

Molecular Self-Assembly: Models and Algorithms

Ashish Goel

Stanford University

MS&E 319/CS 369X; Research topics in optimization; Stanford University, Spring 2003-04

<http://www.stanford.edu/~ashishg>

•Self-Assembly

- Self-Assembly is the process by which *simple* objects *autonomously* assemble into complexes.
 - Geometry, dynamics, combinatorics are all important
 - Inorganic: Crystals, supramolecules
 - Organic: Proteins, DNA, cells, organisms
- Goals: Understand self-assembly, design self-assembling systems
 - A key problem in nano-technology, molecular robotics, molecular computation

What is Molecular Self-assembly?

- Self-assembly is the spontaneous formation of a complex by small (molecular) components under simple combination rules
 - Geometry, dynamics, combinatorics are all important
 - Inorganic: Crystals, supramolecules
 - Organic: Proteins, DNA
- Goals: **Understand** self-assembly, **design** self-assembling systems
 - A key problem in nano-technology, molecular robotics, molecular computation

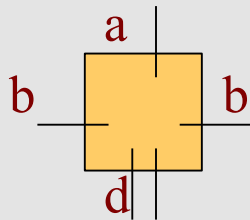
A Matter of Scale

- Question: Why a mathematical study of “molecular” self-assembly specifically?
- Answer: The scale changes everything
 - Consider assembling micro-level (or larger) structures such as robotic swarms. We can attach rudimentary computers, motors, and radios to these structures.
 - Can now implement an intelligent distributed algorithm.
 - In molecular self-assembly, we have nano-scale components. No computers. No radios. No antennas. We **need self-assembly** to make computers, radios, antennas, motors.
 - Local rules such as “attach to another component if it has a complementary DNA strand”
 - Self-assembly at larger scales is interesting, but is more a sub-discipline of distributed algorithms, artificial intelligence etc.

The Tile Model of Self-Assembly

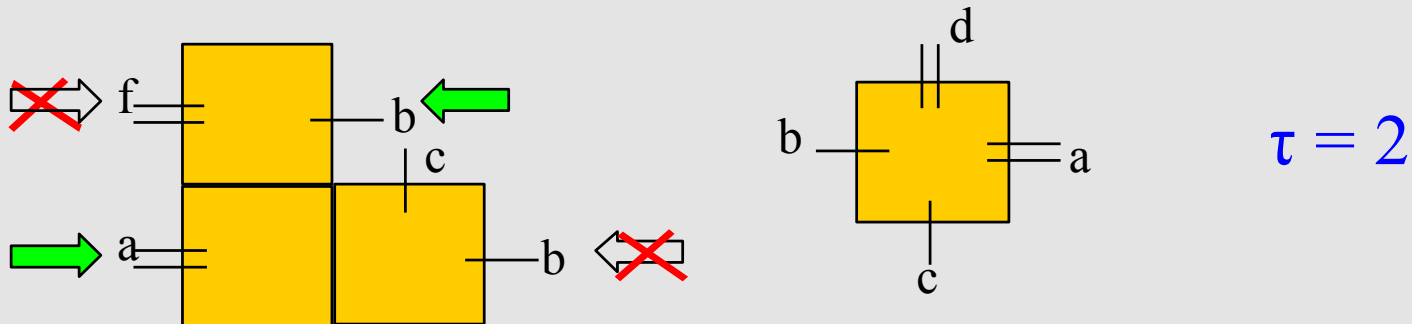
Oriented Tiles with a glue on each side [Wang '61]

- Each glue is labeled by a strength

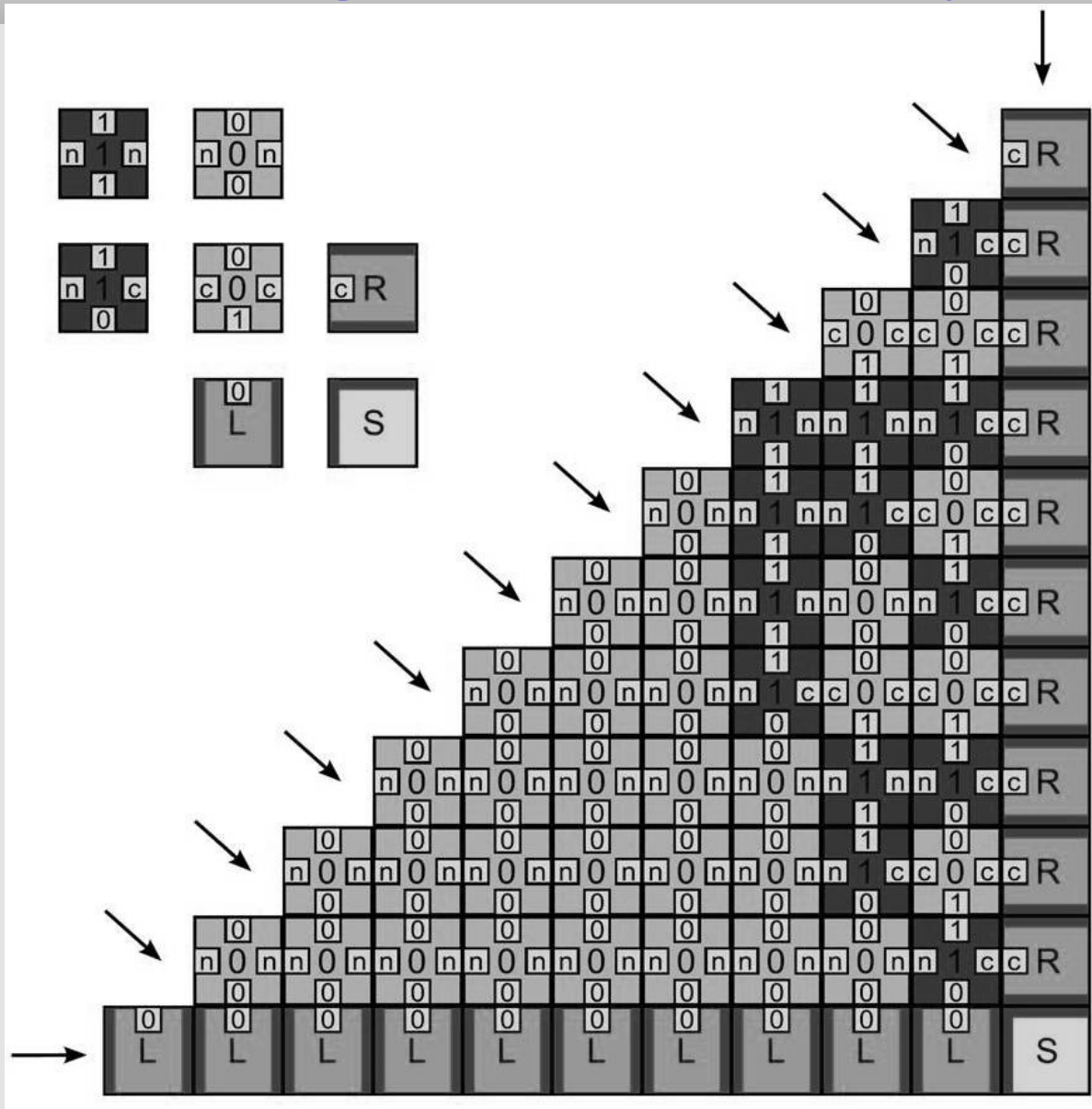


Single bar: strength 1 glue
Double bar: strength 2 glue

- Tiles floating on an infinite grid; Temperature τ
- A tile can add to an existing assembly if
total strength of matching glues $\geq \tau$



Tiling Self-Assembly of Counter



•Counter made by self-assembly

•[Adleman, Cheng, Goel, Huang '01]

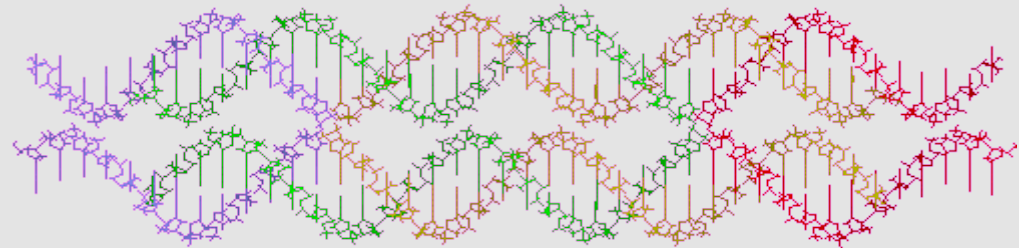
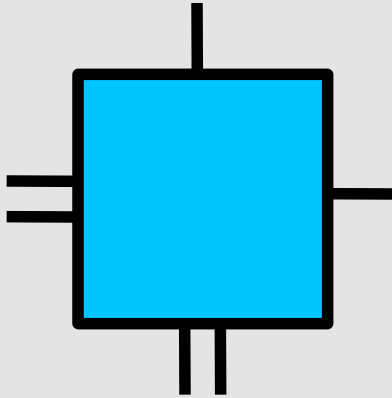
•(Chen, Goel, Cheng)

Synthesized Tile Systems – I

- Styrene molecules attaching to a Silicon substrate
 - Coat Silicon substrate with Hydrogen
 - Remove **one** Hydrogen atom and bombard with Styrene molecules
 - One Styrene molecule attaches, removes another Hydrogen atom, resulting in a chain
 - Suggested use: Self-assembled molecular wiring on electronic circuits

[\[Wolkow et al. '00\]](#)

DNA Tiles



[Winfree]

Glues = sticky ends

Tiles = molecules

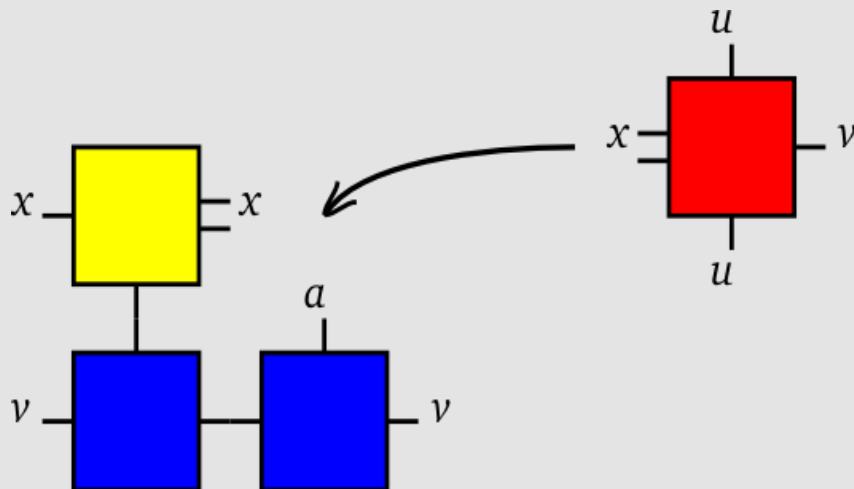
Tile System:

•[Rothemund, Winfree, '2000]

Temperature: A positive integer.

A set of tile types: Each tile is an oriented square with glues on its edges. Each glue has a non-negative strength.

An initial assembly (seed).



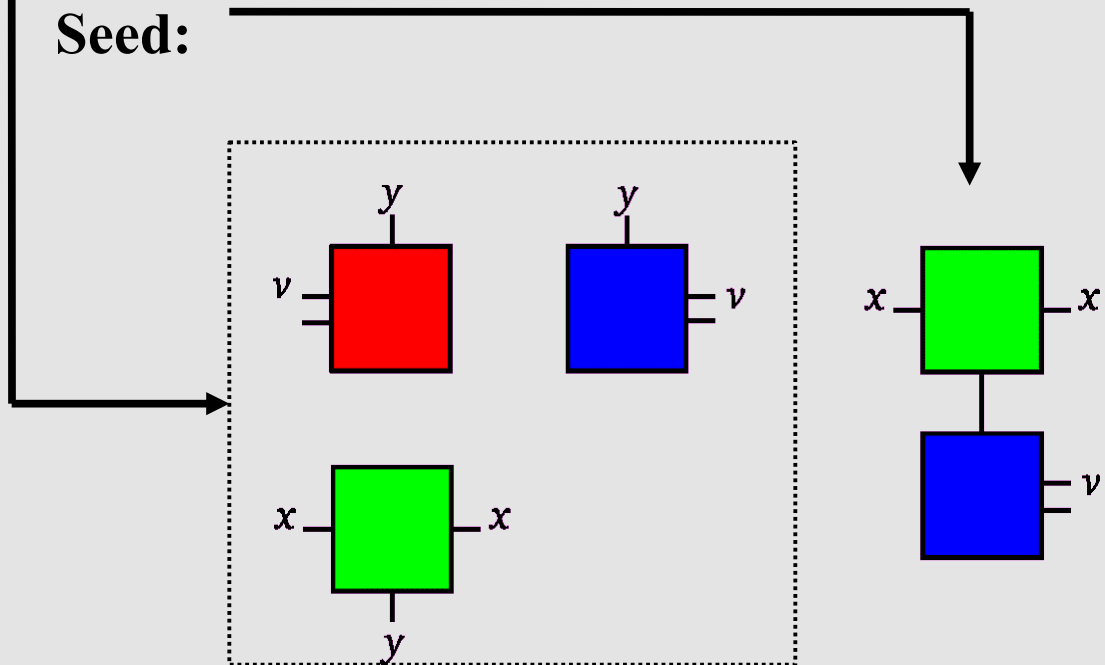
A tile can attach to an assembly iff the combined strength of the “matchings glues” is greater or equal than the temperature.

Example of a tile system

Temperature: 2

Set of tile types:

Seed:

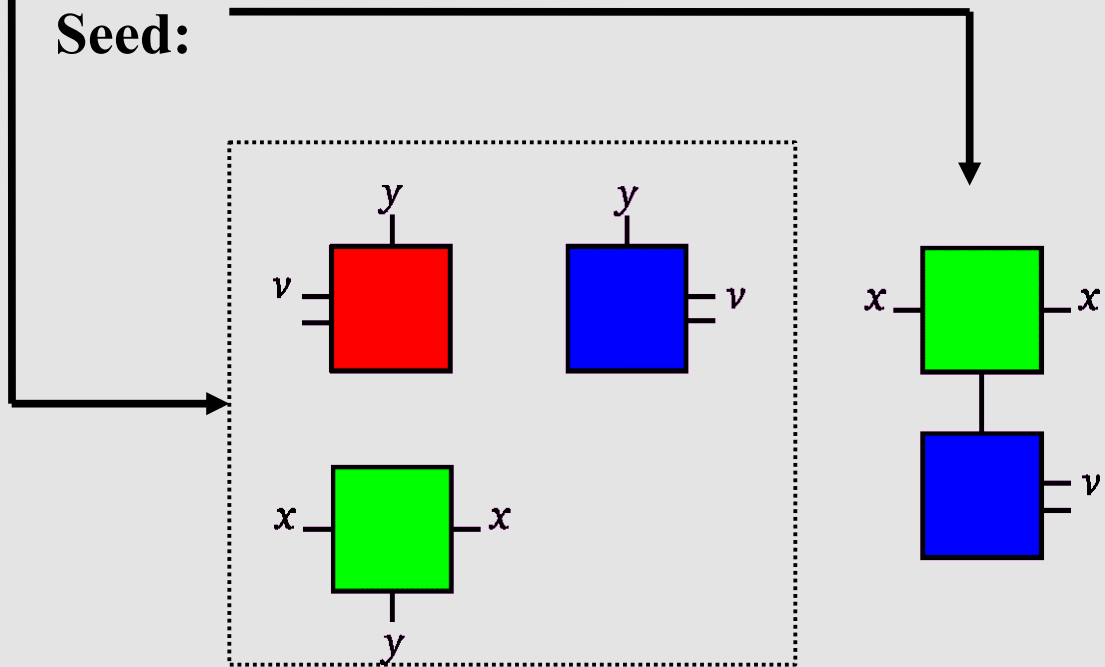


Example of a SA process

Temperature: 2

Set of tile types:

Seed:

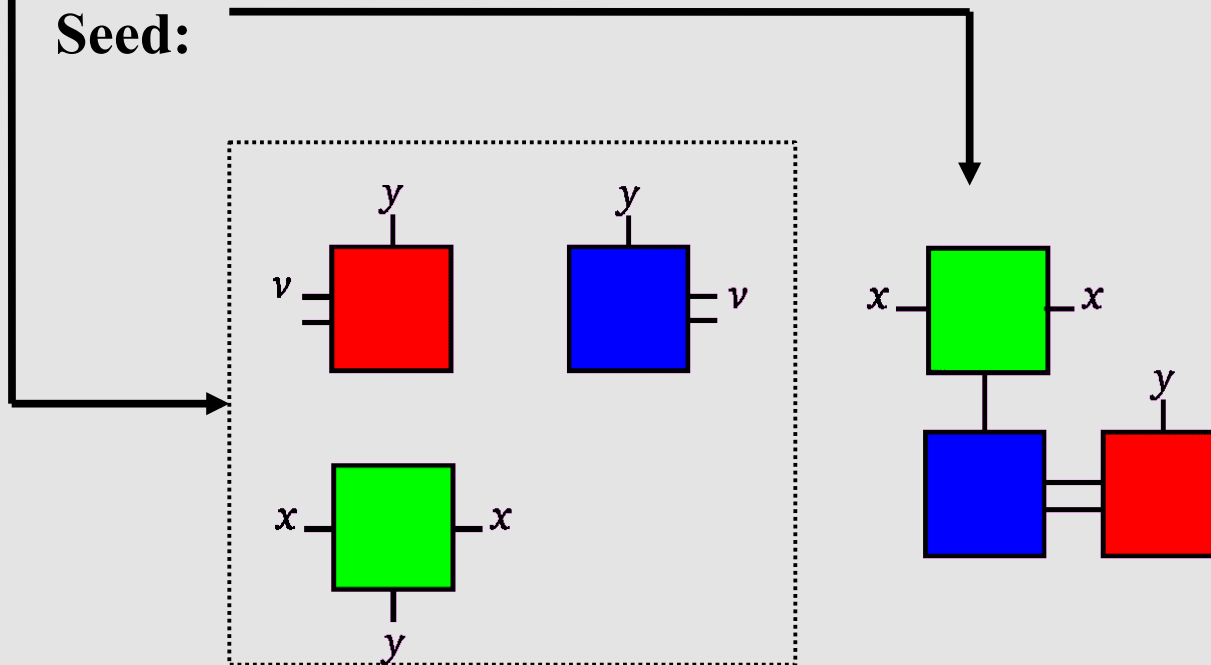


Example of a SA process

Temperature: 2

Set of tile types:

Seed:

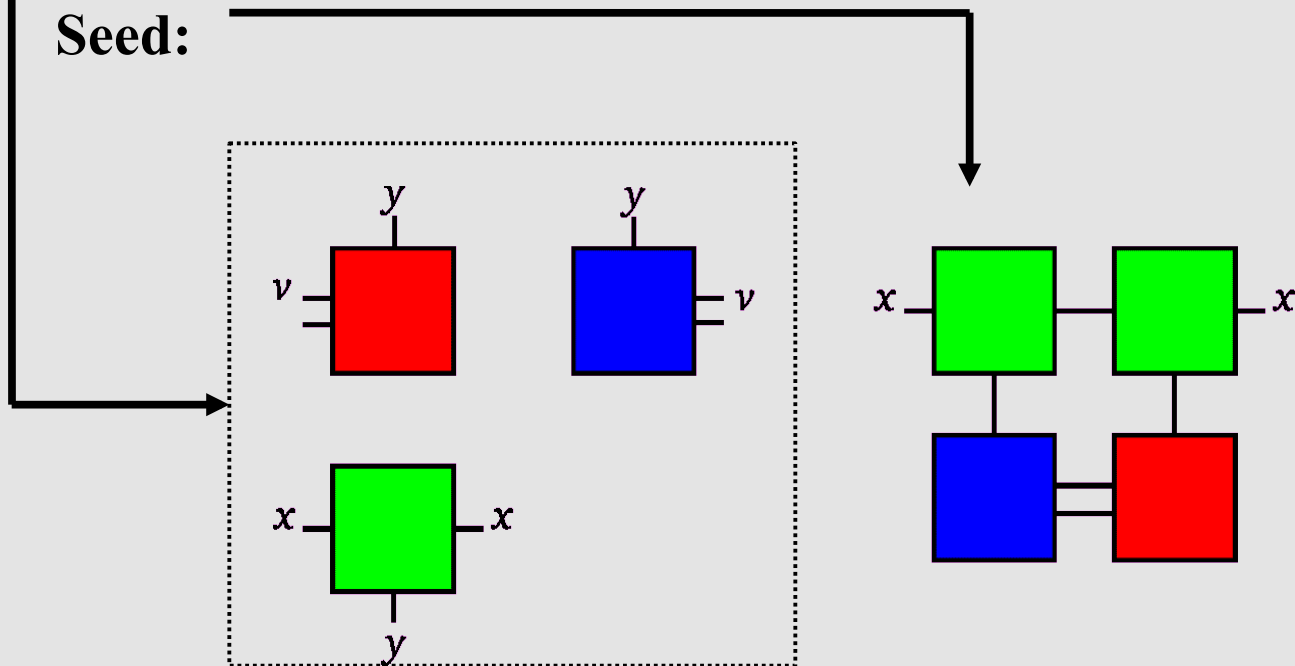


Example of a SA process

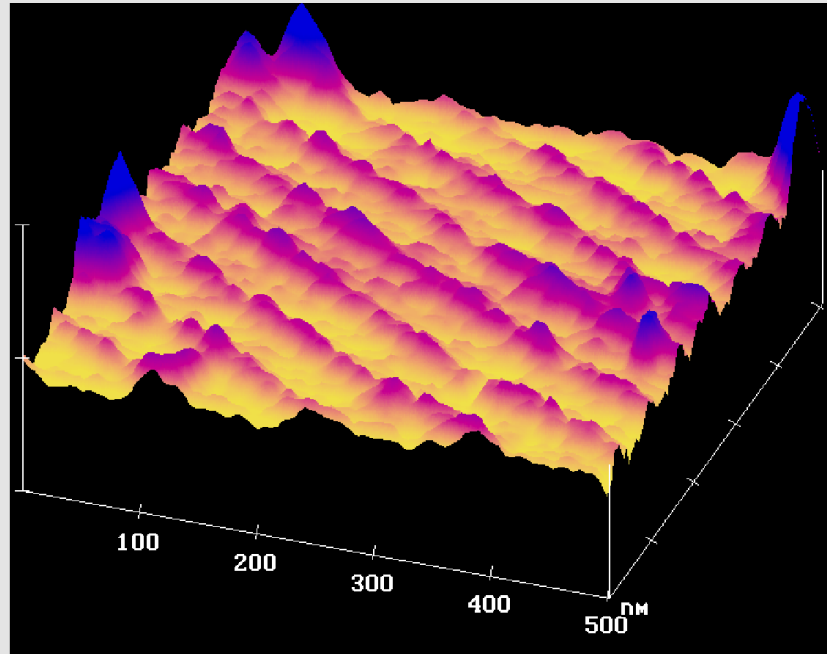
Temperature: 2

Set of tile types:

Seed:



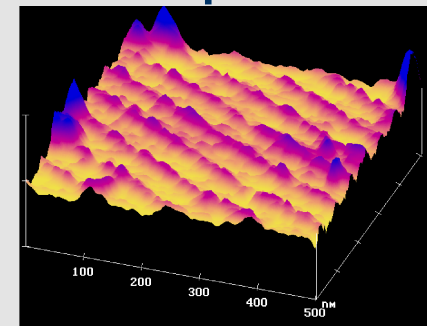
Synthesized Tile Systems - II



A DNA “rug” assembled using DNA “tiles”
The rug is roughly 500 nm wide, and is assembled using
DNA tiles roughly 12nm by 4nm (false colored)
(Due to Erik Winfree, Caltech)

Applications of Self-Assembly

- Self-assembly can be used to create small electrical devices such as FLASH memory. [Black et. Al. ' 03]
- Self-assembly can create nanostructures which “steer” light in the same way computer chips steer electrons. [Percec et. Al. ' 03]
- DNA strands can self-assemble into tiles and those tiles can further self-assemble into larger structures. This has many potential applications. [Winfree ' 96]



DNA “rug” by Winfree

Toy Tile Systems in the Lab



Performing XOR using self-assembling tiles
(Square tiles floating on oil with “glues” on each side)
(Due to Paul Rothmund, USC)

Abstract Tile Systems

- **Tile:** the four glues and their strengths
- **Tile System Γ :**
 - K tiles
 - Infinitely many copies available of each tile
 - Temperature τ
 - A seed tile s
- **Accretion Model:**
 - Assembly starts with a single seed tile, and proceeds by repeated addition of single tiles e.g. **Crystal growth**
 - Are interested primarily in tile systems that assemble into a **unique terminal structure**

[Rothemund and Winfree '00] [Wang '61]

Is Self-Assembly Just Crystallization?

- No. Crystals do not grow into unique terminal structures!
 - A sugar crystal does not grow to precisely 20nm
- Crystals are typically made up of a small number of different types of components
 - Two types of proteins; a single Carbon molecule
- Crystals have regular patterns
 - Computer circuits, which we would like to self-assemble, don't
- Self-assembly = combinatorics + crystallization
 - Can count, make interesting patterns
 - Nature doesn't count too well, so molecular self-assembly is a genuinely new engineering paradigm. Think **engines**. Think **semiconductors**.

A Roadmap for This Course

- Self-assembly as a combinatorial process
 - The computational power of self-assembly
 - Self-assembling interesting shapes and patterns, efficiently
 - Automating the design process?
 - Analysis of design complexity, experiment complexity, and assembly time
- Self-assembly as a chemical reaction
 - Entropy, Equilibria, and Error Rates
 - Reversibility
- Review of experimental progress, interesting applications
- Self-assembly at larger scales
 - Ant colony optimization; assembling autonomous robots

DNA Self-Assembly

- We will tacitly assume that the tiles are made of DNA strands woven together, and that the glues are really free DNA strands
 - DNA is combinatorial, i.e., the functionality of DNA is determined largely by the sequence of ACTG bases. Can ignore geometry to a first order.
 - Trying to “count” using proteins would be hell – proteins have a complicated geometry and it is hard to predict what shape a single protein will take, let alone an entire assembly
 - Proof-of-concept from nature: DNA strands can attach to combinatorially matching sequences
 - DNA tiles have been constructed in the lab, and DNA computation has been demonstrated
 - Can simulate arbitrary tile systems, so we do not lose any theoretical generality, but we get a concrete grounding in the real world
 - The correct size (in the nano range)

1-D, 2-D, and 3-D

- Linear self-assemblies are very different from 2-D self-assemblies
- 2-D and 3-D are essentially identical in terms of the theory
- Will focus on 2-D in this course, and will passingly refer to 1-D

Design Complexity of a Tile System

- Suppose the tile system has k different tiles
 - Assume that we have $\theta(k)$ different glues
 - Assume for simplicity that all glues are represented by DNA strands of the same length, L
 - How many different glues can we have?
 - Each position can be one of the bases A, C, T, or G. An A pairs with T on the complementary strand, and a C pairs with G. So we can think of each position as corresponding to an AC base pair or a CG base pair.
 - Two choices) Binary encoding
 - Number of glues $\cdot 2^L$
 - Total length of all the DNA strands in all the distinct tiles = $\theta(k \log k)$. We will refer to $k \log k$ as the **design complexity**
 - Does not depend on the size of the final structure. This is how much time and expertise and resources you would have to invest into designing the components (like program complexity)

Notes

1. We will sometimes informally use the quantity k to refer to the design complexity.
2. The rules of assembly are easy to code up in a simulator. Thus if a tile system of k tiles can assemble into a certain shape, then there is a computer program of **size** $\theta(k \log k)$ which generates that shape.
3. Design complexity is only interesting for assembling larger and larger shapes of a particular type, eg. squares, lines, spirals. Design complexity of an infinite line, infinite square etc. is just 1.

Three Exercises

1. Suppose we did not assume that the number of glues is $\theta(k)$. Can you still prove that the total length of all the strands in all the tiles is $\theta(k \log k)$?
2. We assumed that our tiles are oriented, i.e., east is always east and north is always north, and the tiles are not allowed to rotate. Show how an oriented tile system can be simulated using a system where tiles can rotate by ± 90 degrees.
3. Now use the fact that the glues are really DNA strands to show how an oriented tile system can be simulated using a system where tiles can rotate by ± 90 as well as 180 degrees.

Some Lower Bounds on Design Complexity

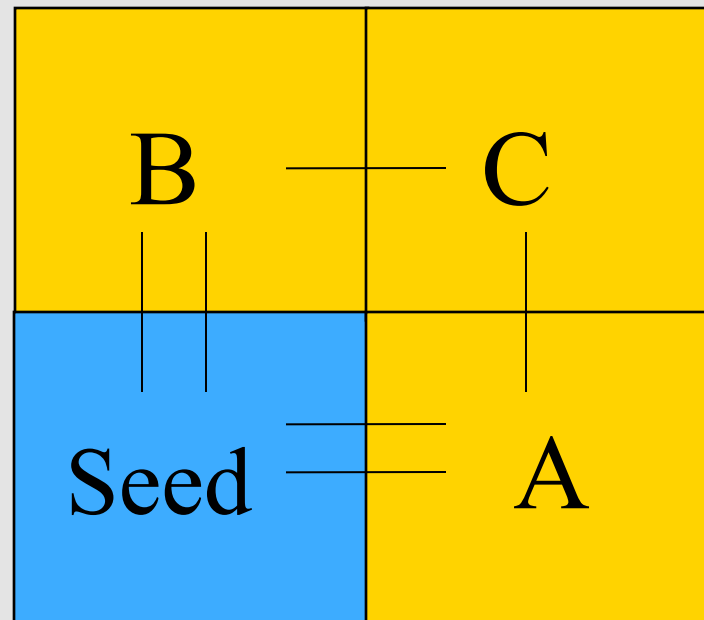
- Consider assembling a line of length n
 - Need at least n different tiles (high design complexity)



- Suppose tiles B and F are the same. Then we can “pump” the line segment BCDE into an infinite line
 - Are we doomed? No. Can assemble thicker rectangles more efficiently
- Consider assembling an $n \times n$ square
 - The average Kolmogorov complexity (the smallest program size to produce a desired output) is $\log n$ bits
 - Thus, $k \log k = \Omega(\log n)$, or $k = \Omega(\log n / \log \log n)$

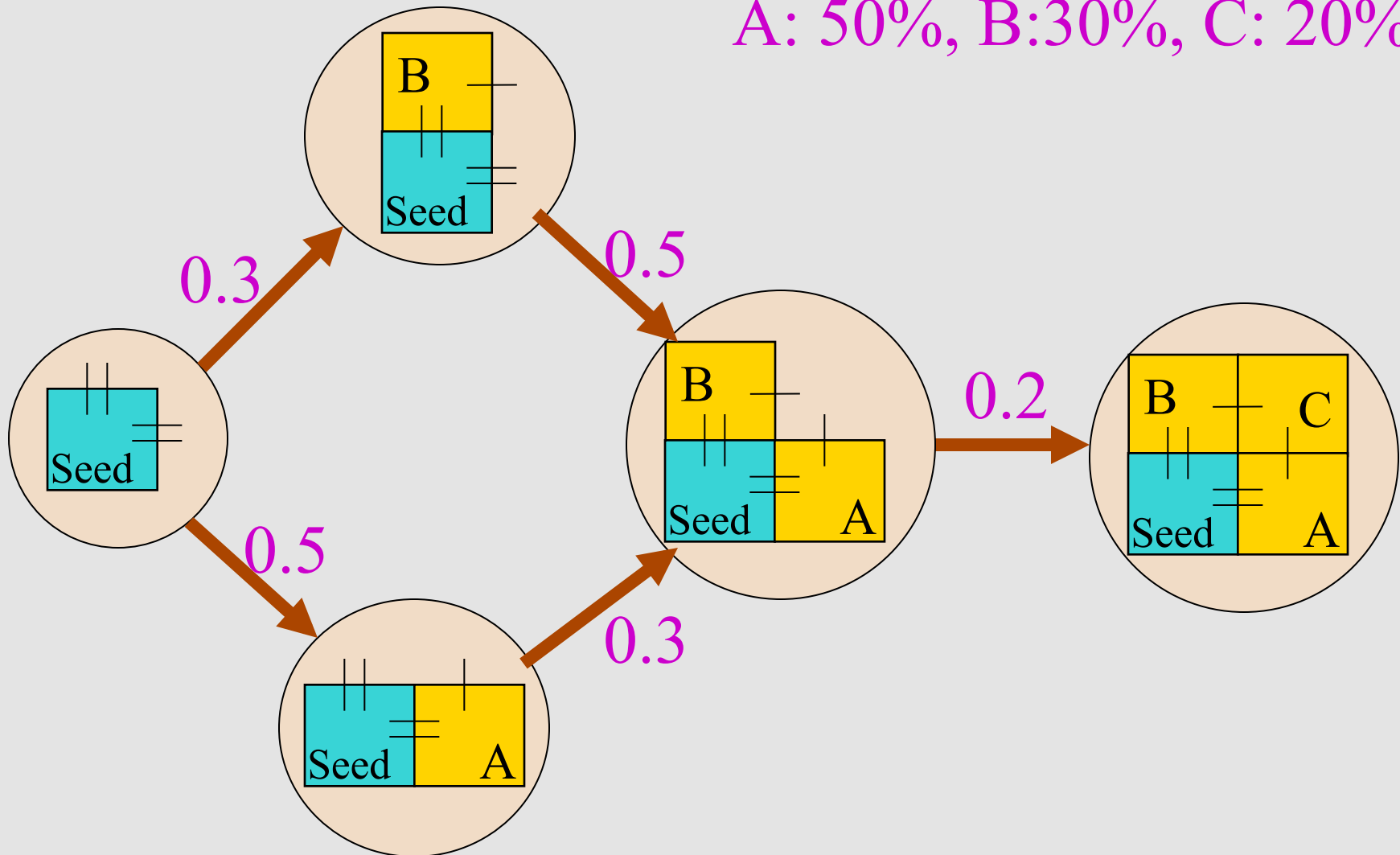
Tile Systems and Running Time

$$\tau = 2$$



Tile Systems and Running Time

A: 50%, B: 30%, C: 20%



Tile Systems and Running Time

- Define a continuous time Markov chain M
 - State space S : set of all structures that a tile system can assemble into
 - Tiles of type T_i have concentration c_i
 - $\sum_i c_i = 1$
 - Unique terminal structure \Rightarrow unique sink in M
 - Seed tile is the unique source state
- **Assembly time**: the hitting time for the sink state from the source state in the Markov Chain

Old Example: Assembling Lines

- Very simple Markov Chain. Average time for the i^{th} tile to attach is $1/c_i$
 - Assembly time = $\sum_i 1/c_i$
- For fastest assembly, all tiles must have the same concentration of $1/(n-1)$
 - Expected assembly time is $\frac{1}{4} n^2$
 - Can assemble thicker rectangles much faster and with much fewer different tiles

A Pretty Picture: Sierpinski Triangles

- Consider the four “XOR” tiles, where the south side is labeled either 0 or 1, and the east side is labeled either 0 or 1, and the west and north sides are labeled $e \text{ XOR } s$ where e and s are the bits on the east and south sides
 - Assume $\tau = 2$, and all glue strengths are 1
- Given the appropriate inputs (i.e. seeds), these tiles can do some very interesting things like
 - Computing the parity
 - Assembling into a very pretty fractal
 - They have already been designed in a lab setting

Theoretical and Algorithmic Issues

- Efficiently assembling basic shapes with precisely controlled size and pattern
 - Constructing $N \times N$ squares with $\Omega(\log n / \log \log n)$ tiles
[Adleman, Cheng, Goel, Huang, '01]
 - Perform universal computation by simulating BCA
[Winfree '99]
- Library of primitives to use in designing nano-scale structures [Adleman, Cheng, Goel, Huang, '01]
- Automate the design process [Adleman, Cheng, Goel, Huang, Kempe, Moisset de Espanes, Rothemund '01]
- **Robustness**