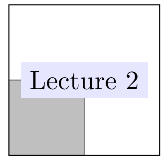


LectureOutline :

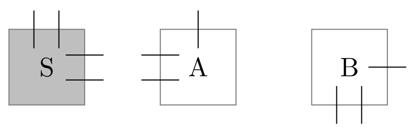
(i) Markov Chains

(ii) Assembly Rates

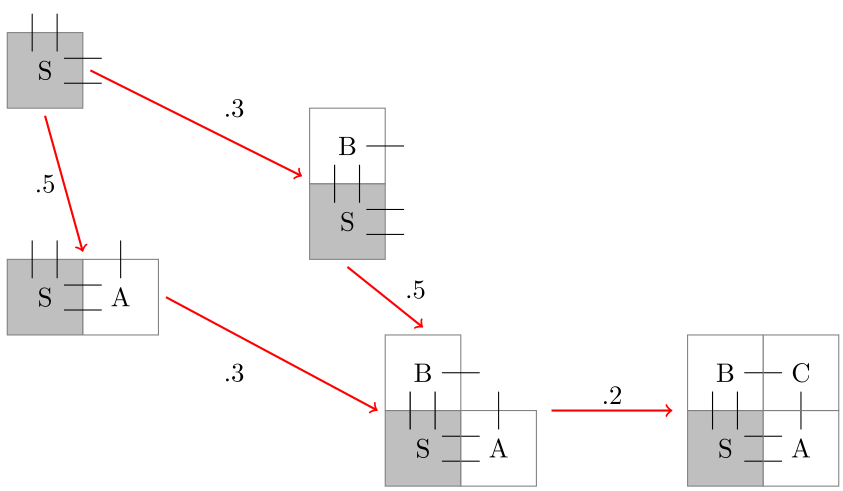
(iii) Building a Square with Counters

I. MarkovChains

Consider the case where $t = 2$ – that is, a tile will disosociate (un-attach) unless it is held to another tile by 2 bonds. Suppose we have the following tiles in solution:



Where S is a Seed, and the concentrations of each of A,B, and C are (A:50%,B:30%,C:20%) by volume. This sets up the following Markov Chain:



Where the values on the edges are probabilities. Note that there are 2 (in this case, equivalent) paths to get to the completed square. Some examples using this diagram:

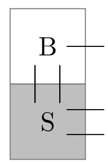
- The rate at which B attaches is $\frac{3}{10}$. The rate at which A attaches is $\frac{1}{2}$.
- The probability that A will attach before B is $\frac{\frac{5}{10}}{\frac{5}{10}+\frac{3}{10}} = \frac{5}{8}$. Similarly, the probability that B will attach before A is $\frac{3}{8}$.

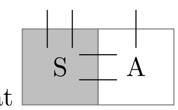
This is an example of a continuous time Markov chain. In general, with state space S and transition matrix P,

$$\text{for i,j} \in S$$
$$P(i,j) = \begin{Bmatrix} \text{Rate at which state} \\ \text{i changes to state j} \end{Bmatrix}$$

This forms an exponential random variable s.t. if T is the time before you make a transition,

$$P_n[T > t] = e^{-tR}$$
$$\text{Expected time is } \frac{1}{R}$$

For example, the expected time it would take to get to 

assuming that 

did not exist is $\frac{1}{R} = \frac{1}{.3}$.

II. Assembly Time

Definition: Assembly time is the average time required to go from seed to final state.

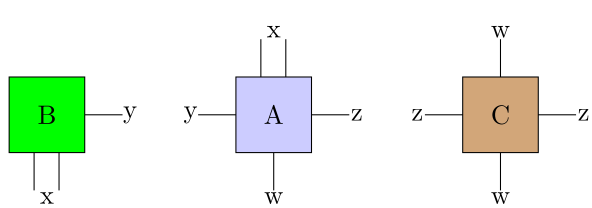
Example: Assmebly of lines

- It requires n distinct tiles to assmeble a line of length n
- If any tiles was not unique, the line would continue infinitely (up to chemistry)
- For fastest assembly, all tiles must have the same concentration $\left(\frac{1}{n}\right)$. In this case expected assembly time is n^2.
- You can assemble thicker rectangles faster and with less tile types!

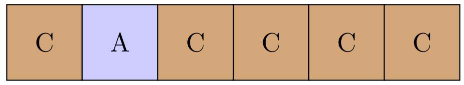
Example: Self Assembling Squares

- Kolmogorov Lower bound ($\Omega_{io}(\frac{\log n}{\log(\log n)}$) on the program size (number of tiles)
- Assembly times has to be $\Omega(n)$ in accretion model

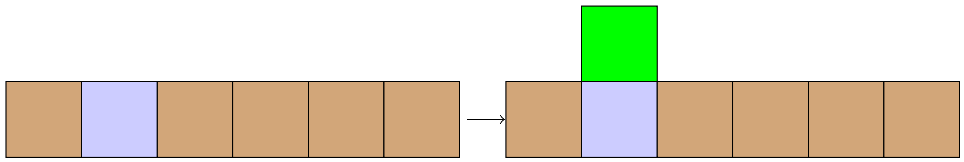
Example: The Rothemund Winfree assembly for constructing Squares
The basic idea is that a $1{\times}n$ rectangle can self-assemble into squares: We have the following rule tiles:



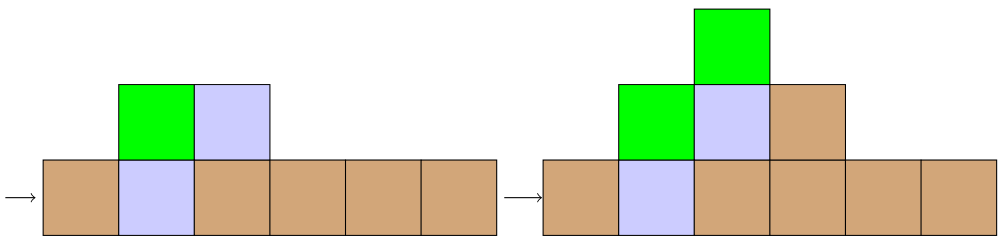
and the flowing seed:

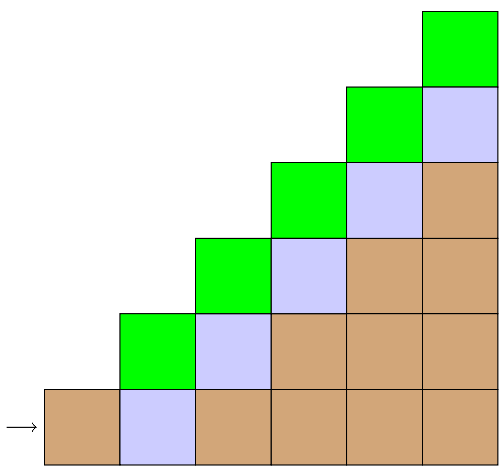


Then with $t = 2$ and a seed of an appropriate $1 \times n$ rectangle (in our example $n = 6$), we can generate a staicase as follows (the method extends directly to the square shape):



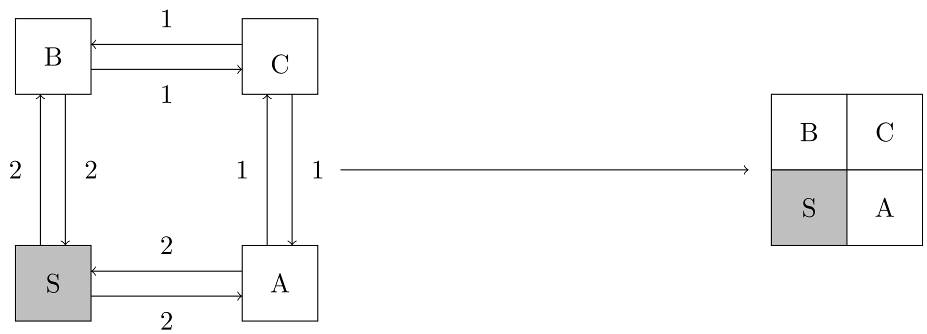
Note that because $t = 2$, the only possible square that could bind is B; so the second state is the only possible path. Once B binds, then A can bind. At this state there are are 2 paths available – C can bind to the right of A, or B can bind above A. This will continue until staircase is of height N. for example:



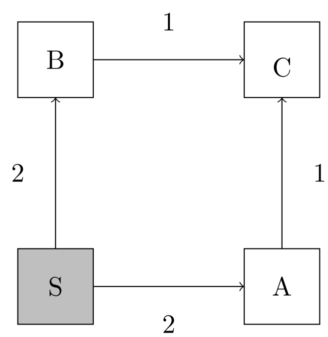


Note that assembling the staircase of size n, given a suitable starting seed of length n, only requires 3 distinct tiles, while a line requires n distinct tiles!

Analysis of assembly: The following chart illustrates the formation of the square in section I.

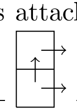


This is equivalent to the directed graph



Where the directional nature of the graph stems from the bond strengths.

Theorem :
Suppose that L is the length of the longest path in an equivalent acyclic subgraph of G. Let c_m be the smallest concetration c_i. The assembly time is $\mathcal{O}(L/c_m)$.

Proof:[1]
Let $T_{i,j}$ be the time to attach at position i, j after it becomes attachable. Using our example, the maximum number of paths possible at a given node is 3 –  – so the # of paths is less than or equal to 3^L, where L is the length of the longest path. For any p in the graph, let $T_p = \sum_{i,j \in p} T_{i,j}$. The assembly times is $\leq \max_p T_p$, which gives us $E[t_p] \leq L/C$, in sharp concentration by Chernoff bounds (i.e., the variance cancels and you get close to the mean).

$$\Rightarrow \text{Expected assembly time is } \mathcal{O}(L/C)$$

III. Making A Square with Counters

We saw in section II how to make a square with an appropriate seed. Here we employ a counter to make a square. The rule tiles are:

The seed is 3 tiles with (n, n, b) glues pointing 'upward'. For example, we will use the 3 tile set $\boxed{\mathbf{0}_Z}\boxed{\mathbf{0}_Z}\boxed{\mathbf{0}^*_E}$.
Note that with $t = 2$, only double bonds can form *above* a row. Observing that only the $\mathbf{0}^*_E$ and $\mathbf{0}^*_M$ tiles

[1]Chen et. all

tiles.png

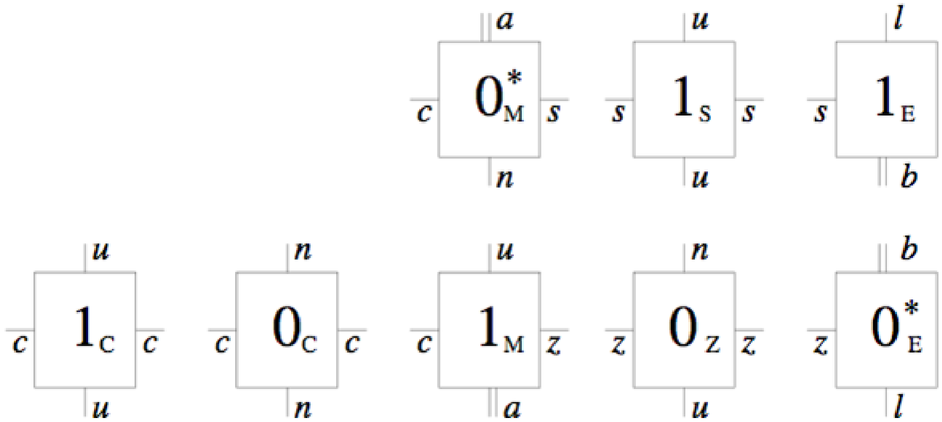


Figure 1: From Cheng, Moisset, Goel

have double bonds, we will focus on these tiles as they assemble. The first few steps are analyzed below (new attachments are shaded):

		$\mathbf{1}_E$
$\mathbf{0}_Z$	$\mathbf{0}_Z$	$\mathbf{0}^*_E$

$\longrightarrow$

$\mathbf{0}_C$	$\mathbf{0}^*_M$	$\mathbf{1}_E$
$\mathbf{0}_Z$	$\mathbf{0}_Z$	$\mathbf{0}^*_E$

$\longrightarrow$

	$\mathbf{1}_{\mathrm{M}}$	
$\mathbf{0}_C$	$\mathbf{0}^*_M$	$\mathbf{1}_E$
$\mathbf{0}_Z$	$\mathbf{0}_Z$	$\mathbf{0}^*_E$

$\longrightarrow \cdots$

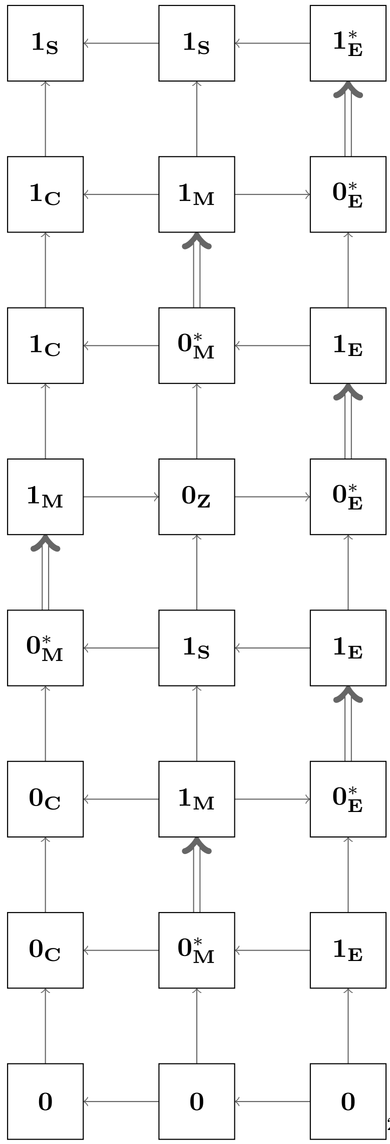
This will not propogate infinitely. To see this, note the tile set

$\mathbf{1}_C$	$\mathbf{1}_M$	$\mathbf{0}^*_E$

binds the following next 3 tiles:

$\mathbf{1}_S$	$\mathbf{1}_S$	$\mathbf{1}_E$
$\mathbf{1}_C$	$\mathbf{1}_M$	$\mathbf{0}^*_E$

This row has no tiles with strength 2 bonds, and thus the counter stops. The full assembly analysis is given by:



[2]

Notes: This counter generates an 8×3 rectangle and stops growing. By choosing a seed with different glues (simulating another position) it is possible to create a smaller counter. For example, if you seed had 'upward' glues as in the 3 tile set $\boxed{\mathbf{0}^*_E}\boxed{\mathbf{0}_Z}\boxed{\mathbf{1}_M}$, you would generate a 4×3 rectangle.

This basic idea can extended as follows:

- Assemble $n \times n$ square in time $\mathcal{O}(n)$ using $\mathcal{O}(\frac{\log n}{\log(\log n)})$ tiles (provably optimum)
- count optimally in binary using the same assembly time and program size
- *general optimization techniques:* A library of subroutines (counting, base-conversion,triangulation of the line)

[2]Cheng, Moisset, Goel

Why did nature not learn to count if this is so efficient? One argument is that it is not evolvable.

– Open problems: general analysis techniques for assembly time in *reversible* models.

WorksCited:

Optimal self-assembly of counters at temperature two. http://www.stanford.edu/%7Eashishg/papers/counter-t2.pdf H. Chen, Q.Cheng, A. Goel, M.-D. Huang, and P. Moisset.