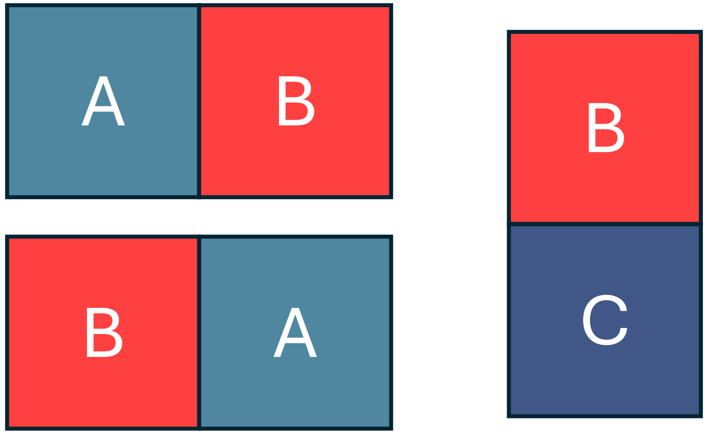
Tiles:

System:



Affinities:

Temperature: τ

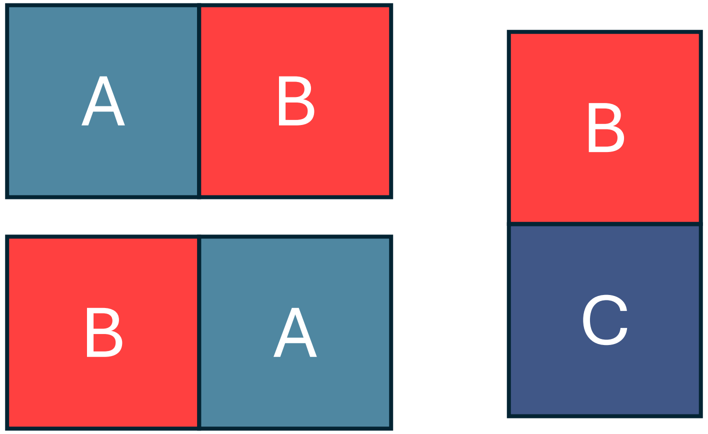


The aTAM (abstract Tile Assembly Model)

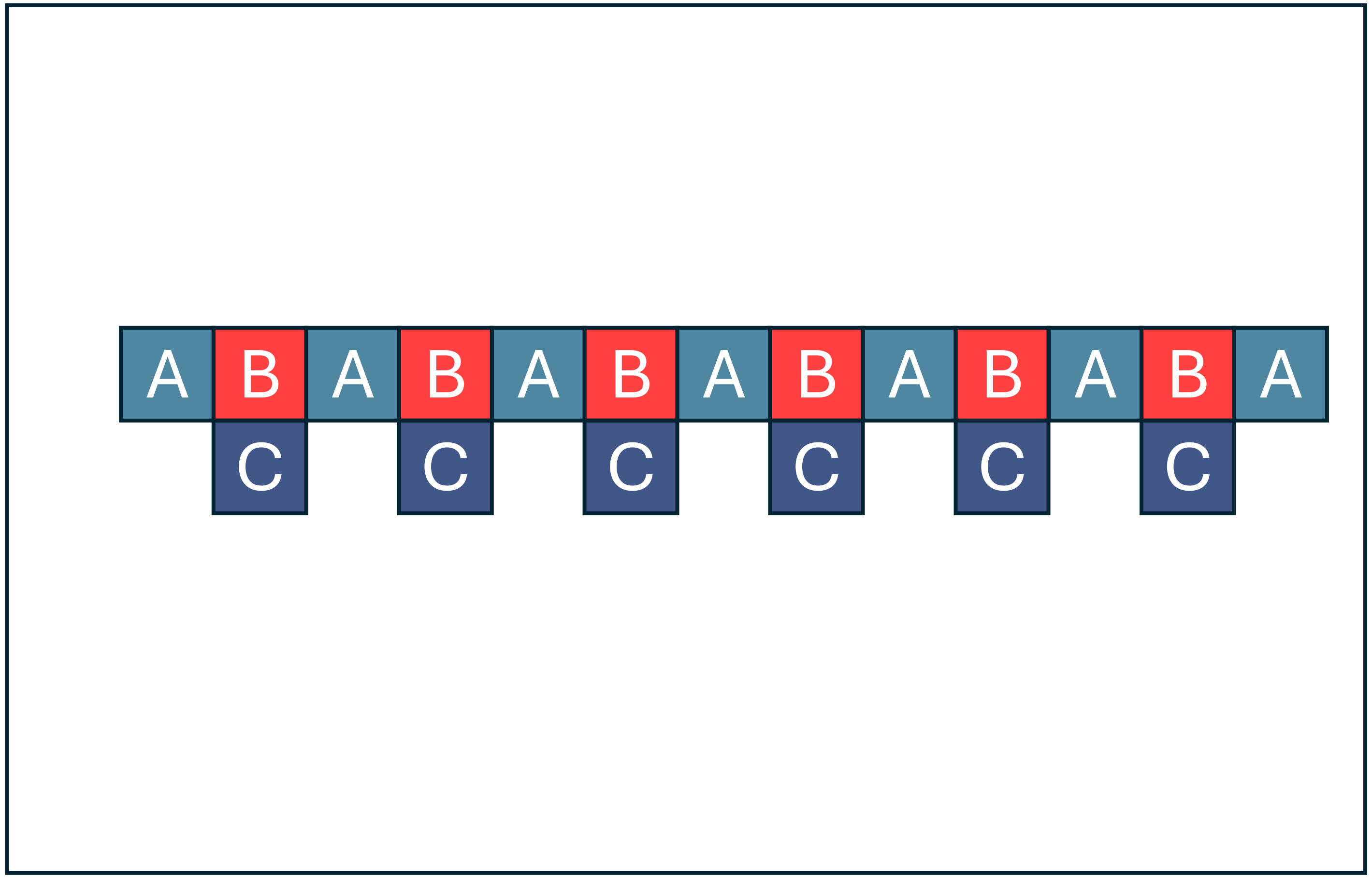
Tiles:

Affinities:

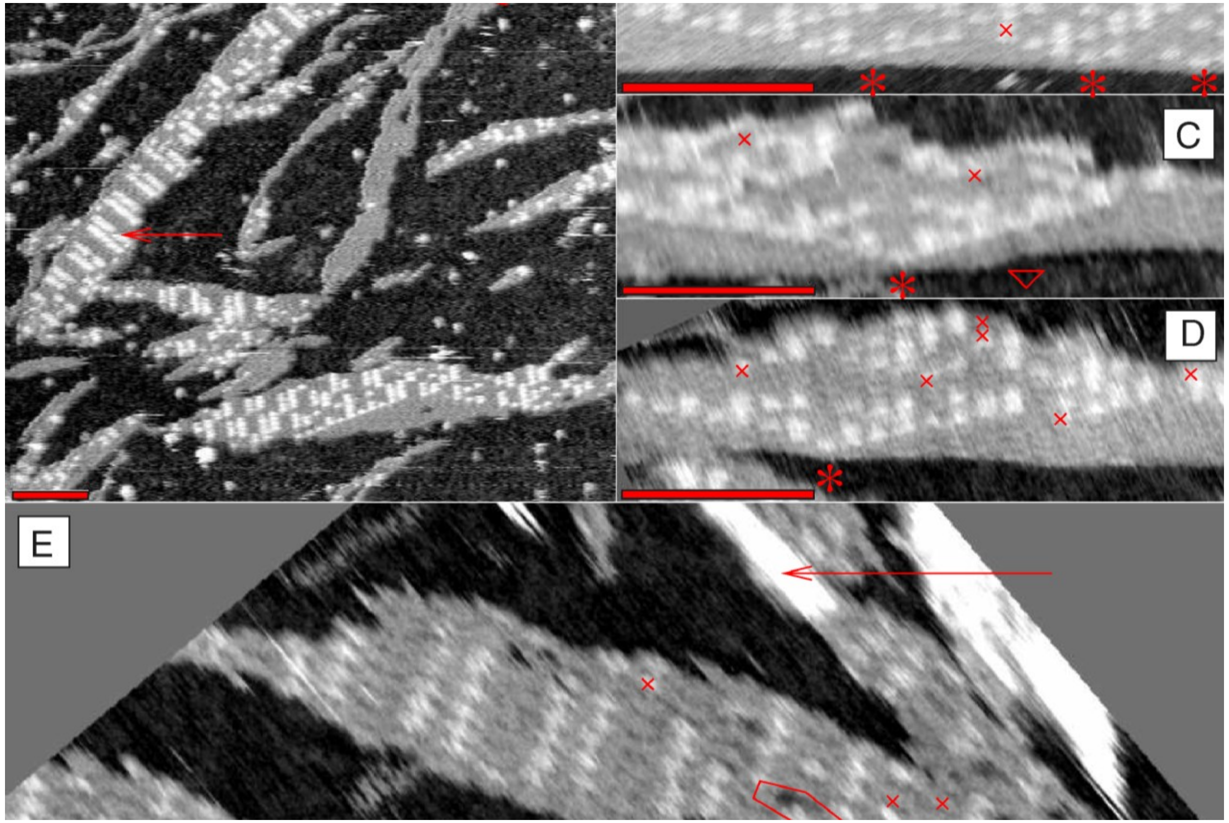
Temperature: τ

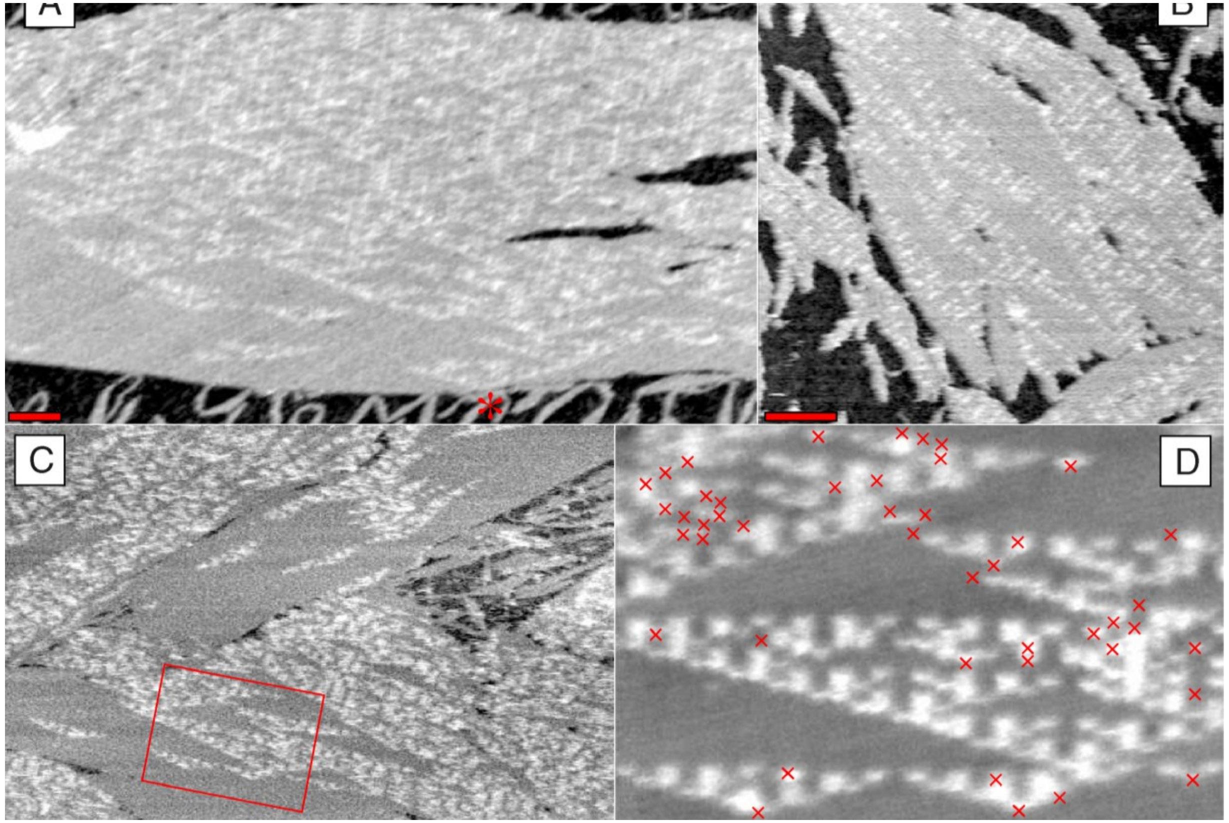


System:



What is known (aTAM)

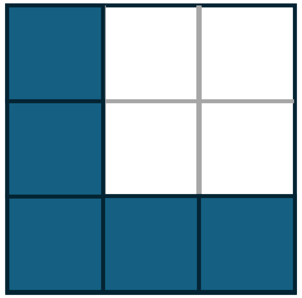


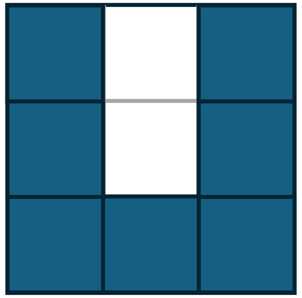


Rothemund PWK, Papadakis N, Winfree E (2004) Algorithmic Self-Assembly of DNA Sierpinski Triangles. PLoS Biol 2(12): e424.

What is known (aTAM)

- 2010 [M. J. Patitz, S. M. Summers. Self-assembly of discrete self-similar fractals]:
  1. Any "L" shape does not strictly self assemble in the aTAM (any temp)
  2. No fractal weakly self assembles at temperature 1
  3. "Nice" fractals can be weakly self assembled at higher temps

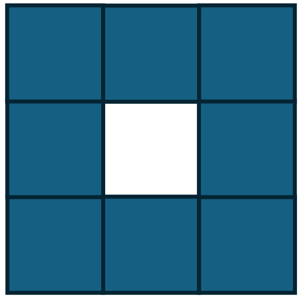




- 2025 [F. Becker, D. Hader, M. J. Patitz. Strict self-assembly of discrete self-similar fractals in the abstract tile-assembly model]:
  1. A full characterization of which DSSFs are buildable in the aTAM.

Variations of the aTAM

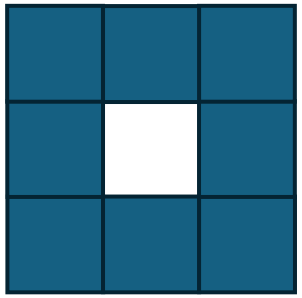
- 2HAM (2-Handed Assembly Model):
  1. 2 different assemblies can attach
  2. Can finitely assemble a larger class of shapes



- STAM (Signal-Passing Tile Assembly Model):
  1. Glues can turn “on” and “off”
  2. Detachment is allowed

Variations of the aTAM

- 2HAM (2-Handed Assembly Model):
  1. 2 different assemblies can attach
  2. Can finitely assemble a larger class of shapes



- STAM (Signal-Passing Tile Assembly Model):
  1. Glues can turn “on” and “off”
  2. Detachment is allowed
  3. Any arbitrary fractal is buildable
  4. Without detachments, some shapes impossible

A Comparison

	aTAM	2HAM	STAM
Description	- Single tile attachments - No detachments - Fixed glues	- Up to 2 assembly attachments - No detachments - Fixed glues	- Up to 2 assembly attachments - Detachments - Glues turn on/off
Results	- No fractals weakly buildable (temp 1) - Some fractals not strictly buildable (temp > 1)	- Finitely assemble larger class of shapes - Still some non-buildable fractals	- All fractals strictly buildable - **Without detachment:** some impossible fractals

Question: What if adjacent tiles can infinitely change states?

Seeded Tile Automata

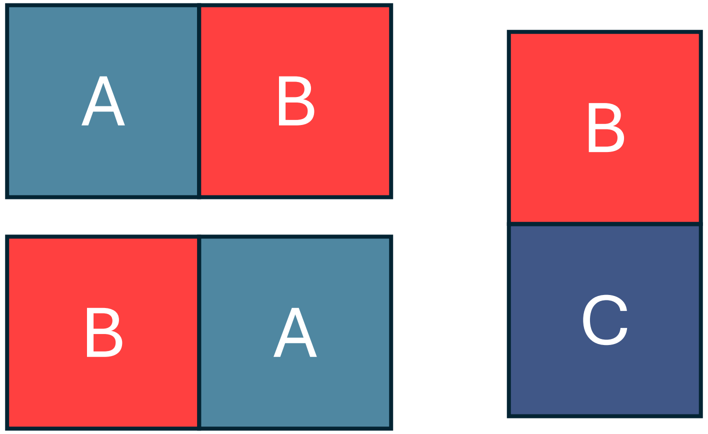
What is the model?

Seeded TA

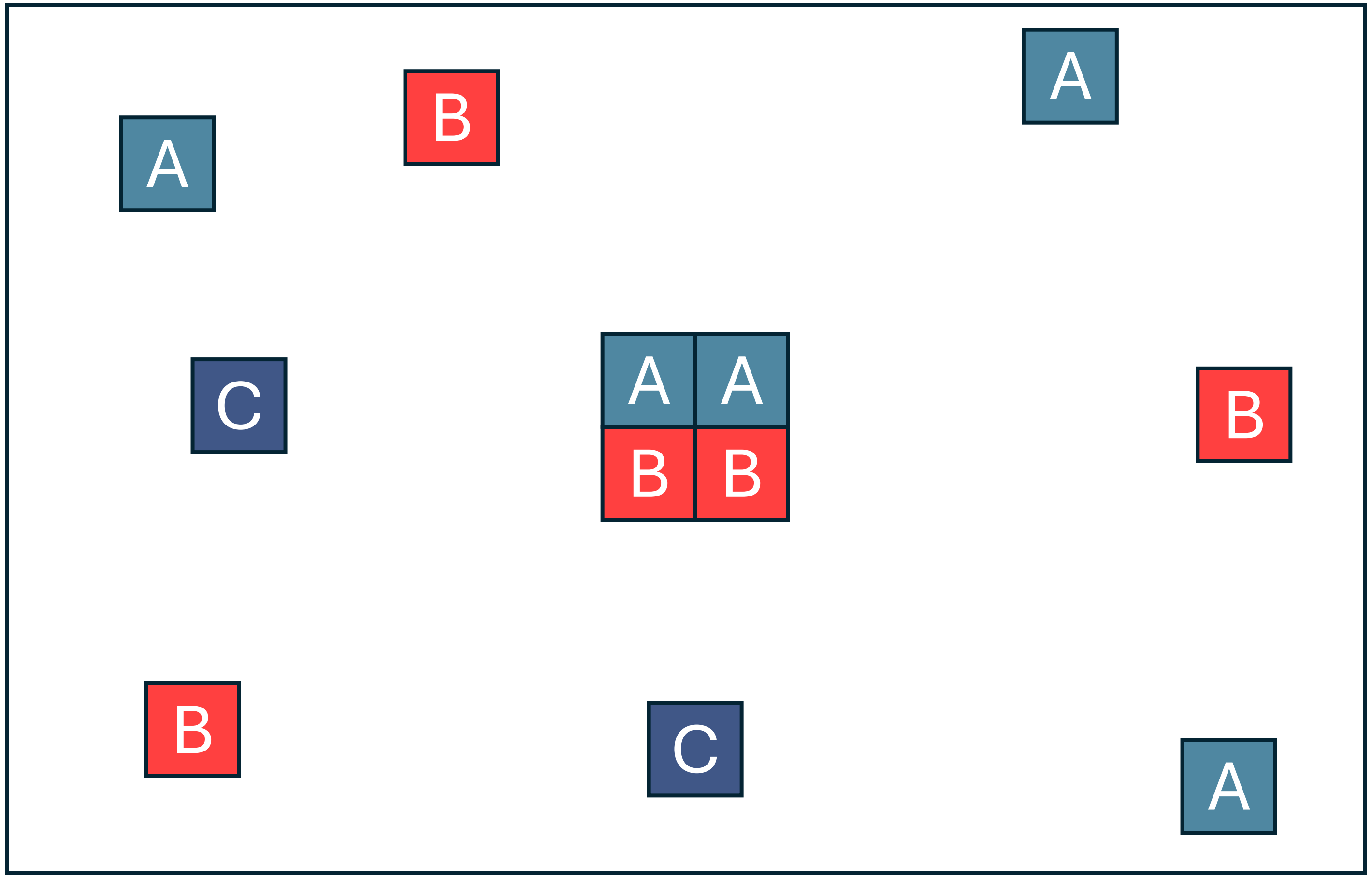
Tiles:

Affinities:

Temperature: τ



System:

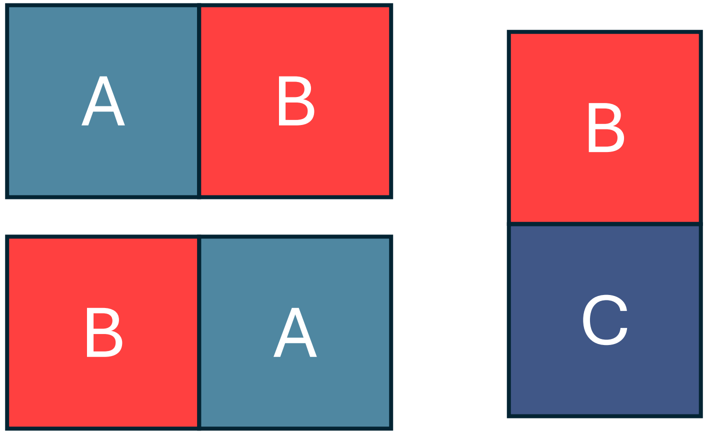


Seeded TA

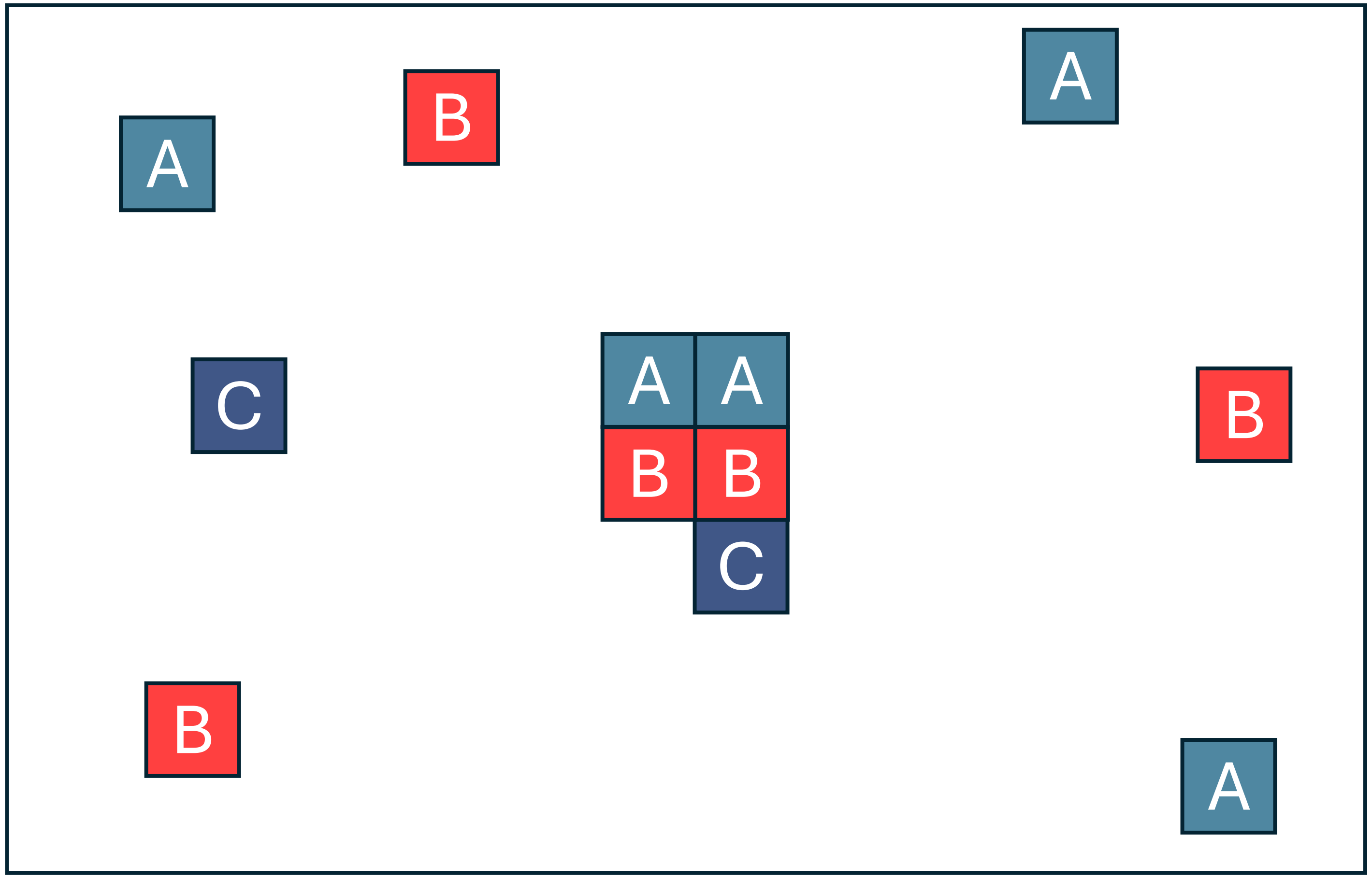
Tiles:

Affinities:

Temperature: τ



System:

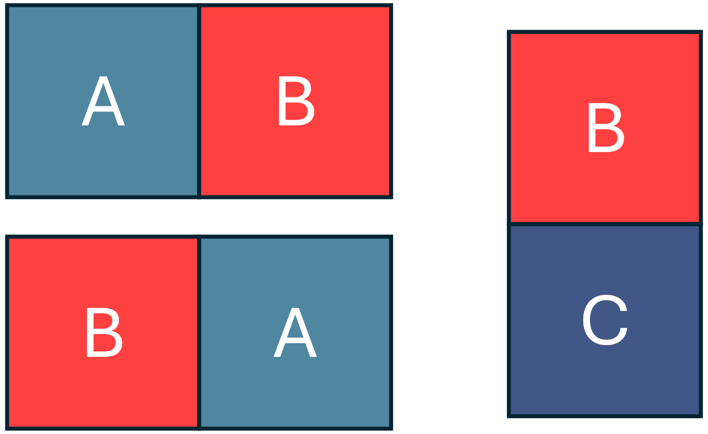


Seeded TA

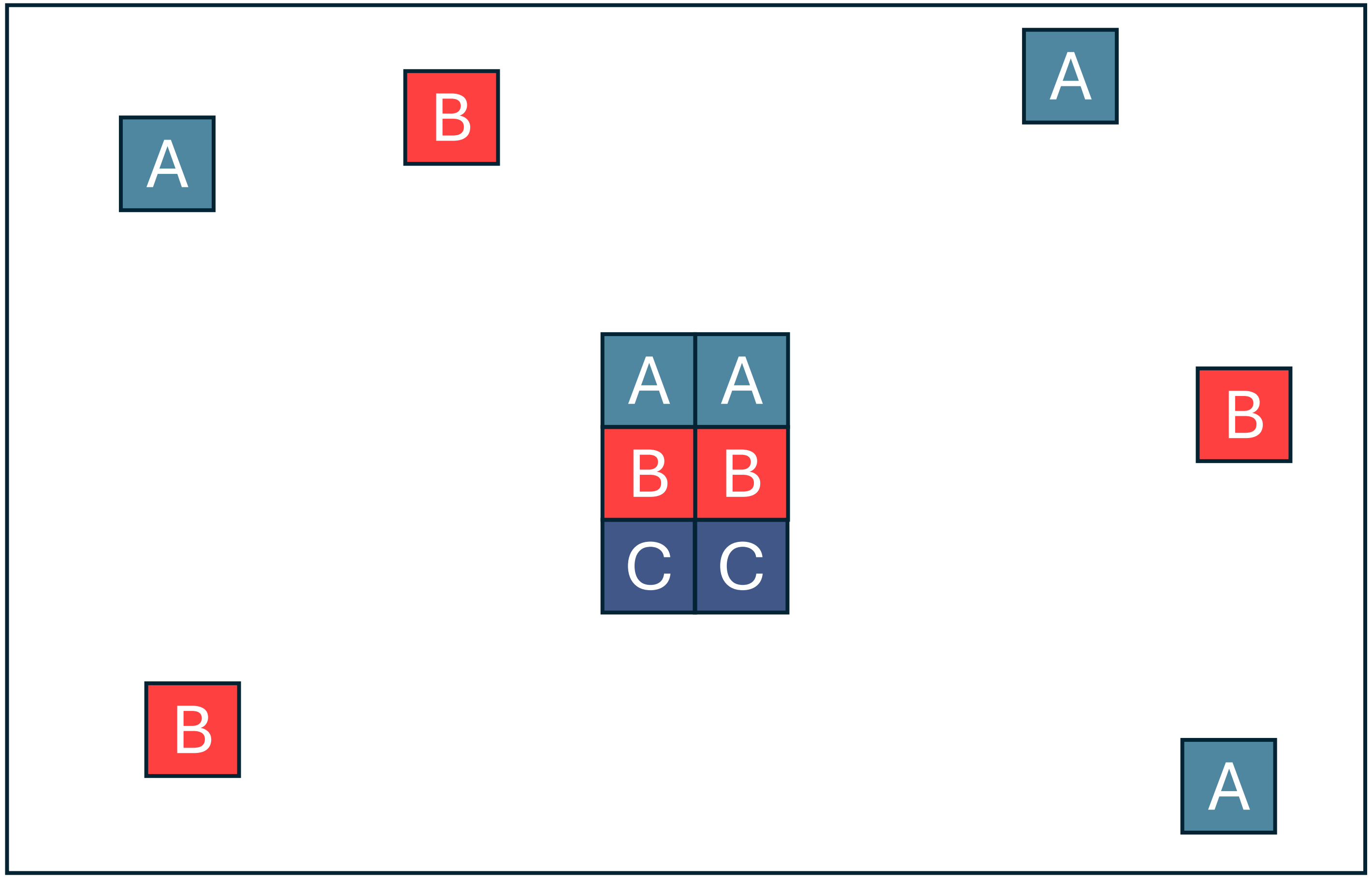
Tiles:

Affinities:

Temperature: τ



System:

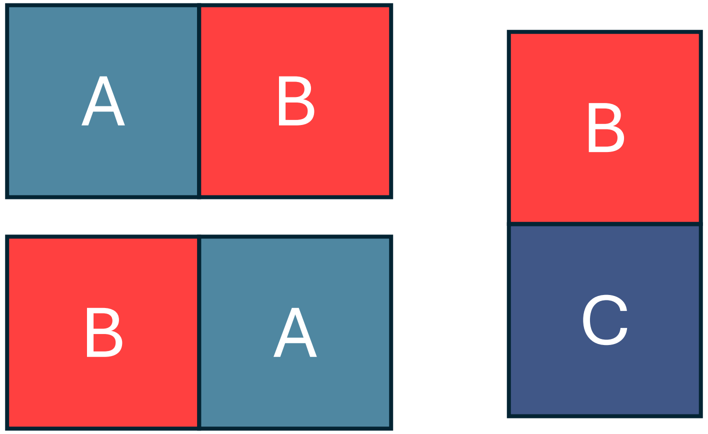


Seeded TA

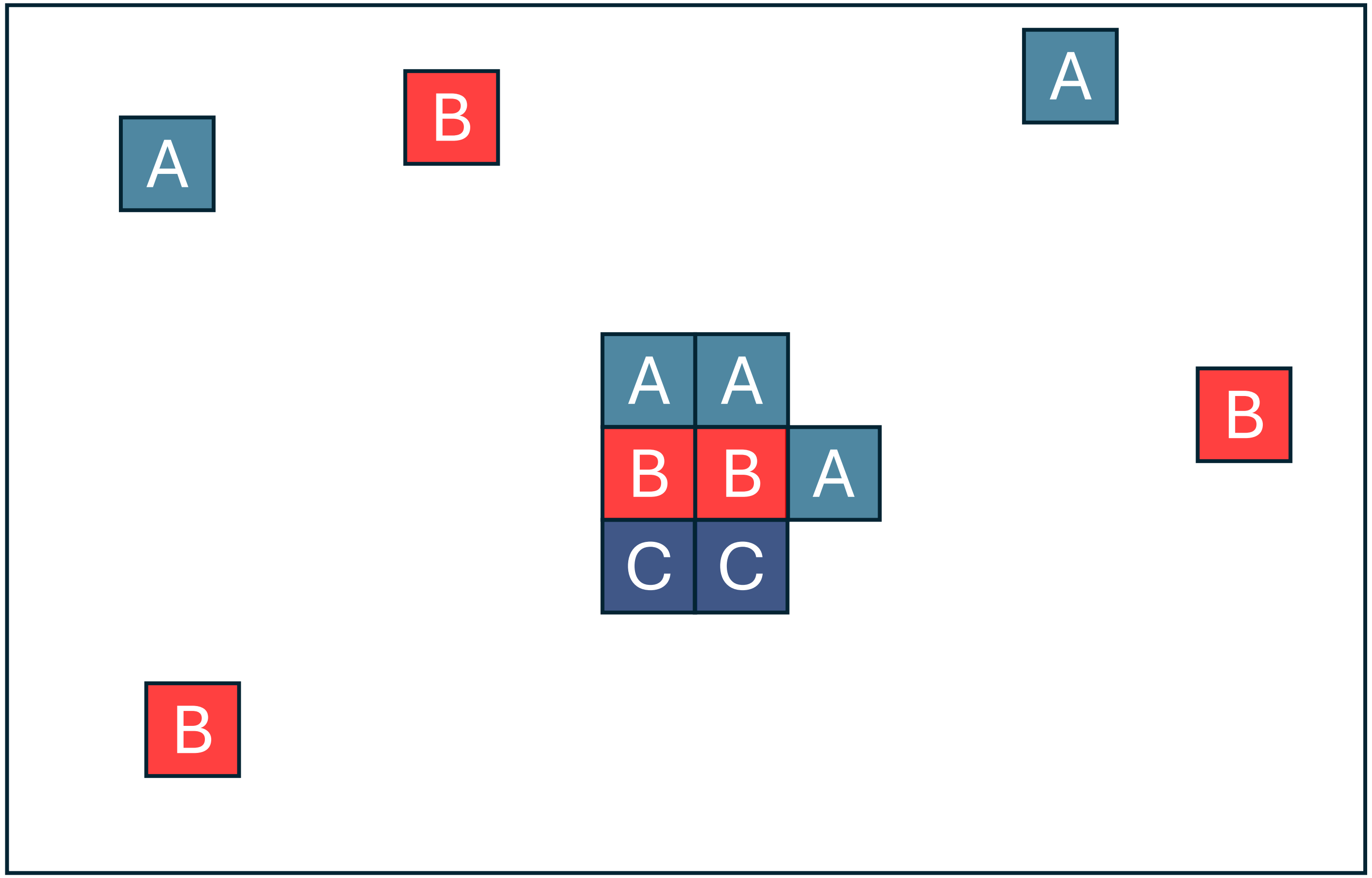
Tiles:

Affinities:

Temperature: τ



System:

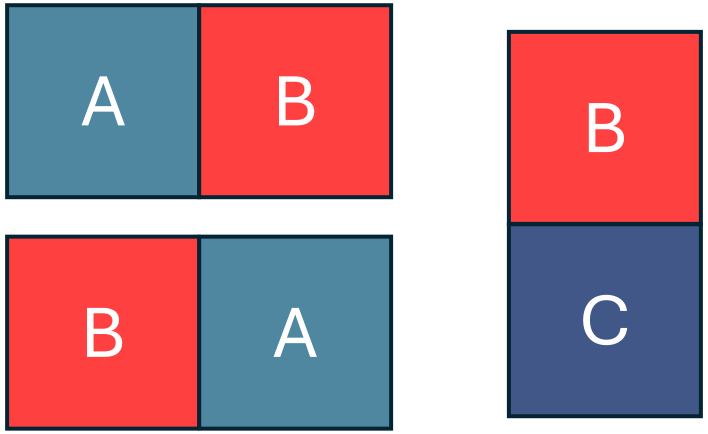


Seeded TA

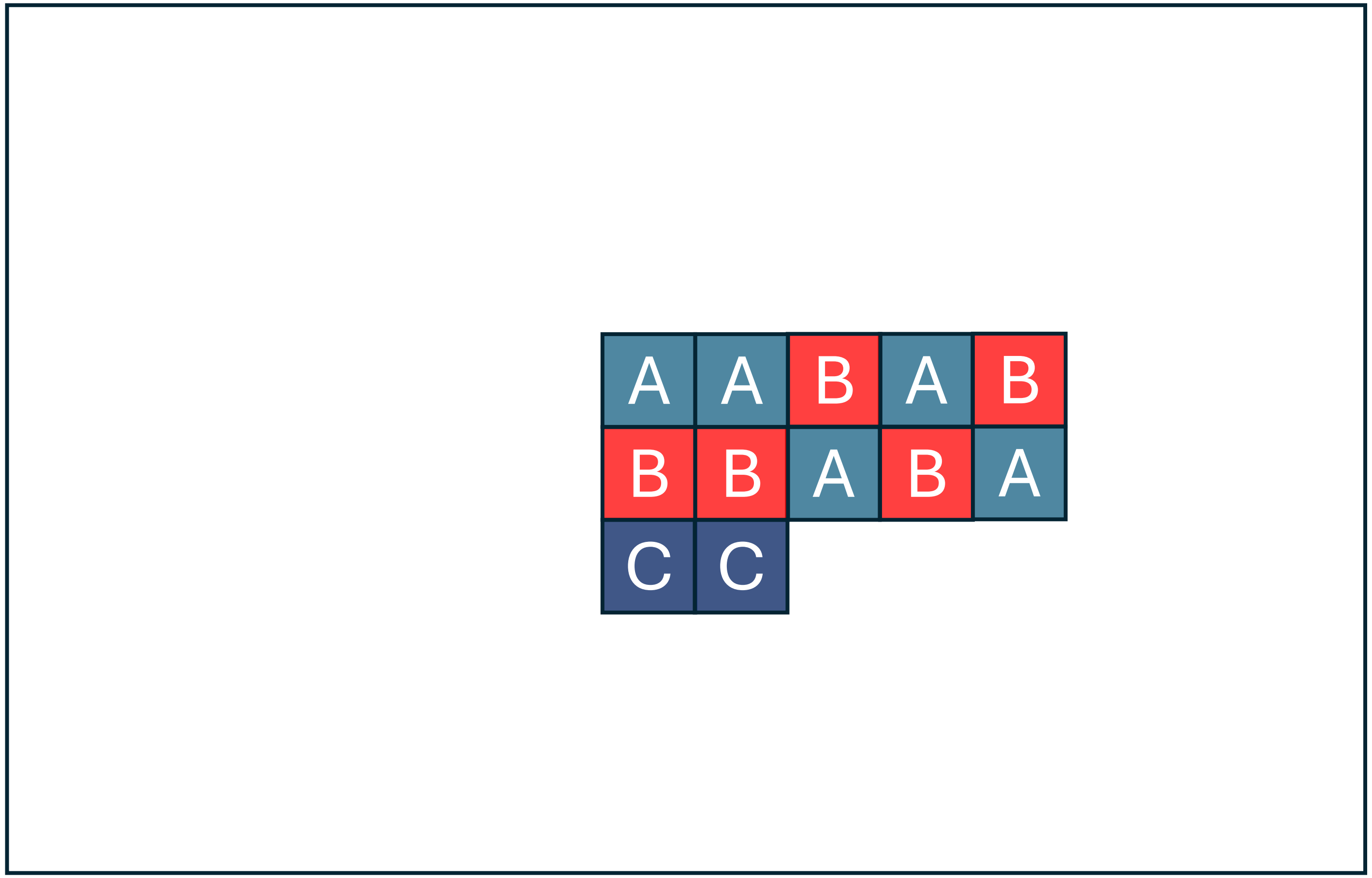
Tiles:

Affinities:

Temperature: τ

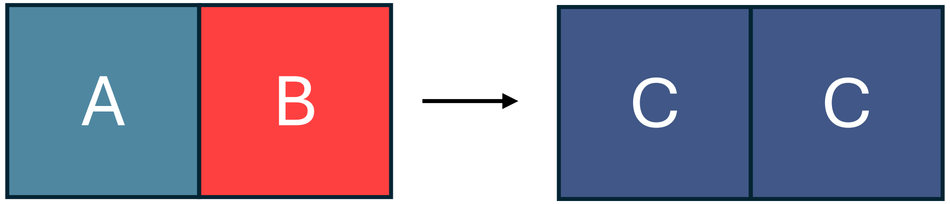


System:



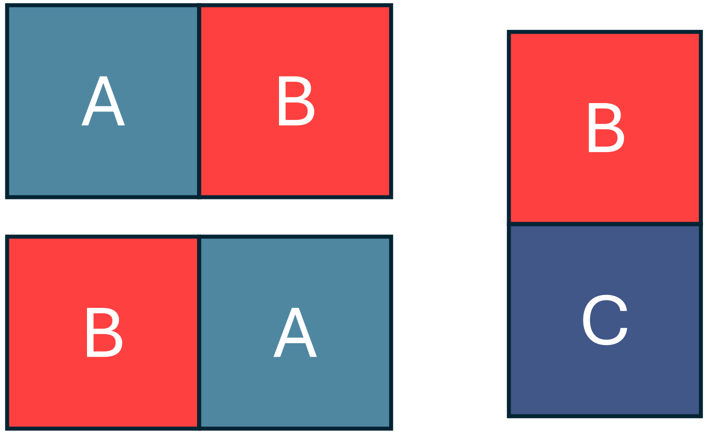
Seeded TA

Transitions:

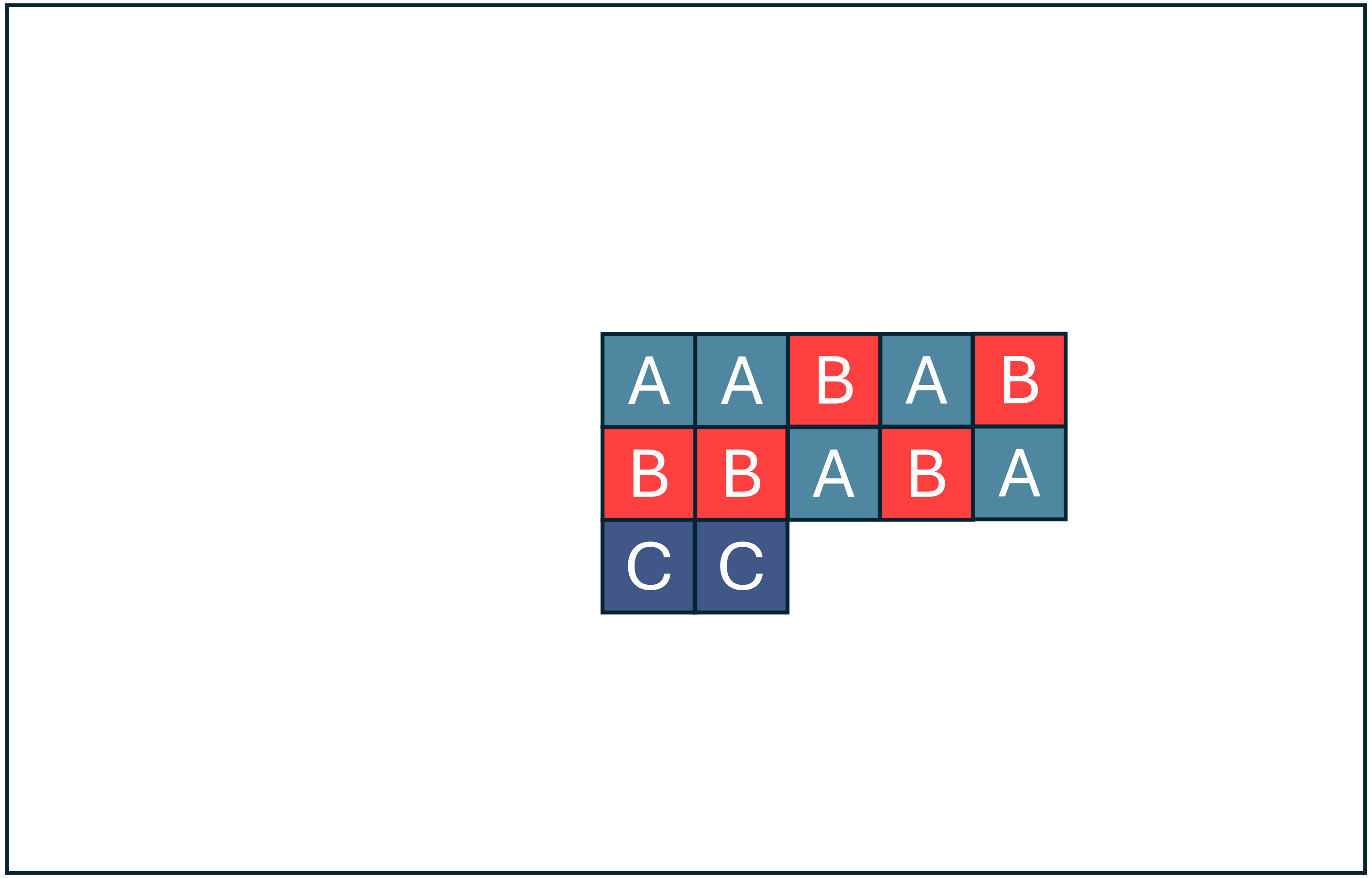


Affinities:

Temperature: τ

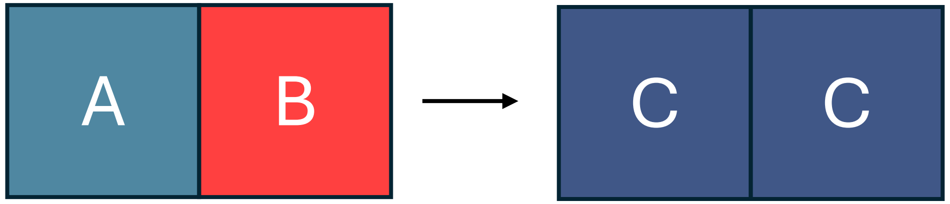


System:



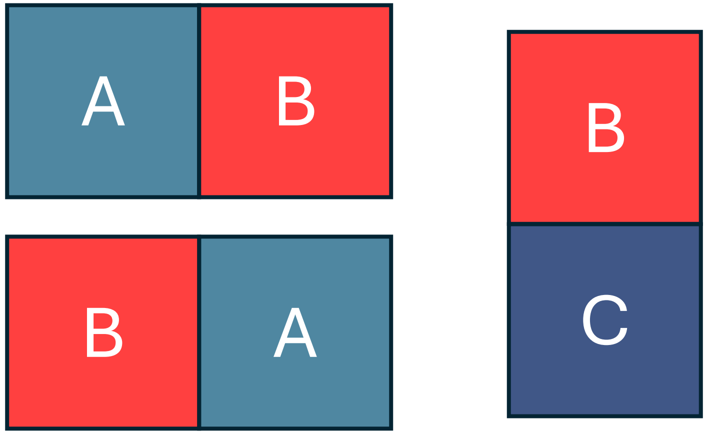
Seeded TA

Transitions:

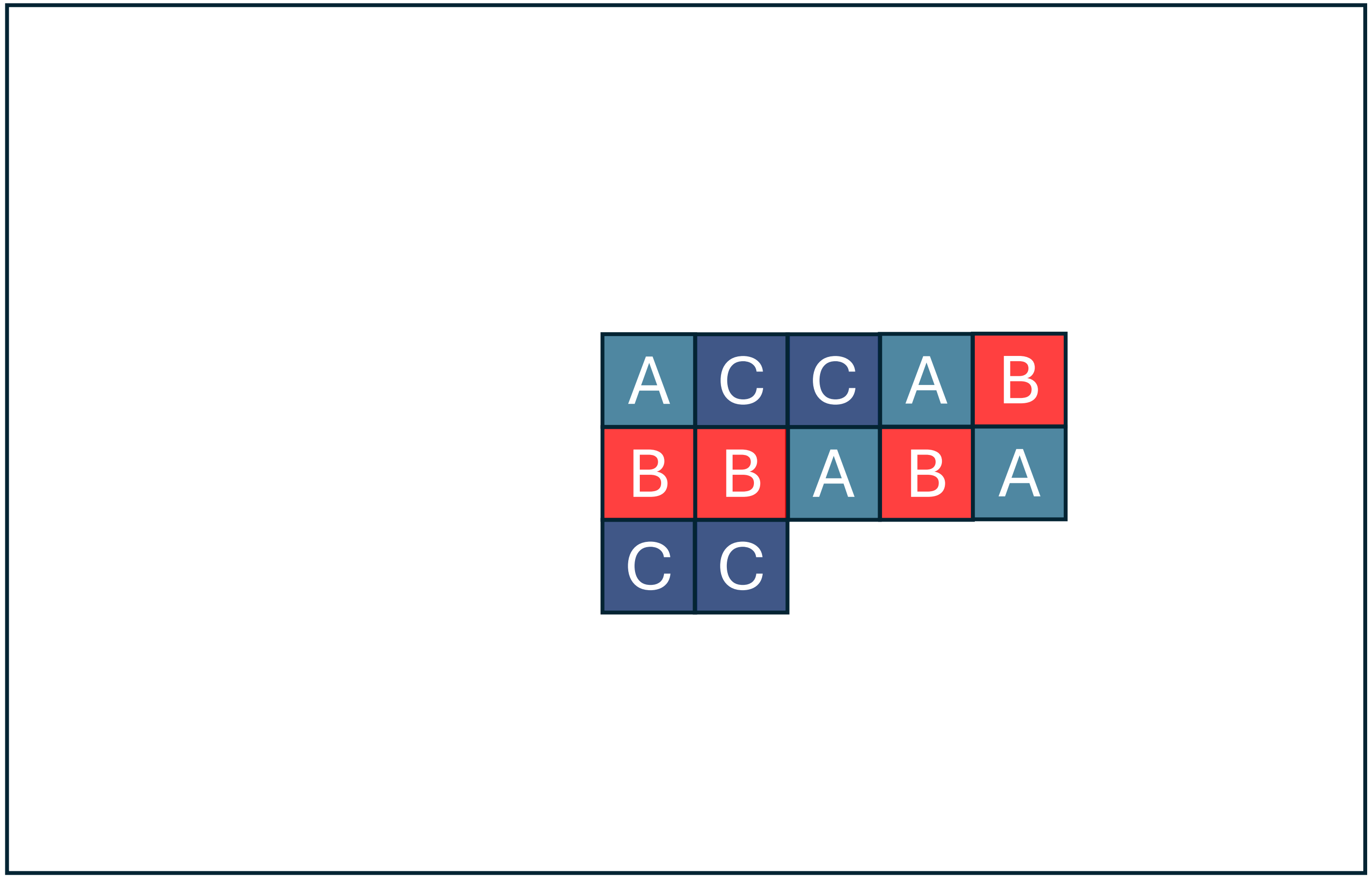


Affinities:

Temperature: τ

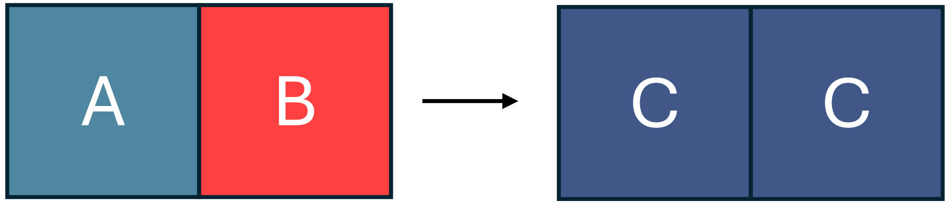


System:



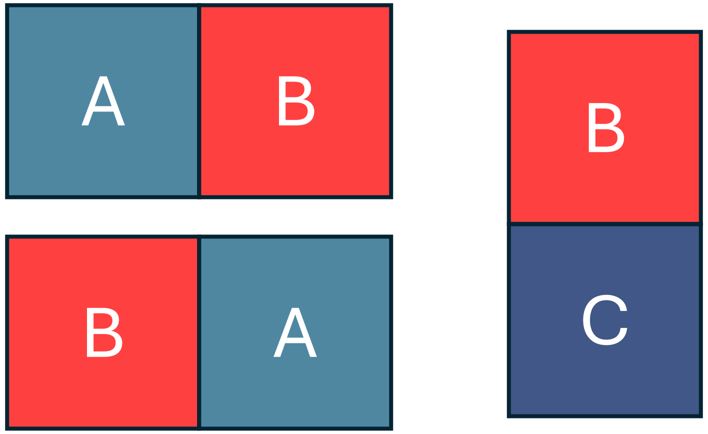
Seeded TA

Transitions:

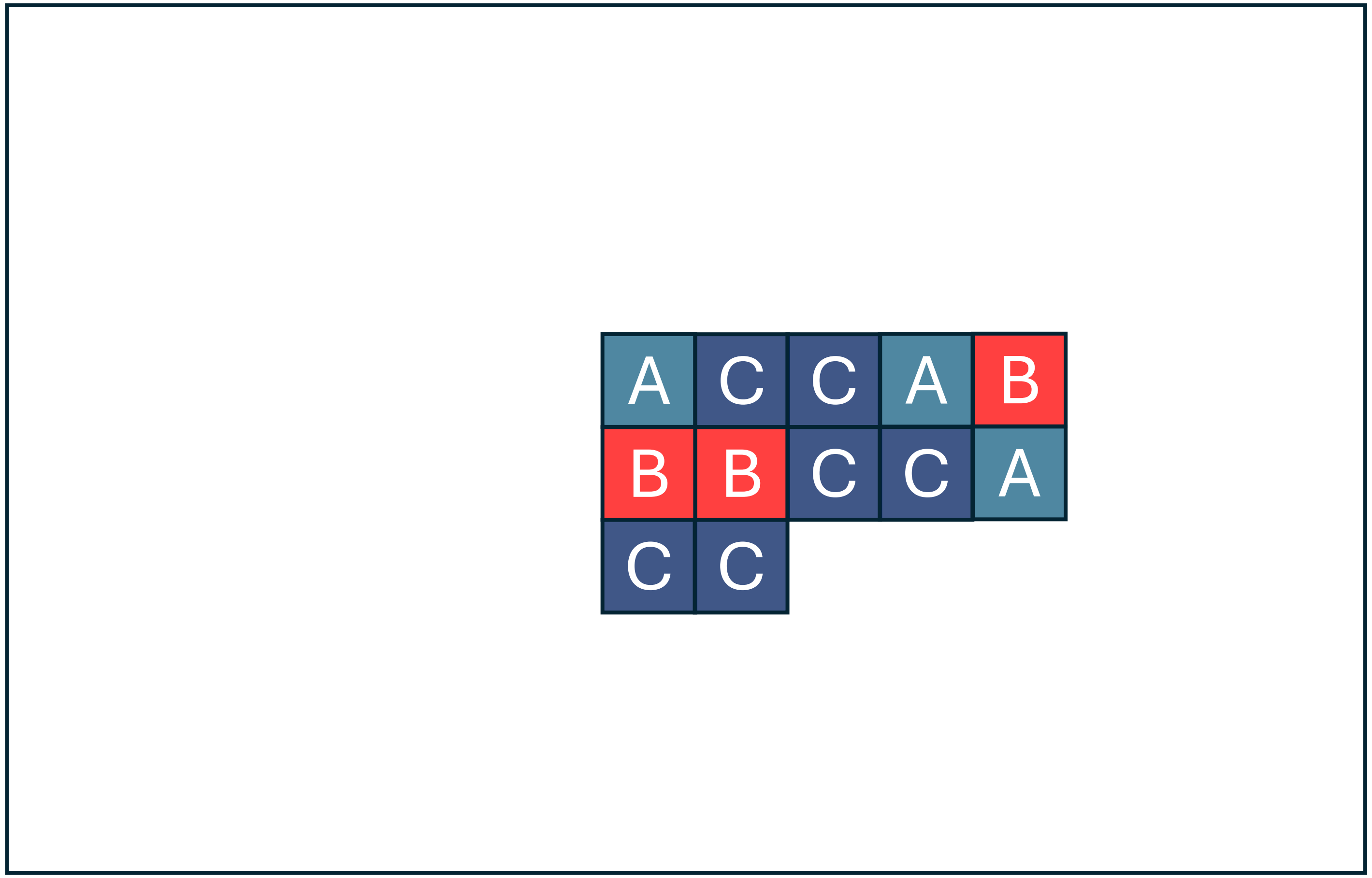


Affinities:

Temperature: τ

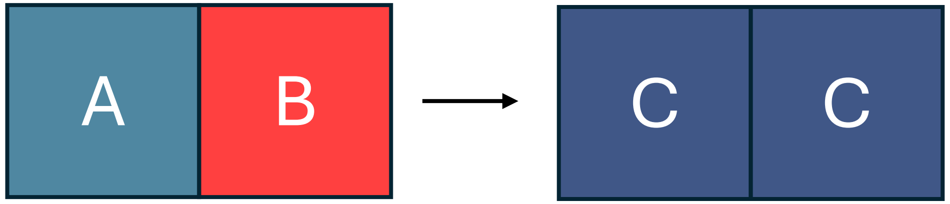


System:



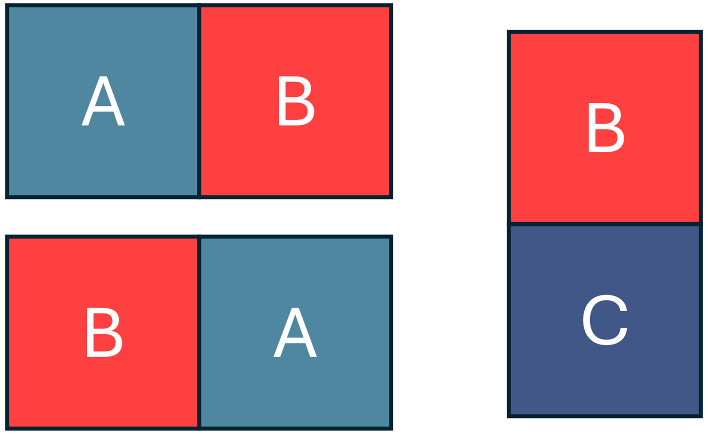
Seeded TA

Transitions:

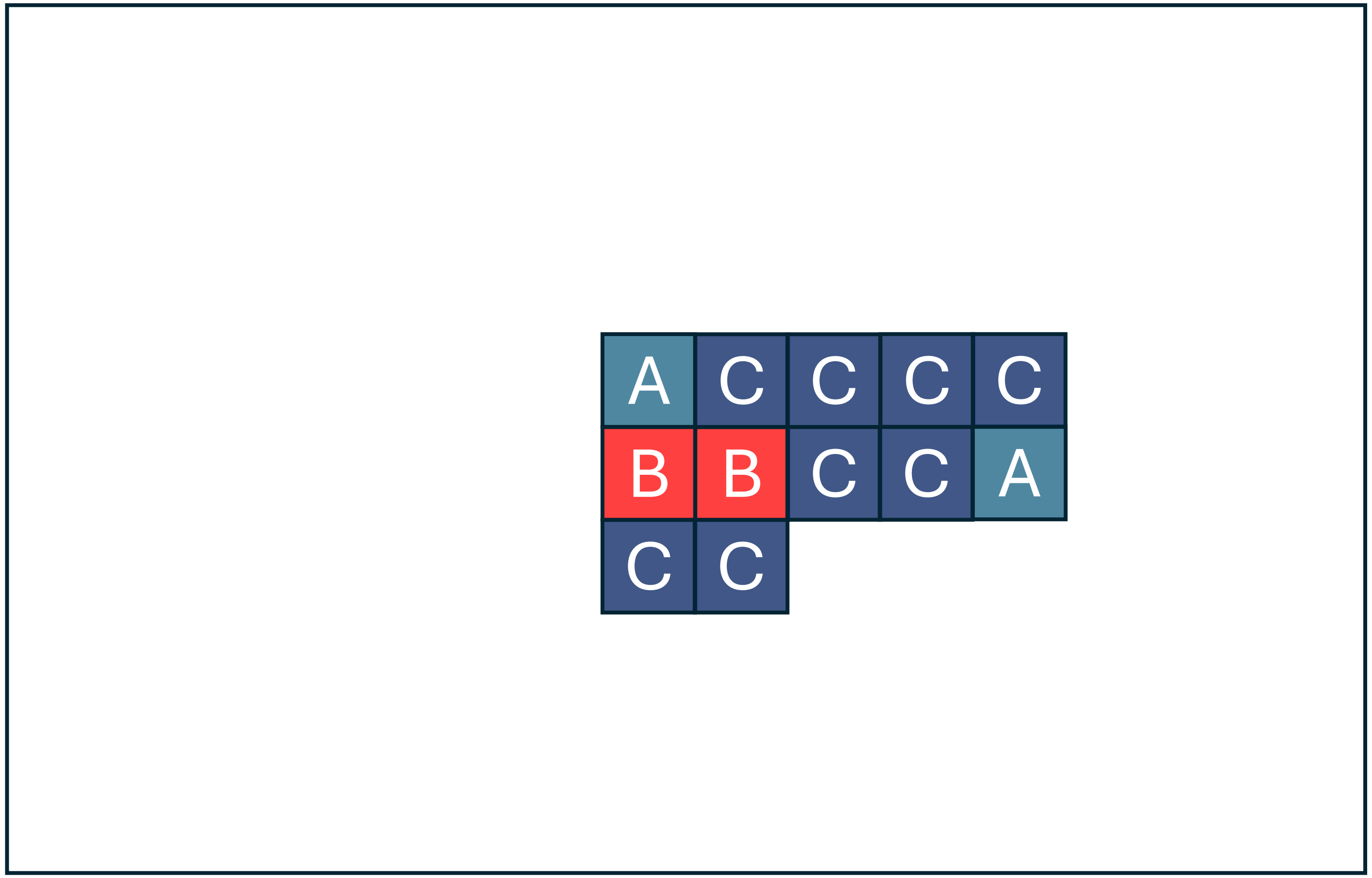


Affinities:

Temperature: τ



System:



What is known

	aTAM	2HAM	STAM	Seeded TA
Description	- Single tile attachments - No detachments - Fixed glues	- Up to 2 assembly attachments - No detachments - Fixed glues	- Up to 2 assembly attachments - Detachments - Glues turn on/off	- **Single tile attachments** - **No detachments** - **Transitions**
Results	- No fractals weakly buildable (temp 1) - Some fractals not strictly buildable (temp > 1)	- Finitely assemble larger class of shapes - Still some non-buildable fractals	- All fractals strictly buildable - Without detachment: some impossible fractals	- **Fractals with generators that have a Ham-path are strictly buildable (temp 1)**

Our Contribution

	aTAM	2HAM	STAM	Seeded TA
Description	- Single tile attachments - No detachments - Fixed glues	- Up to 2 assembly attachments - No detachments - Fixed glues	- Up to 2 assembly attachments - Detachments - Glues turn on/off	- **Single tile attachments** - **No detachments** - **Transitions**
Results	- No fractals weakly buildable (temp 1) - Some fractals not strictly buildable (temp > 1)	- Finitely assemble larger class of shapes - Still some non-buildable fractals	- All fractals strictly buildable - Without detachment: some impossible fractals	- **Fractals with generators that have a Ham-path are strictly buildable (temp 1)**

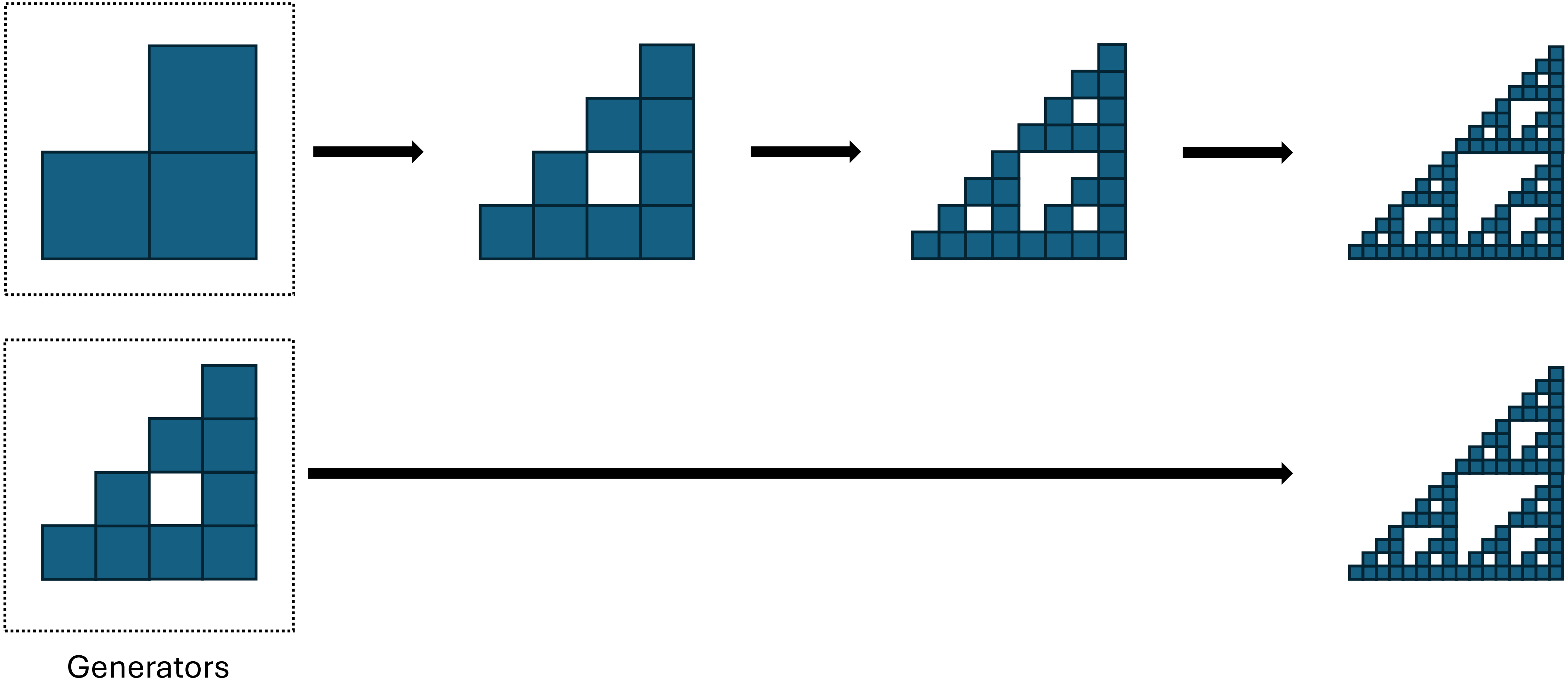
Our Contribution

	aTAM	2HAM	STAM	Seeded TA
Description	- Single tile attachments - No detachments - Fixed glues	- Up to 2 assembly attachments - No detachments - Fixed glues	- Up to 2 assembly attachments - Detachments - Glues turn on/off	- **Single tile attachments** - **No detachments** - **Transitions**
Results	- No fractals weakly buildable (temp 1) - Some fractals not strictly buildable (temp > 1)	- Finitely assemble larger class of shapes - Still some non-buildable fractals	- All fractals strictly buildable - Without detachment: some impossible fractals	- **ALL fractals are strictly buildable (temp 1)** - **There exists a single system that does so**

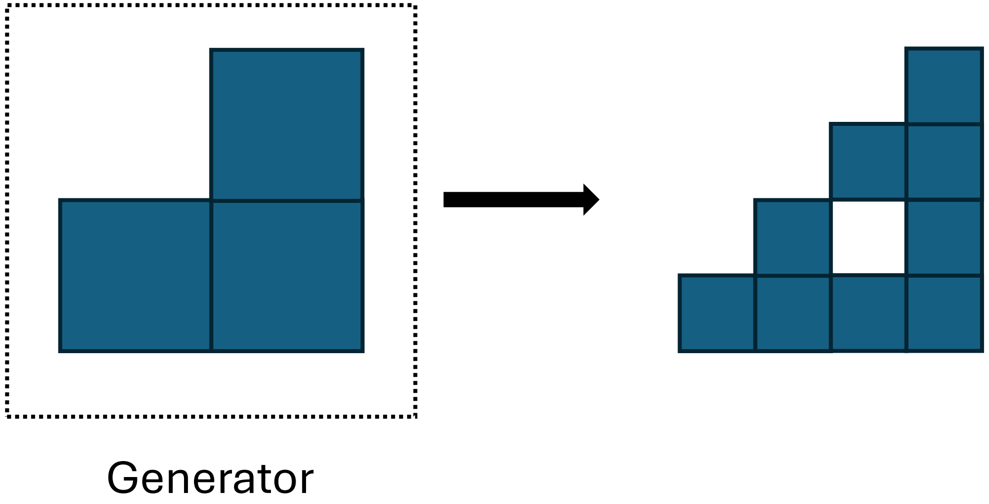
The Construction

How did we do it?

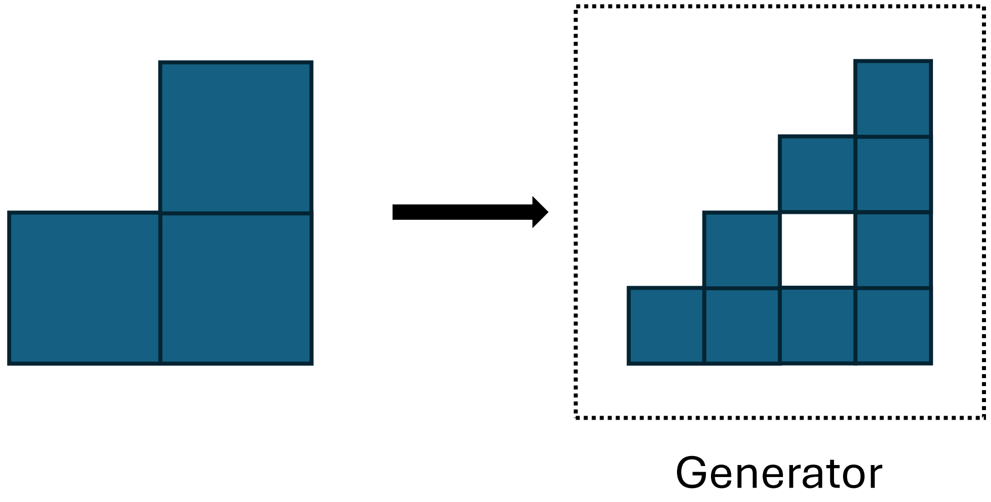
Important Observation



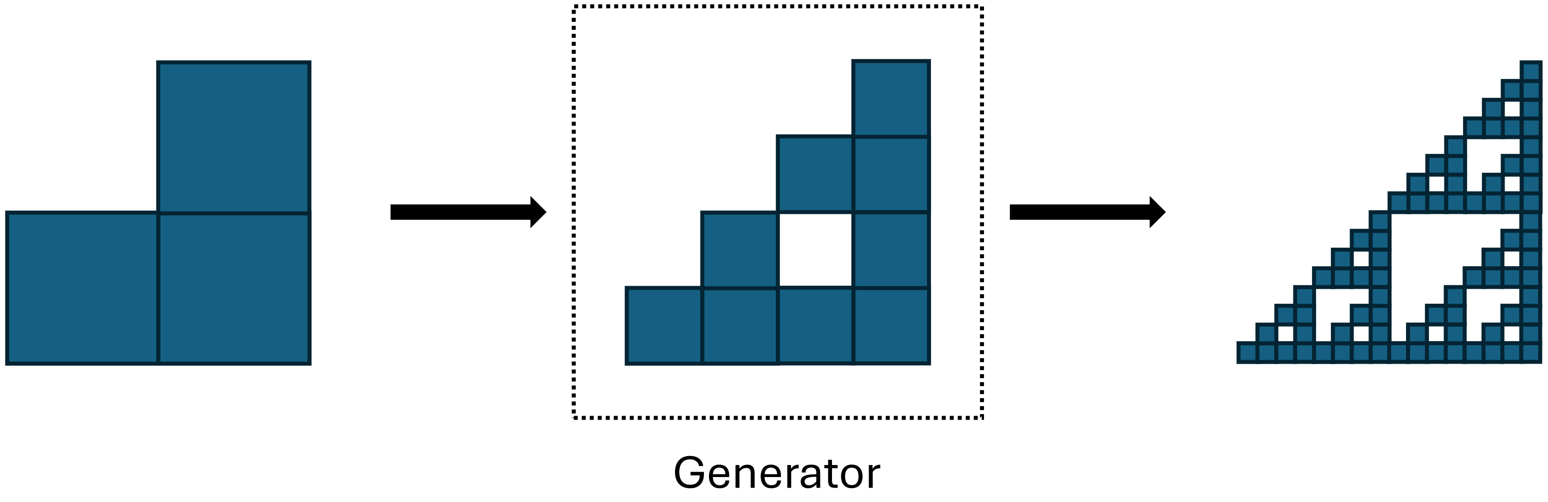
Important Observation



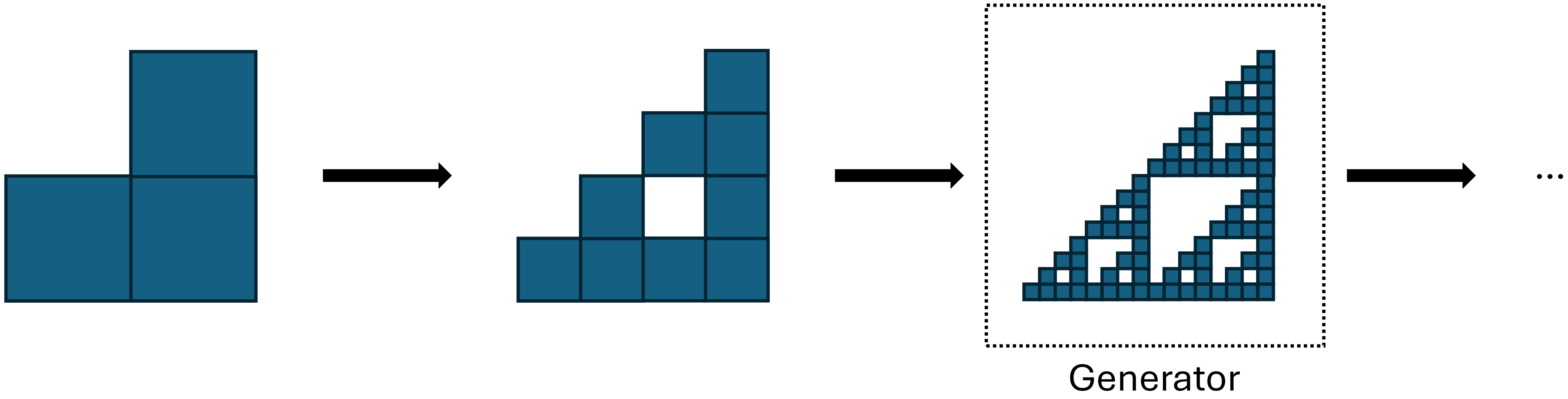
Important Observation



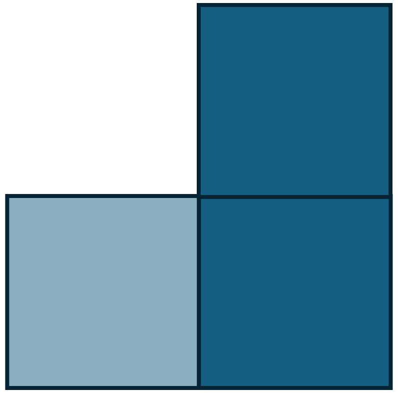
Important Observation

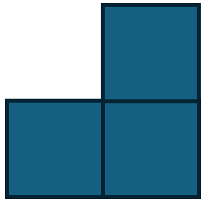


Important Observation



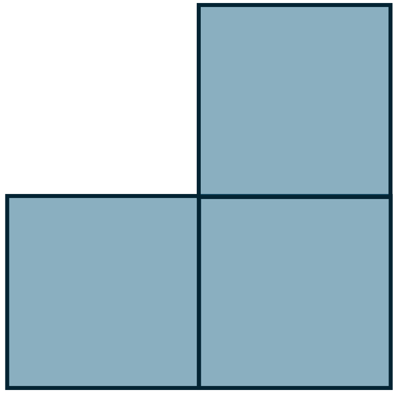
High Level Idea

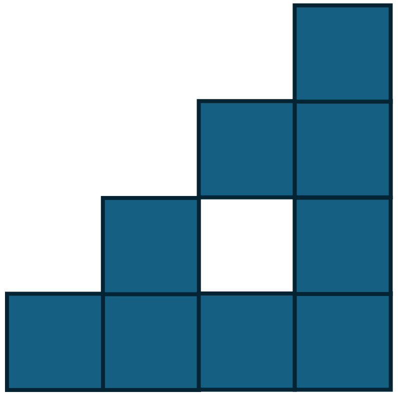




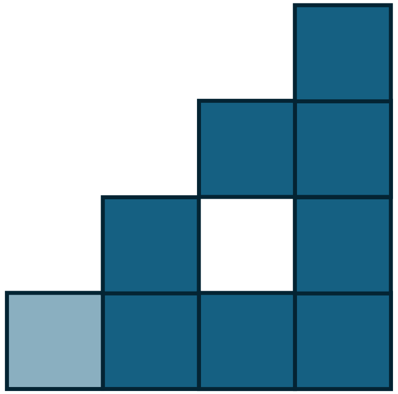
High Level Idea

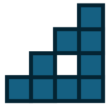
High Level Idea



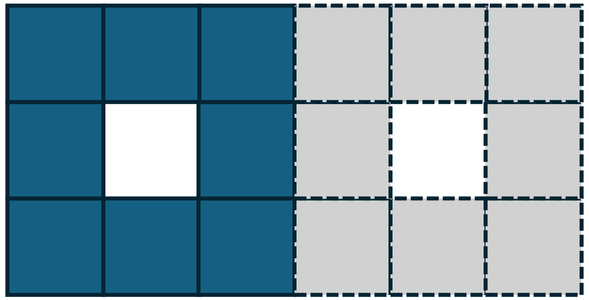


High Level Idea

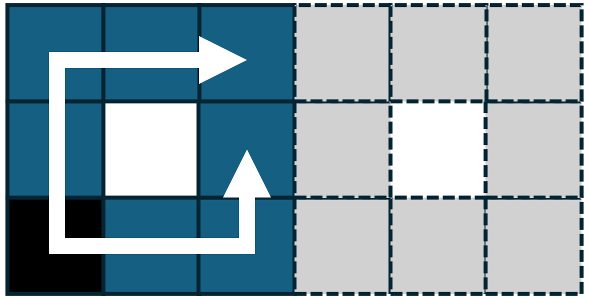




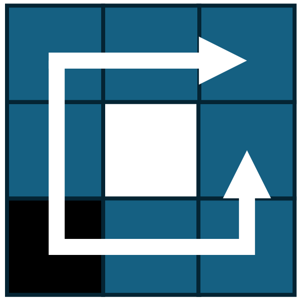
Step 1: Devising an Order

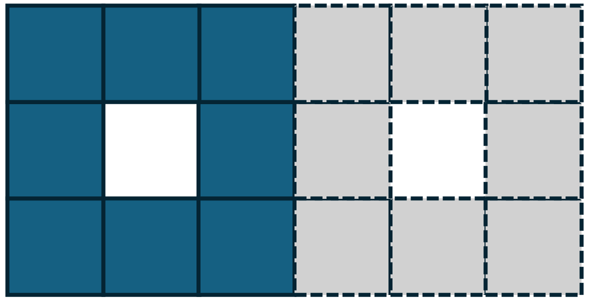


Step 1: Devising an Order

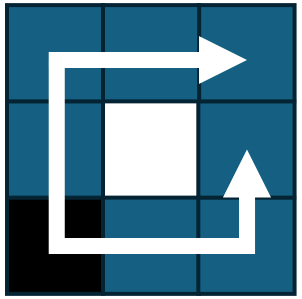


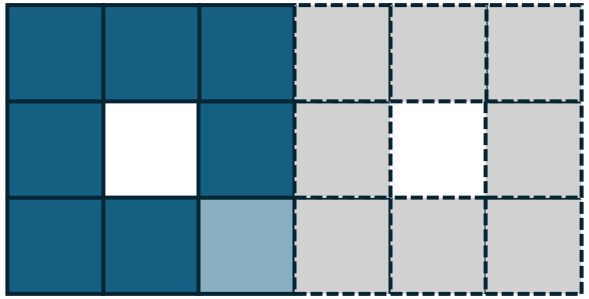
Step 2: A Copying Procedure



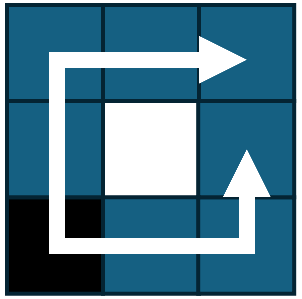


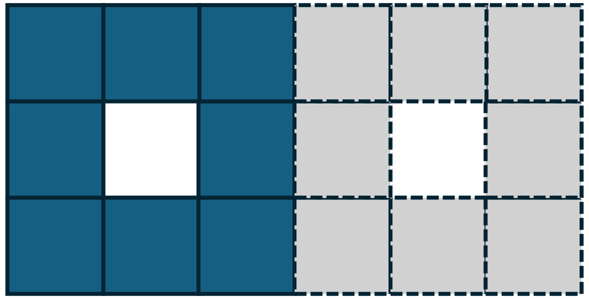
Step 2: A Copying Procedure



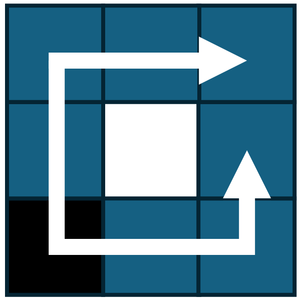


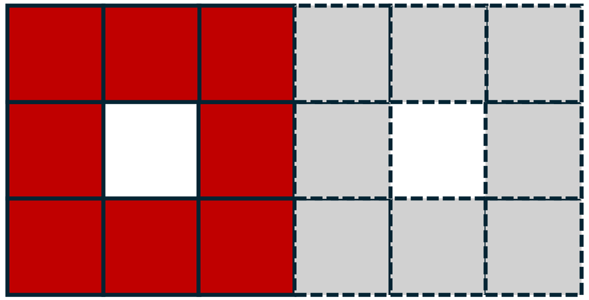
Step 2: A Copying Procedure



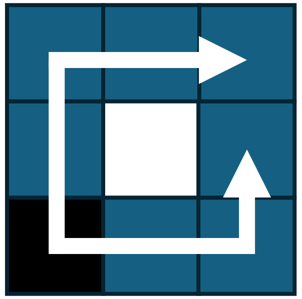


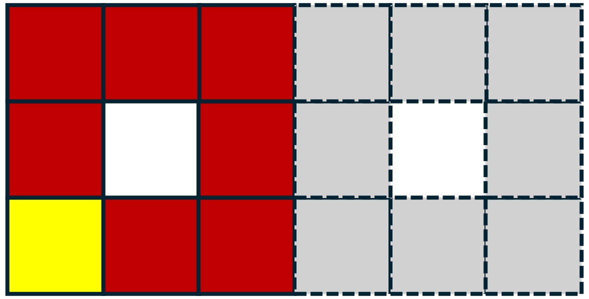
Step 2: A Copying Procedure



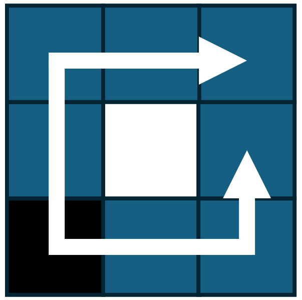


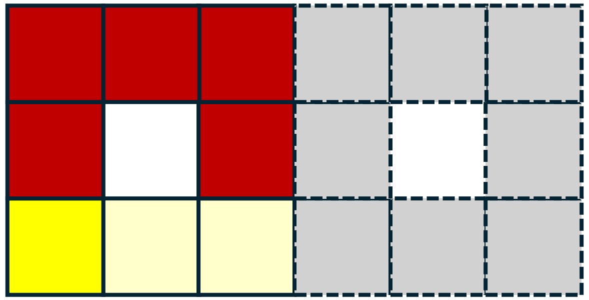
Step 2: A Copying Procedure



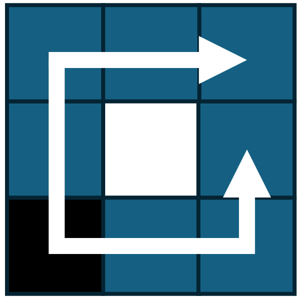


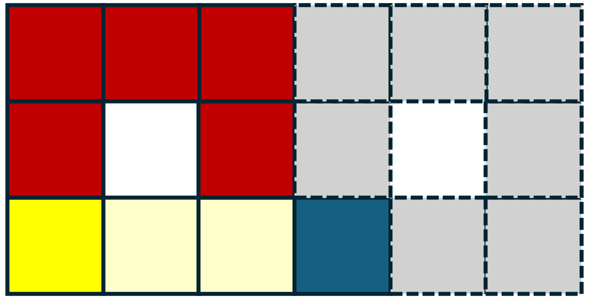
Step 2: A Copying Procedure



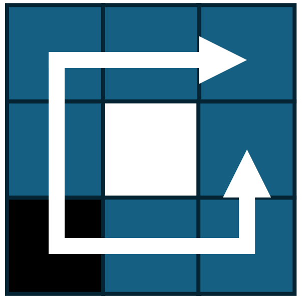


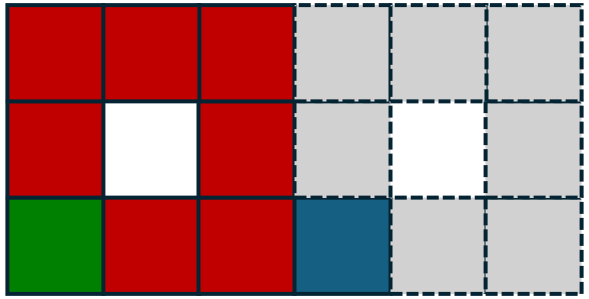
Step 2: A Copying Procedure



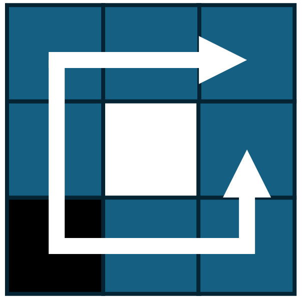


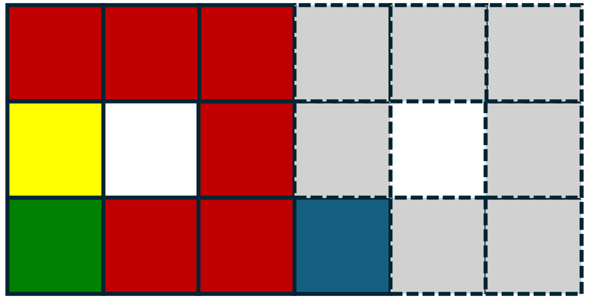
Step 2: A Copying Procedure



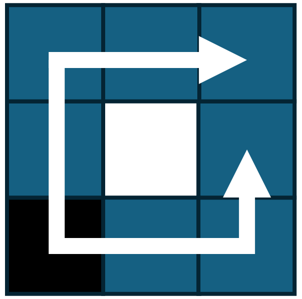


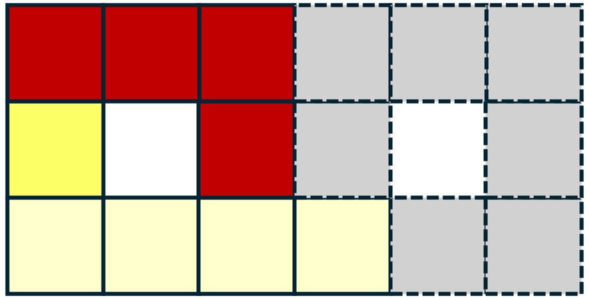
Step 2: A Copying Procedure



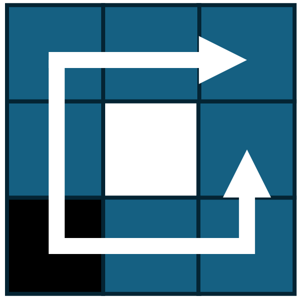


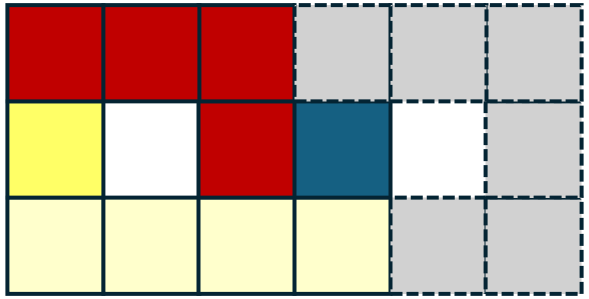
Step 2: A Copying Procedure



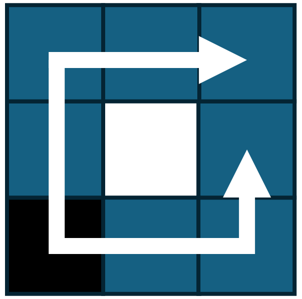


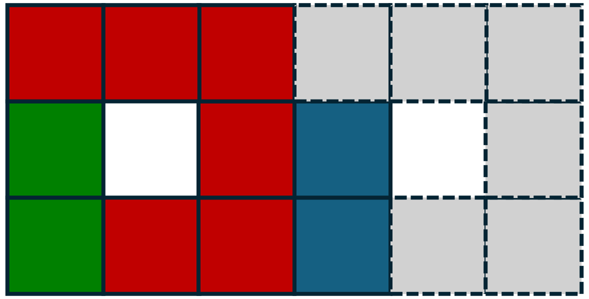
Step 2: A Copying Procedure



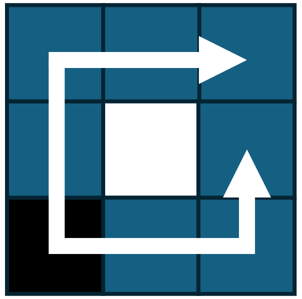


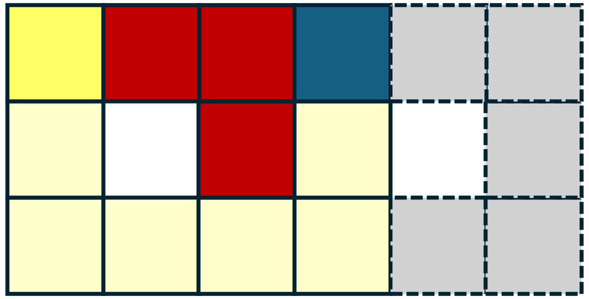
Step 2: A Copying Procedure



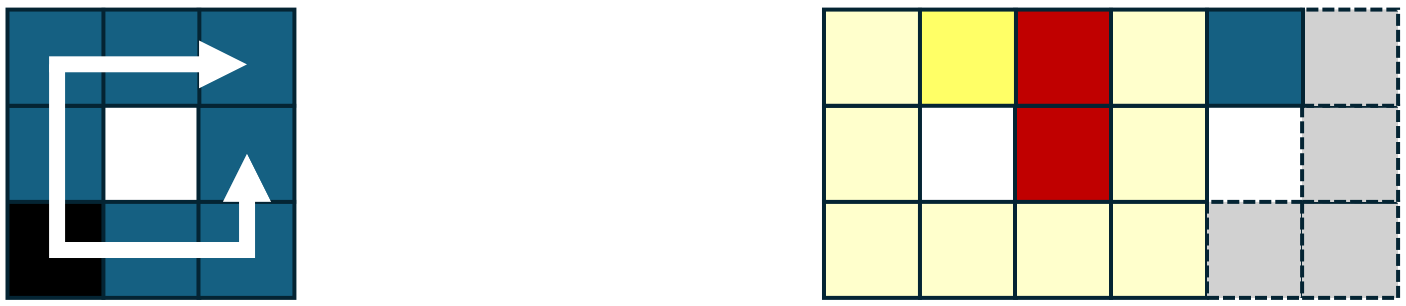


Step 2: A Copying Procedure

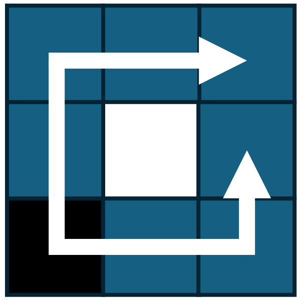


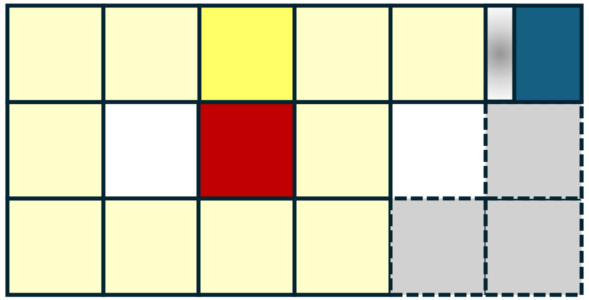


Step 2: A Copying Procedure

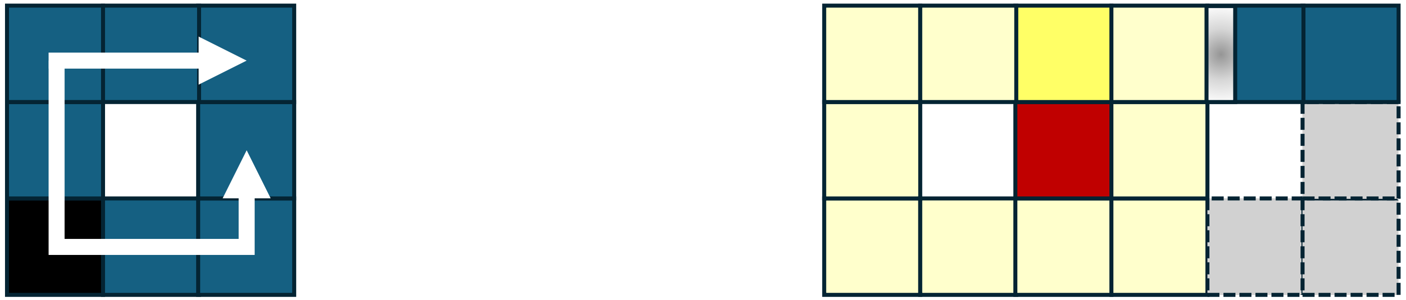


Step 2: A Copying Procedure

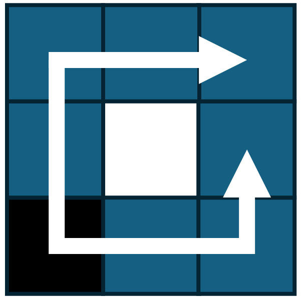


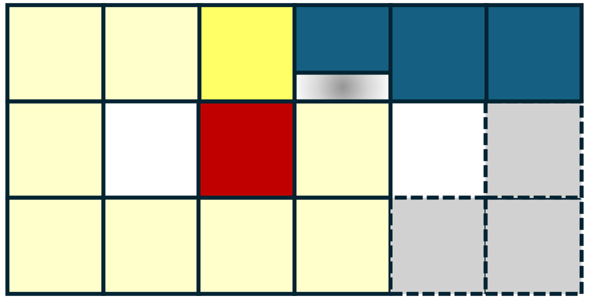


Step 2: A Copying Procedure

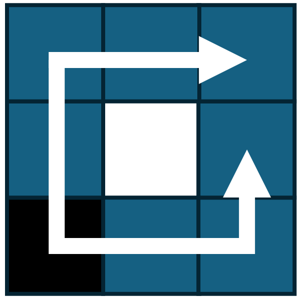


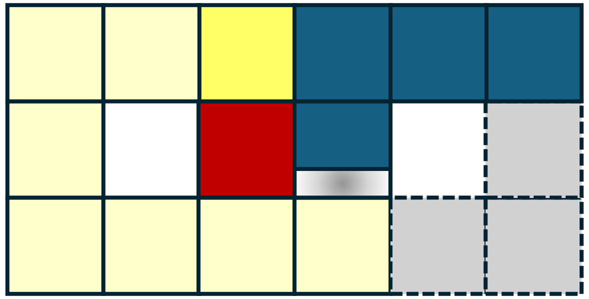
Step 2: A Copying Procedure



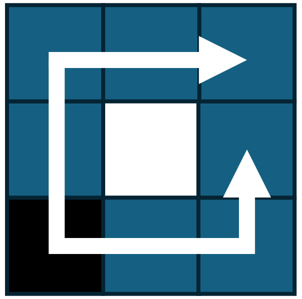


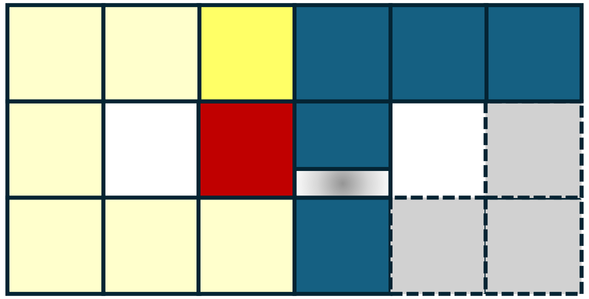
Step 2: A Copying Procedure



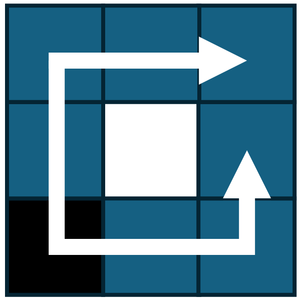


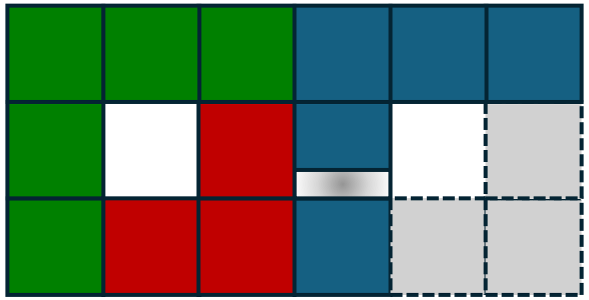
Step 2: A Copying Procedure



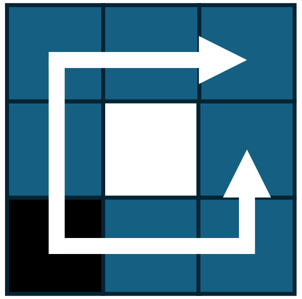


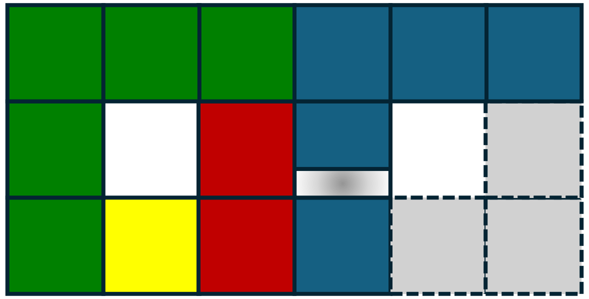
Step 2: A Copying Procedure



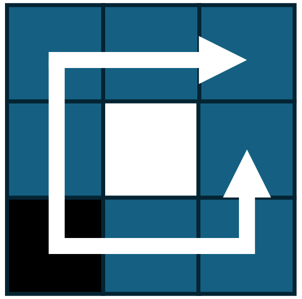


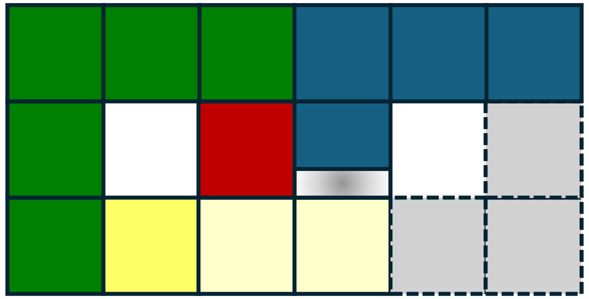
Step 2: A Copying Procedure



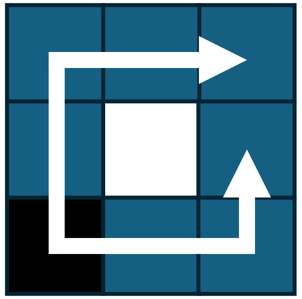


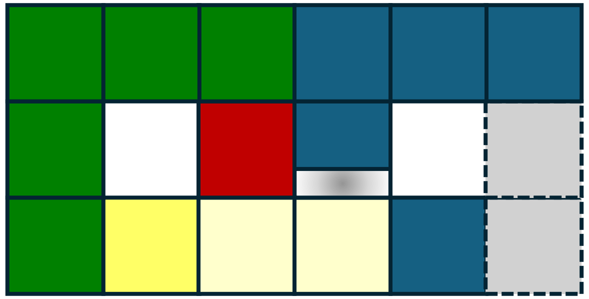
Step 2: A Copying Procedure



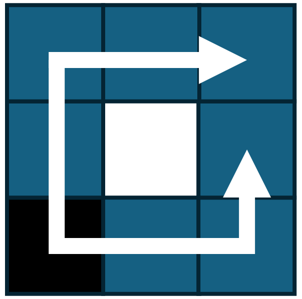


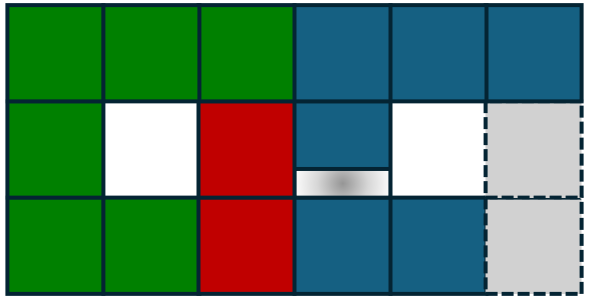
Step 2: A Copying Procedure



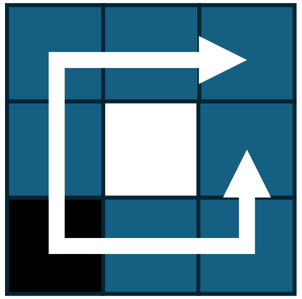


Step 2: A Copying Procedure

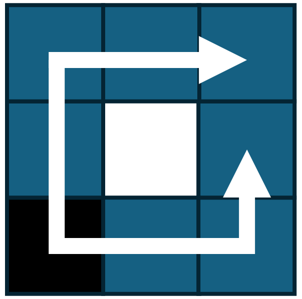




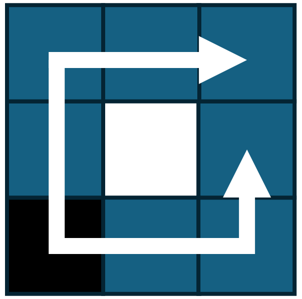
Step 2: A Copying Procedure

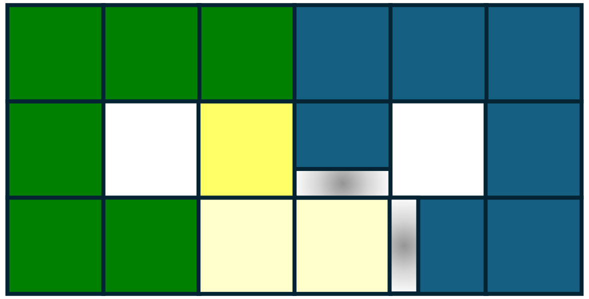


Step 2: A Copying Procedure

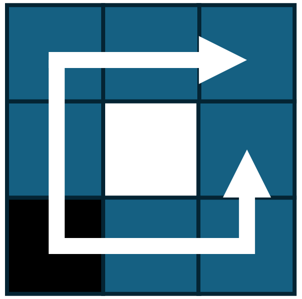


Step 2: A Copying Procedure

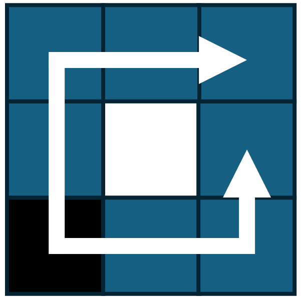


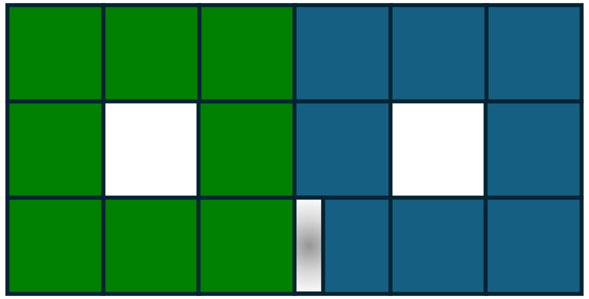


Step 2: A Copying Procedure

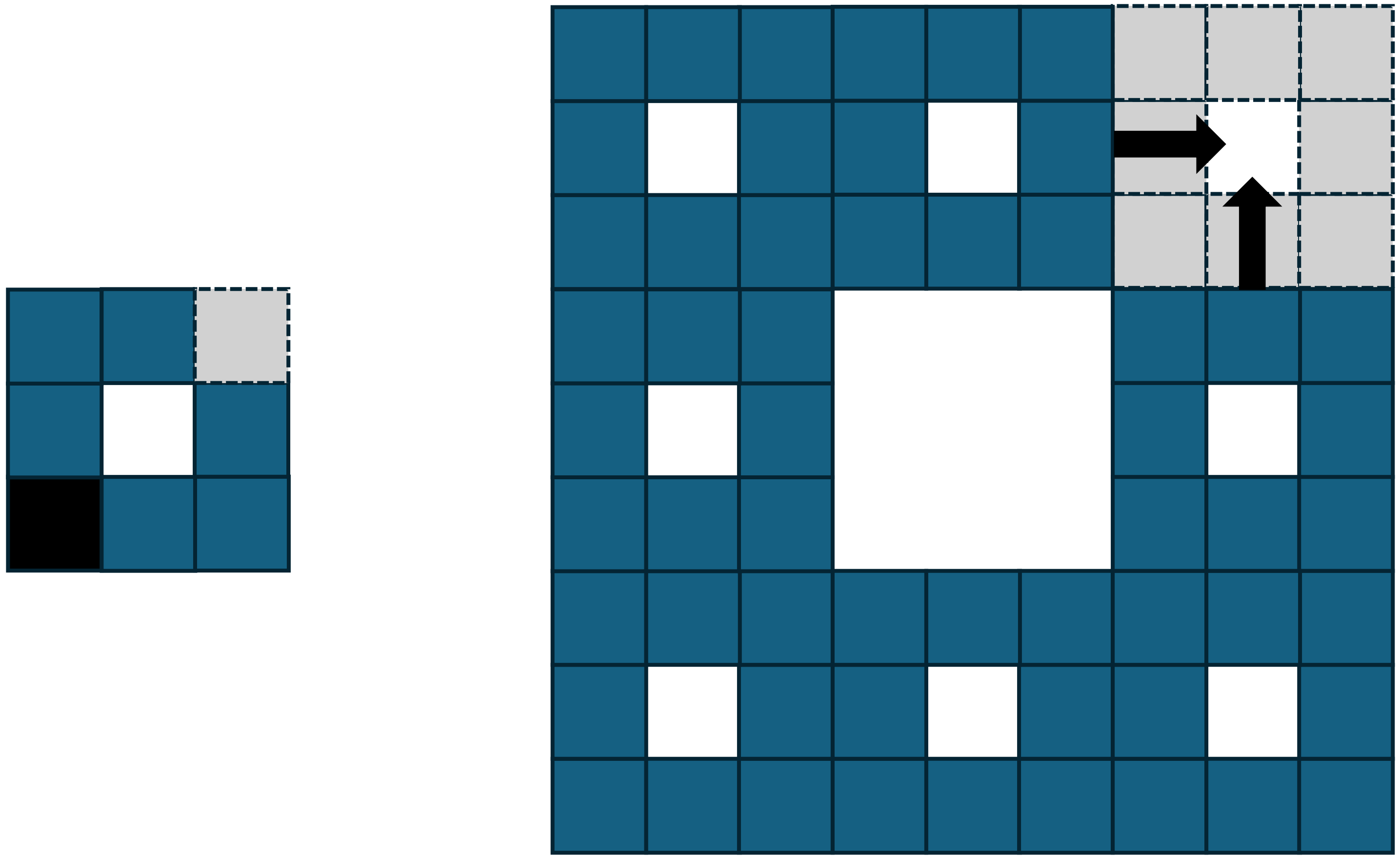


Step 2: A Copying Procedure

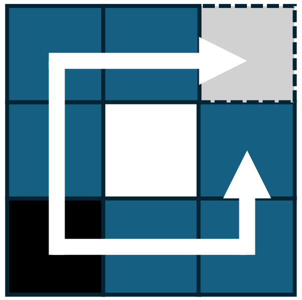


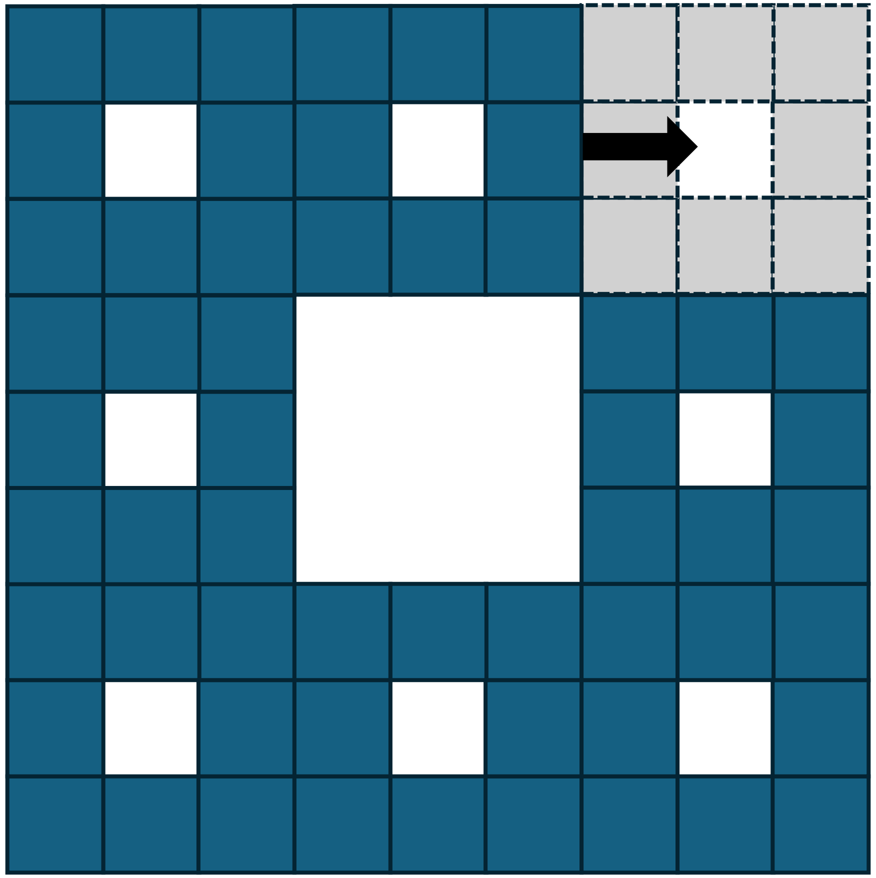


Step 3: Growing the Full Assembly

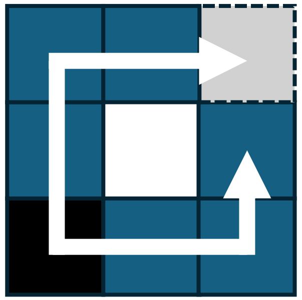


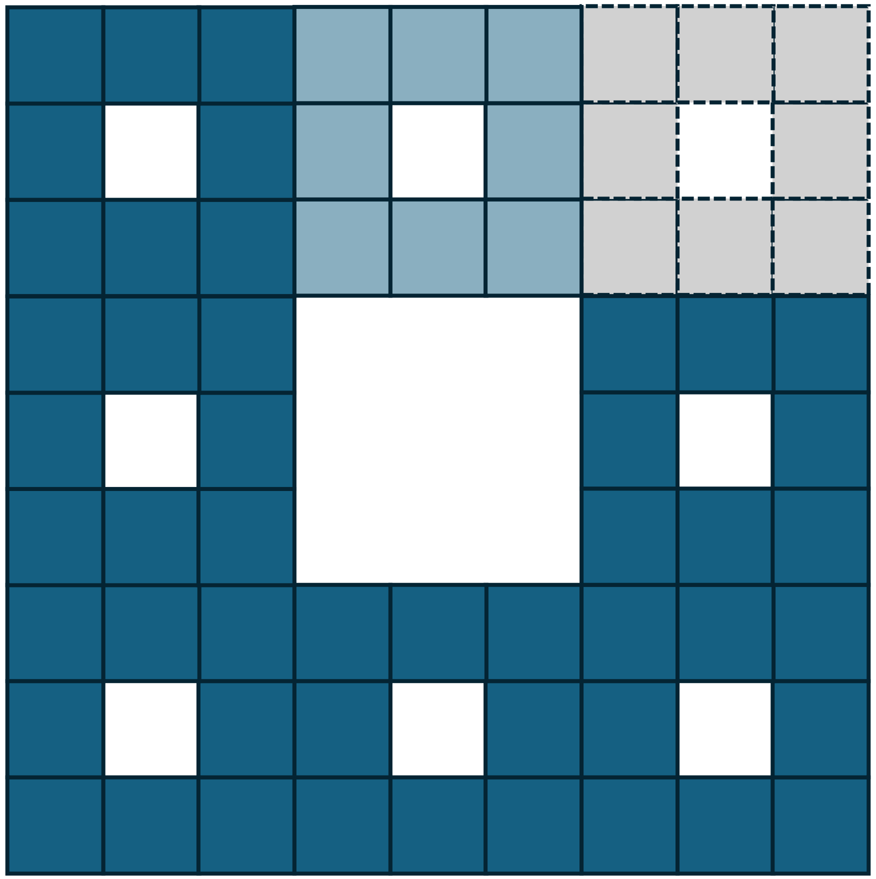
Step 3: Growing the Full Assembly



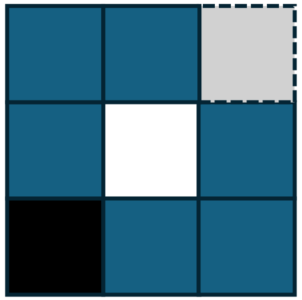


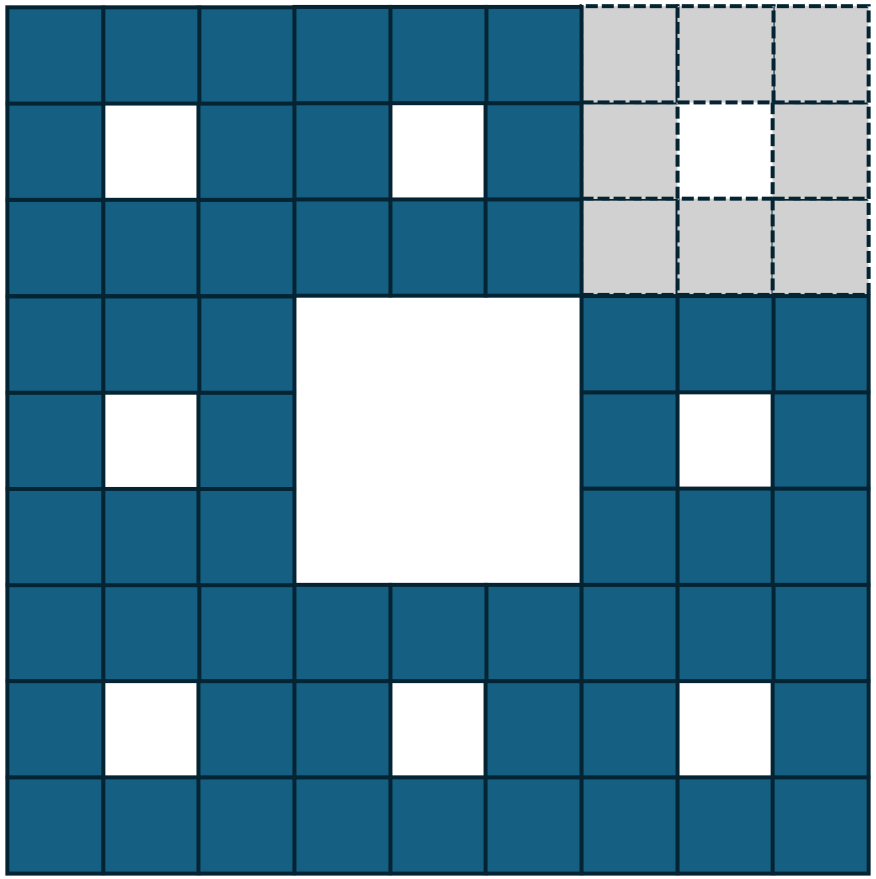
Step 3: Growing the Full Assembly



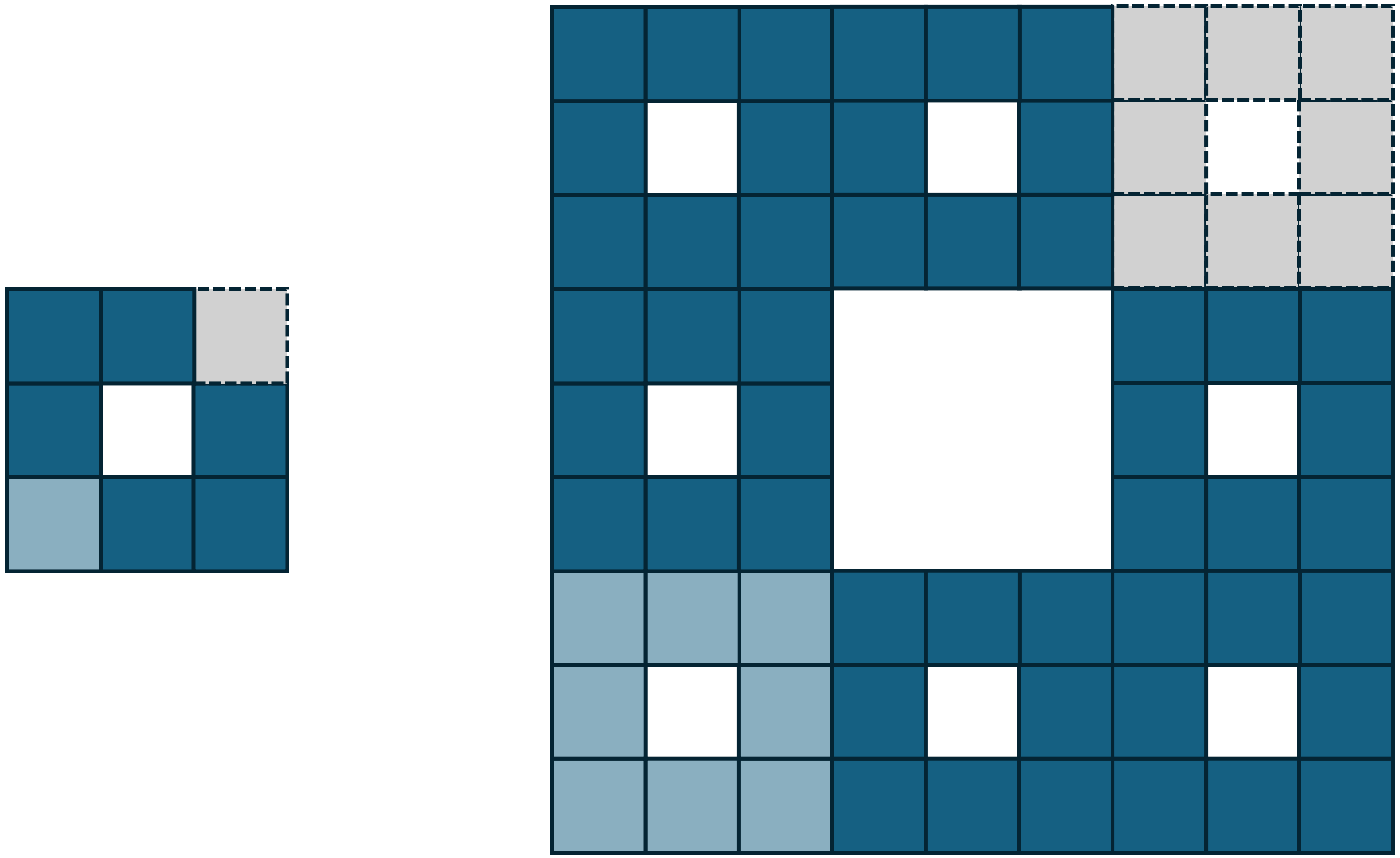


Step 3: Growing the Full Assembly

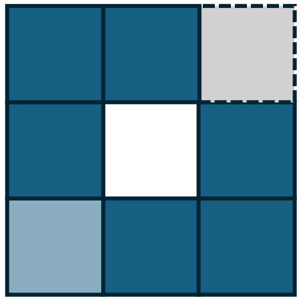


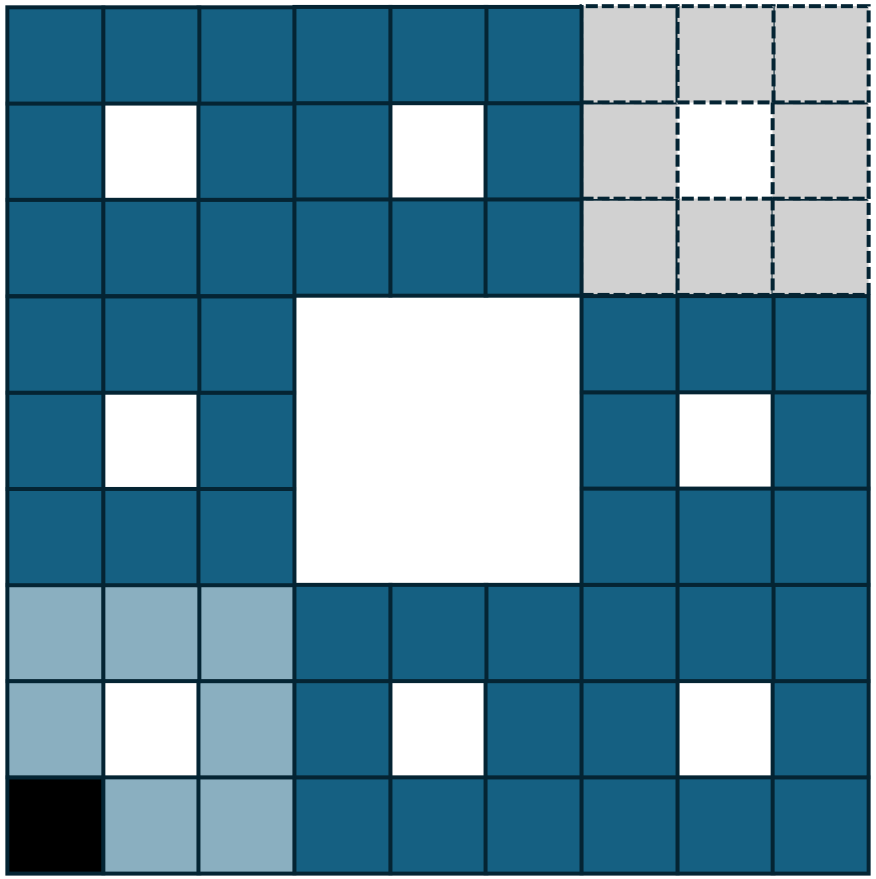


Step 3: Growing the Full Assembly

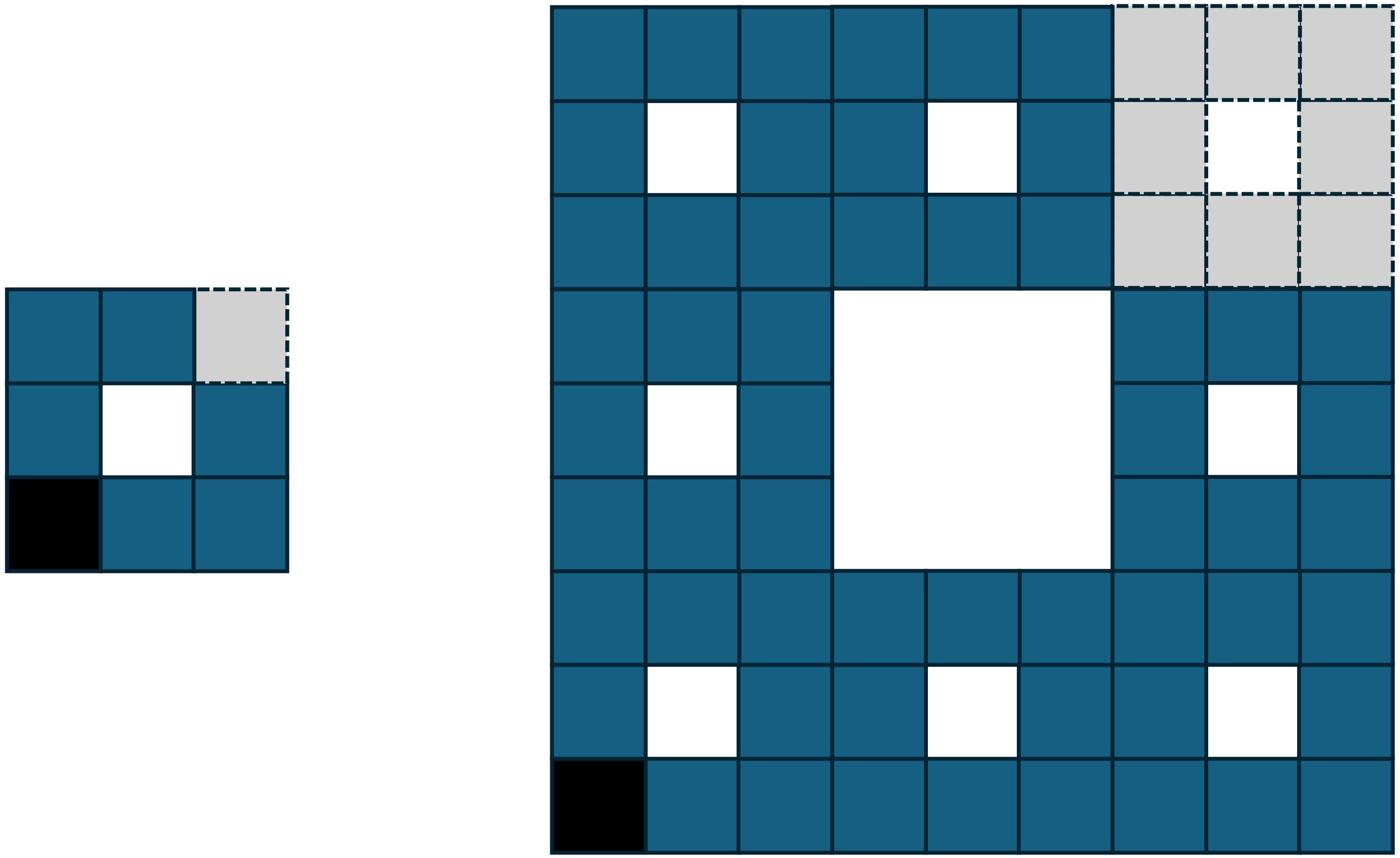


Step 3: Growing the Full Assembly

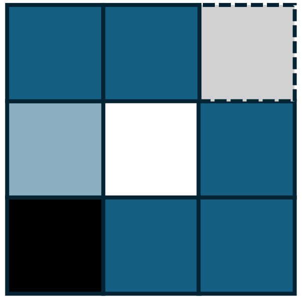


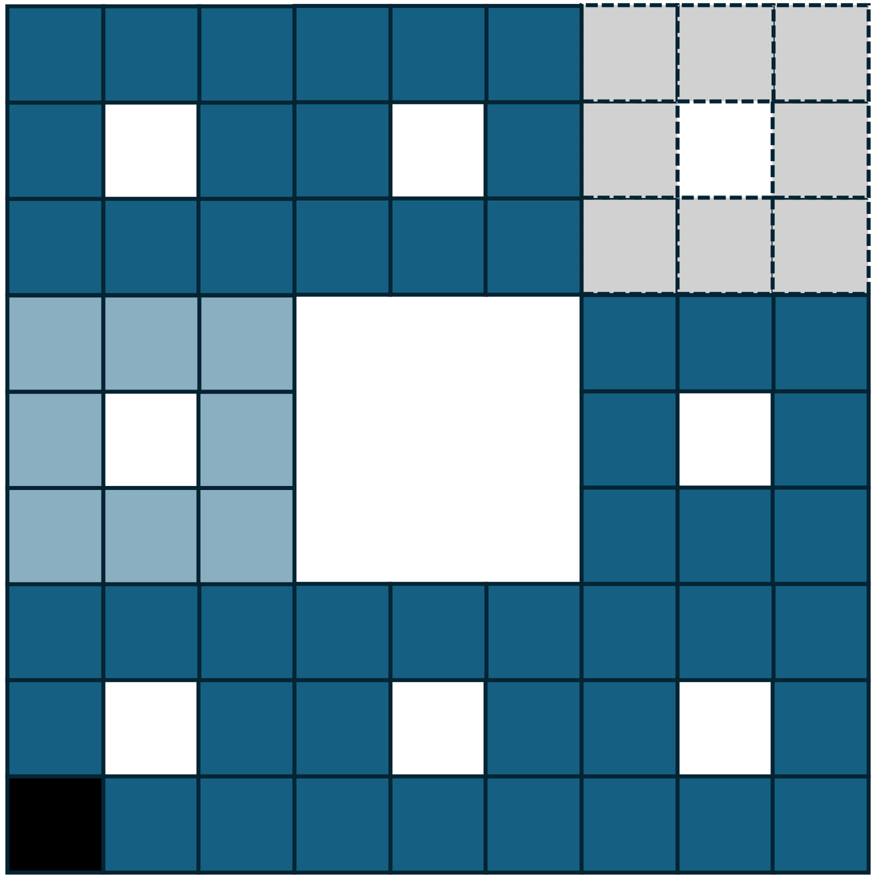


Step 3: Growing the Full Assembly

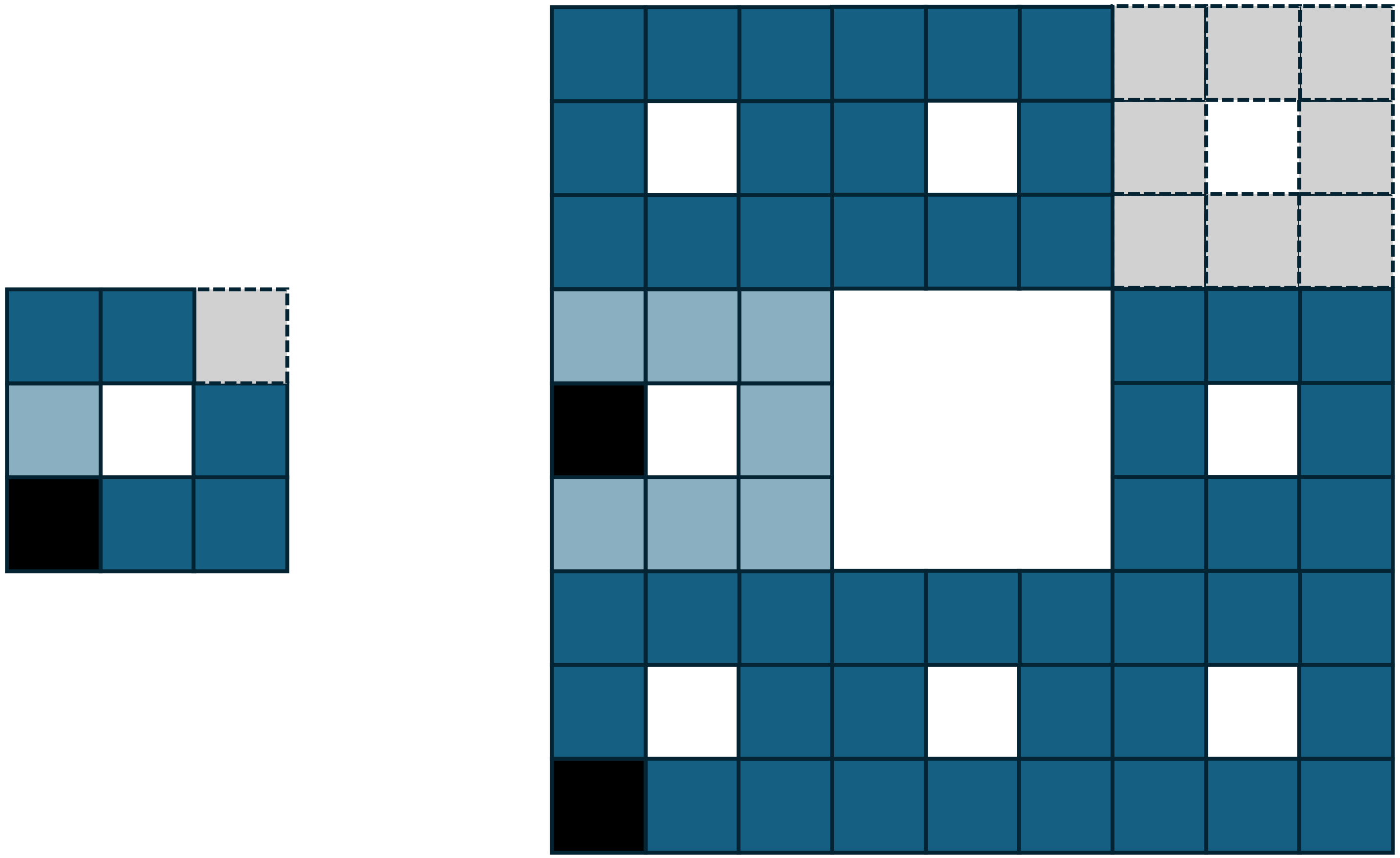


Step 3: Growing the Full Assembly

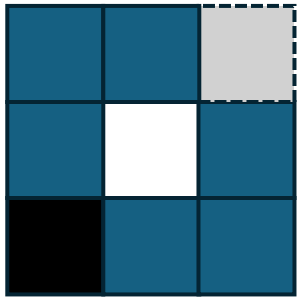


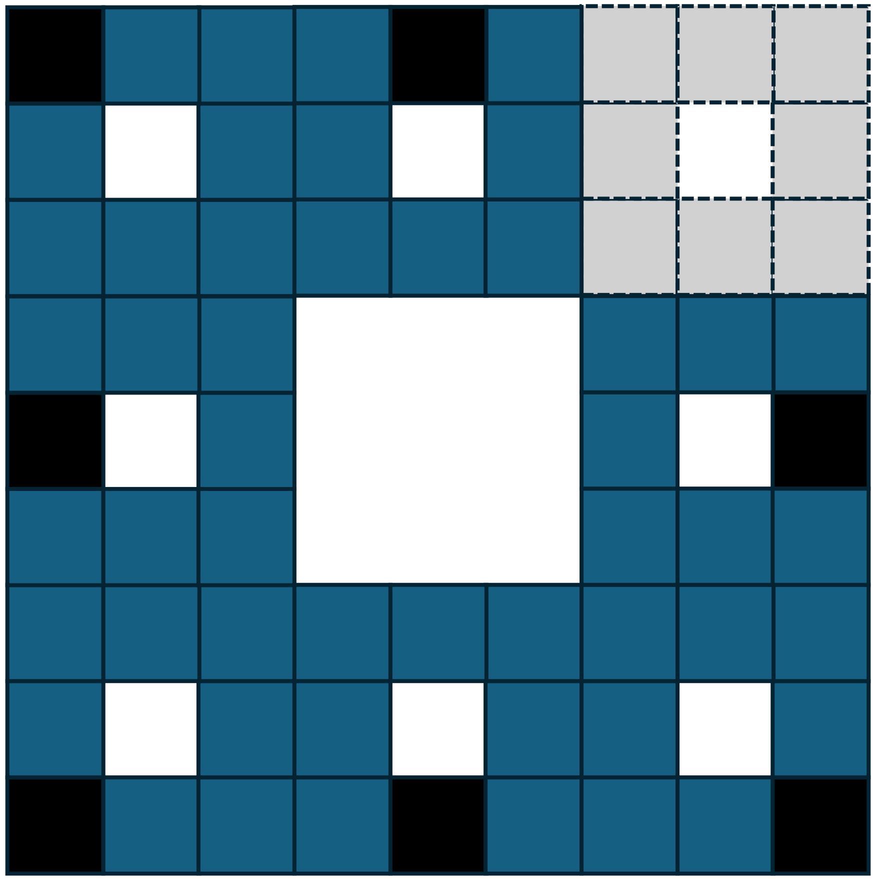


Step 3: Growing the Full Assembly

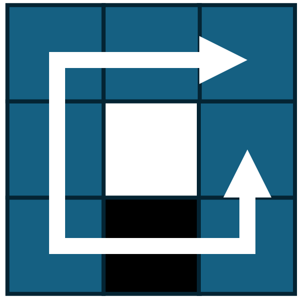


Step 3: Growing the Full Assembly

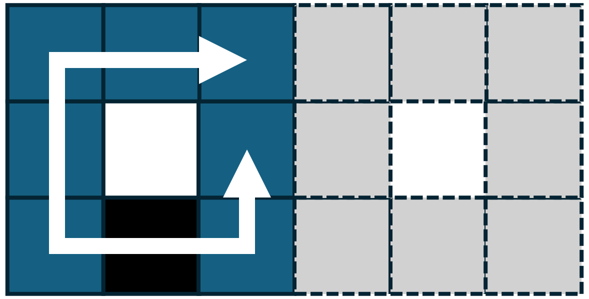




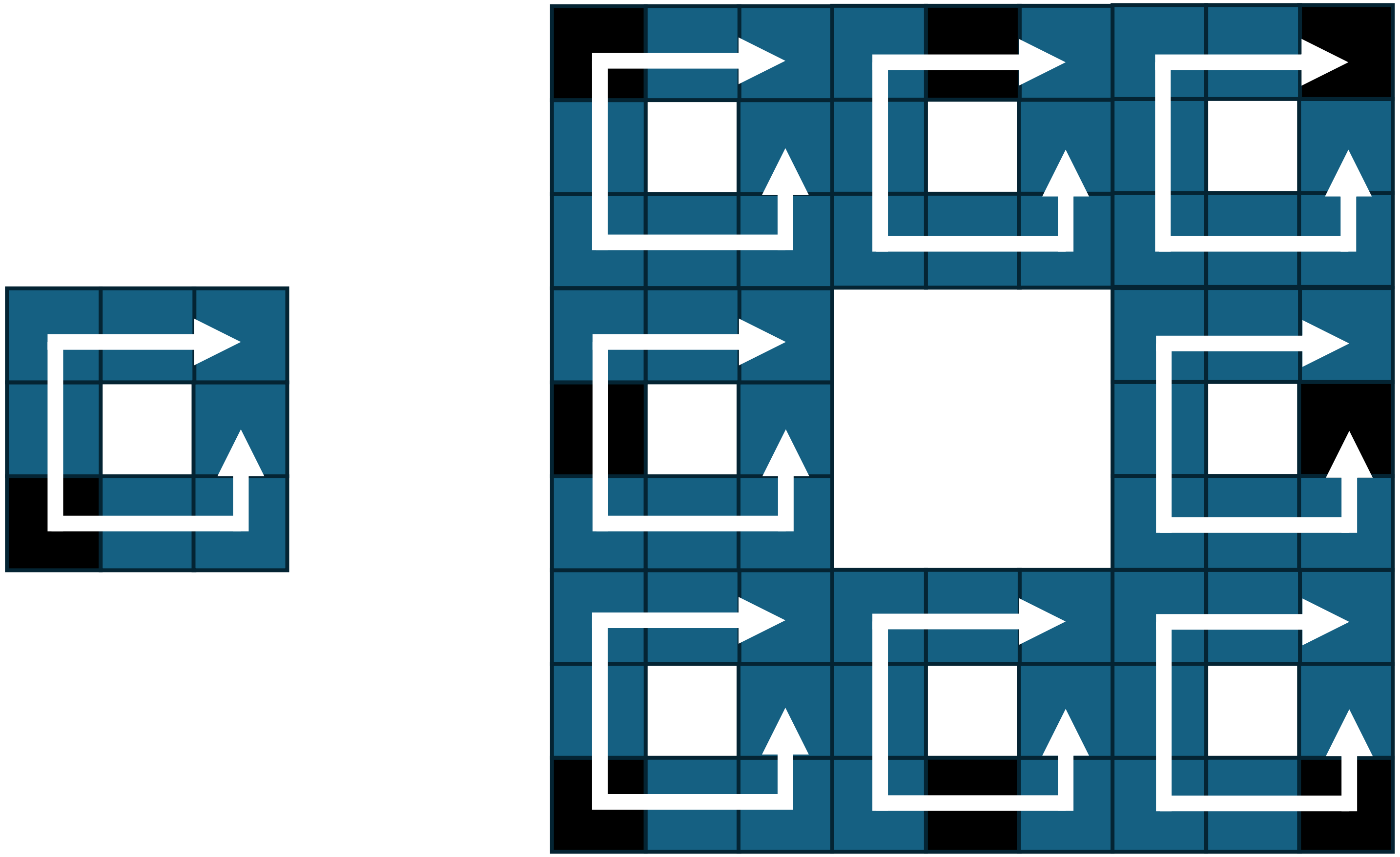
Step 3: Growing the Full Assembly



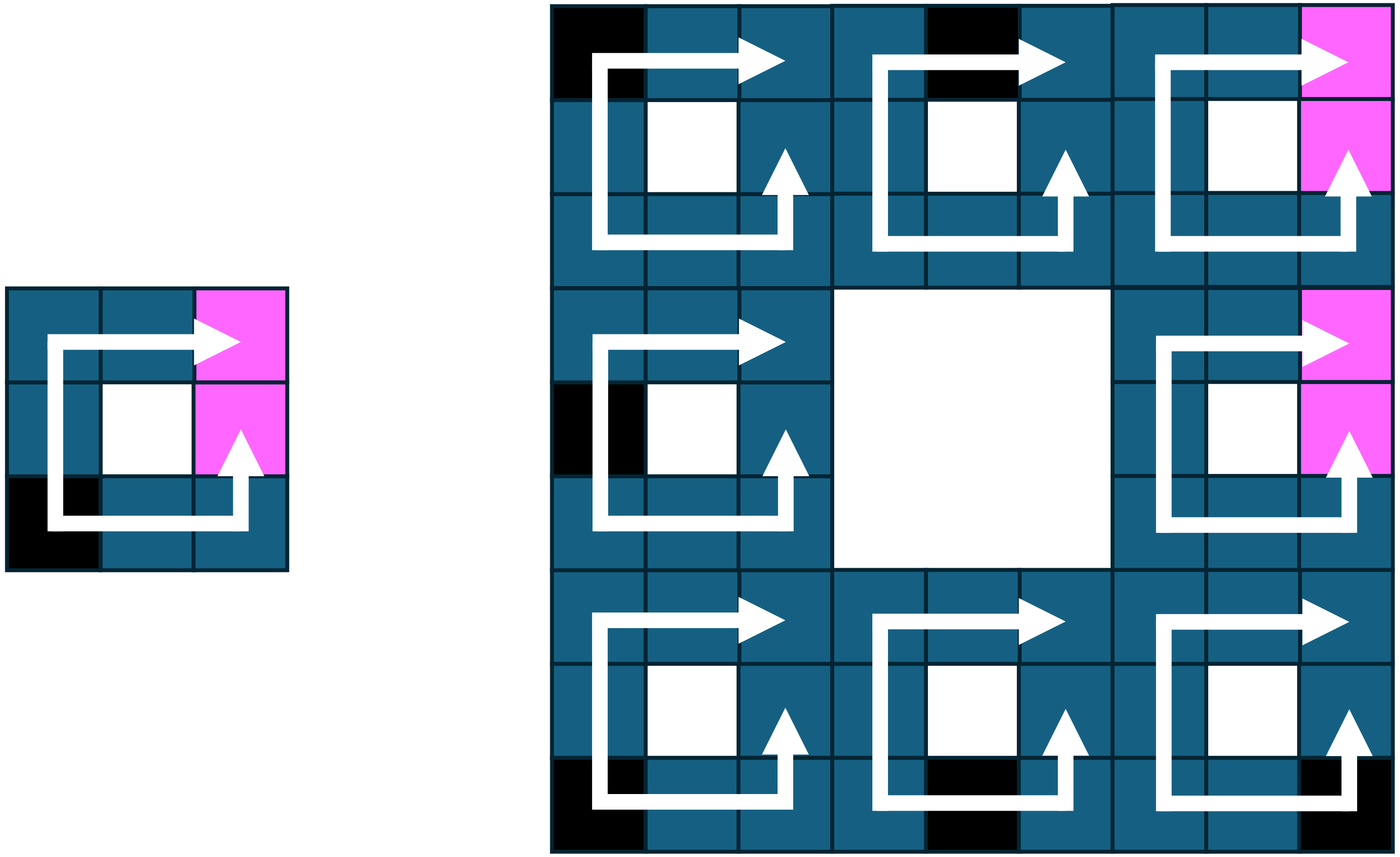
Step 3: Growing the Full Assembly



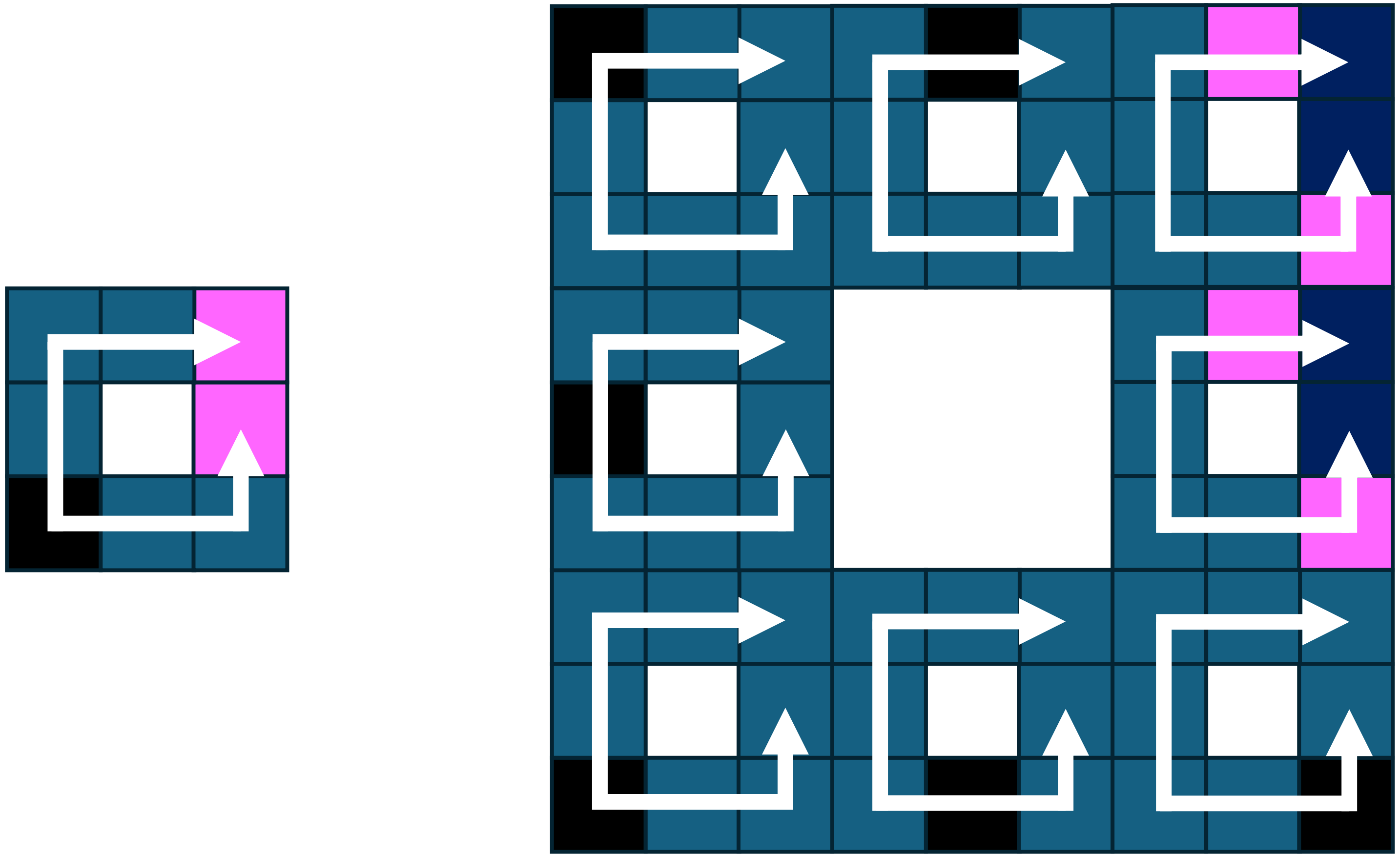
Step 4: Resetting



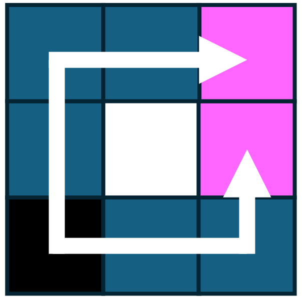
Step 4: Resetting

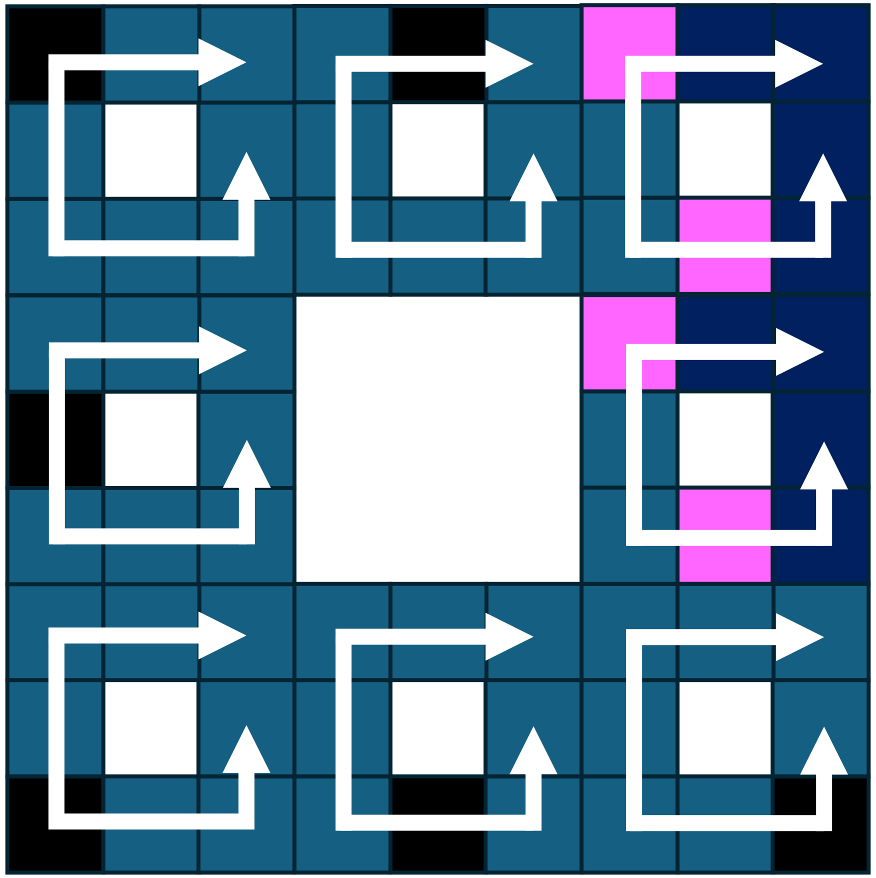


Step 4: Resetting

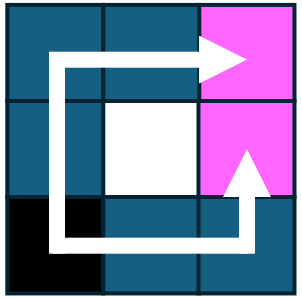


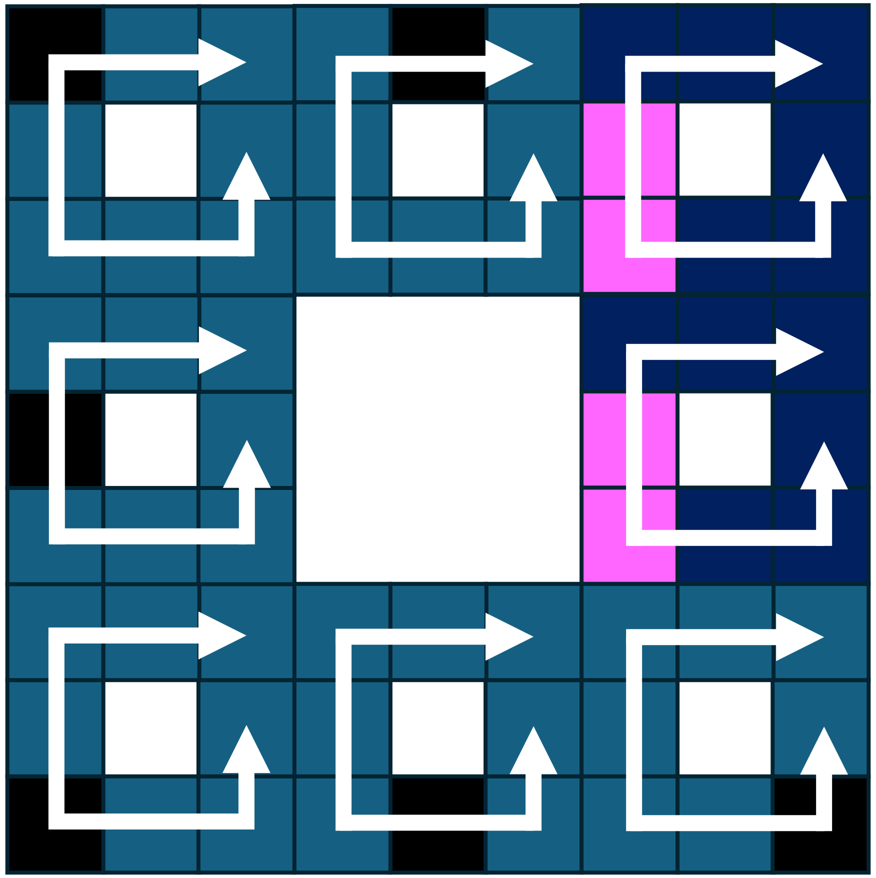
Step 4: Resetting



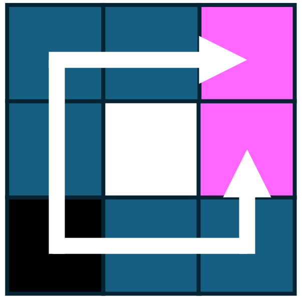


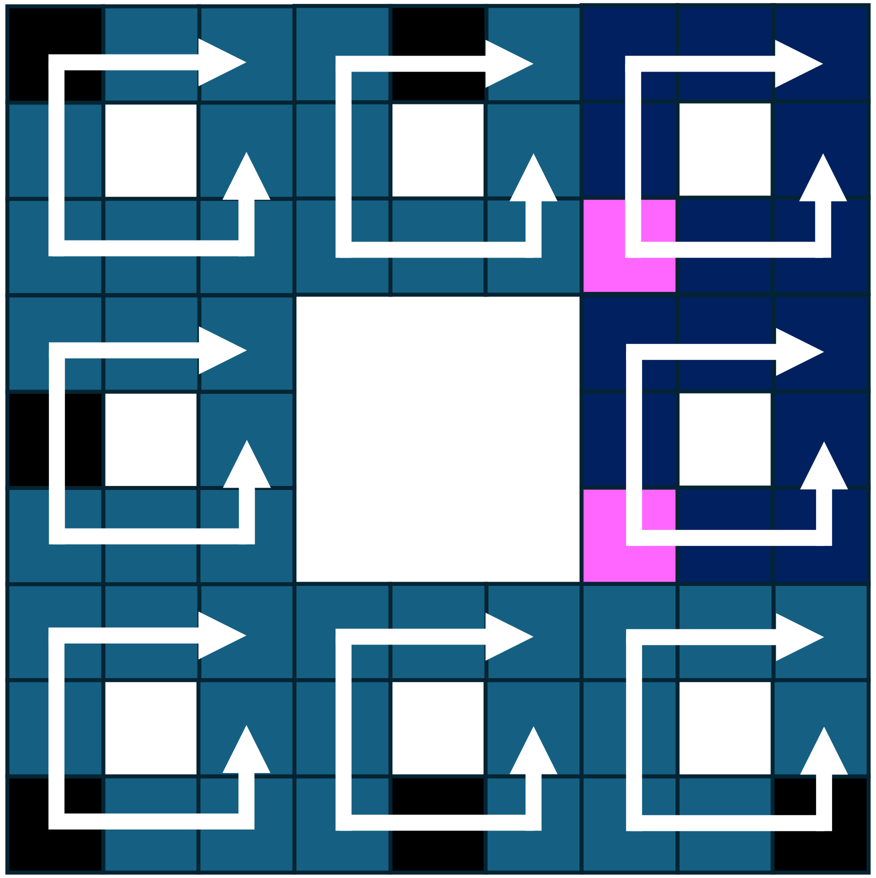
Step 4: Resetting



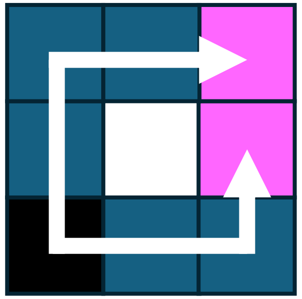


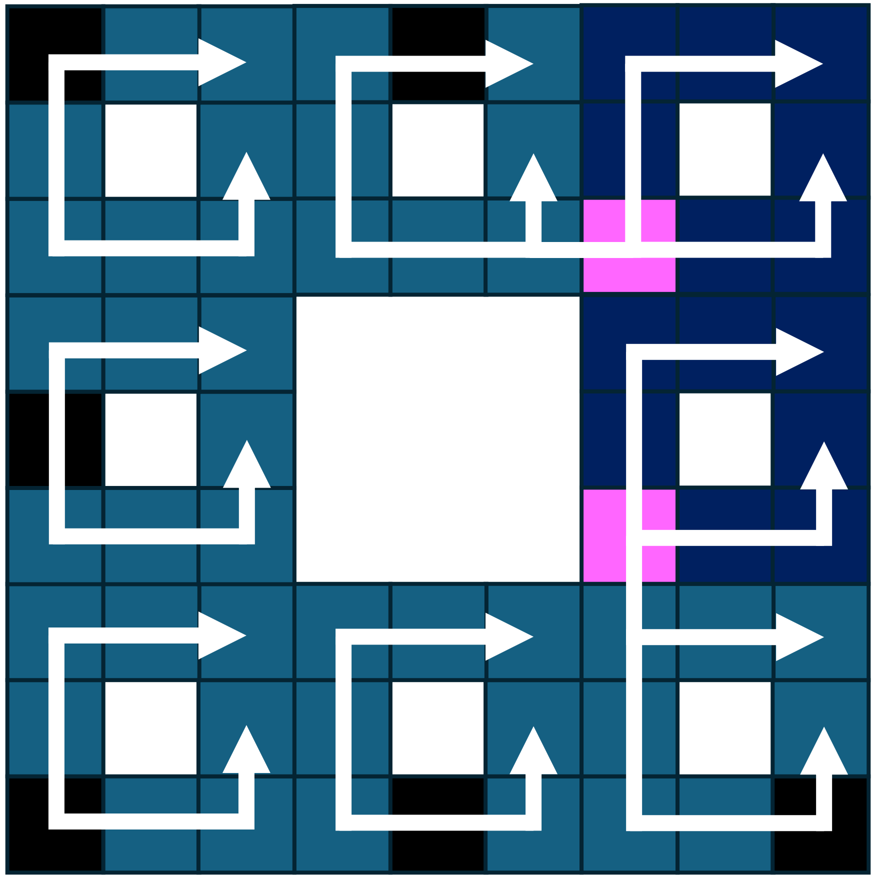
Step 4: Resetting



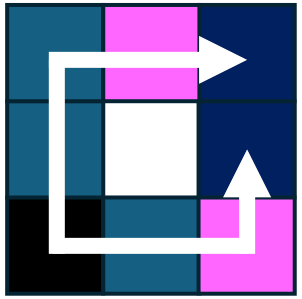


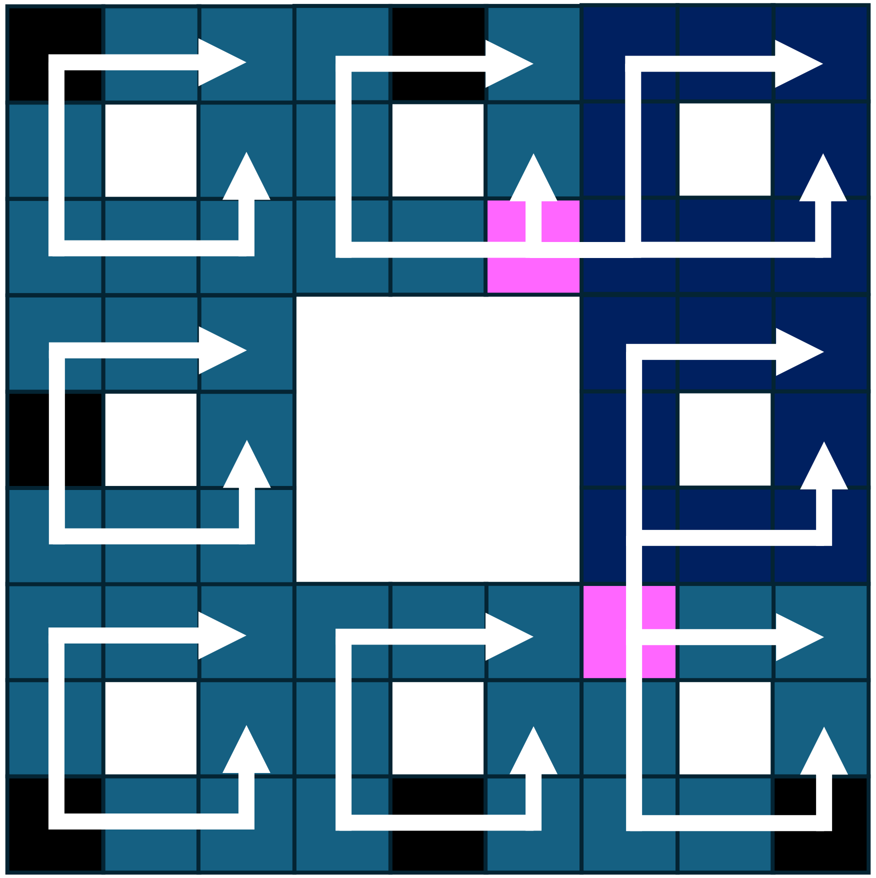
Step 4: Resetting



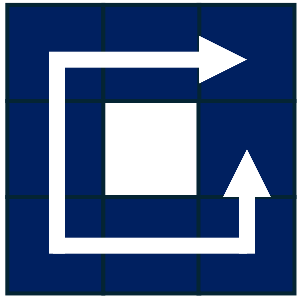


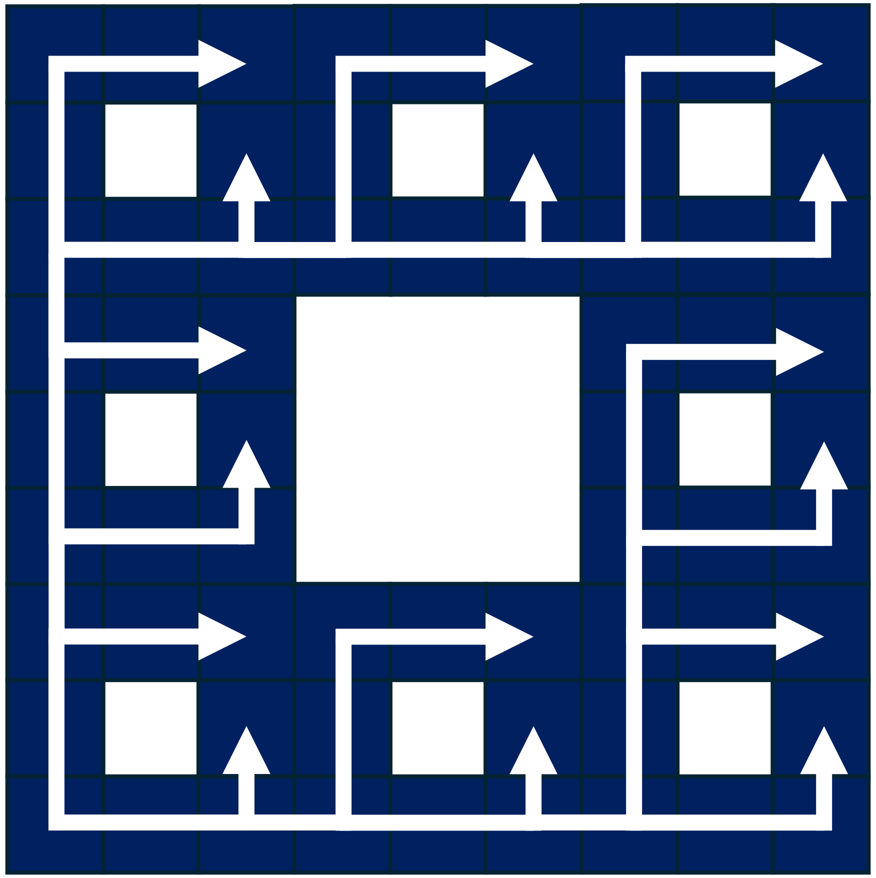
Step 4: Resetting



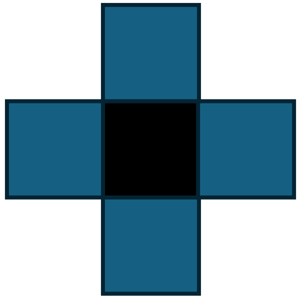


Step 4: Resetting



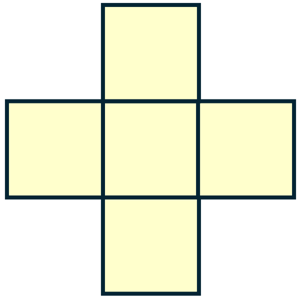


Synchronization

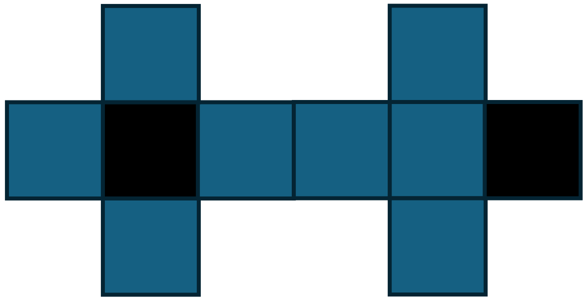


Synchronization

Synchronization

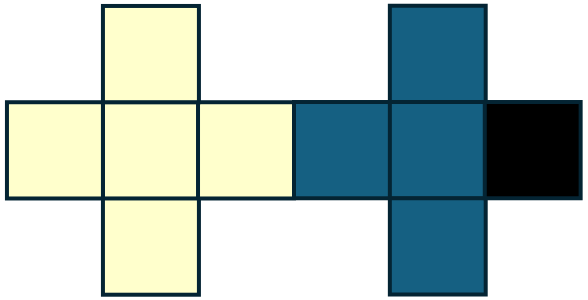


Synchronization

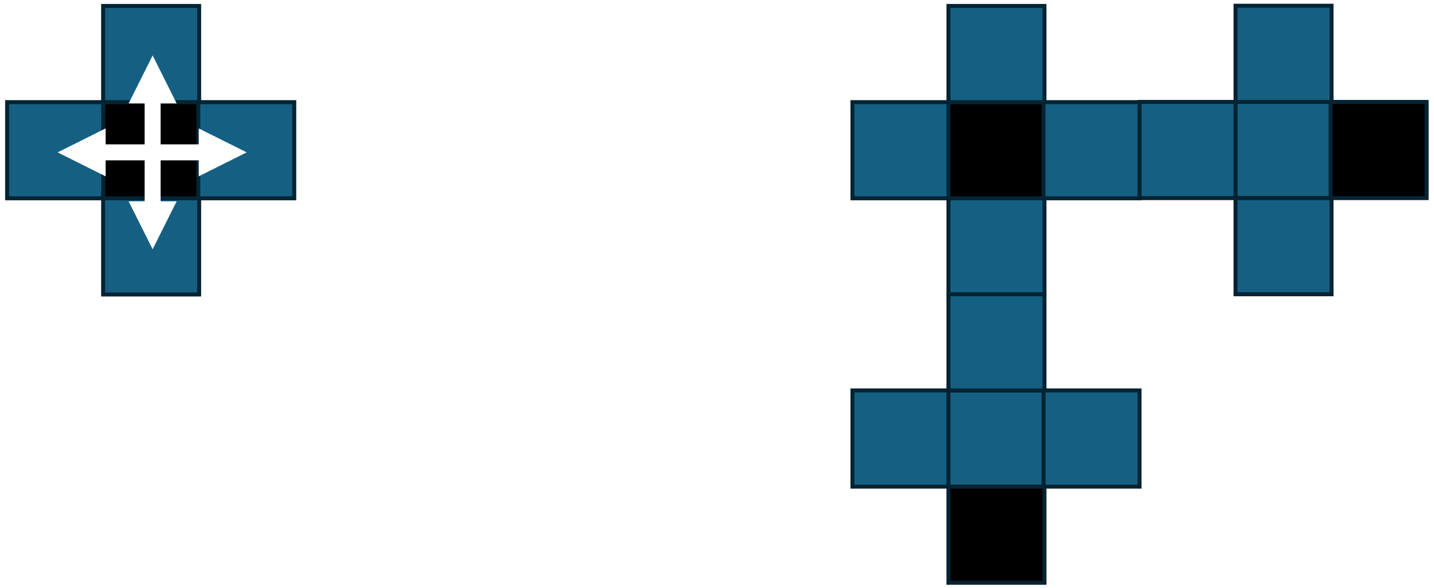


Synchronization

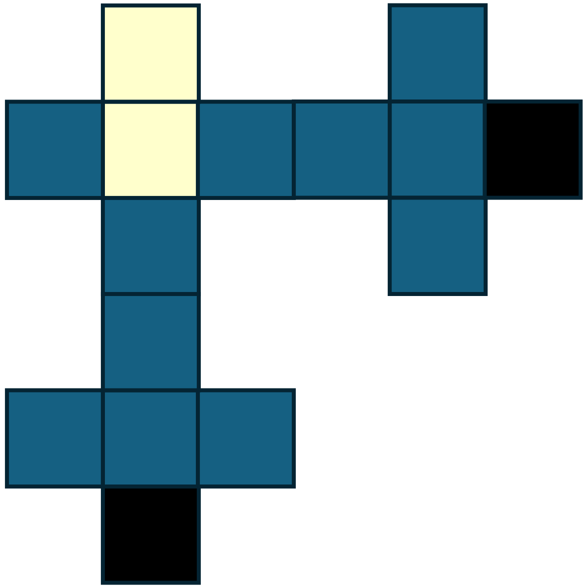
Synchronization



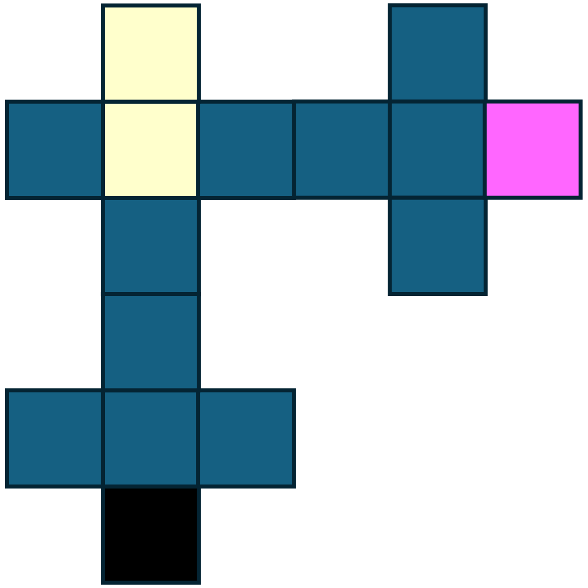
Synchronization



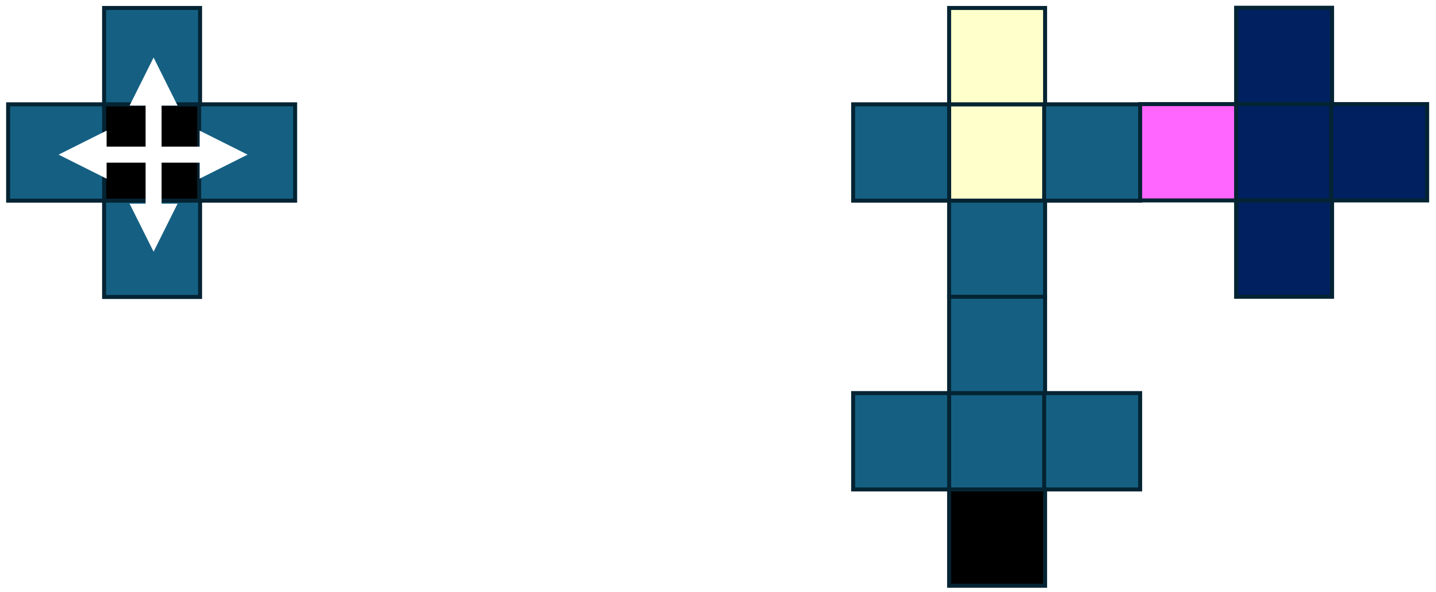
Synchronization



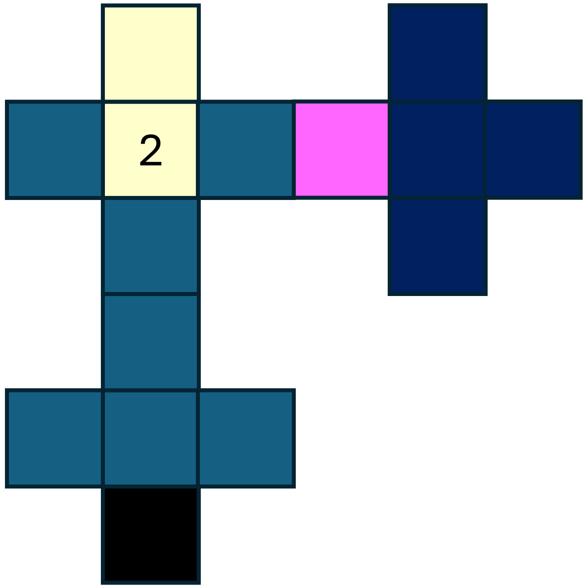
Synchronization



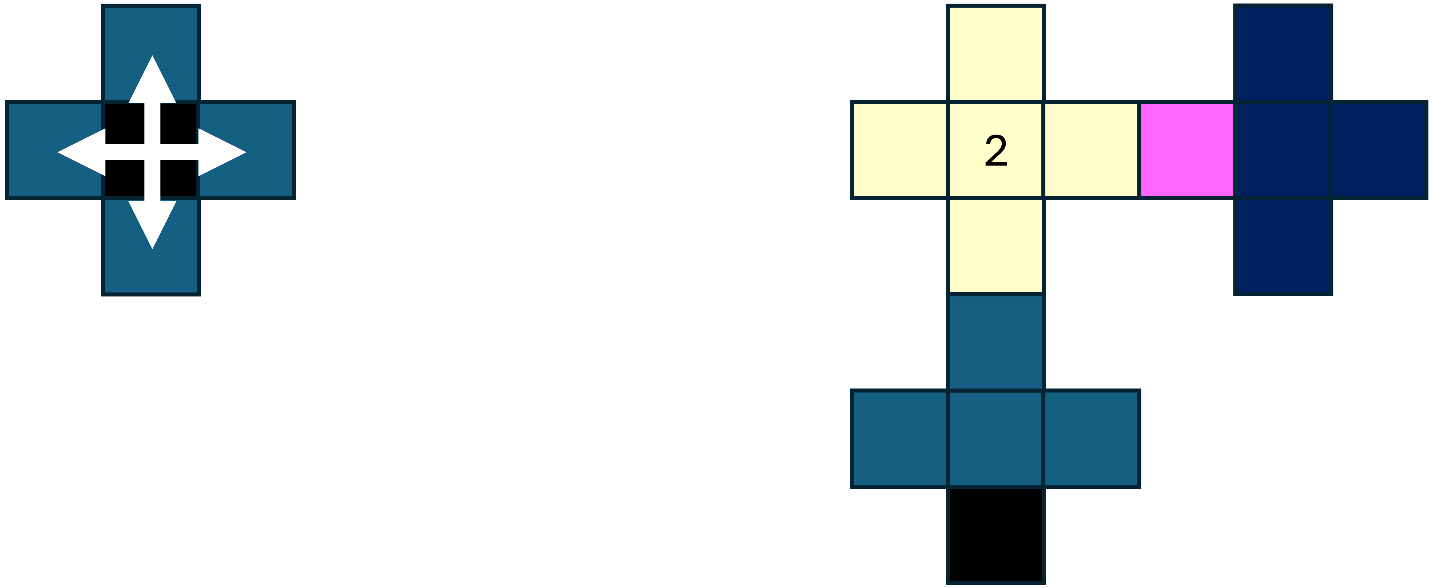
Synchronization



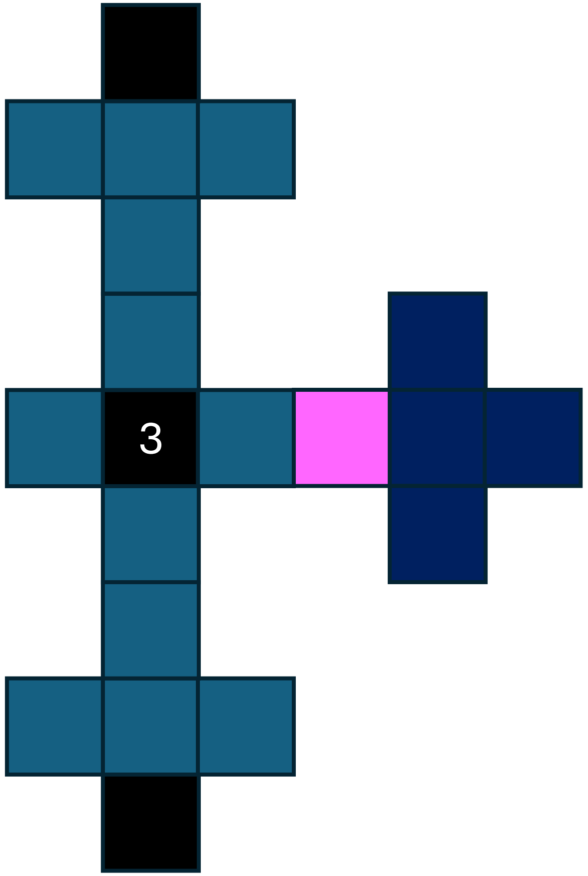
Synchronization



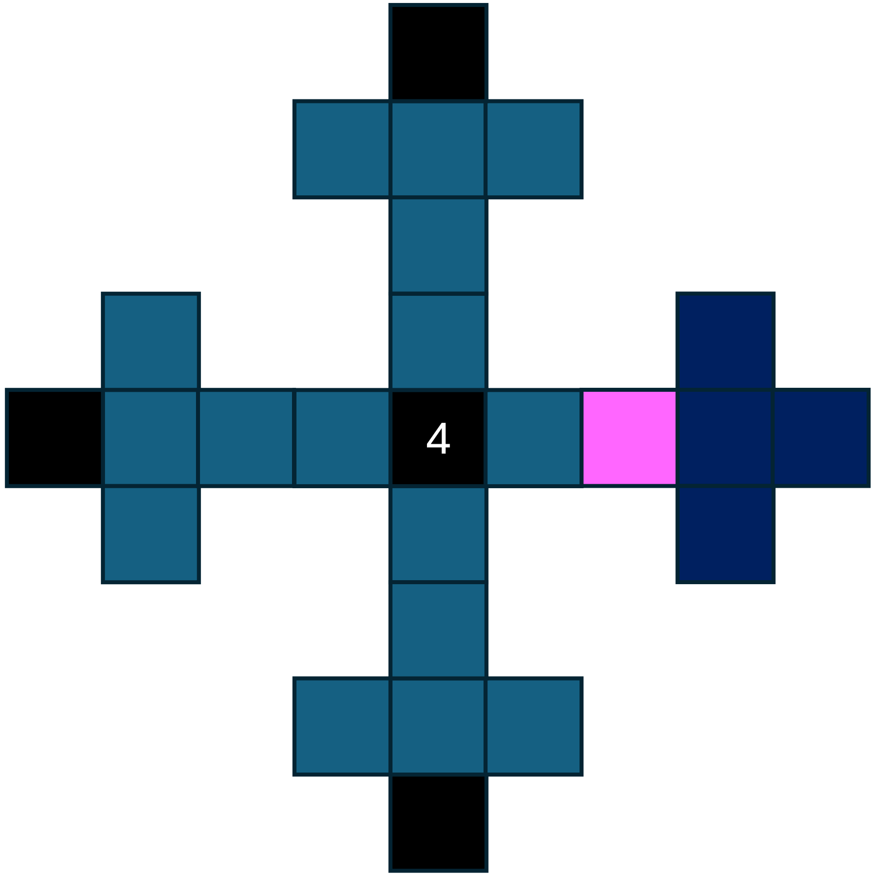
Synchronization



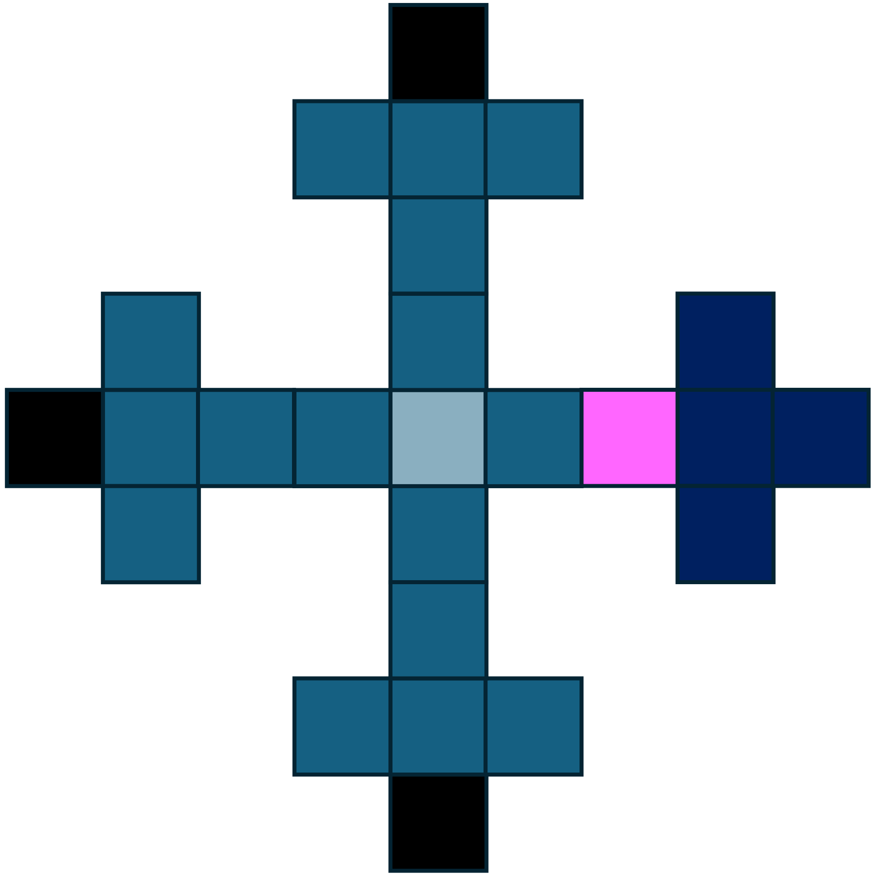
Synchronization



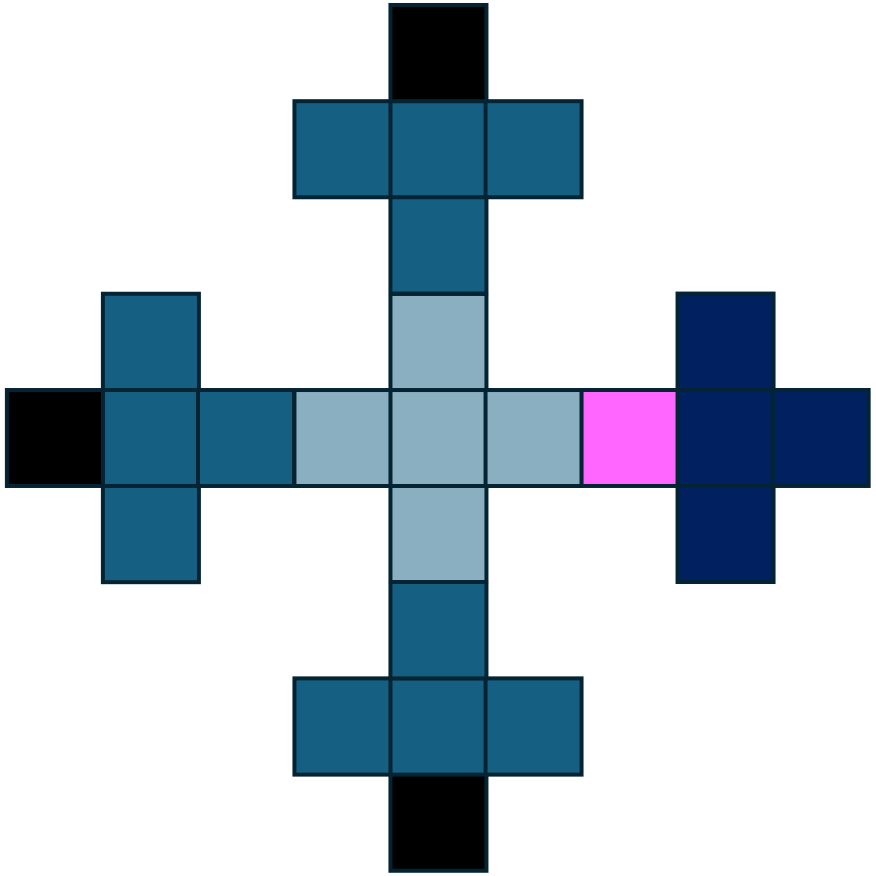
Synchronization



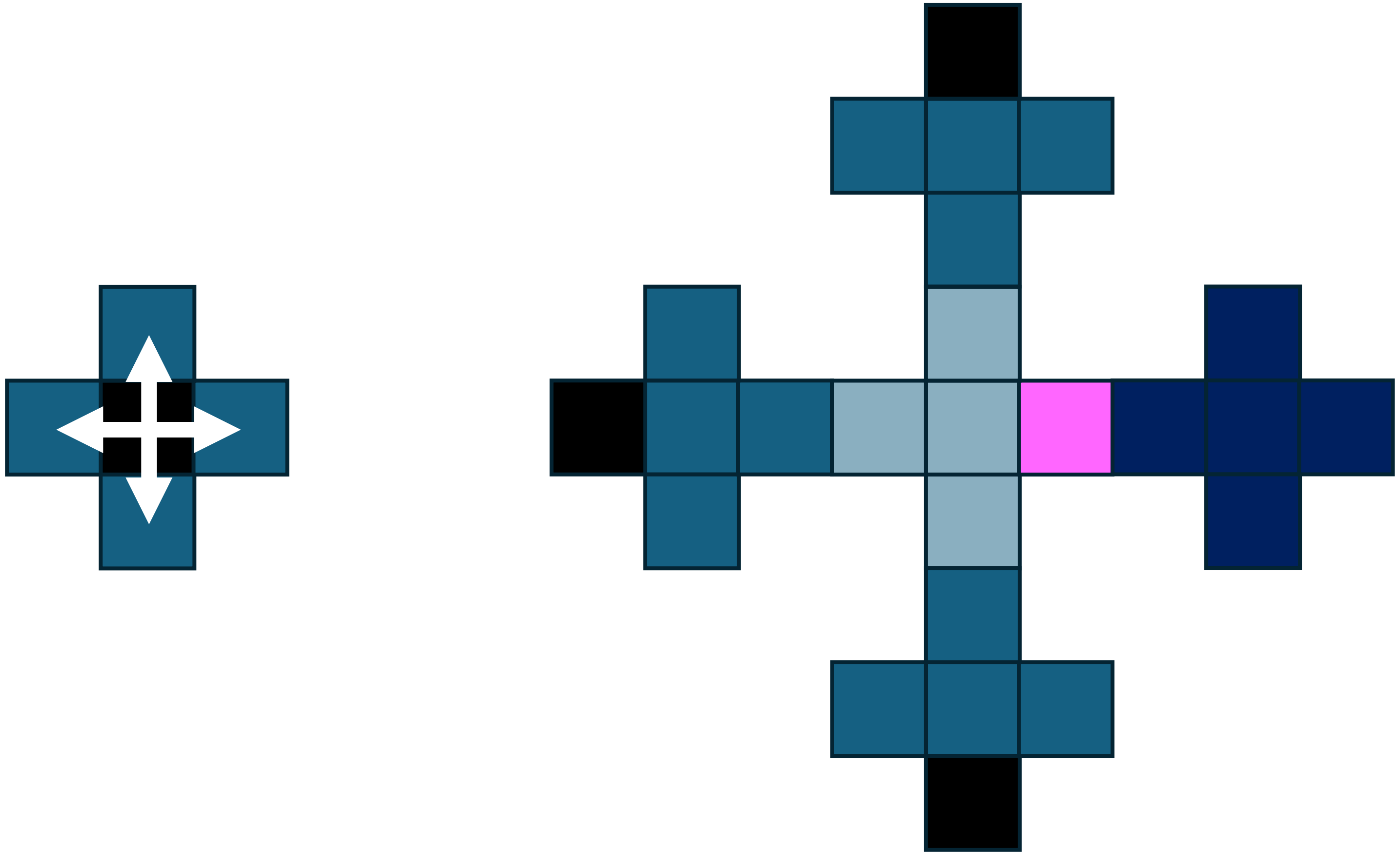
Synchronization



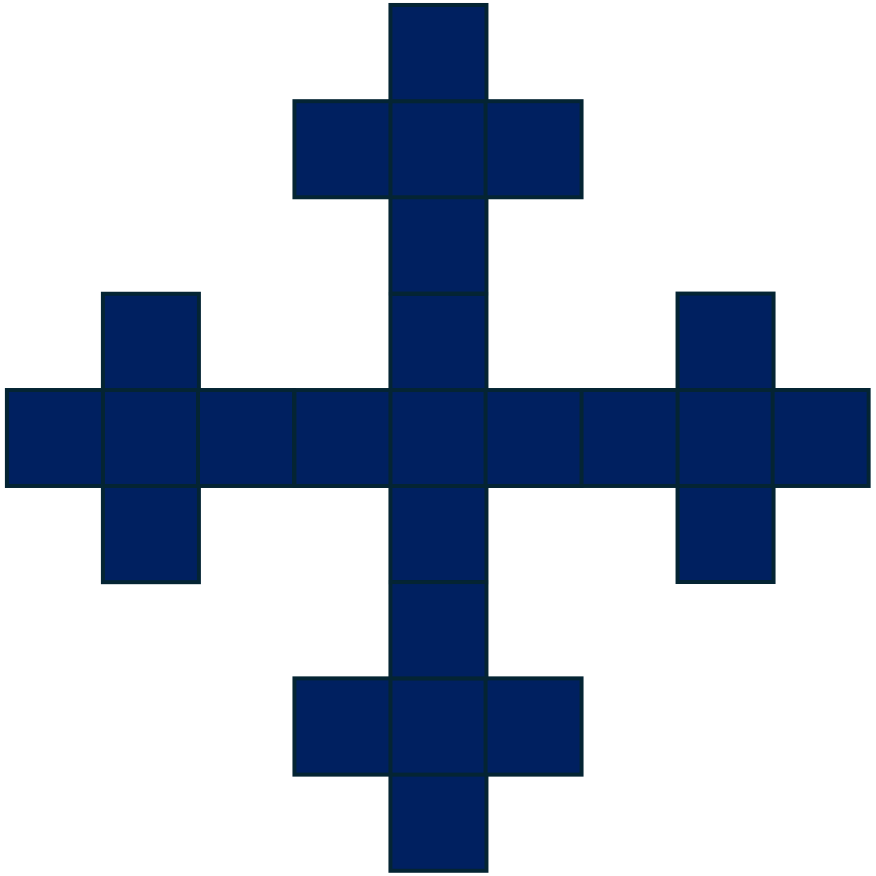
Synchronization



Synchronization



Synchronization



Final Result (Sierpinski Triangle)

Conclusion

What is left?

Conclusion

- All fractals are buildable at temperature 1
- A single system exists that builds all these fractals
- Do higher temperatures allow for more efficient fractal constructions?
- Are there special classes of fractals which can be built with small systems?

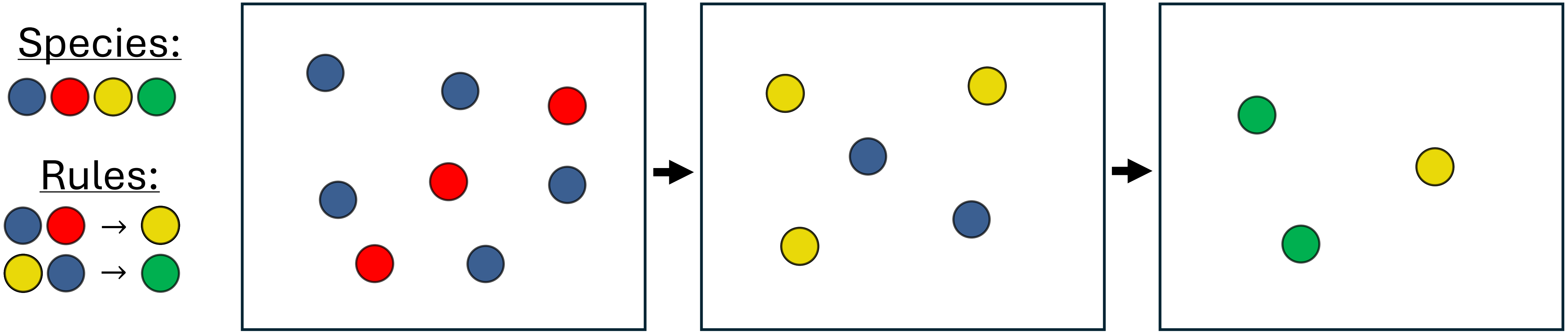
Reachability in Chemical Reaction Networks

Introduction to the Model

Chemical Reaction Networks

- Ordered pair (Λ, Γ)
  - Λ = Set of input species
  - Γ = Set of rules
- Configuration
  - Length $|\Lambda|$ vector of non-negative integers
  - Represents count of each species

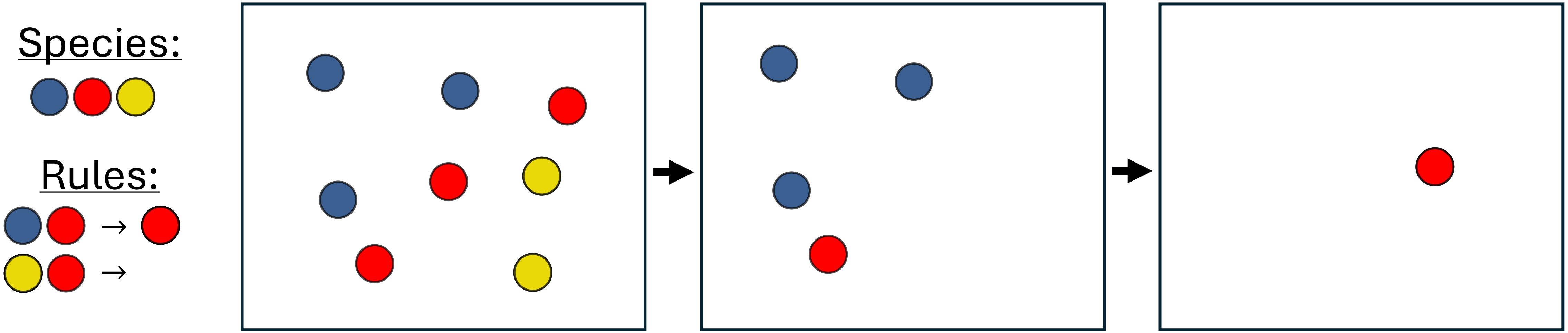
Example CRN System



Configuration *I*:
[5, 3, 0, 0]

Configuration *D*:
[0, 0, 1, 2]

Example Void CRN System

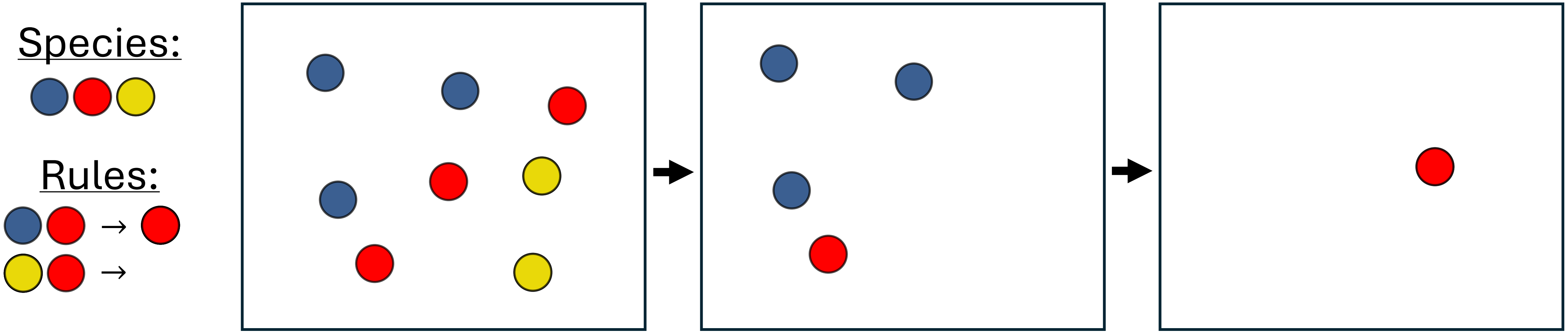


Configuration *I:*
[3, 3, 2]

Configuration *D:*
[0, 1, 0]

Reachability

- Given initial configuration *I* and target configuration *D*, can *D* ever be reached from *I*?



Configuration *I:*
[3, 3, 2]

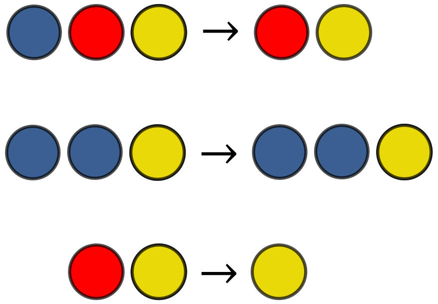
Configuration *D:*
[0, 1, 0]

Our Contributions

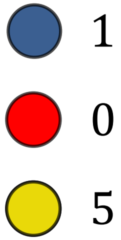
	Basic CRNs (1-step)	
Void Rules	**Complexity**	**Ref.**
$(2,1)$	NL	Thm. 11
$(k,k-1)^+$	$O(\vert\Lambda\vert^2\vert\Gamma\vert)$	Thm. 21
$(2,0)$	$O(\vert\Lambda\vert^2\vert\Gamma\vert\log(\vert\Lambda\vert))$	[1]
$(2,0)$, $(2,1)$	$O(\vert\Lambda\vert^2\vert\Gamma\vert\log(\vert\Lambda\vert))$	Thm. 18
$(k\geq 3, g\leq k-2)$	NPC	Cor. 24

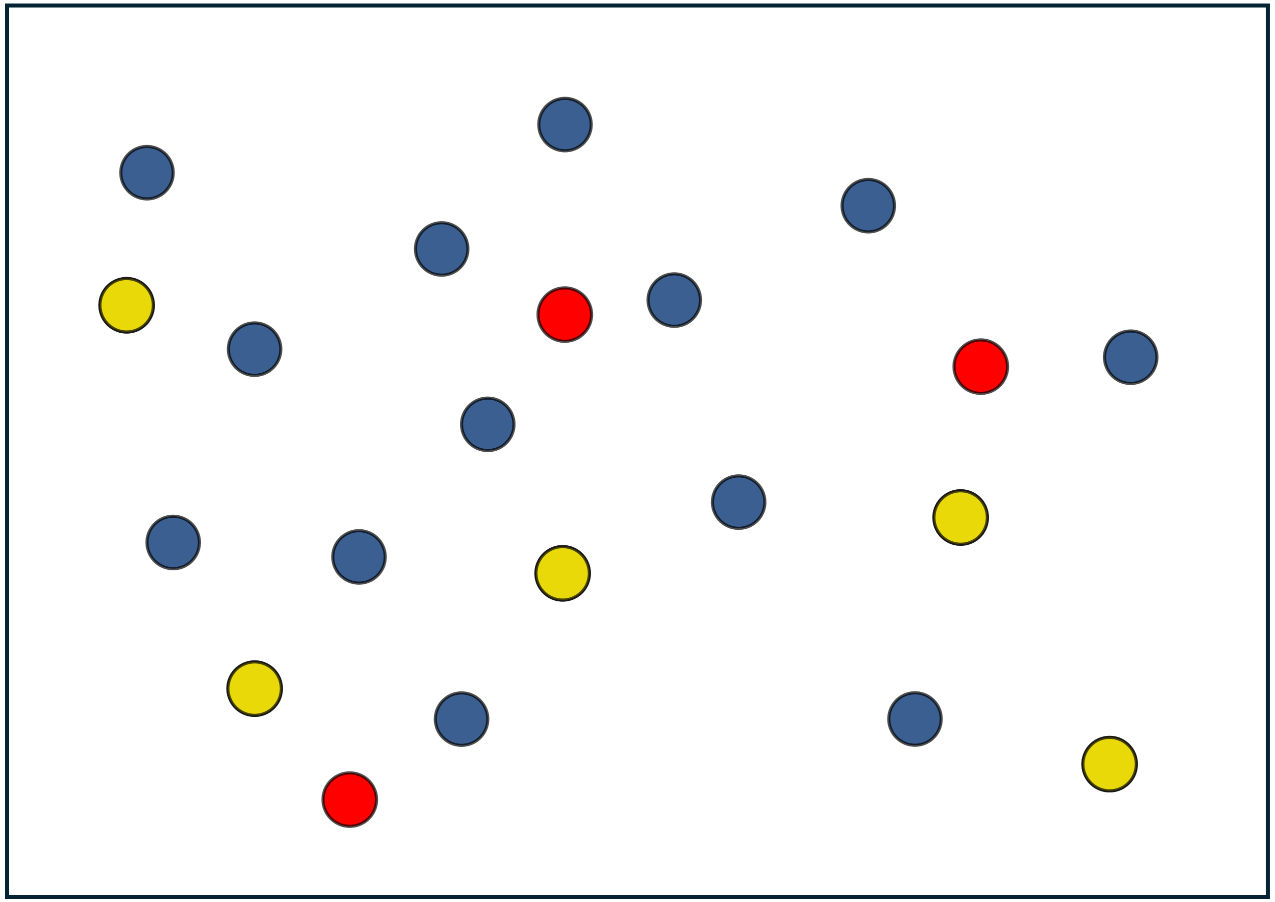
Bin Fu, Timothy Gomez, Ryan Knobel, Austin Luchsinger, Aiden Massie, Marco Rodriguez, Adrian Salinas, Robert Schweller, and Tim Wylie. In 31th International Conference on DNA Computing and Molecular Programming (DNA 25).

High Level Idea

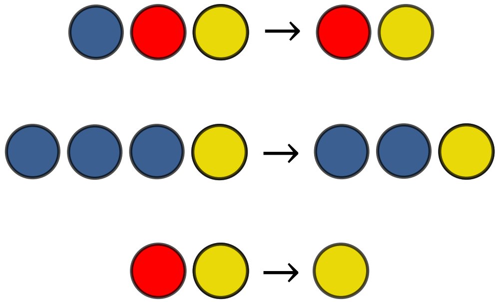


Final Counts:

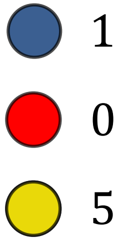


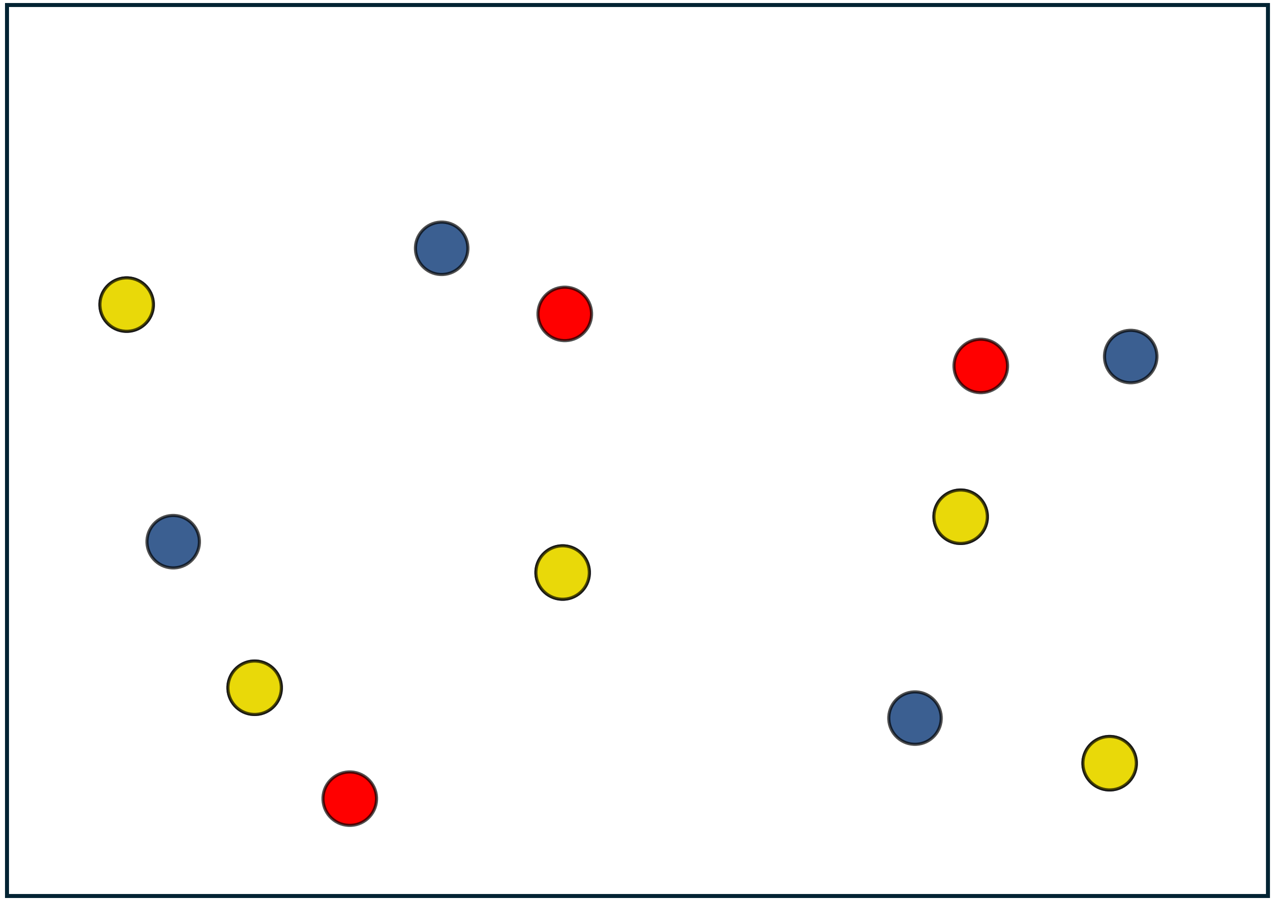


High Level Idea

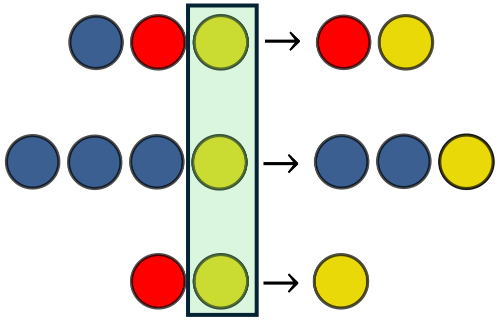


Final Counts:

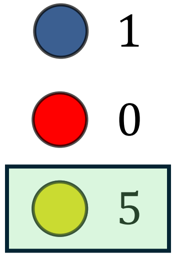


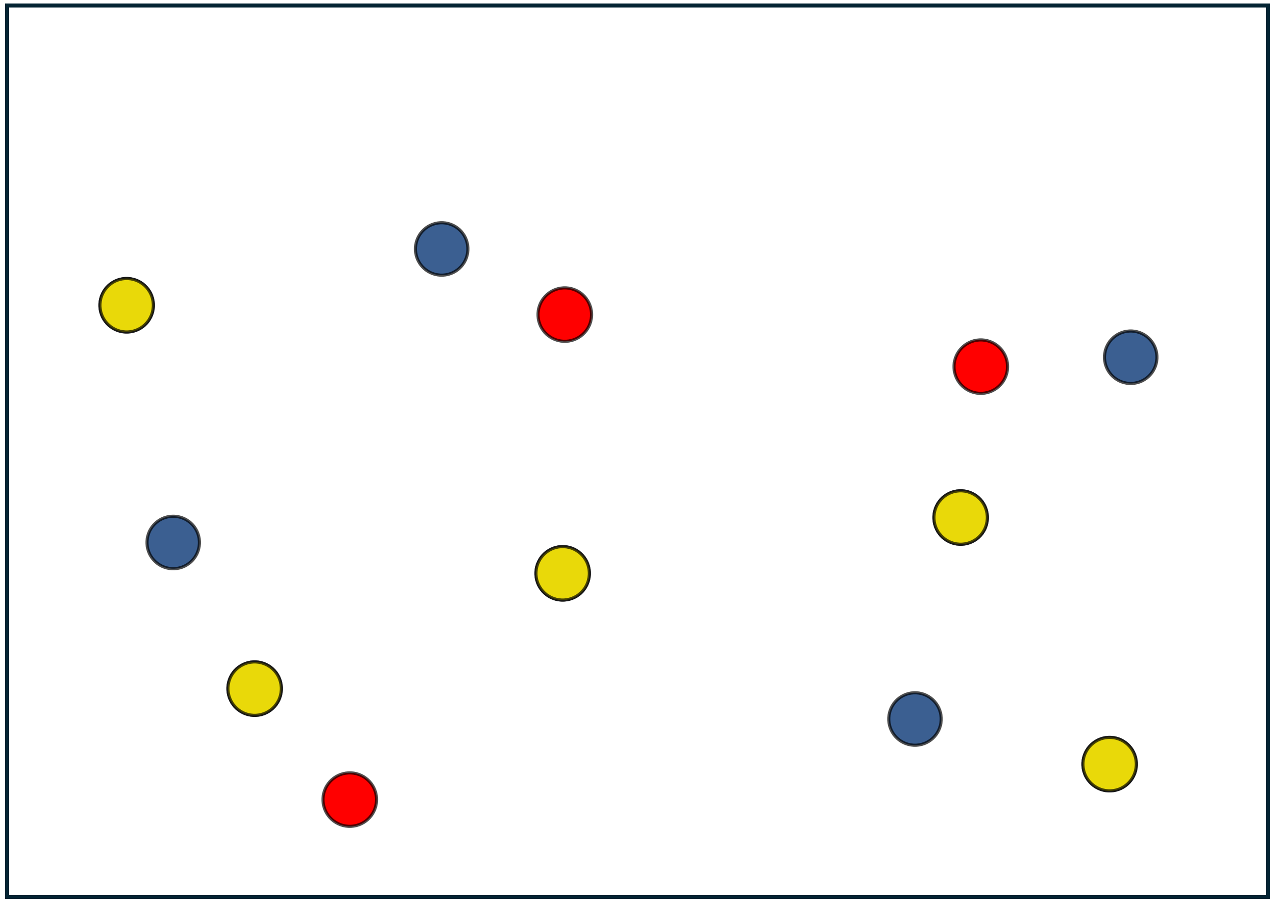


High Level Idea

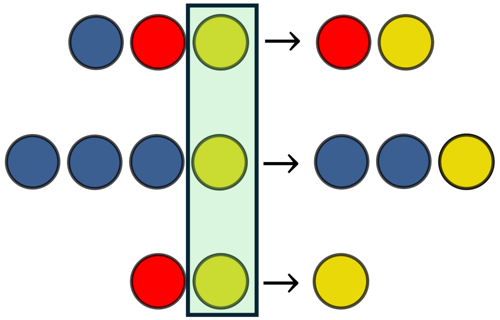


<u>Final Counts:</u>

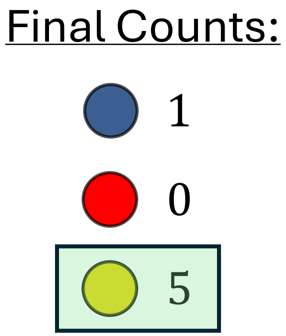


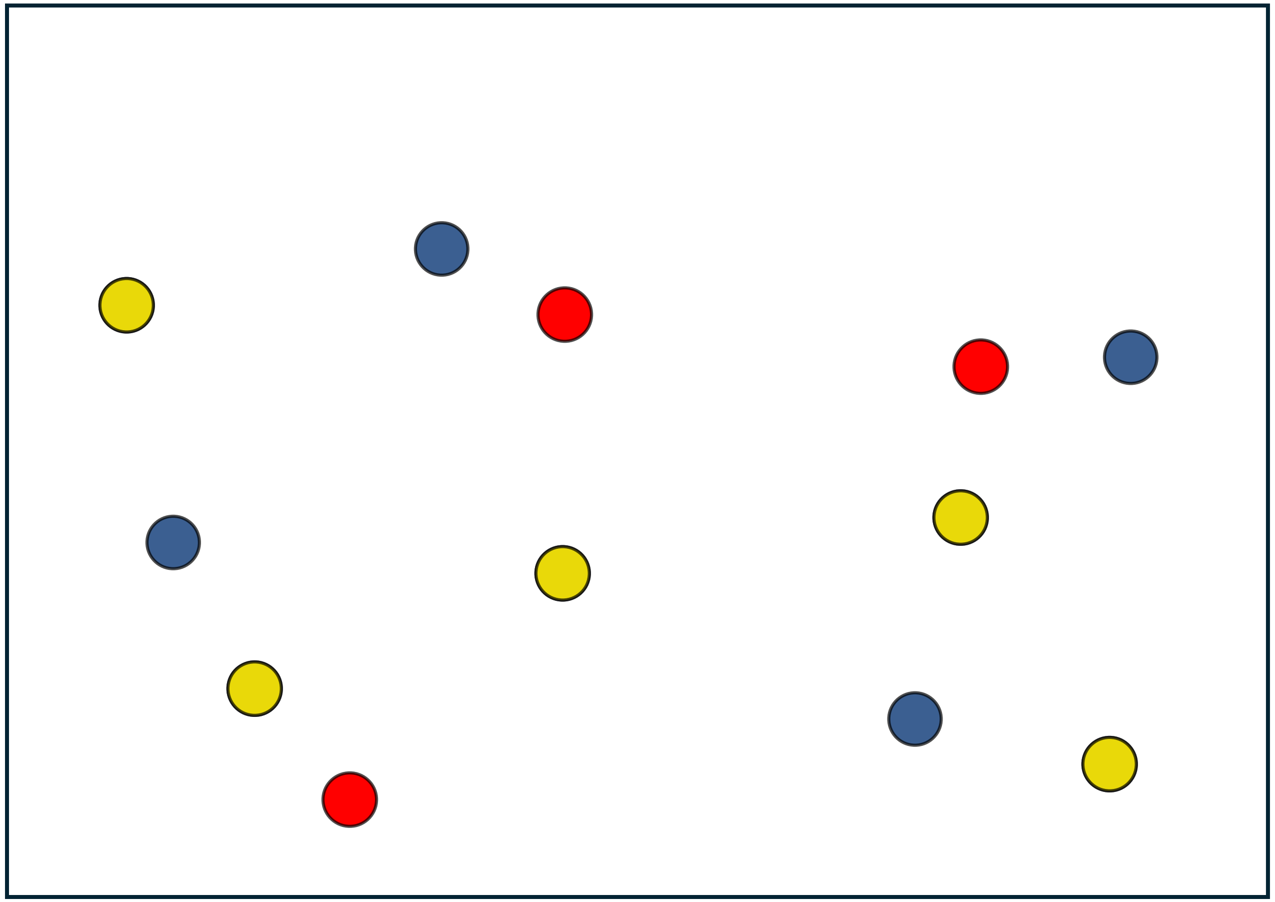


High Level Idea

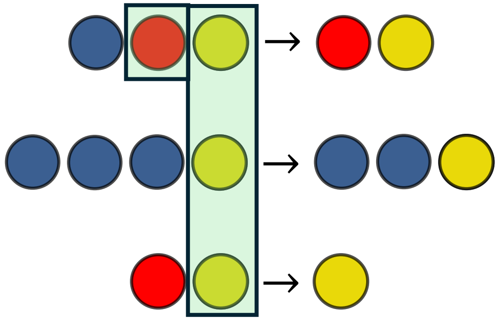


(1)

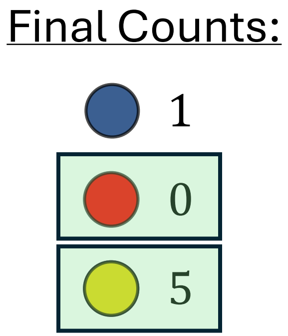


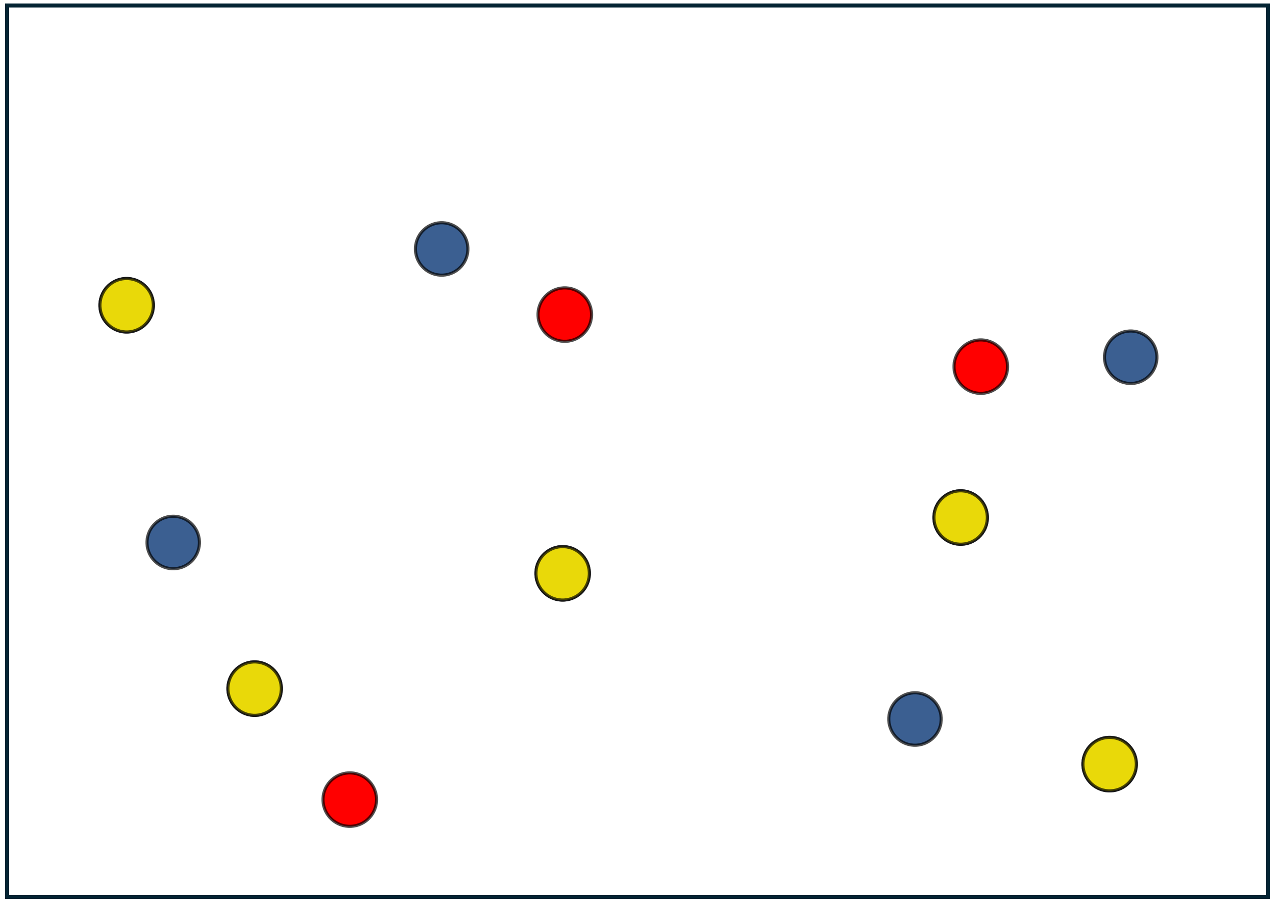


High Level Idea

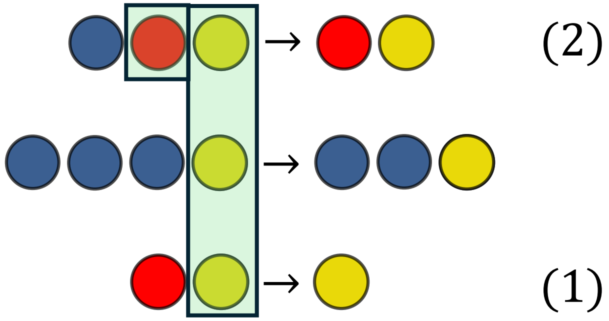


(1)

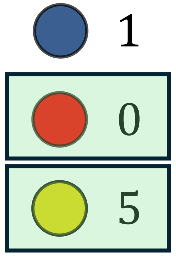


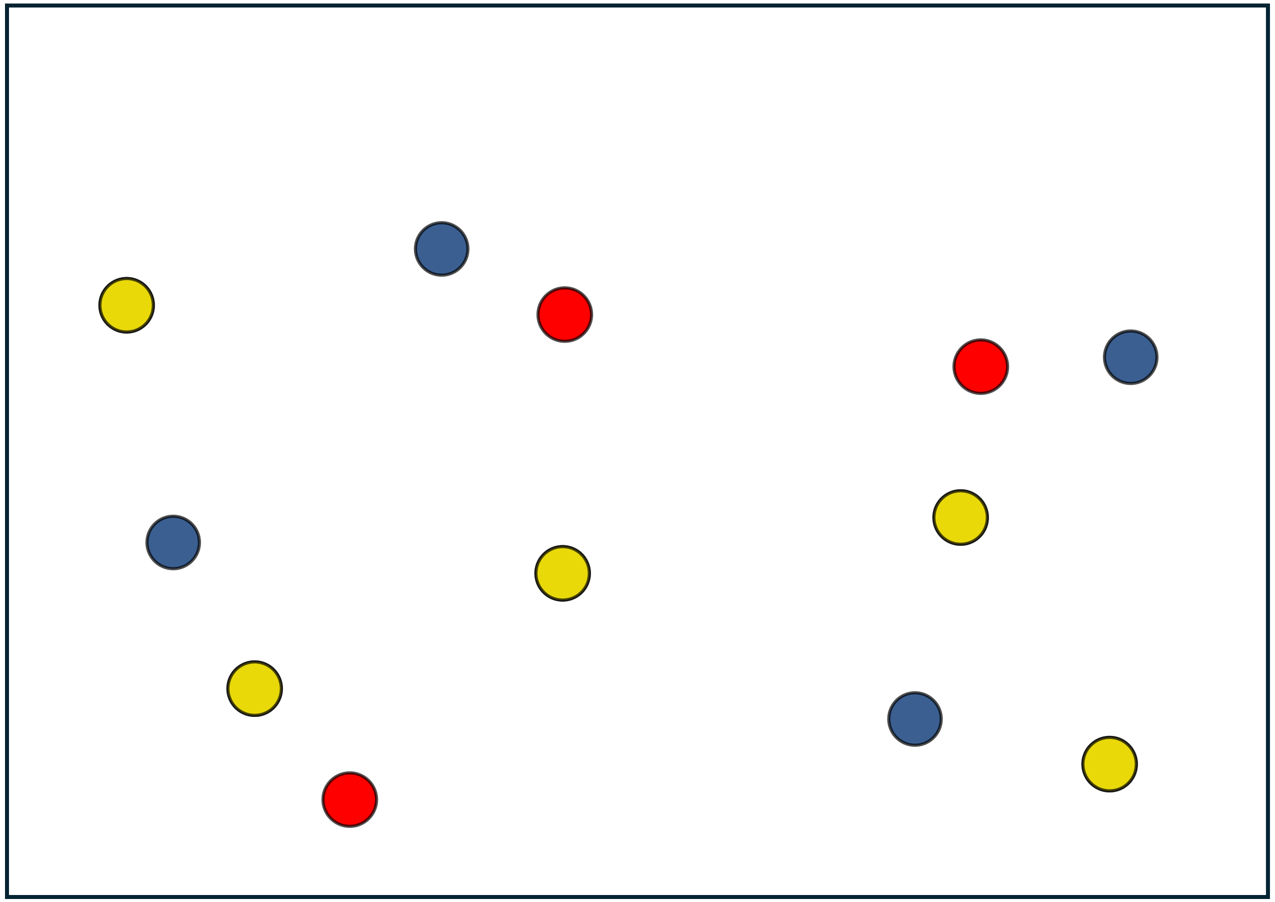


High Level Idea

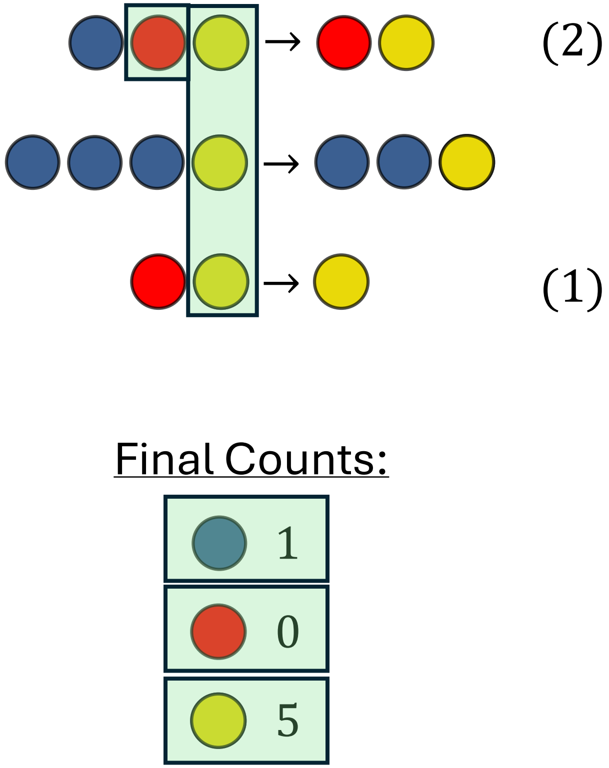


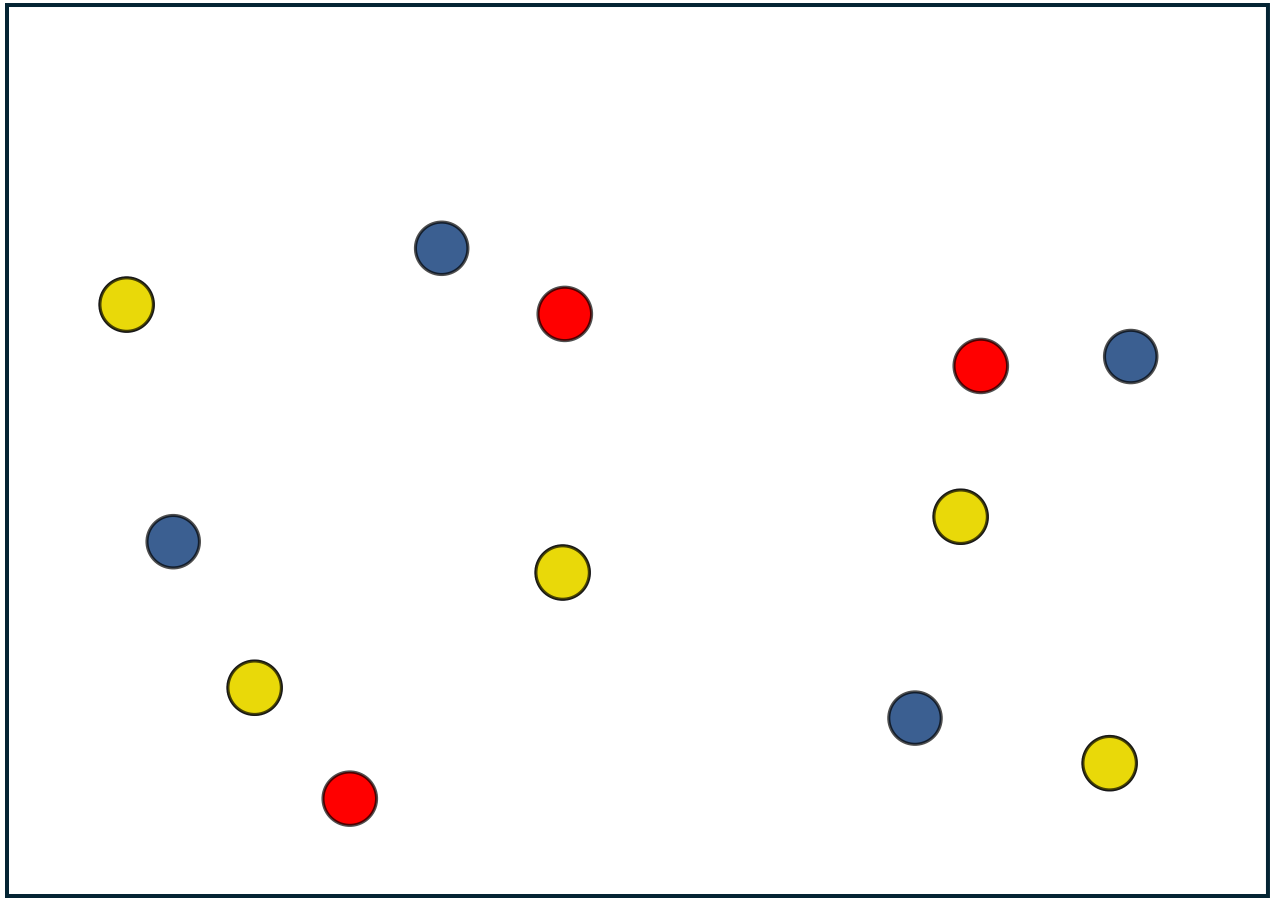
Final Counts:



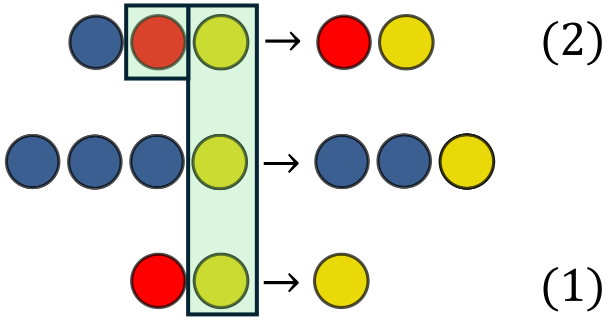


High Level Idea

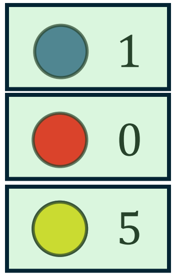


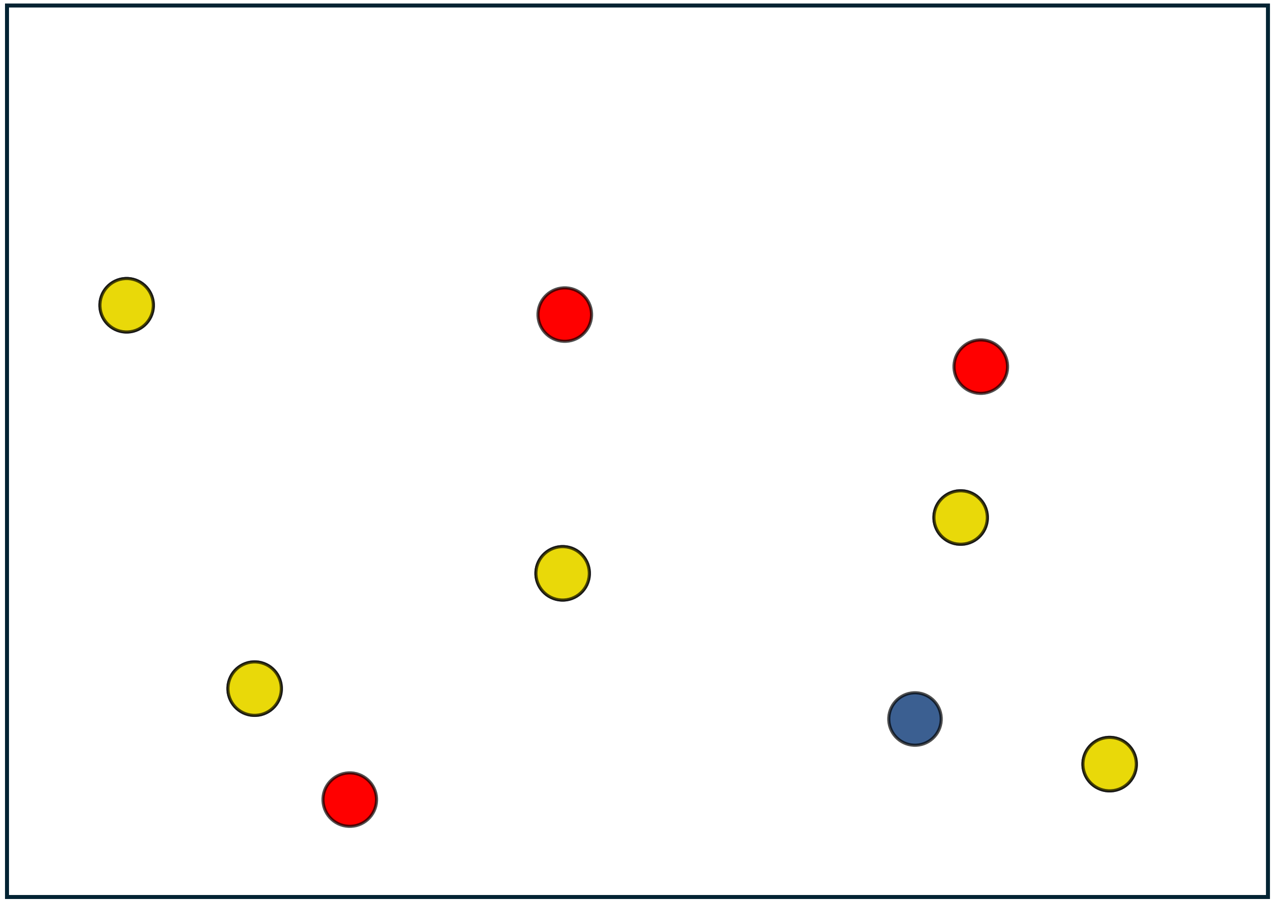


High Level Idea

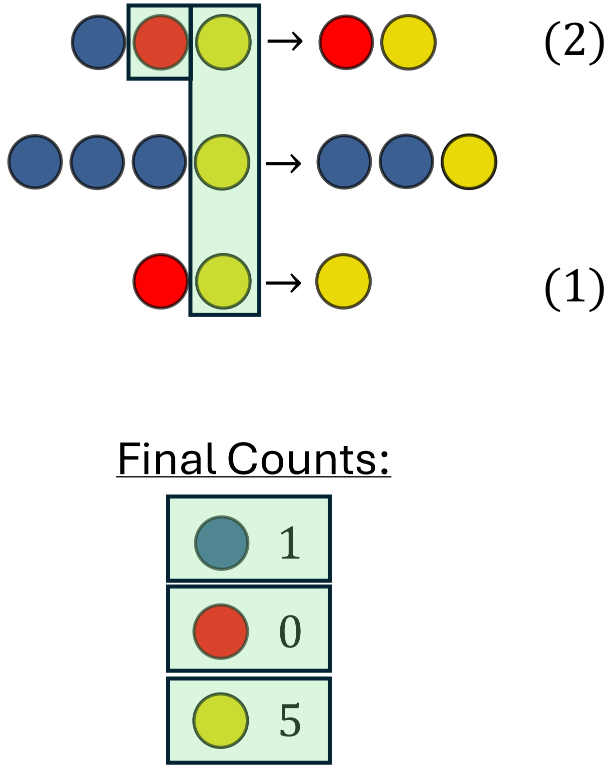


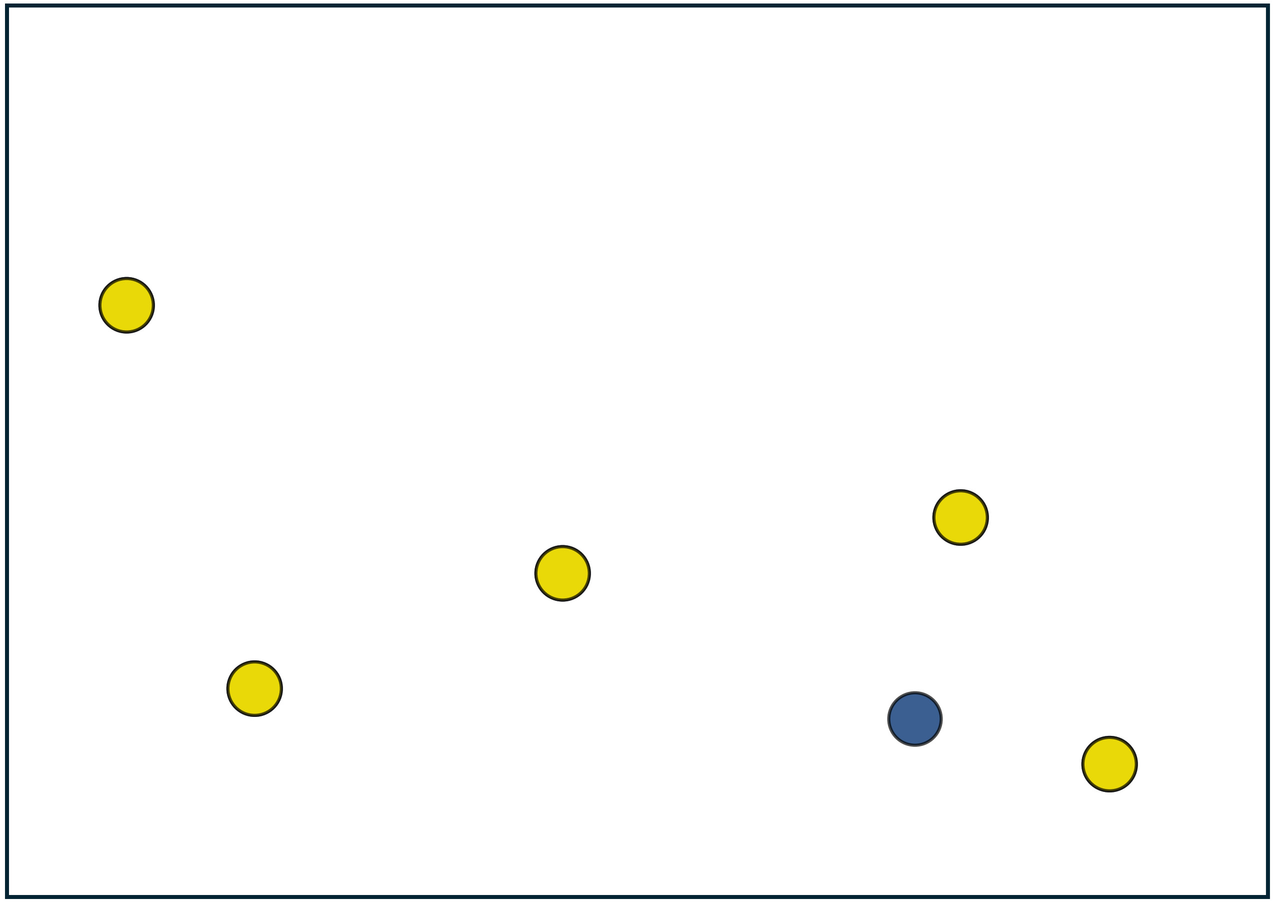
Final Counts:





High Level Idea

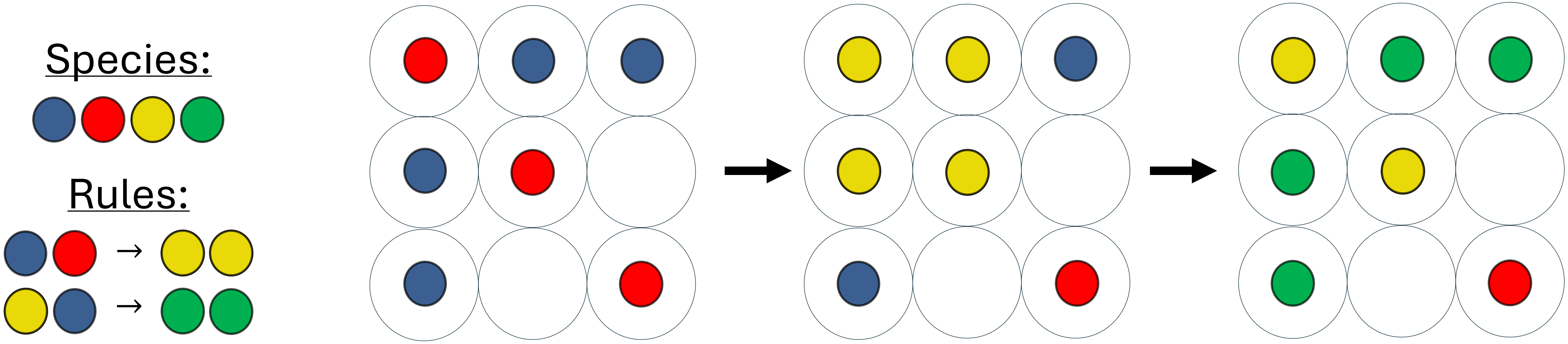




Surface Chemical Reaction Networks

- Ordered pair (Λ, Γ)
  - Λ = Set of input species
  - Γ = Set of rules
- Surface
  - Undirected graph
- Configuration
  - Mapping of input species to vertices on the surface
  - Each vertex has no more than 1 species at a time

Example sCRN System

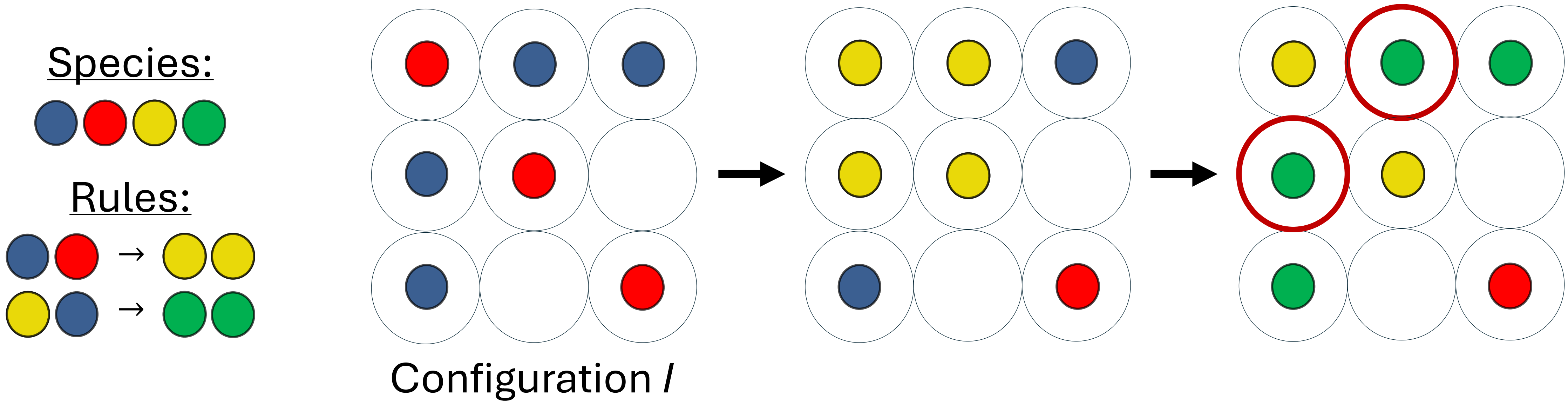


Configuration
I

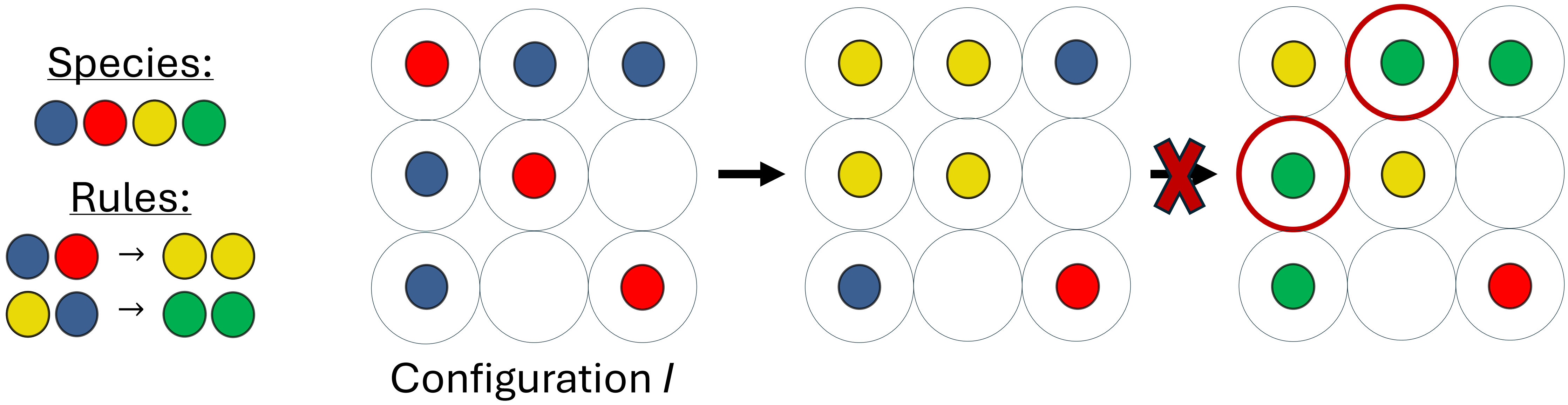
Surface Chemical Reaction Networks

- Ordered pair (Λ, Γ)
  - Λ = Set of input species
  - Γ = Set of rules
- Surface
  - Undirected graph
- Configuration
  - Mapping of input species to vertices on the surface
  - Each vertex has no more than 1 species at a time
- Burnout
  - Limit to how many times a vertex can change states

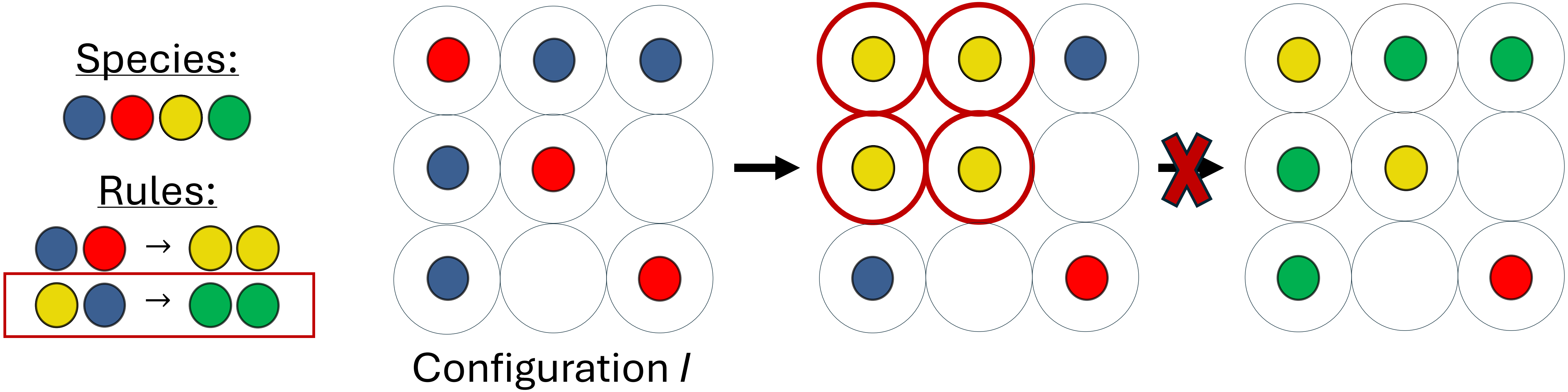
Example sCRN System: 2 burnout



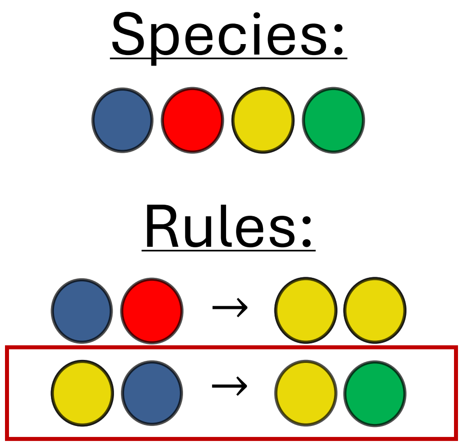
Example sCRN System: 1 burnout

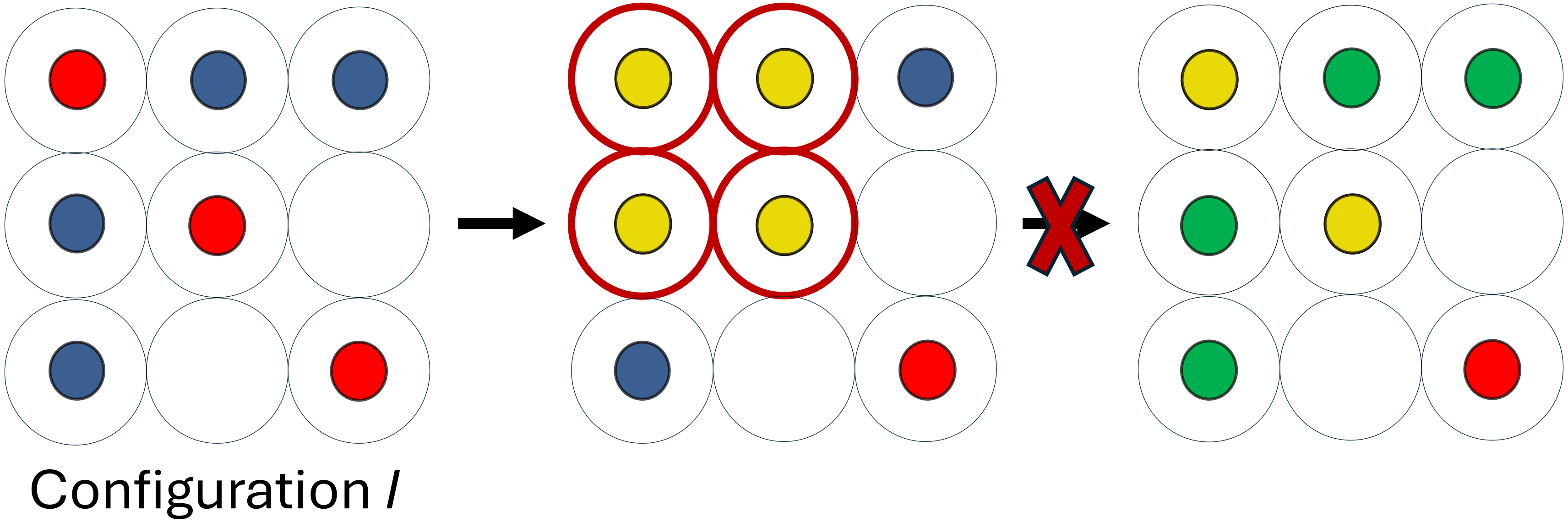


Example sCRN System: 1 burnout

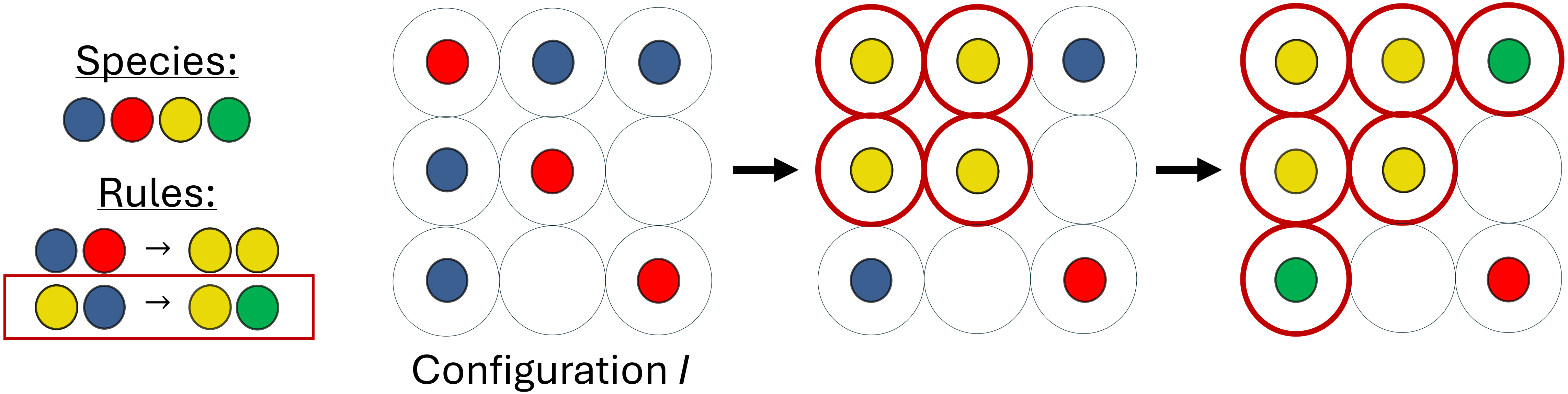


Example sCRN System: 1 burnout



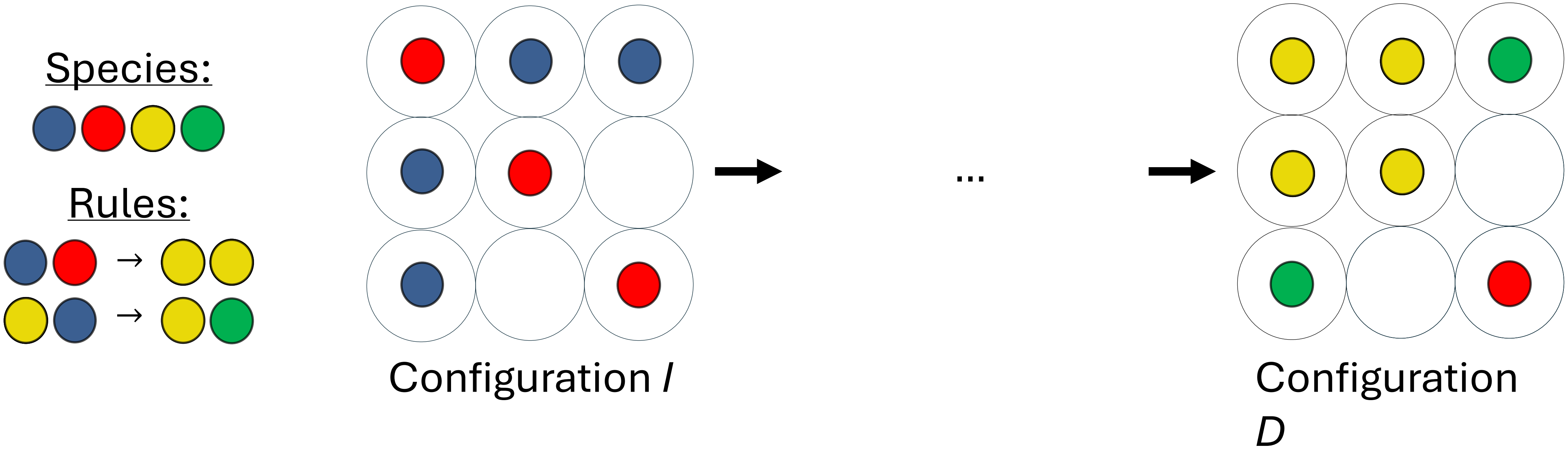


Example sCRN System: 1 burnout



Reconfiguration

- Given initial configuration *I* and target configuration *D*, can *D* ever be reached from *I*?

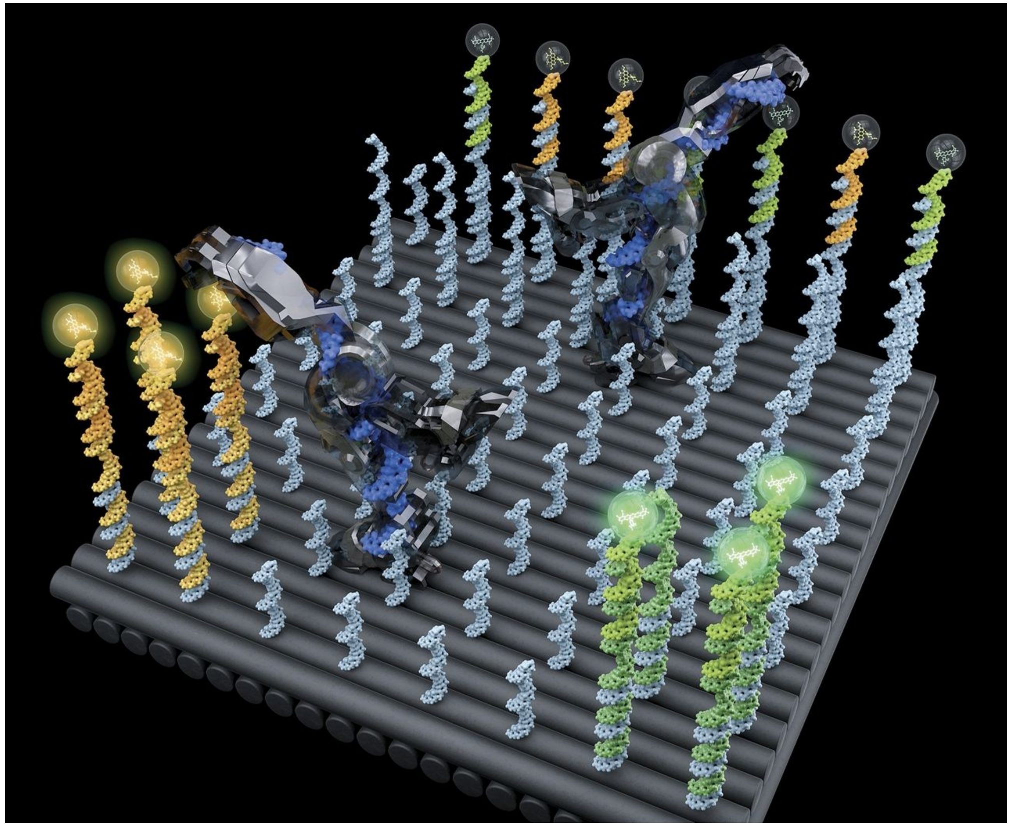


Motivation

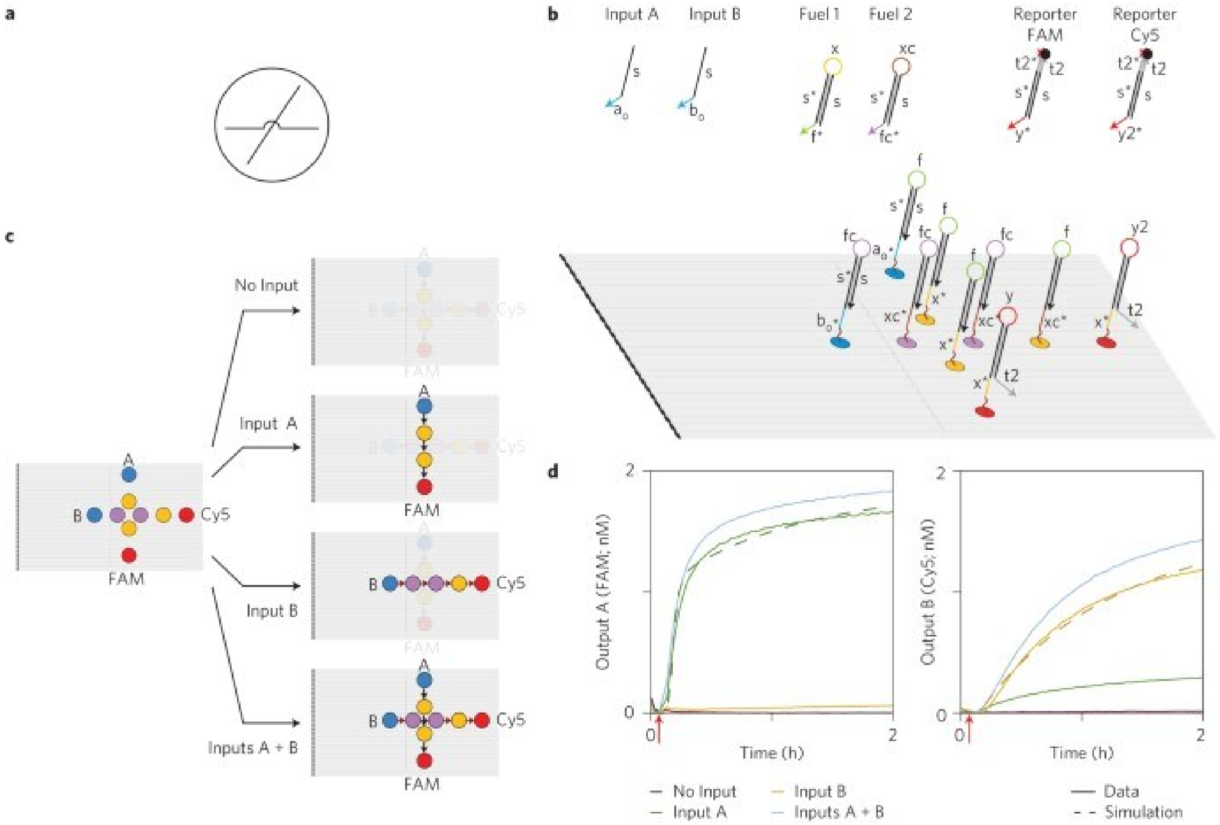
Motivation

- CRNs are practical but limited in computational power
  - Deterministically compute semi-linear functions
- sCRNs help overcome some of these limitations
  - Some geometry
- Burnout reflects how molecules may interact
  - Molecules with limited lifetimes
  - Reactions that consume large quantities of energy

Motivation



A. J. Thubagere, W. Li, R. F. Johnson, Z. Chen, S. Doroudi, Y. L. Lee, G. Izatt, S. Wittman, N. Srinivas, D. Woods, et al. **A cargo-sorting DNA robot**. Science, 357(6356):eaan6558, 2017.



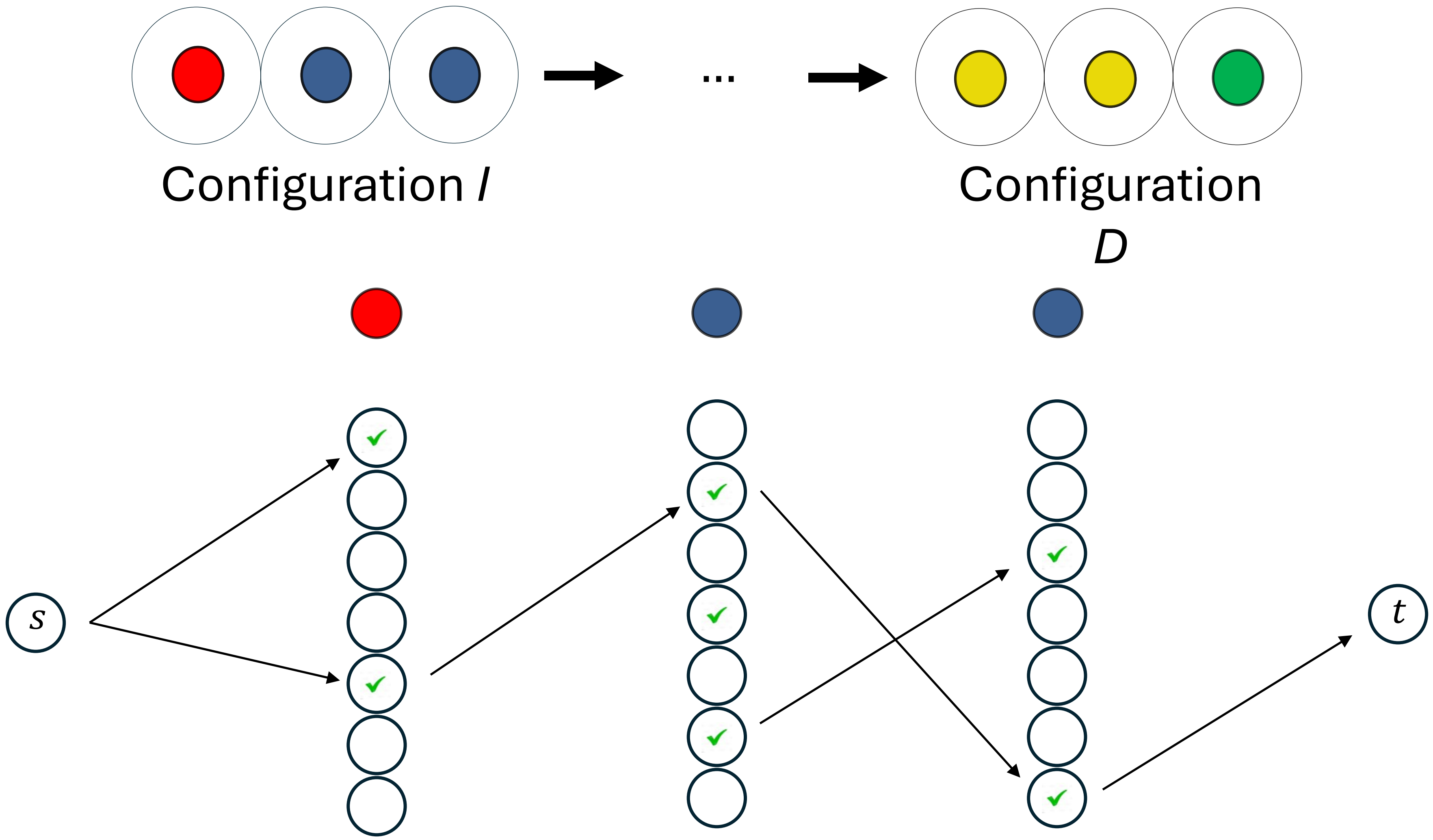
G. Chatterjee, N. Dalchau, R. A. Muscat, A. Phillips, and G. Seelig. **A spatially localized architecture for fast and modular DNA computing**. Nature nanotechnology, 12(9):920–927, 2017.

Linear Surfaces

Problem	Burnout	Result
Reconfiguration	$\mathbf{1}$	$\boldsymbol{O(n + \|R\|)}$
Reconfiguration	$\mathbf{2}$	$\boldsymbol{O(n \cdot \|S\|^2 \cdot \|R\|^4)}$
Reconfiguration	$\boldsymbol{O(1)}$	**P**
Reconfiguration	$\boldsymbol{k\ (unary)}$	$\boldsymbol{NP - Complete}$
Reconfiguration	$None$	$PSPACE - Complete$

Robert Alaniz, Michael Coulombe, Erik Demaine, Bin Fu, Tim Gomez, Elise Grizzell, Ryan Knobel, Andrew Rodriguez, Robert Schweller, Tim Wylie. Proc. of the 35th Candian Conf. on Computational Geometry (CCCG 23).

1-burnout



Conclusion

Known + New Results

Linear Surfaces

Problem	Burnout	Result
Reconfiguration	**1**	$\boldsymbol{O(n + \lvert R \rvert)}$
Reconfiguration	**2**	$\boldsymbol{O(n \cdot \lvert S \rvert^2 \cdot \lvert R \rvert^4)}$
Reconfiguration	$\boldsymbol{O(1)}$	**P**
Reconfiguration	$\boldsymbol{k\ (unary)}$	$\boldsymbol{NP - Complete}$
Reconfiguration	$None$	$PSPACE - Complete$

General Surfaces

Problem	Burnout	Result
Reconfiguration	**1**	$\boldsymbol{O(V^{1.5} + \lvert R \rvert)^*}$
Reconfiguration	**1**	$\boldsymbol{NP - Complete}$
Reconfiguration	2	$NP - Complete$

* Non-catalytic reactions

Grid Surfaces

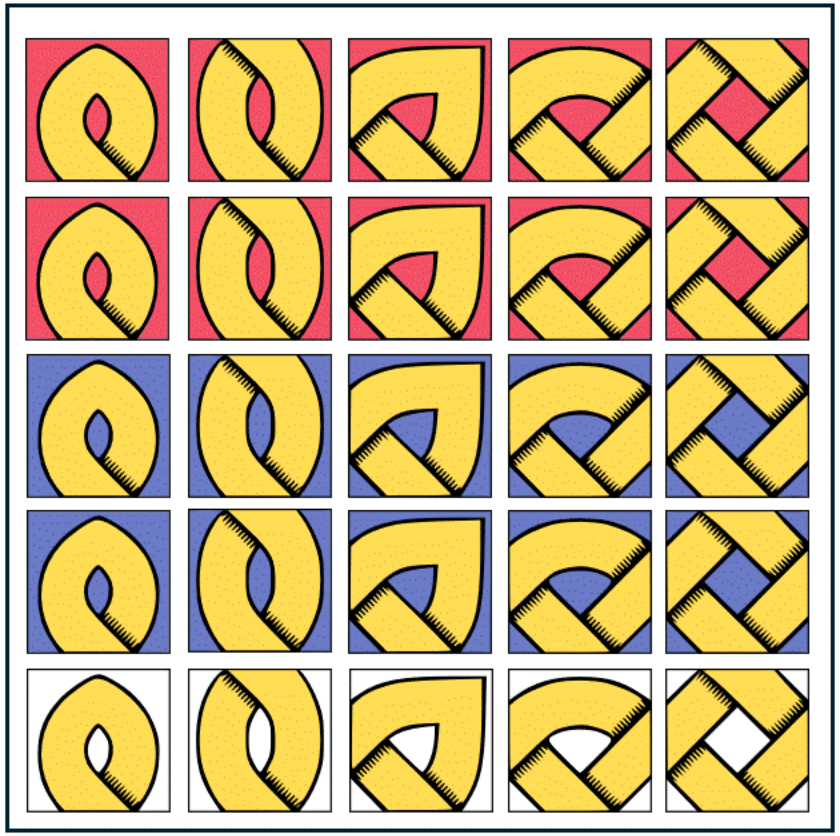
Problem	Burnout	Result
1-Reconfiguration	**1**	$\boldsymbol{O(n \cdot (\lvert S \rvert \lvert R \rvert)^{2m} \cdot f(m))}$
1-Reconfiguration	1	$NP - Complete$

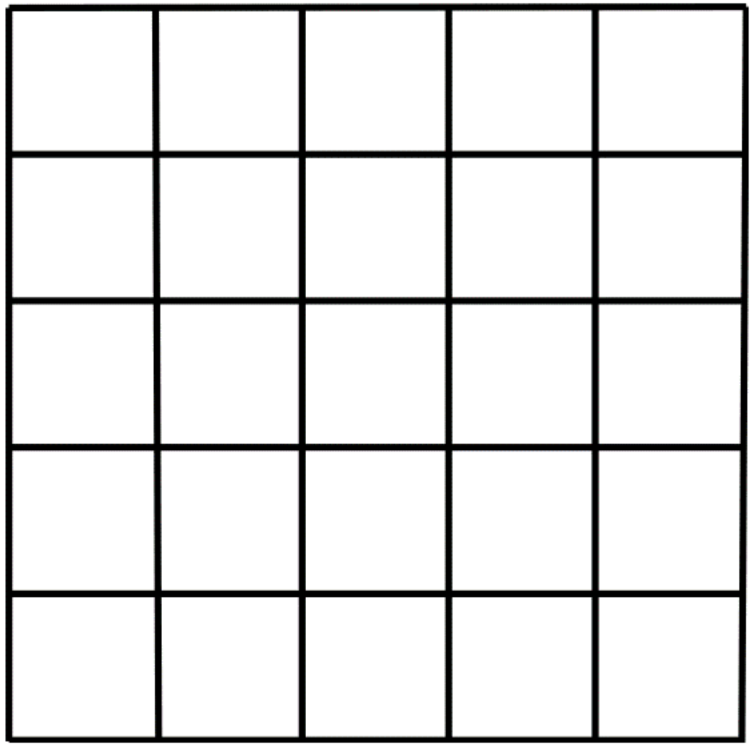
Open Problems

- Lower bounds for k-burnout on linear surfaces
- Other than size of surface, what about:
  - Species, rules and burnout
- Any FPT algorithms for reconfiguration

Game Complexity

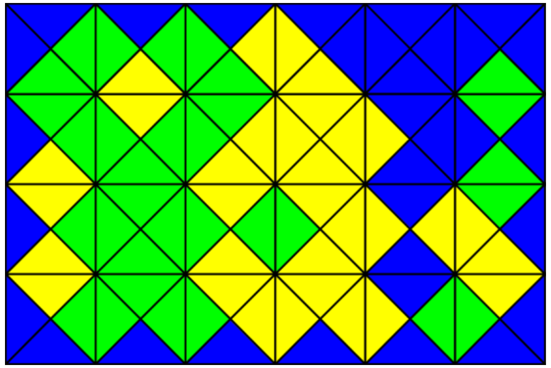
Tile-based Knot Assembly with Celtic!





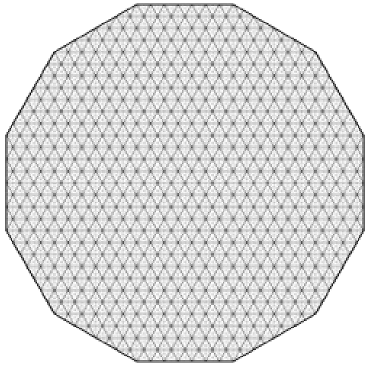
Authors: Divya Bajaj, **Ryan Knobel**, Juan Manuel Perez, Rene Reyes, Ramiro Santos, Tim Wylie

Background

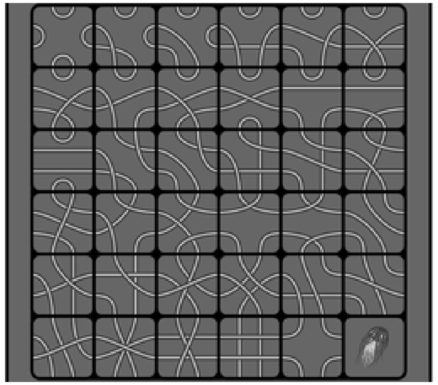


MacMahon Squares

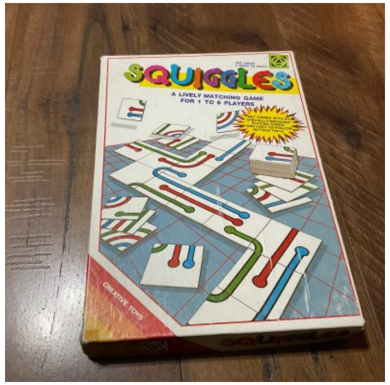
Jigsaw Puzzles



Eternity

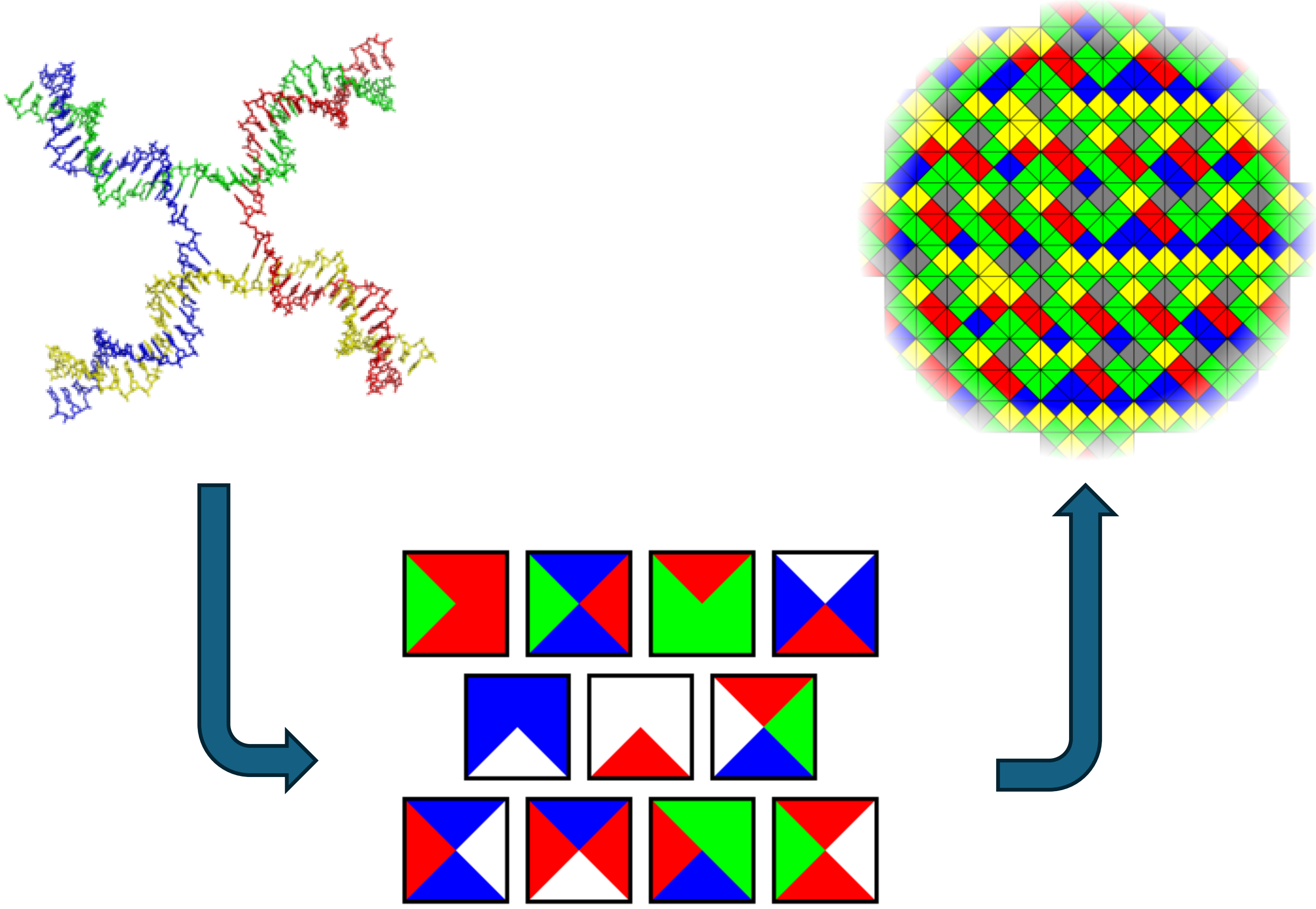


Tsuro

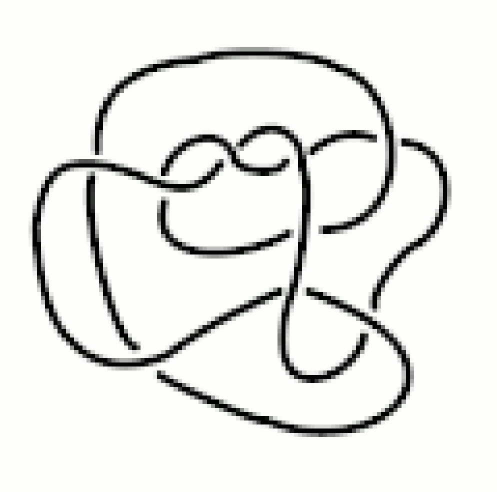


Squiggles

Background

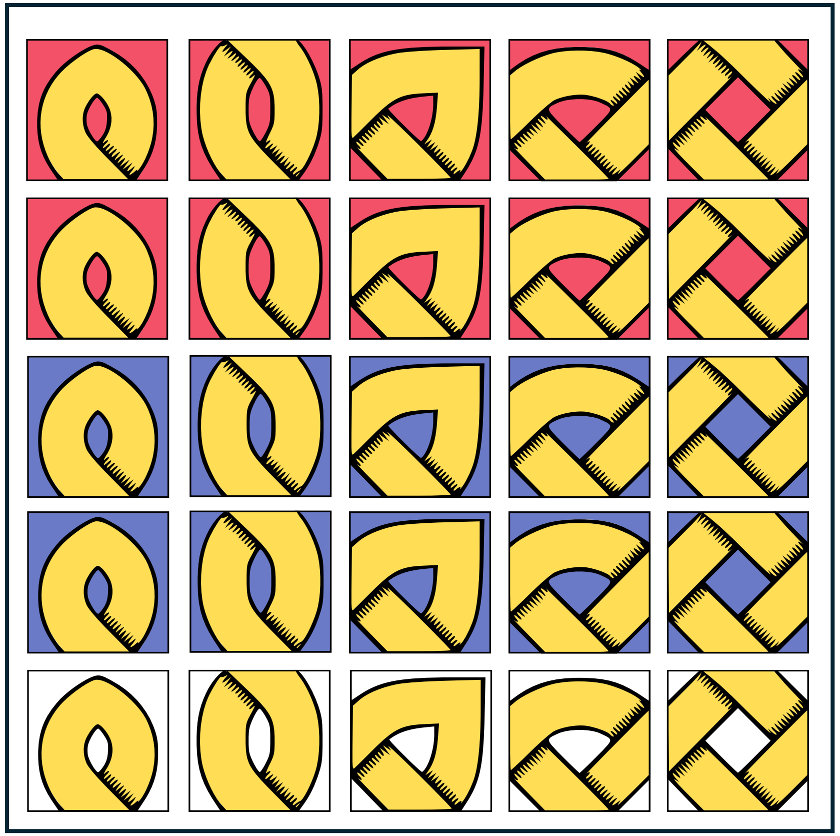


Wang tiles

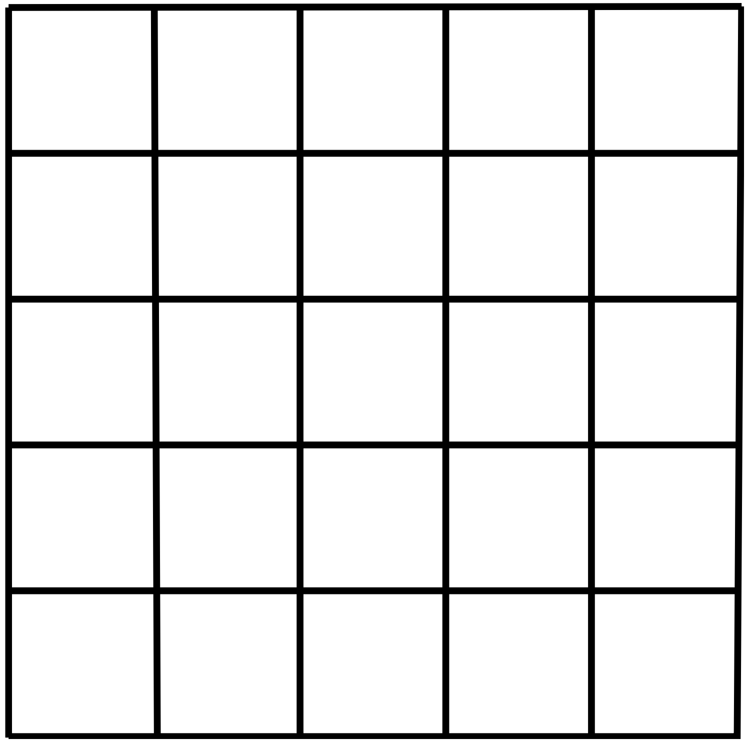


[Knot Theory Wikipedia]

Celtic!

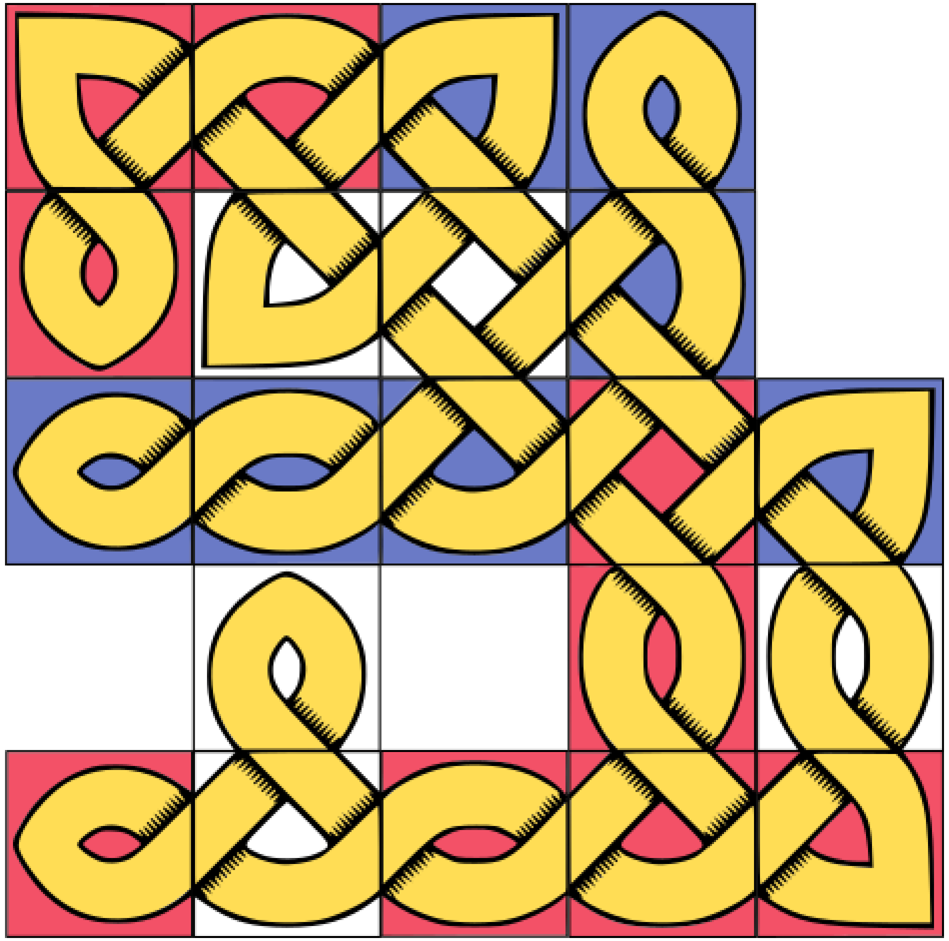


Pieces

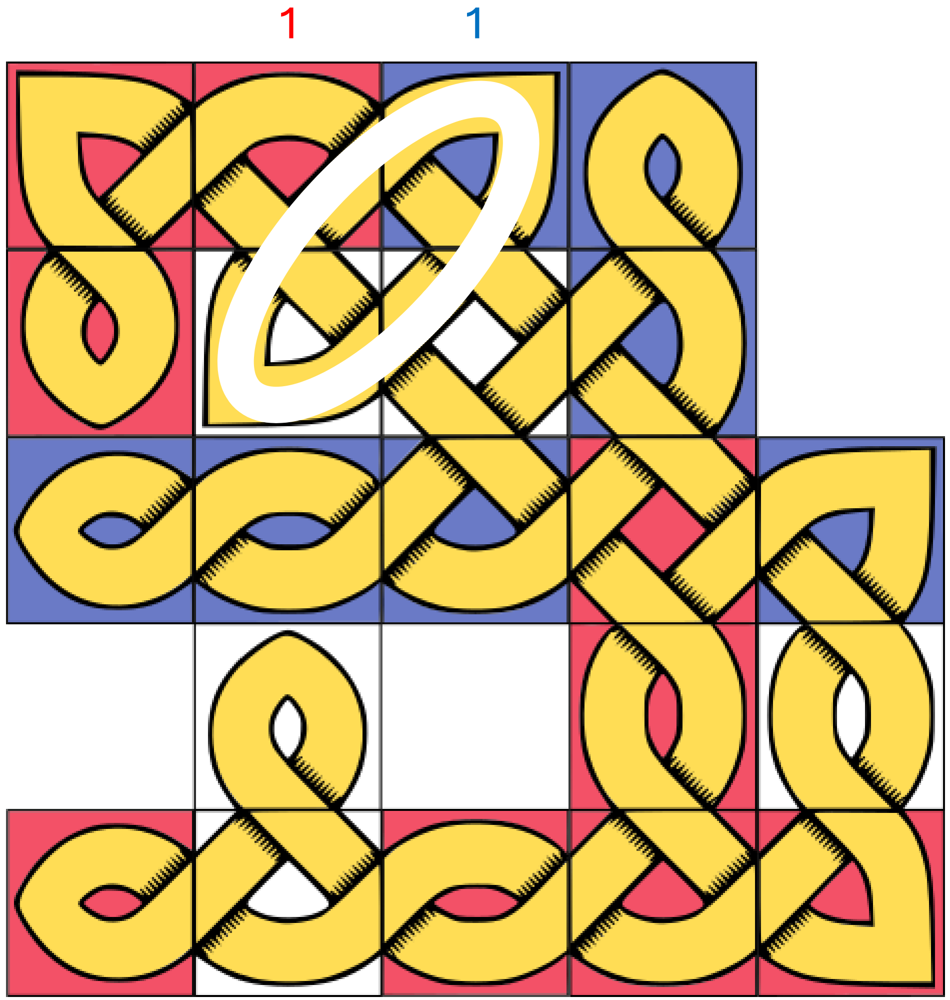


Board

Celtic! - Scoring



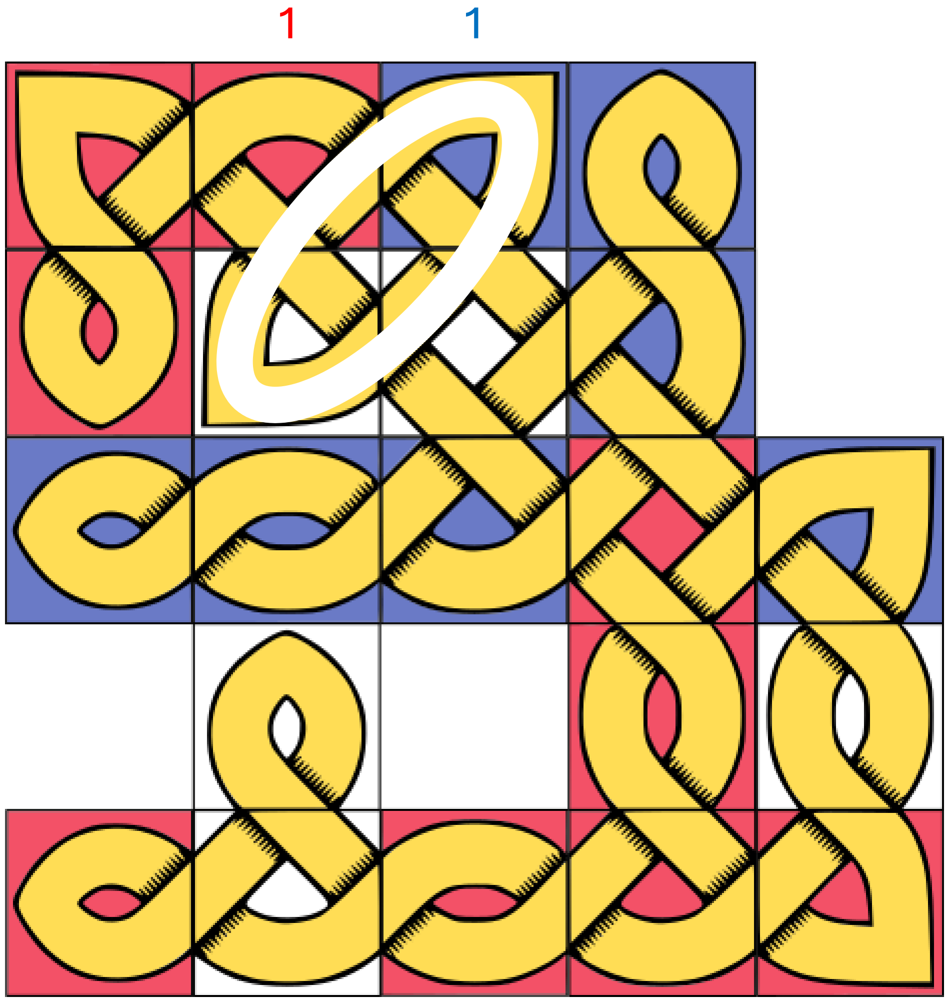
Celtic! - Scoring



Celtic! - Scoring

Score

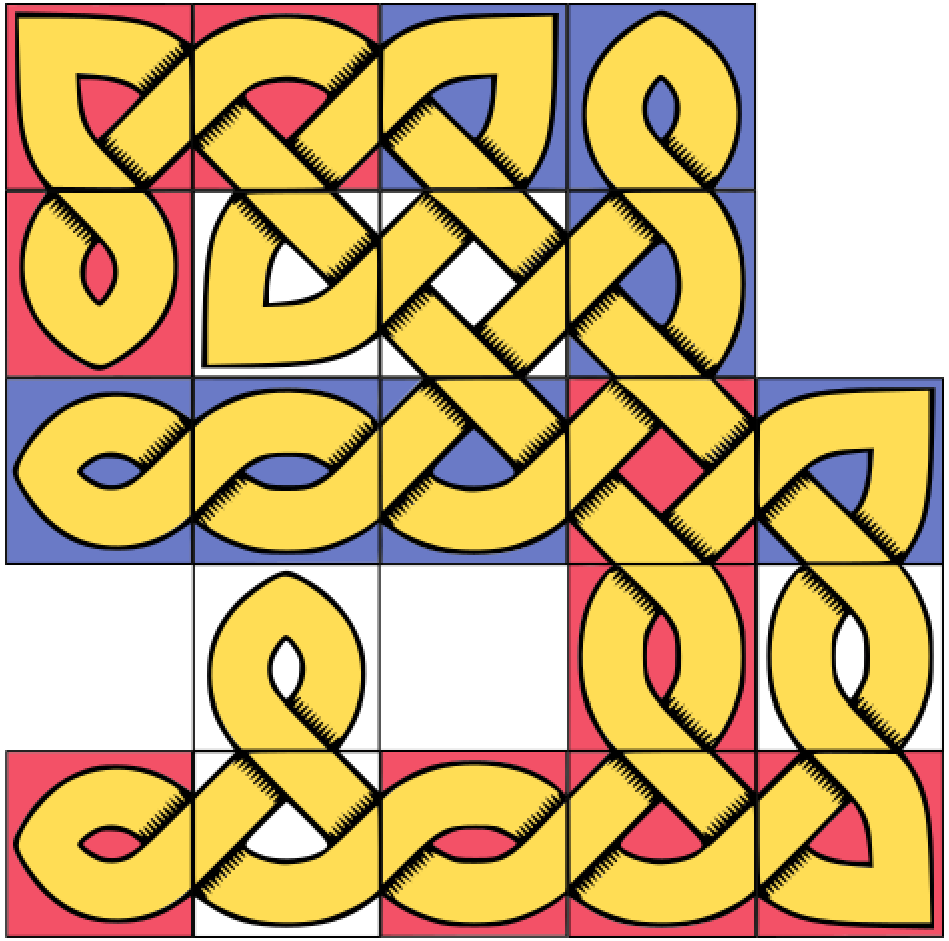
Knot 1: Tie



Celtic! - Scoring

Score

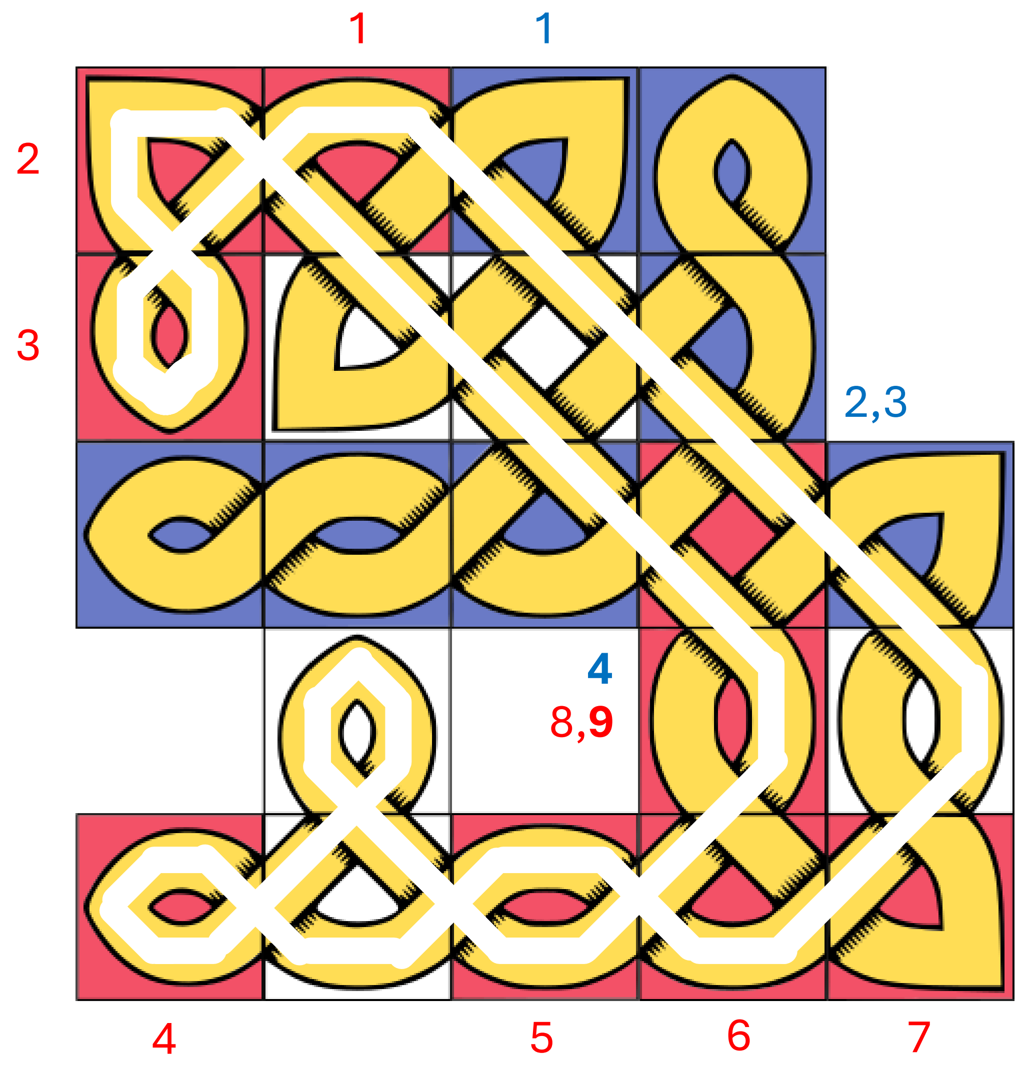
Knot 1: Tie



Celtic! - Scoring

Score

Knot 1: Tie

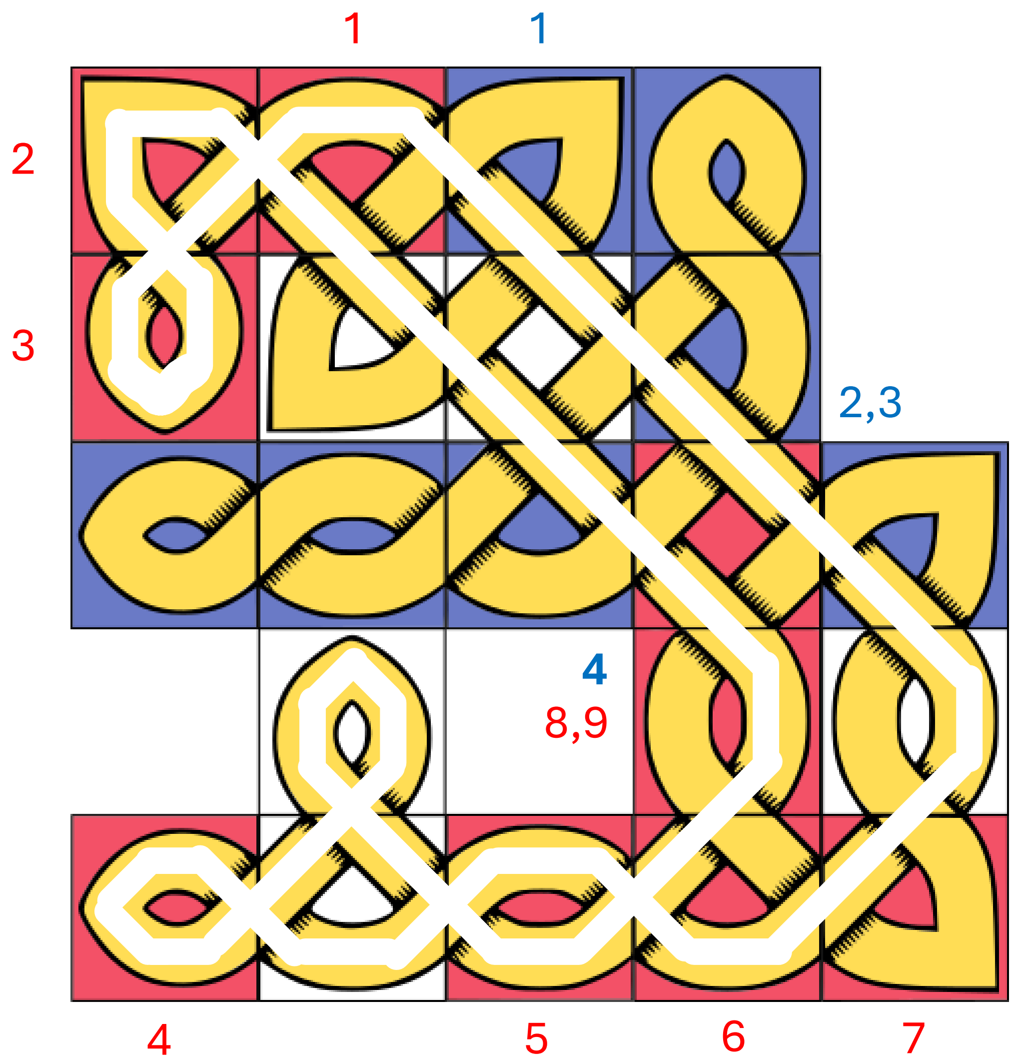


Celtic! - Scoring

Score

Knot 1: Tie

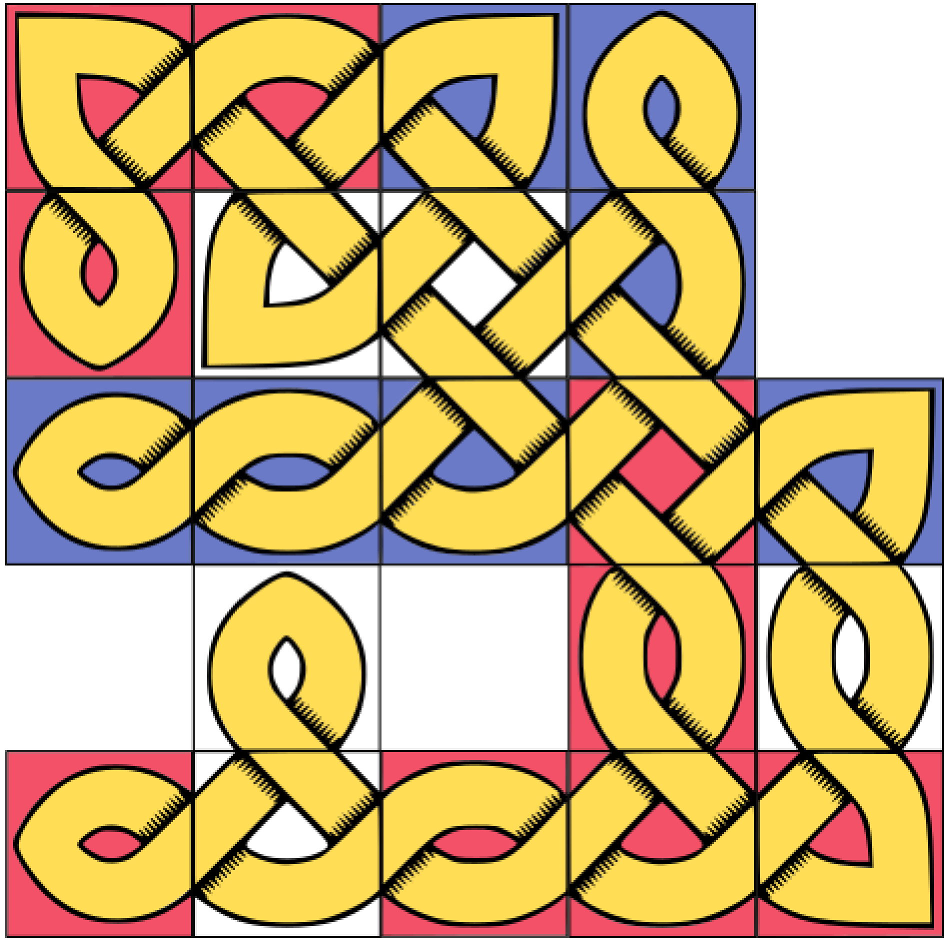
Knot 2: Red (9)



Celtic! - Scoring

Score

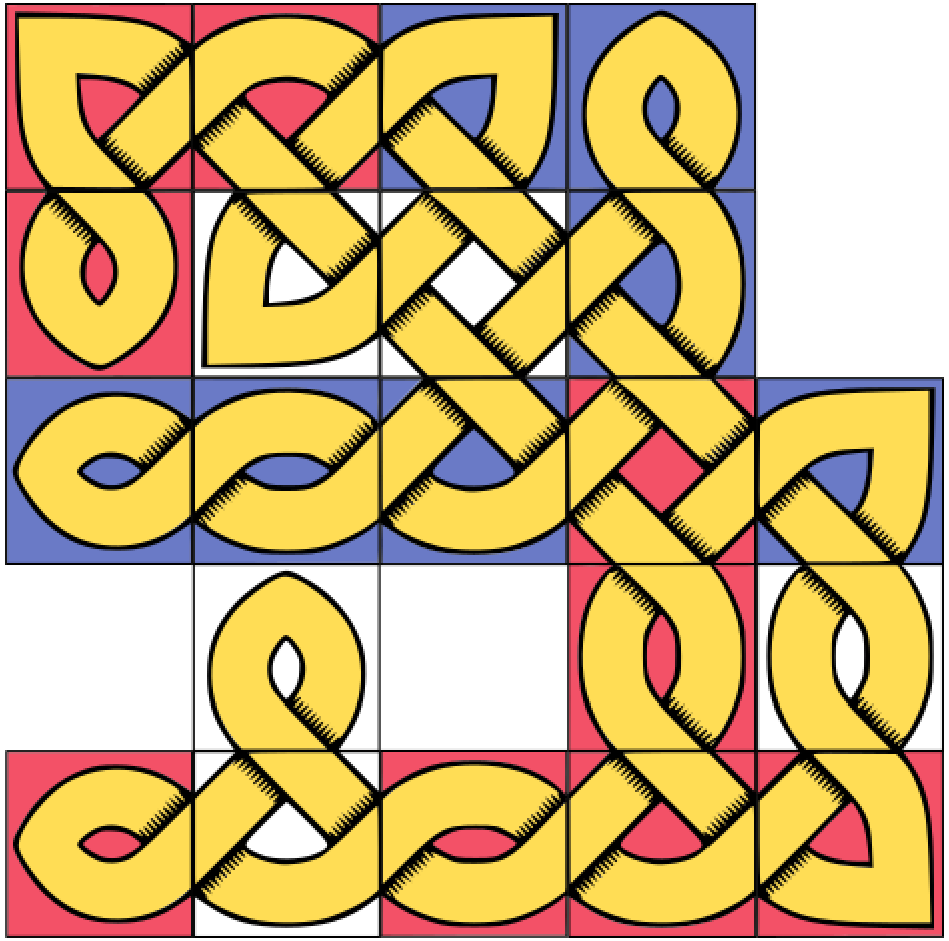
Knot 1: Tie
Knot 2: Red (9)



Celtic! - Scoring

<u>Score</u>

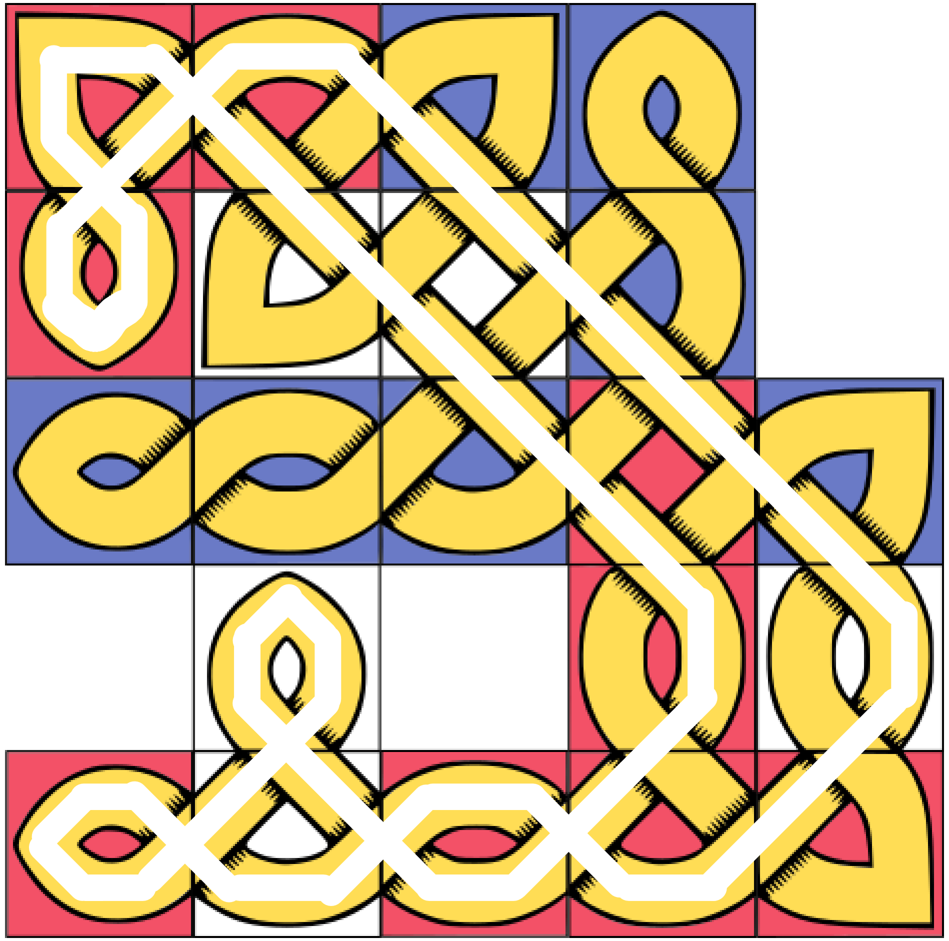
Knot 1: Tie
Knot 2: Red (9)
Knot 3: Blue (5)
Knot 4: Red (4)



Celtic! - Scoring

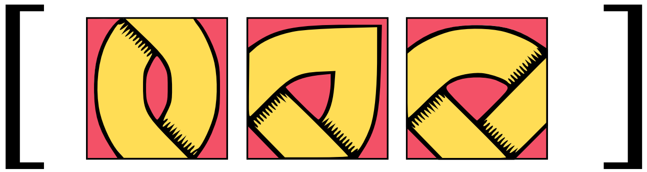
Score

Knot 1: Tie
Knot 2: Red (9)
Knot 3: Blue (5)
Knot 4: Red (4)

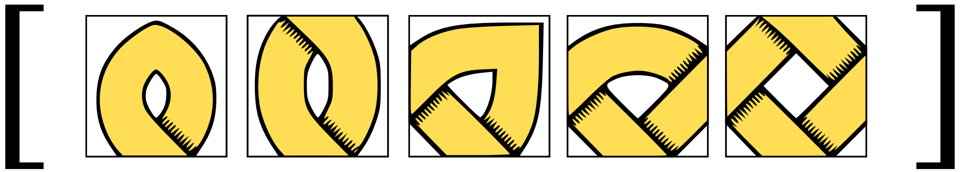


Generalized 1-Player Celtic!

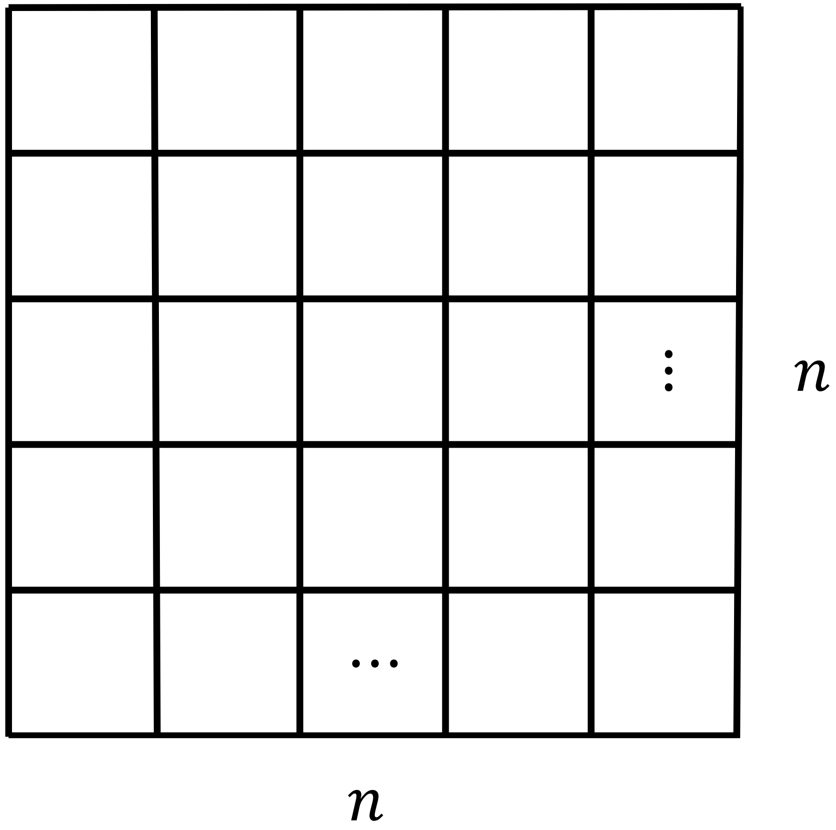
Generalization: 1-Player



Player Pieces

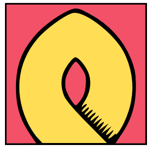
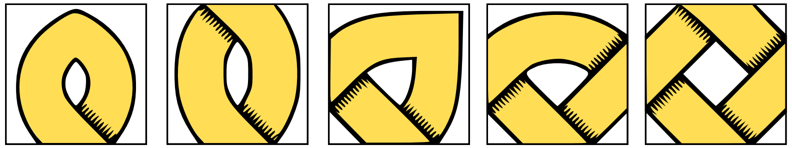
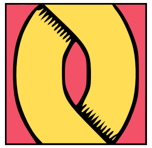


Board Pieces



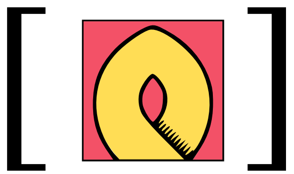
Problem: Can the player build a knot of size $\geq L$ given k pieces?

1-Player Celtic!: The “Easy” problems

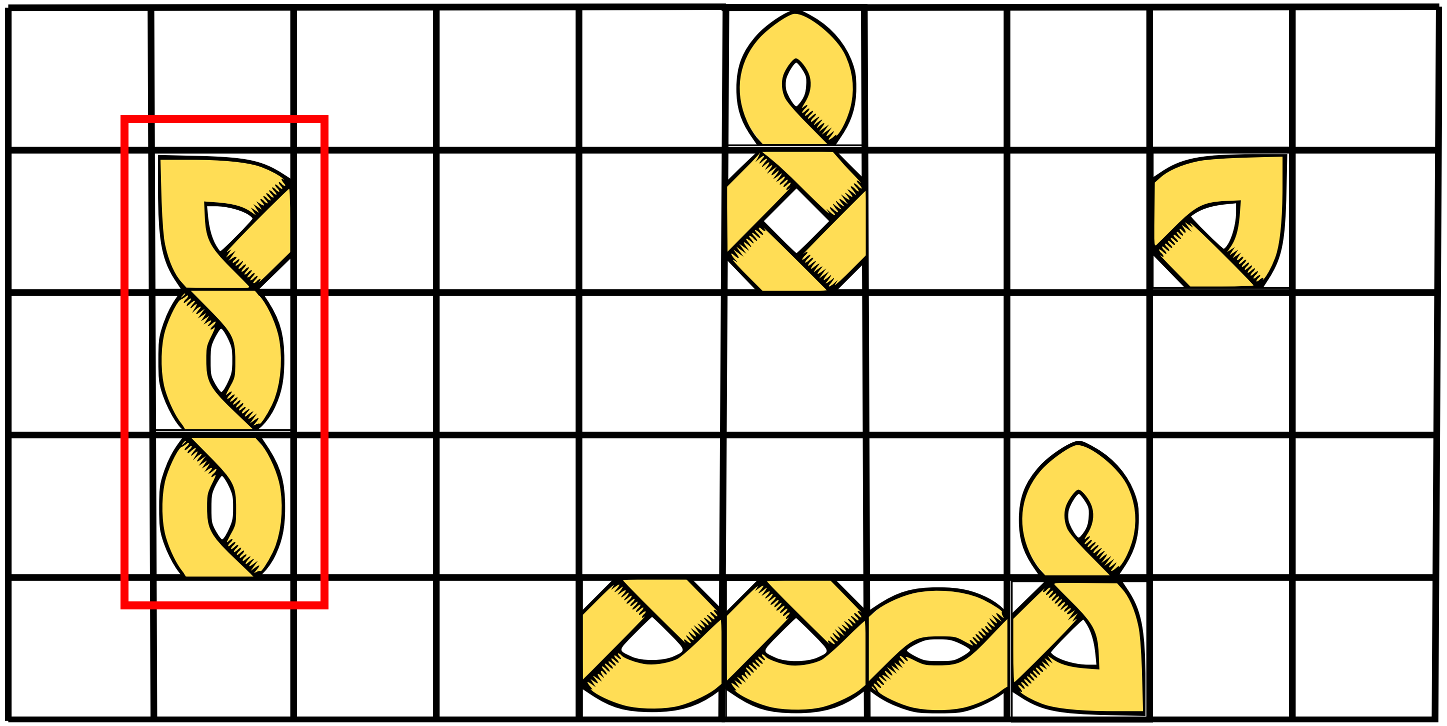
Player Pieces	Board Pieces	Complexity
		$O(n^2)$
		
		

Problem: Can the player build a knot of size $\geq L$ given k pieces?

1-Player Celtic!: The "Easy" problems

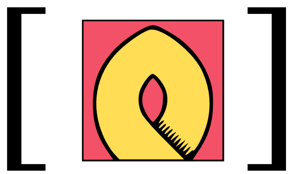


Player Pieces

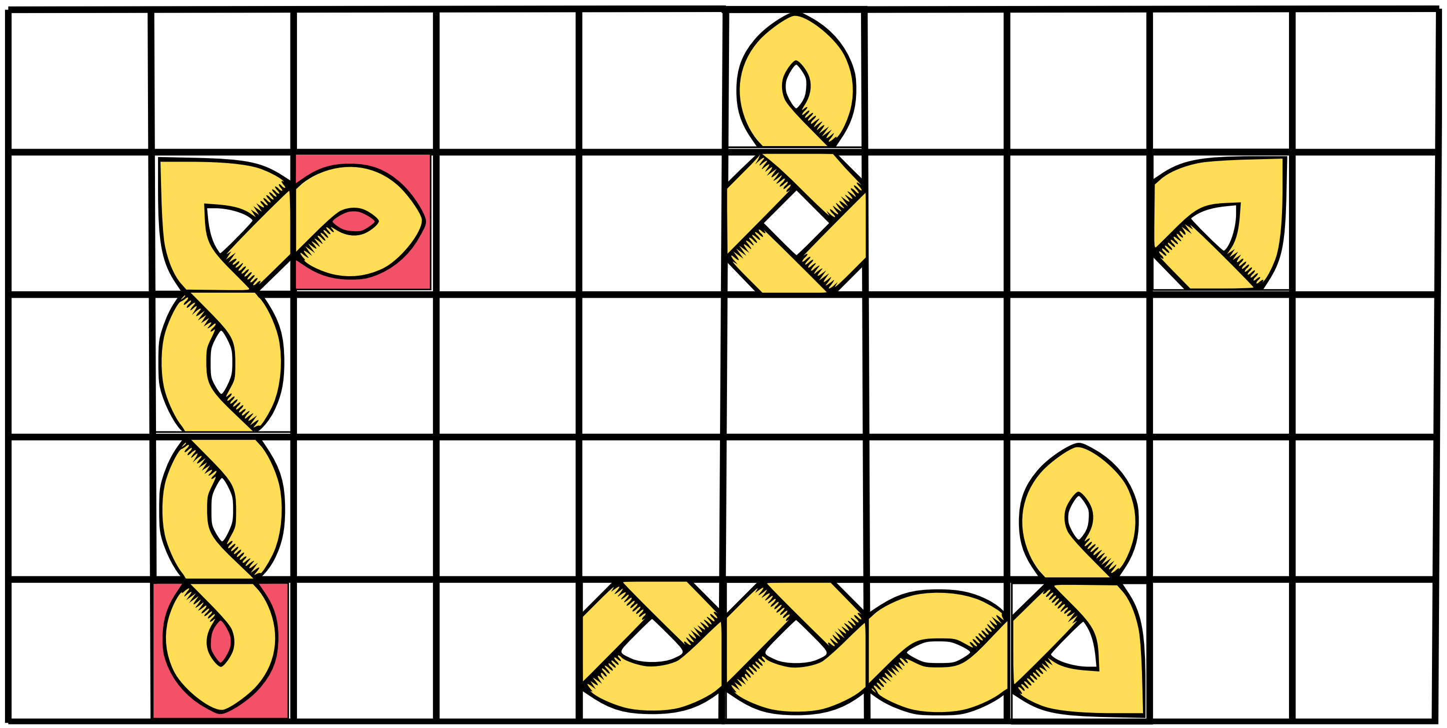


Given: $L = 6, \ k = 4$

1-Player Celtic!: The “Easy” problems

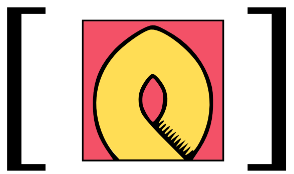


Player Pieces

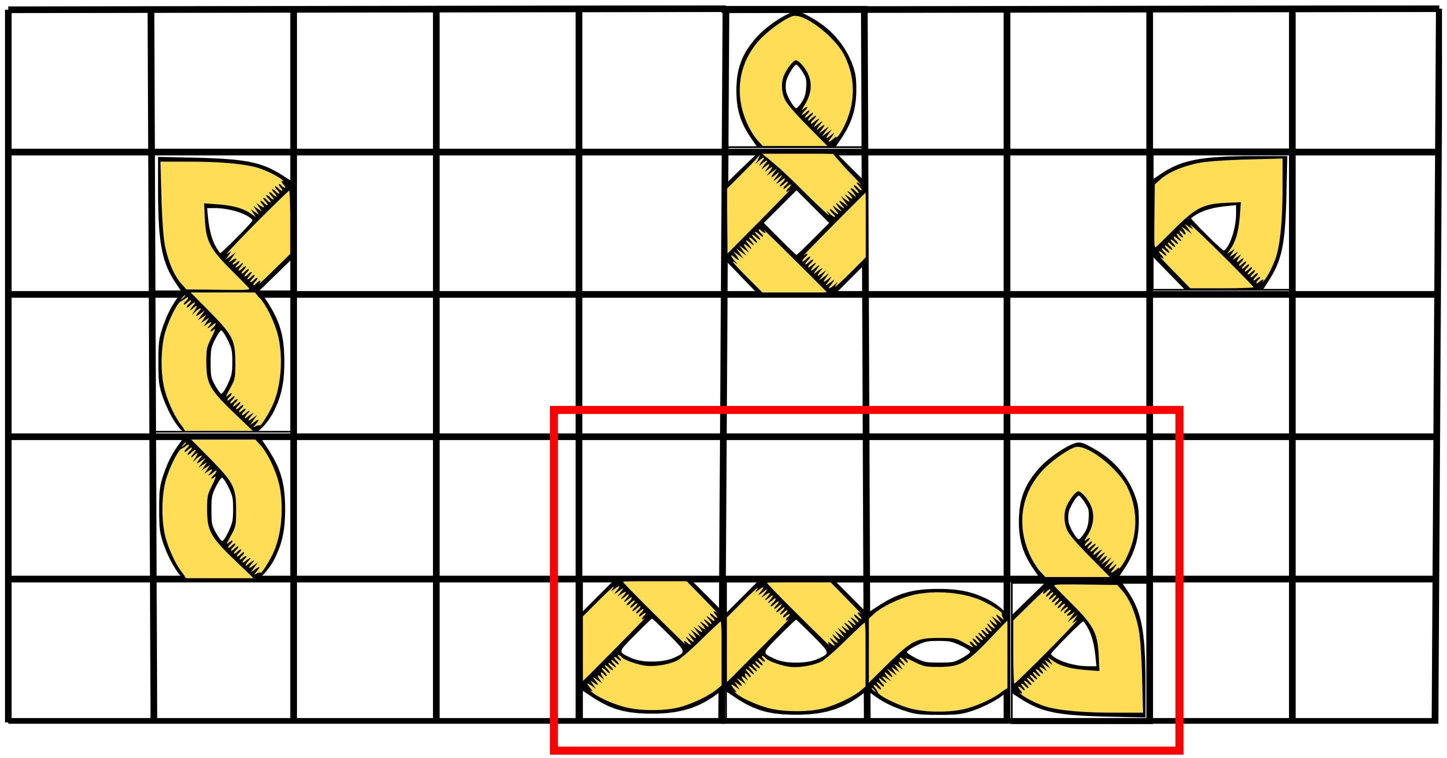


Given: $L = 6,\ k = 4$

1-Player Celtic!: The “Easy” problems

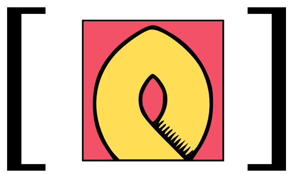


Player Pieces

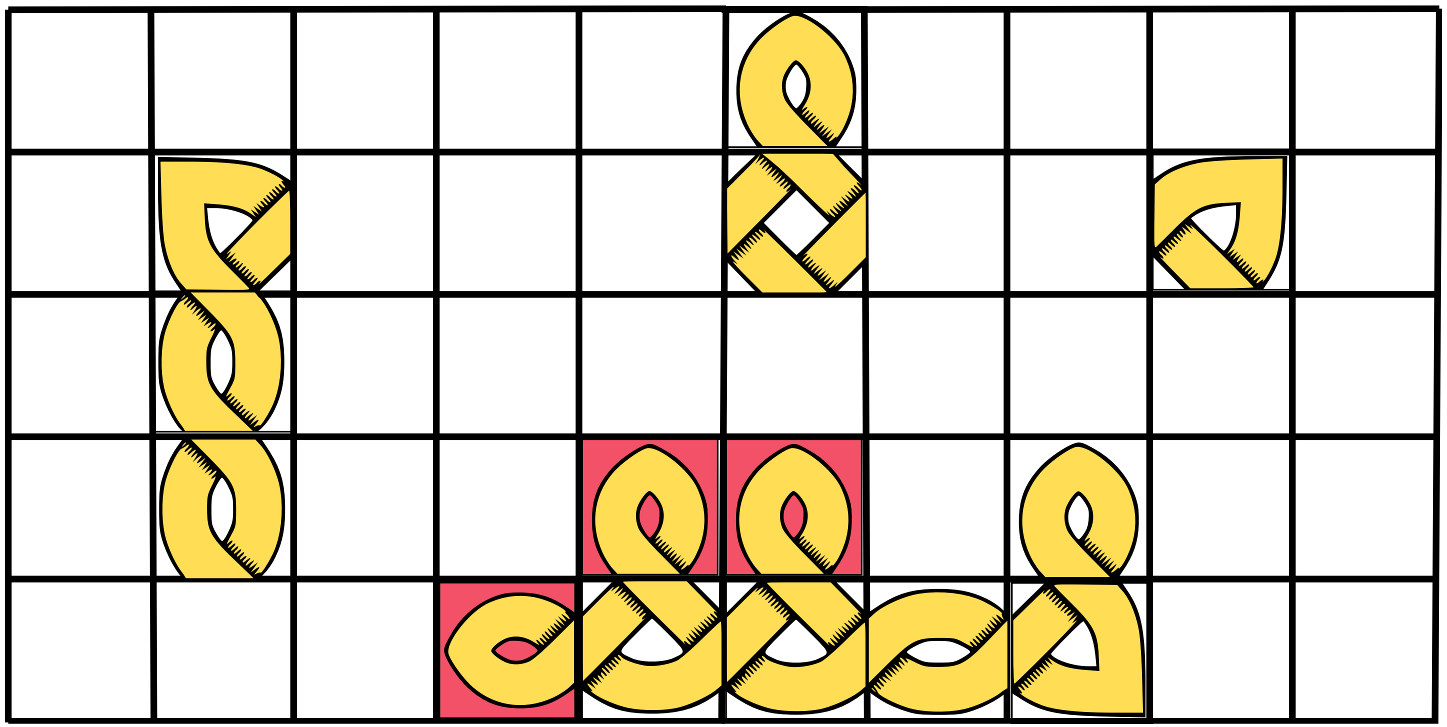


Given: $L = 6,\ k = 4$

1-Player Celtic!: The "Easy" problems

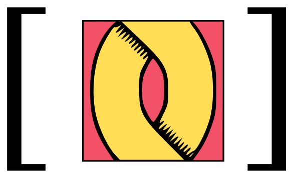


Player Pieces

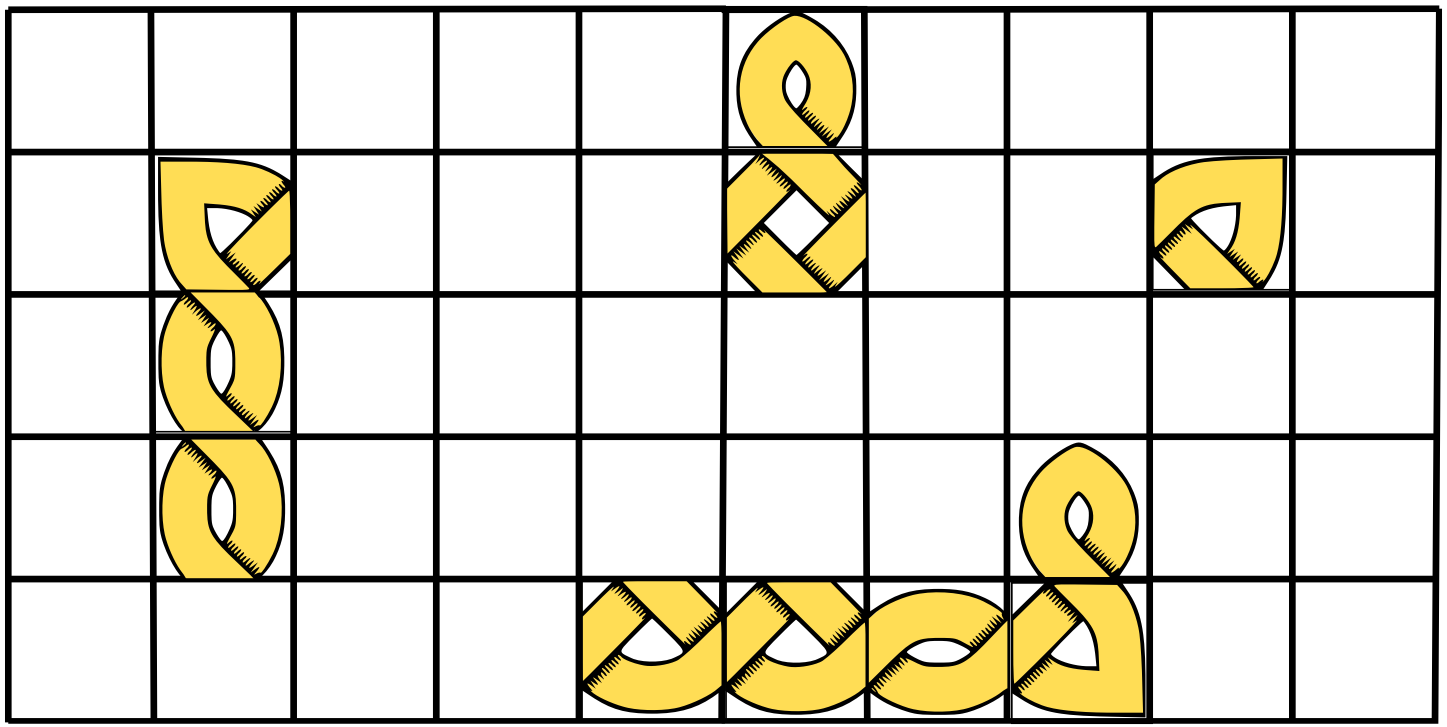


Given: $L = 6, \; k = 4$

1-Player Celtic!: The “Easy” problems

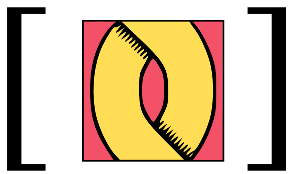


Player Pieces

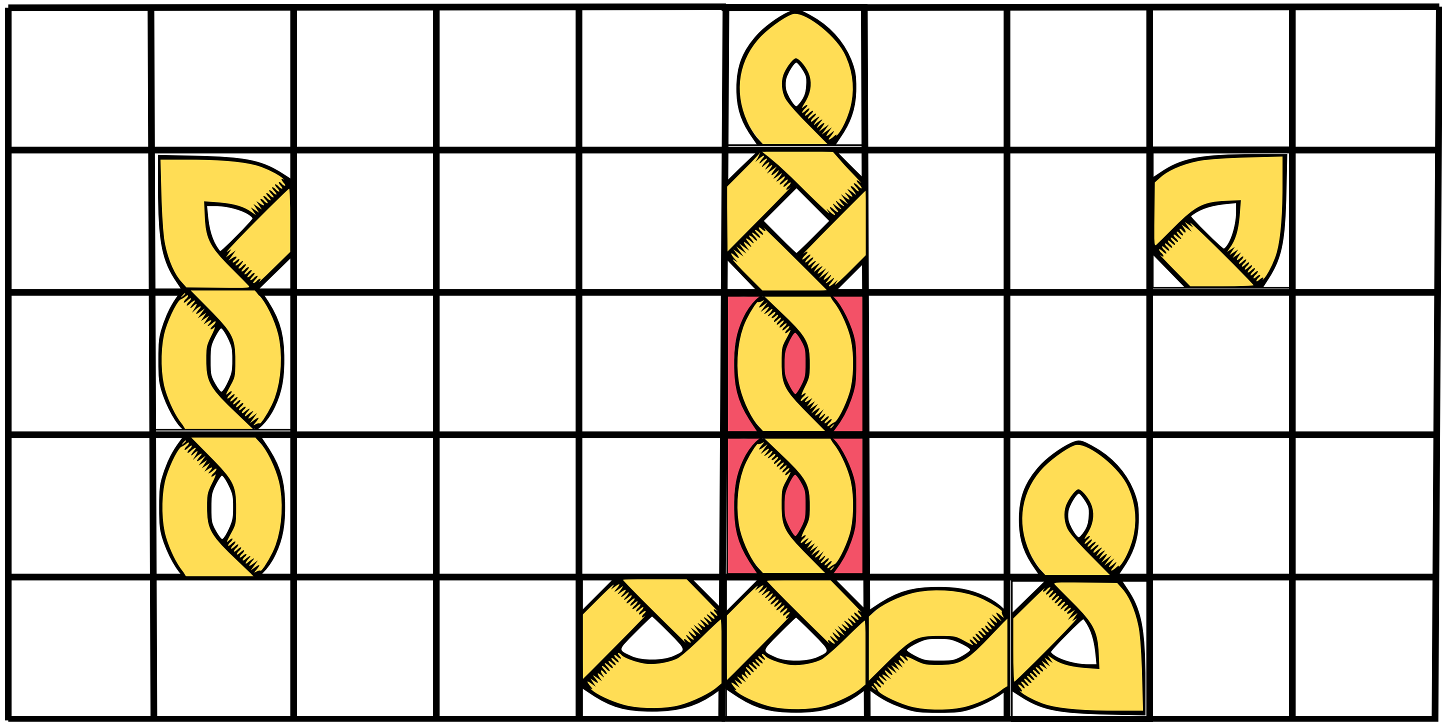


Given: $L = 6,\ k = 4$

1-Player Celtic!: The “Easy” problems

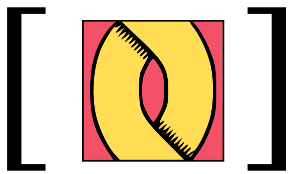


Player Pieces

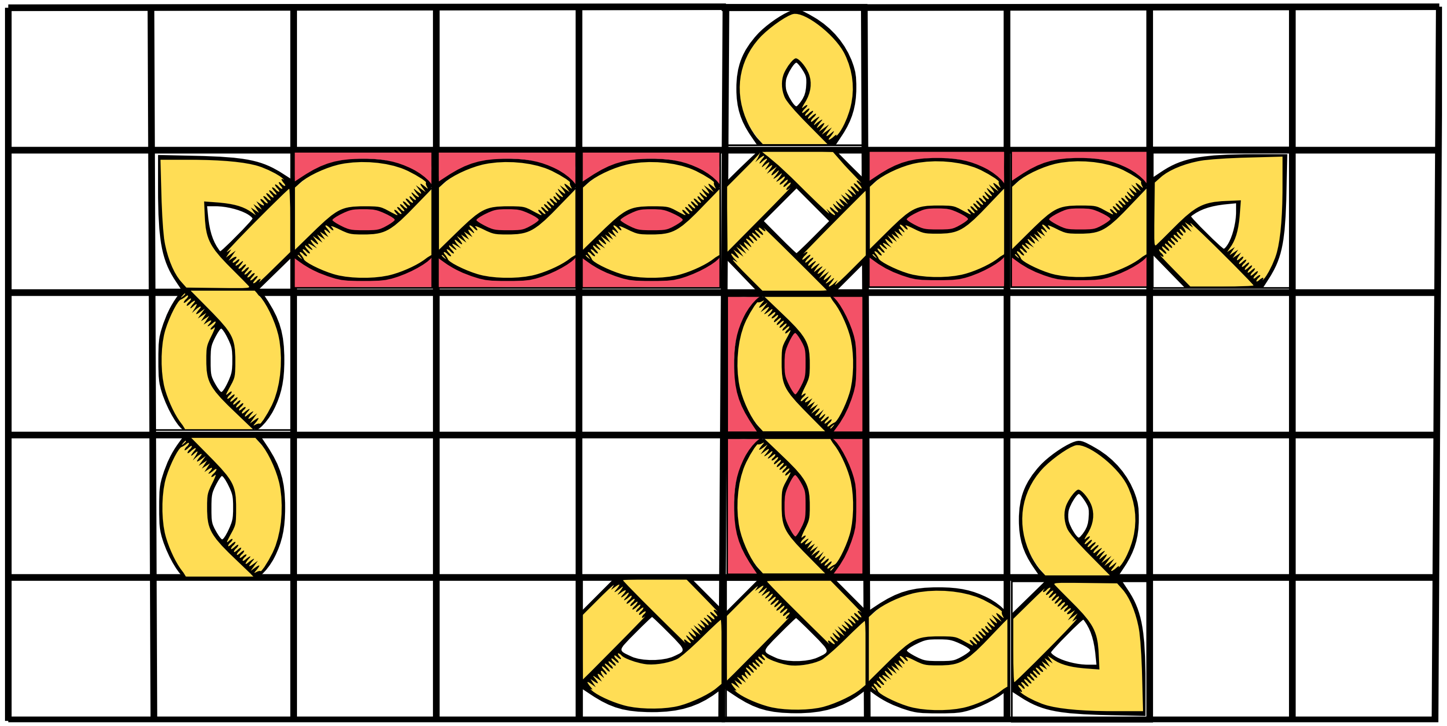


Given: $L = 6, \; k = 4$

1-Player Celtic!: The "Easy" problems

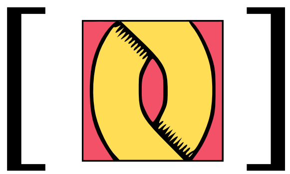


Player Pieces

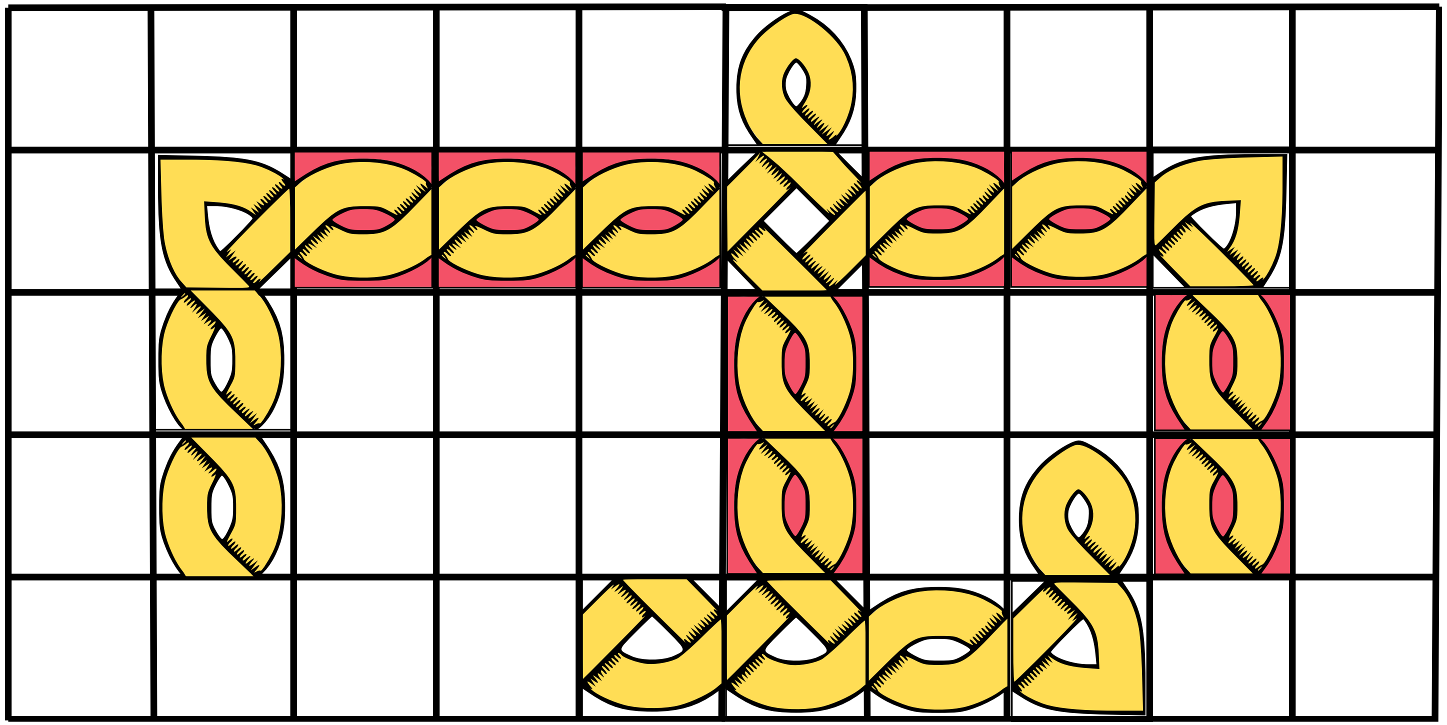


Given: $L = 6,\ k = 4$

1-Player Celtic!: The “Easy” problems

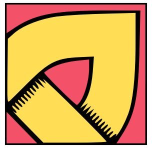
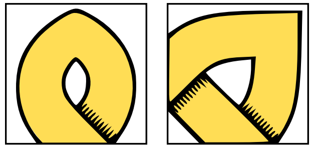
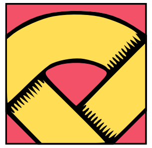
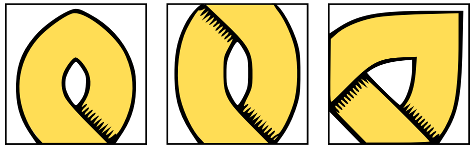
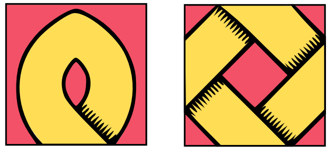


Player Pieces



Given: $L = 6, \; k = 4$

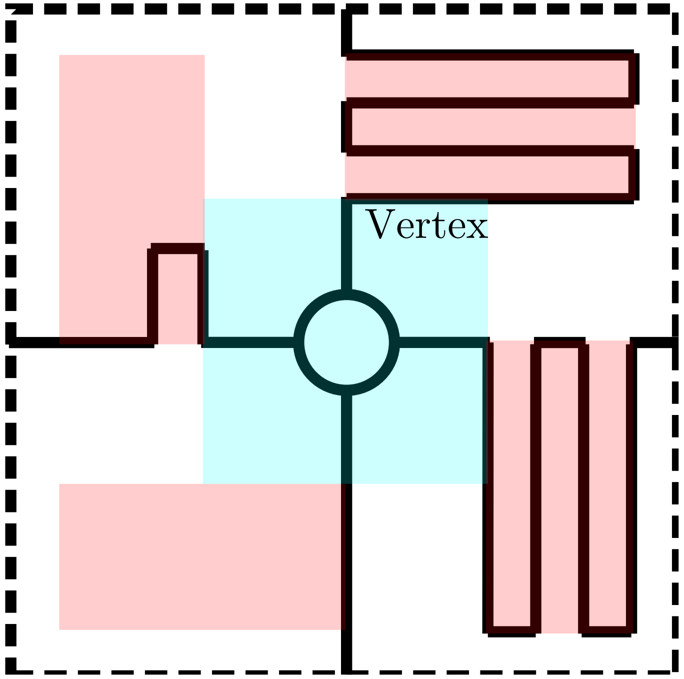
1-Player Celtic!: The “Hard” problems

Player Pieces	Board Pieces	Complexity
		*NP-complete*
		
		

Problem: Can the player build a knot of size $\geq L$ given k pieces?

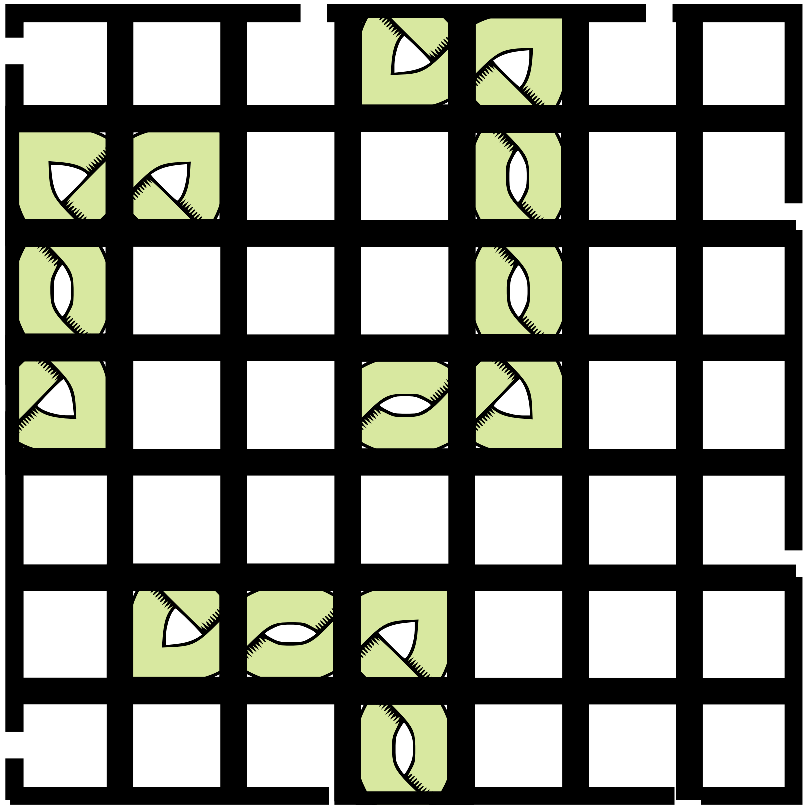
1-Player Celtic!: The “Hard” problems

- Planar, directed Hamiltonian Cycle with max degree 3
- Transform graph into equivalent rectilinear embedding with area $(|V| + 1) \times (|V| + 1)$
- Scale new graph by $O(|V|)$
- Edges are pumped to be roughly the same



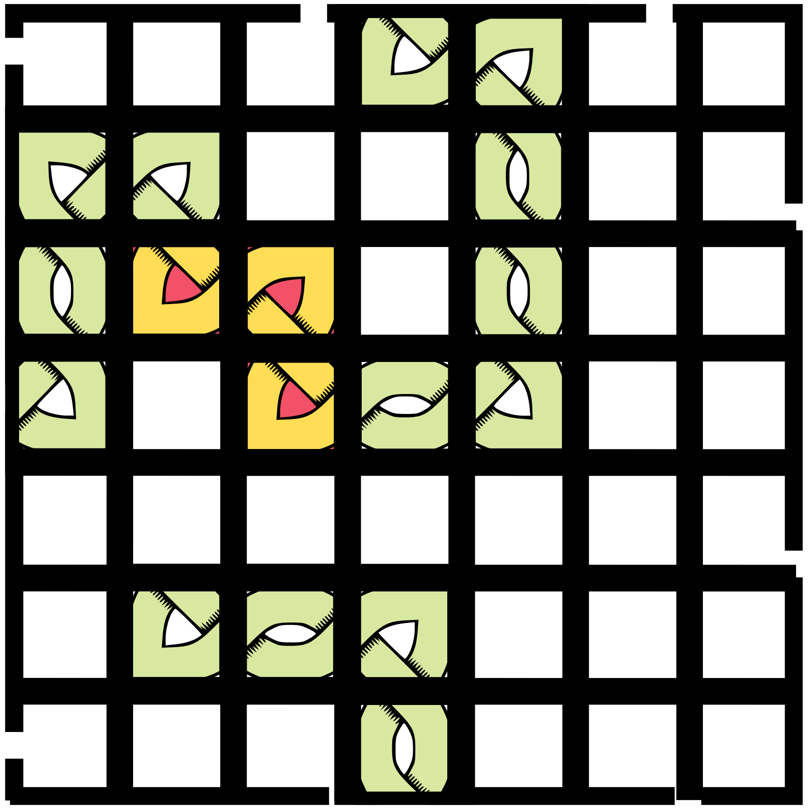
1-Player Celtic!: The “Hard” problems

- Center piece represents the “vertex”
- Incoming edge(s) are moved to one side
- Outgoing edge(s) are moved to the other side
- The “choice” is made by the played pieces

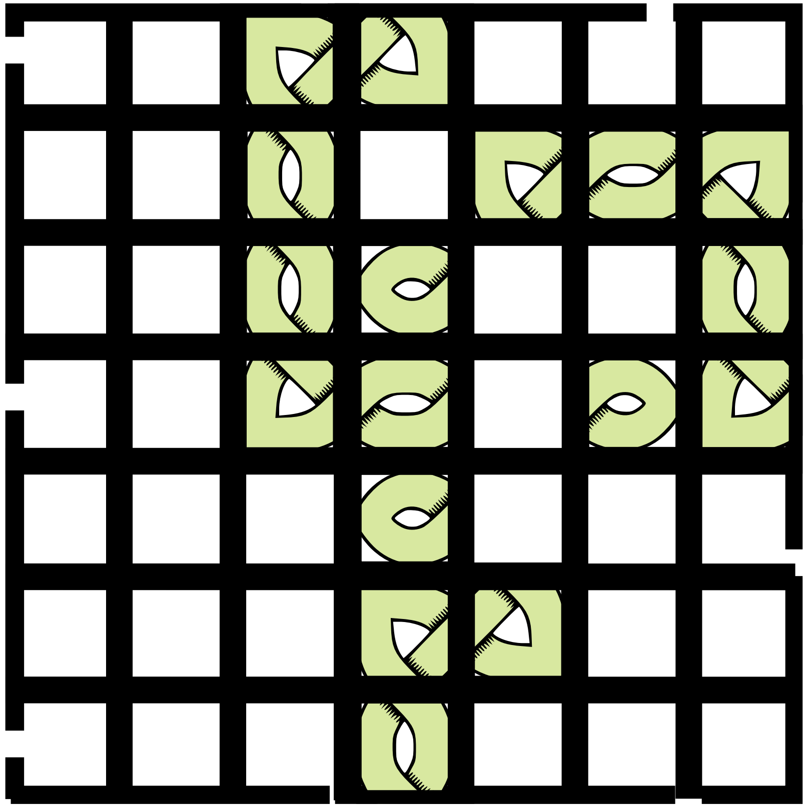


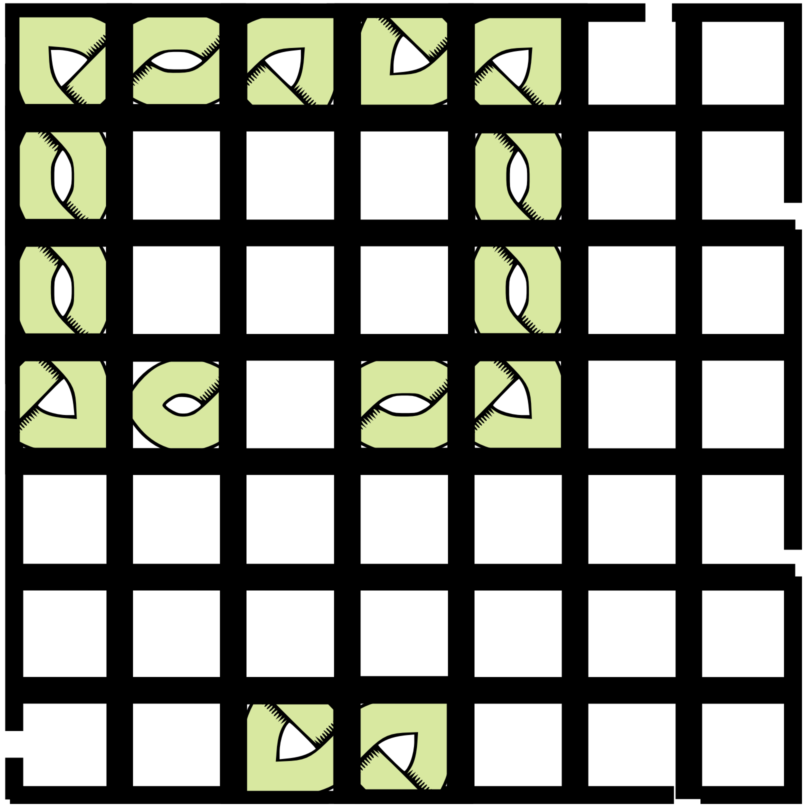
1-Player Celtic!: The “Hard” problems

- Center piece represents the “vertex”
- Incoming edge(s) are moved to one side
- Outgoing edge(s) are moved to the other side
- The “choice” is made by the played pieces

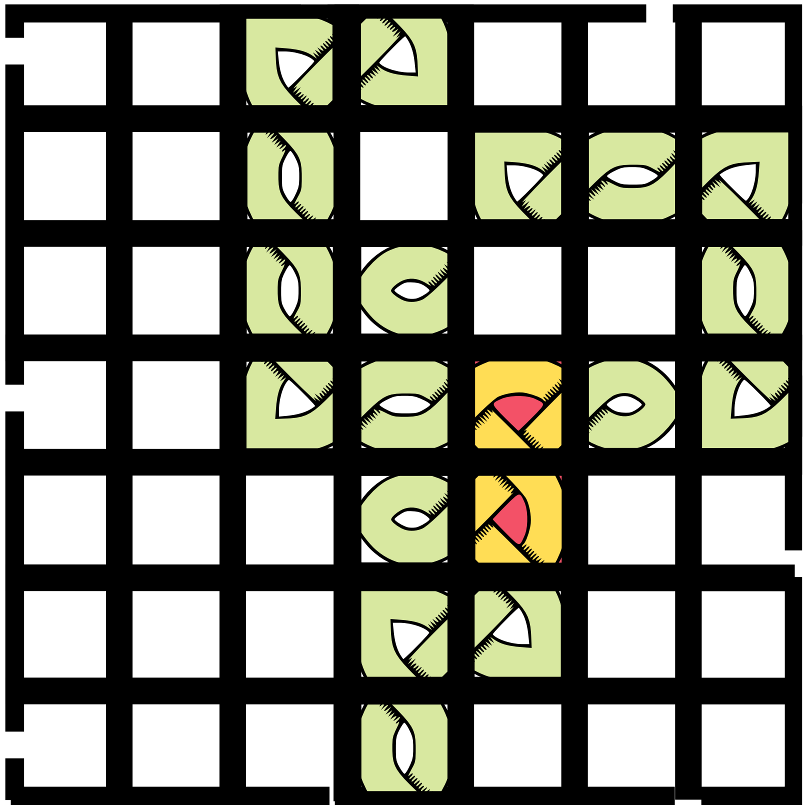


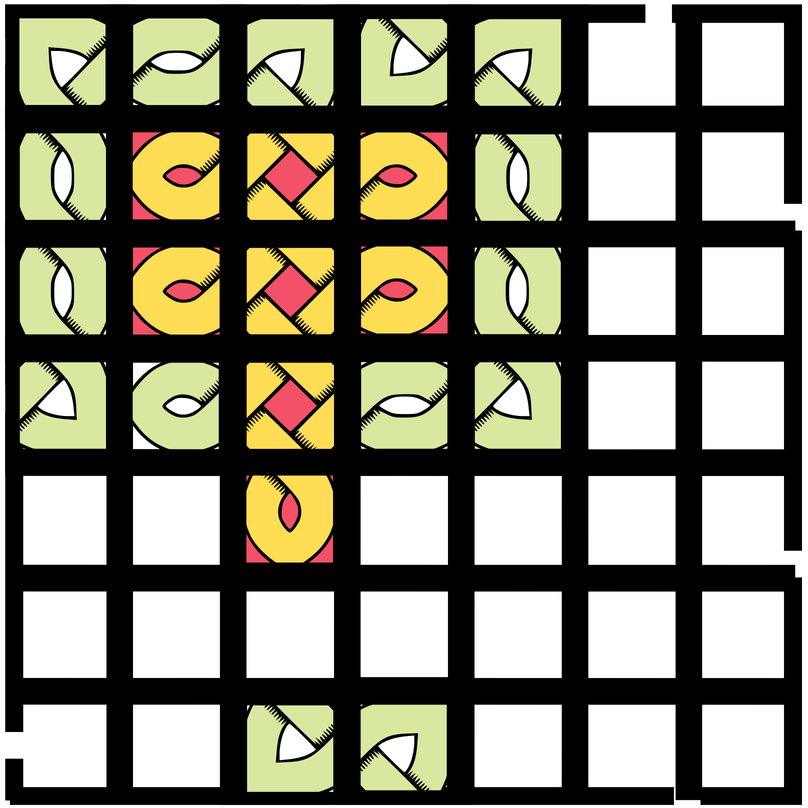
1-Player Celtic!: The “Hard” problems





1-Player Celtic!: The “Hard” problems

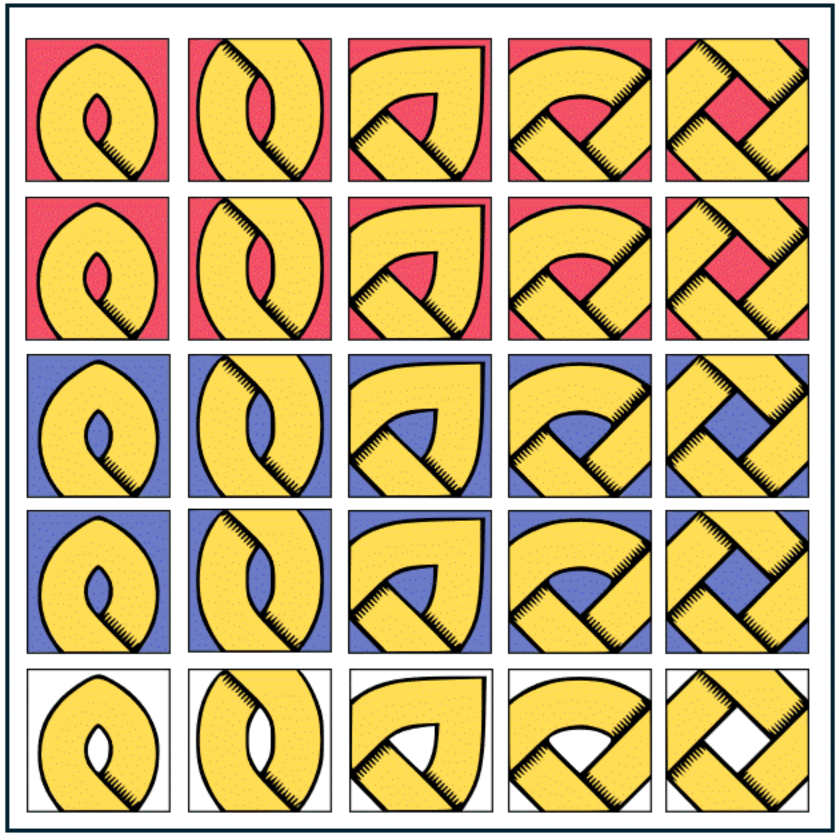


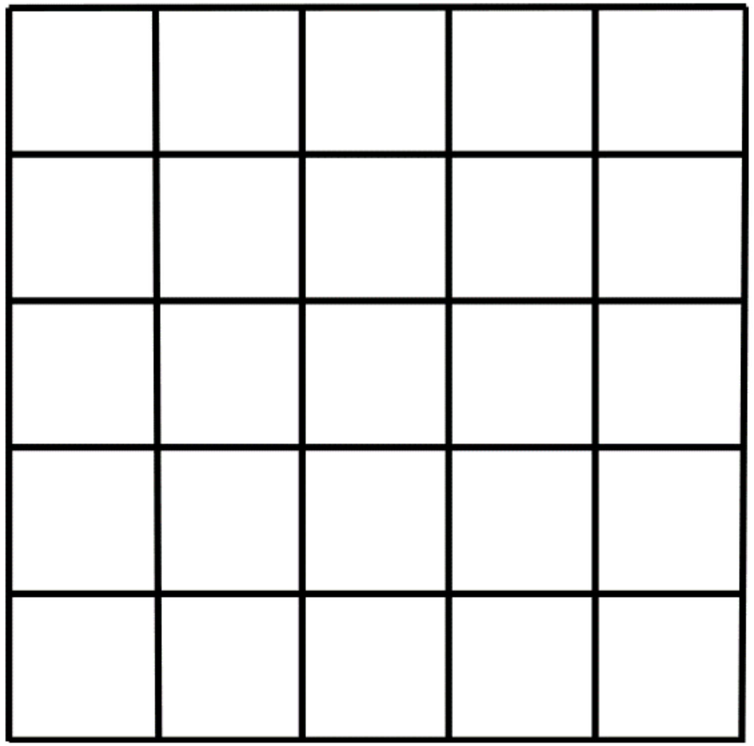


Conclusion

- Some piece combinations missing for Generalized 1-Player Celtic!
- More balanced pieces for Generalized 2-Player Celtic!
- FPT, approximations for maximally sized knots
- Is Celtic! a 1st player win, 2nd player win, or a draw?
- Pattern based complexity of the knots.
- Consider > genus-1 knot assembly.

Tile-based Knot Assembly with Celtic!





Authors: Divya Bajaj, **Ryan Knobel**, Juan Manuel Perez, Rene Reyes,
Ramiro Santos, Tim Wylie