Counting Classes Of Matrices and More Using Experimental Mathematics

Presented by: George Spahn

Thesis Advisor: Doron Zeilberger

Outline

- Intro
- Social Distancing
- Baxter Matrices
- New York Times Puzzles
- Voting Districts
- Do it Yourself Guide!

The Role Of Computers In Mathematics

- In many ways, mathematical progress has gotten harder throughout history.
- Computers give us an edge over our ancestors!
- 4-color theorem, famous early contribution of computers.

The Role Of Computers In This Thesis

- We construct matrices with millions of entries
- We perform matrix multiplication and inverse computations with them
- Will the output of our computations be understandable by a person?
- Yes!

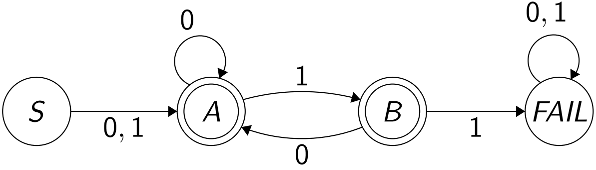
The Role Of Computers In This Thesis

- Often in Experimental Mathematics big computations have simple answers
- A combinatorial interpretation can give rise to conjectures and theorems!
- Computers are not limited to helping us check proofs, they can also provide conjectures and theorems themselves.

Background: Finite State Machine!

- A finite state machine can be used to describe a set of valid "words"
- The machine reads a word symbol by symbol and then when it's done reading it outputs ACCEPT or REJECT according to whether the word was valid

Binary Strings Avoiding 2 Consecutive 1s



Background: Finite State Machine!

- A finite state machine (FSM) is a directed graph.
- The edges (transitions) are labeled with symbols from an alphabet
- The vertices (states) are used to store information as we read a sequence of symbols
- Some of the states are labelled as ACCEPT states
- If a sequence of symbols leads to an ACCEPT state, than the word formed by that sequence of symbols is accepted

Related Questions

- How many ways can people sit in an auditorium so that no two are adjacent but no more people can be added?
- How many ways can a rectangular grid of towns be divided into two connected voting districts?
- How many solutions to a Ring-Ring puzzle are there on an empty grid?
- How many Baxter Matrices exist for specific dimensions?

Counting the number of ways to arrange objects in a rectangular grid!

Framework

We would like to count the number of ways to arrange objects in a rectangular grid.

- Let r be the number of rows and c be the number of columns.
- Key idea: Fix r
- Let $A_r(c)$ be the number or valid arrangements on the $r \times c$ grid.
- Analyze the sequence A_r
- Limitation: r is fixed

Columns As Symbols

- For r rows, a column of our arrangement is a string of length r.
- Now consider the whole column to itself be a single symbol.
- Our State Machine will read a sequence of columns, and then ACCEPT or REJECT.

Social Distancing

- This chapter is joint work with Doron Zeilberger

Proctoring

- No two students may sit adjacent horizontally or vertically
- Perhaps the students sit randomly without violating the rules
- Will we run out of space?
- What density can we expect?

x		x		
				x
	x		x	
x		x		x

Definitions

- Input: The dimensions of the grid of seats $r \times c$
- Input: A set of forbidden patterns, S
- A seating assignment can be represented as an $r \times c$ matrix of 0s and 1s (1s represent occupied seats)
- An assignment is said to be maximal if it satisfies 2 properties:
  - None of the forbidden patterns are present.
  - Changing any 0 to a 1 causes a forbidden pattern to be present.

Example

Not Maximal

1	0	0	0	0
0	0	1	0	1
0	1	0	1	0
1	0	1	0	1

8 1s

Maximal

1	0	0	1	0
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0

7 1s

Questions

- Given r,c, and S:
- How many maximal configurations are there?
- If I were to select a maximal configuration uniformly at random, what is the expected density?

Finite State Machine!

- For us, the symbols are possible columns
- 2^r symbols in our alphabet
- A r by c maximal assignment will be an accepted word of length c

When to REJECT

There are two ways that a seating assignment can fail to be maximal:

- There are two adjacent 1s
- There is a 0 with no adjacent 1

A 0 with an adjacent 1 is said to be a satisfied 0. If we encounter an unsatisfied 0 we should REJECT!

Detecting unsatisfied 0s

- We don't have enough information to determine whether a 0 in the current column is satisfied.
- Instead check that each 0 in the previous column is satisfied
- Need to store the previous TWO columns in the state.
- Total of 2^{2r} states, one for each possible contents of the previous two columns

Which states are ACCEPT states?

- If we reach the end of input, should we ACCEPT or REJECT?
- Still need to check 0s in most recent column!
- If all those 0s are satisfied, then ACCEPT

Counting Paths

- Each maximal assignment corresponds to a path from the START state to the ACCEPT state.
- The number of maximal assignments with c columns is thus the number of paths from START to ACCEPT with length c!

Transition Matrix

- We can count paths using matrices!
- Let M be the adjacency matrix of the state machine
  - Each state becomes a row and column of the matrix
  - A valid transition from state i to j is represented by a 1 in the $[i, j]$ entry of M
  - All other entries are 0
- $M^2[i, j]$ now counts the number of paths from i to j of length 2
- $M^c[i, j]$ now counts the number of paths from i to j of length c

$r = 2$ sequence

- We now can compute the sequence giving the number of maximal assignments
- For $r = 2$: 2, 2, 4, 6, 10, 16, 26, 42, 68, 110, 178, 288, 466, 754, ...
- Twice the Fibonacci sequence!

$r = 3, 4$

- Maximal assignments with 3 rows:
- 2, 4, 10, 18, 38, 78, 156, 320, 654, . . .
- Maximal assignments with 4 rows:
- 3, 6, 18, 42, 108, 274, 692, 1754, 4442, . . .
- A157049, A157050 is the OEIS

Generating Function

- Using this method we can get generating functions for these sequences without too much extra work. The sequence:

$$f(n) = M^n[1,2]$$

has the generating function:

$$F(x) = \sum f(n)x^n$$

A system of Equations

1. Let $F_i(x)$ be the generating function for the number of ways to reach state i from START in n steps.
2. Then
$$F_i(x) = \sum_j F_j(x) \cdot x$$
where the sum is taken over preceding nodes
3. Big system of equations is great for Maple!

Shortcut

- Ignoring matrices for a second...

$$
\begin{aligned}
F(x) &= \sum M^n x^n && (1)\\
&= \sum (Mx)^n && (2)\\
&= \frac{1}{1 - Mx} && (3)\\
&\approx (I - Mx)^{-1} && (4)
\end{aligned}
$$

- This matrix, N, contains all of our desired generating functions!
- $N[1,2]$ gives the generating function for the number of paths from START to ACCEPT.

Results

- Here is the generating function for $r = 3$:

$$-\frac{2x^6 - x^5 + x^4 - x^3 - x^2 - x - 1}{x^5 + x^4 - 3x^3 - x^2 - x + 1}$$

Back to Density

- What if want to compute the average density over all these maximal assignments?
- Modify the transition matrix M.
- Previously it had entries either 1 or 0 indicating edges in the graph.
- Now replace the ones with powers of z.
- z^t will indicate that the corresponding transition added t 1s to the assignment.

Density Polynomials

- Previously $M^n[1,2]$ counted the number of maximal assignments with n columns.
- Now it is a polynomial in z.
- The coefficient of z^k gives the number of maximal assignments with n columns and k total 1s.

Example

For 3x3 assignments we get the polynomial:

$$g(z) = z^5 + z^4 + 8z^3$$

	1	
1		1
	1	

1		1
	1	
1		1

		1
1		
		1

1		
		1
	1	

The average density is 0.37. One way to compute this is:

$$\frac{g'(1)}{9g(1)}$$

Bivariate Generating Functions

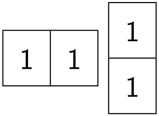
- We can also include z in the generating function!
- The coefficient of $x^j z^k$ now gives the number of maximal assignments with j columns and k total 1s.
- For 3 rows: $-\frac{2x^6z^5 - x^5z^4 + x^4z^3 + x^3z^4 - 2x^3z^3 - x^2z^2 - xz^2 - 1}{x^5z^4 + 2x^4z^4 - x^4z^3 + x^3z^4 - 4x^3z^3 - x^2z^3 - xz + 1}$
- Maple can extract coefficient polynomials using Taylor series!

Limiting Density

- We can look at the roots of the denominator of the generating function to get asymptotics.
- We can compute the limiting average density over all maximal assignments as the number of columns goes to infinity.
- For $r = 3$ we compute $d = 0.352...$
- For $r = 4$ we compute $d = 0.347...$
- For $r = 5$ we compute $d = 0.342...$
- Only slightly smaller than the 3 by 3 case, 0.37...

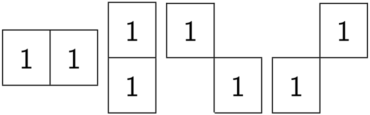
Generalizing S

- Recall that S is the set of violations.
- So far we have looked at the specific case where S has two elements: horizontal and vertical adjacencies
- We represent these violations as polyominoes



Non-Attacking Kings

If the seats are not allowed to be adjacent diagonally, we get the famous non-attacking kings problem.



Housing Developments

The paper that inspired this research was interested in avoiding the T-piece. "Packing density of combinatorial settlement planning models"

1	1	1
	1	

The idea is that you don't want any houses to be totally blocked from the sun (and there is no sun from the North)

Checking Arbitrary Patterns

- It is now not sufficient to only keep track of the previous two columns.
- Let W be the largest width of any polyomino.
- In general we will have to store the previous $2W - 2$ columns.
- This gives a total of $2^{r(2W-2)}$ states.

Down Facing T

1	1	1
	1	

- Maximally avoiding the T with $r = 3$ gives the following sequence:
- 1, 1, 10, 19, 41, 105, 269, 651, 1560, . . .
- Sadly need to make the code faster to compute the generating function, inverting the 189 x 189 matrix was taking too long.

Baxter Matrices

Not only do we count them, we also resolve a conjecture of Donald Knuth!

Background

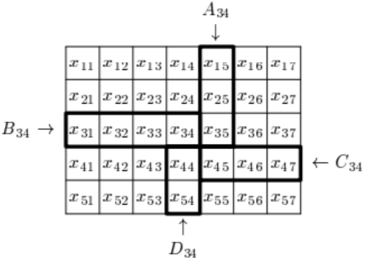
- Don Knuth
- *Baxter Matrices* – an "Unpublication"
- September 5, 2021
- Extension of Baxter Permutations

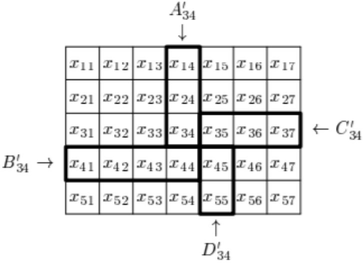
What is a Baxter Matrix?

A $m \times n$ matrix of 0's and 1's satisfying 4 conditions:

1. Each row contains a 1
2. Each column contains a 1
3. Each clockwise pinwheel contains a segment of all 0's
4. Each counterclockwise pinwheel contains a segment of all 0's

Pinwheels





- Each pinwheel requires a segment of zeroes
- Center can be on any vertex in the interior of the matrix
- $(m-1)*(n-1)$ possible centers

An Example?

$$
\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}
\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}
$$

How many 1's can we fit into a Baxter Matrix?

Putting three 1's in a corner:

$$\begin{pmatrix} 1 & 1 & \cdot \\ 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & \cdot \\ 1 & 0 & 0 \\ \cdot & 0 & \cdot \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

It turns out that the maximum number of 1's in a 3×3 matrix is 5.

Conjecture (Knuth)

Conjecture:

The maximum number of 1's in a $m \times n$ Baxter matrix is:

$$m + n - 1$$

- Knuth verified this conjecture for all Baxter Matrices up to size 7×7 by enumerating them

FSM for $2 \times n$ Baxter Matrices

- Matrix as a sequence of columns
- Any sequence of columns can be "accepted" or "rejected"
- possible columns: $\begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Rejecting Early

Suppose we read the columns $\begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

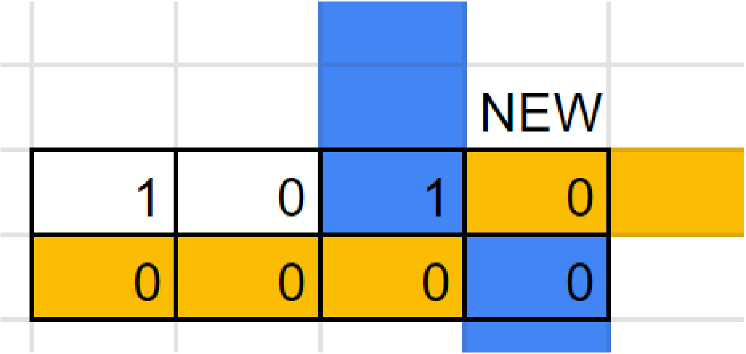
This means that our input matrix starts out like $\begin{pmatrix} 0 & 1 & 1 & \dots \\ 0 & 1 & 1 & \end{pmatrix}$

- Not possible for the pinwheels to ever be satisfied!

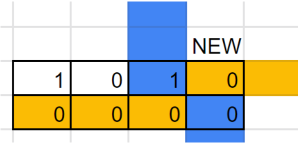
Keeping Track of Data

As we read a column, we check two new pinwheels

- They have center to the immediate left of our column



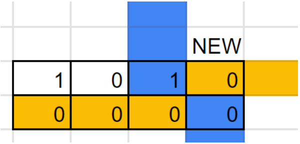
Keeping Track of Data



- In order to check the pinwheel, we need to know the contents of the previous columns
- Could be arbitrarily many columns!
- Finite State machine can only keep track of finitely many things

To check the leftward orange segment, only need to know whether that row has been all 0's so far

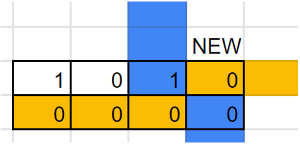
Keeping Track of Data



- Keep track of the exact contents of the previous column
- Keep track of whether each row has only 0's so far
- What about the rightward segment?

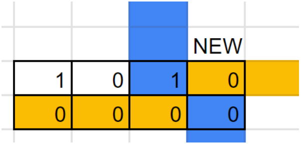
Cannot see the future

Seeing the future



- If any of the other segments are 0's, then we don't care
- Else, we have a row that must be permanently 0's in the future

The 4 rowstates

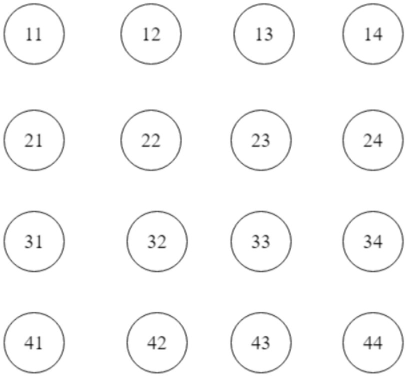


1. This row had a 1 in the previous column
2. This row has only contained 0's so far
3. This row must contain only 0's in the future
4. This row had a 0 in the previous column, but neither 2 or 3 applies

If we know which rowstate each row is in, we can check whether our new pinwheel is satisfied!

Constructing the FSM for 2 rows

- $4 \times 4 = 16$ states, to allow for all combinations of rowstates between the two rows.

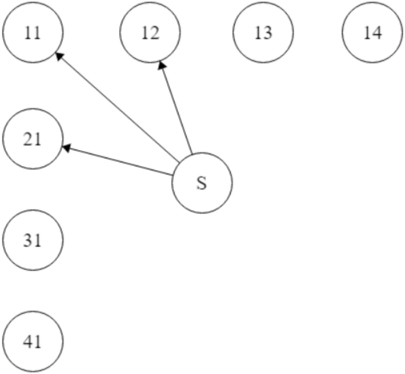


Unused States

- Remove the states which do not have a row in rowstate 1

Start State

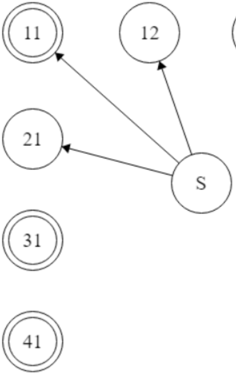
After a single column, each row must be in either rowstate 1 or rowstate 2



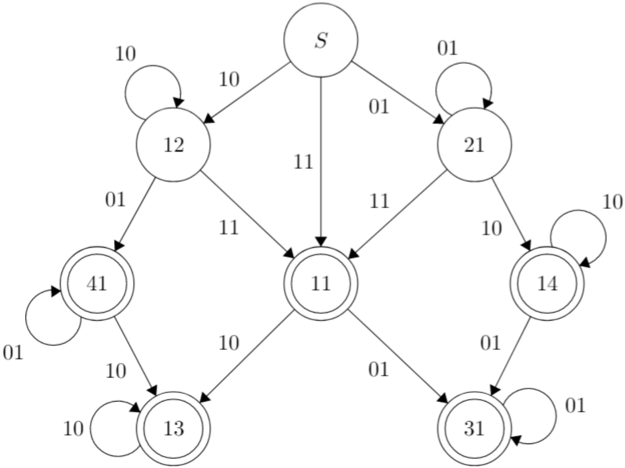
Accept States

In a Baxter Matrix each row must contain a 1.

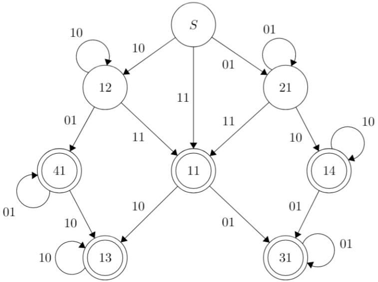
To accept, we must require that every row has left rowstate 2.



Drawing in the Arrows



Correspondence

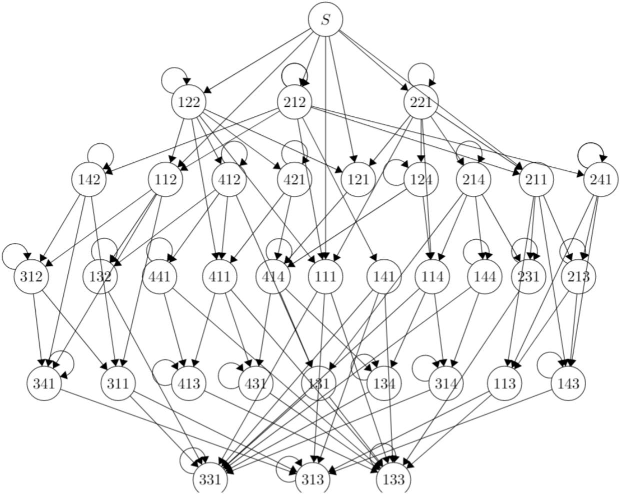


We have a 1-1 correspondence between Baxter Matrices with 2 rows and paths from the start state to an accept state

More rows!

- We can do a similar process for 3 rows, 4 rows, etc.
- Let's fix r as the number of rows
- FSM will have 2^r symbols (one for each column) and $4^r - 3^r$ used states
- Denote the FSM for r rows as A_r

A_3 drawn with downward arrows



Depth of a state

Definition:

The **depth** *of a state is number of 1's plus the number of 4's plus twice the number of 3's that can be found in the rowstates of the rows.*

Min Depth $= 0$ (Start). Max Depth $= 2r - 1$

Lemma:

Any transition in A_r must either be a self arrow or increase depth.

Counting Baxter Matrices

- How many Baxter Matrices of size $r \times k$?
- We don't need a transfer matrix in this case!
- If we ignore self-arrows, the lemma forces there to be only finitely many paths in A_r.
- A self-arrow corresponds to a repeated column in the Baxter Matrix
- Let's say a Baxter Matrix with no repeated columns is "interesting"

Counting Baxter Matrices

- Only finitely many interesting Baxter Matrices with r rows.
- Each non-interesting Baxter Matrix can classified according to the interesting matrix that remains after removing the self-arrows.
- To count the total number of $r \times k$ Baxter Matrices, just need to count the number of non-interesting Baxter Matrices with k columns that correspond to each interesting Baxter Matrix with r rows.

Counting Baxter Matrices with r rows

1. Enumerate the finitely many interesting Baxter Matrices with r rows.
2. Receive a polynomial in k of degree at most $2r - 2$ from each.
3. For any specific k, plug it in to each polynomial. If the output would be negative, set it to 0.
4. Add up the results!

Counting Baxter Matrices with r rows

- For $k \geq r$, the polynomials won't be negative, so we can add up the polynomials before plugging in, to get a single polynomial of degree $2r - 2$

Theorem:

For a fixed number of rows, r, the number of Baxter matrices with r rows and k columns eventually satisfies a polynomial in k of degree $2r - 2$.

Maple Code

I have maple code that does the above process to compute the polynomial for any r.

rows	formula	works for
2	$k^2 + 3k - 4$	$k \geq 2$
3	$(1/3)k^4 + 3k^3 - (16/3)k^2 + 2k + 3$	$k \geq 3$
4	$(1/18)k^6 + (21/20)k^5 - (5/18)k^4 - \ldots$	$k \geq 4$
5	$(23/4032)k^8 + (937/5040)k^7 + \ldots$	$k \geq 5$
6	$(361/907200)k^{10} + (403/20160)k^9 + \ldots$	$k \geq 6$

Returning to Knuth's conjecture

Conjecture:

The maximum number of 1's in a $m \times n$ *Baxter matrix is:*

$$m + n - 1$$

Recall from the definition that each column of a Baxter Matrix must contain a 1.

- Let's say each column with more than one 1 contains extra 1's.

Returning to Knuth's conjecture

Rephrasing the conjecture,

Conjecture:

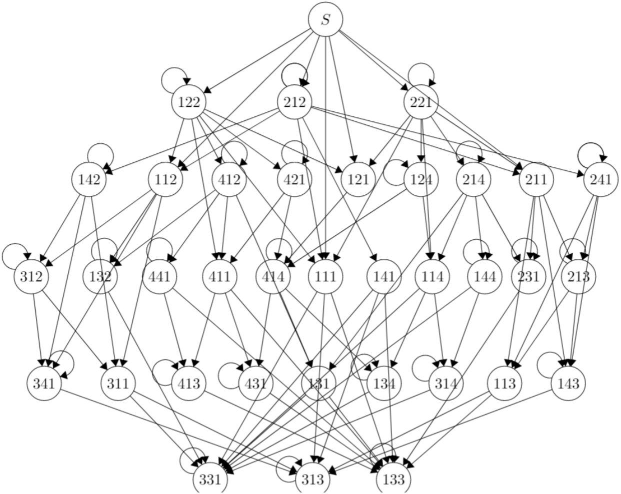
The number of extra 1's in a Baxter Matrix with r rows is less than r.

A discovery

Lemma:

The total number of extra 1's that appear in two consecutive columns is at most the change in depth of the corresponding state transition in A_r.

A discovery


S
122
212
221
142
112
412
421
121
124
214
211
241
312
132
441
411
414
111
141
114
144
231
213
341
311
413
431
131
134
314
113
143
331
313
133

Using the Discovery

Let M be a $r \times k$ Baxter Matrix, p be its corresponding path in A_r, and T be the set of transitions in p.

$$(\#\text{ of extra 1's in } M) = \frac{1}{2}\left(\sum_{\tau \in T}(\#\text{ of extra 1's in the columns associated with } \tau)\right)$$

- This assumes the first and last states do not have extra 1's.

Using the Discovery

$(\#\text{ of extra 1's in } M) = \frac{1}{2}\left(\sum_{\tau \in T}(\#\text{ of extra 1's in the columns associated with } \tau)\right)$

$$\leq \frac{1}{2}\left(\sum_{\tau \in T}(\text{depth increase of } \tau)\right)$$
$$\leq \frac{1}{2}(2r - 1)$$
$$< r$$

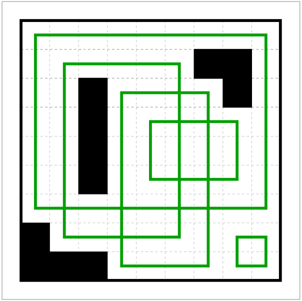
Done!

Ring-Ring

In which we count the number of solutions to puzzles!

Solved Ring-Ring Puzzle

Ring-Ring is a type of puzzle from the New York Times magazine.



The solution is in green.

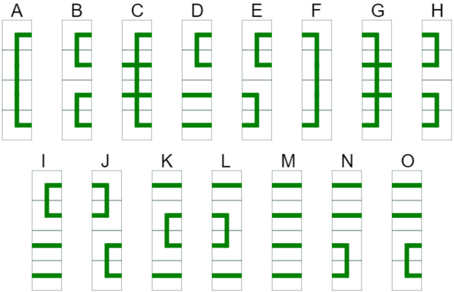
Rules

1. Draw a set of rectangles on the grid.
2. No cell can remain empty.
3. No rectangle may share a side or corner.

Counting Solutions

- A good puzzle has only 1 solution.
- What if we remove the clues?
- How many solutions are there starting from an empty grid?
- We fix the number of rows and apply the same methodology.

4 Row Case

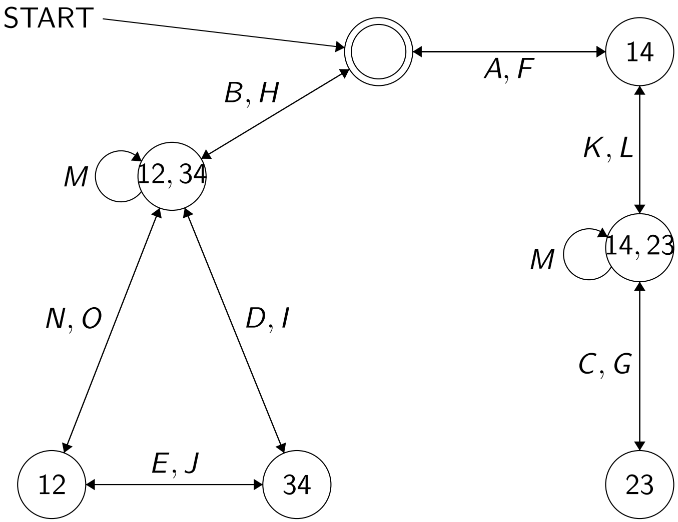


The possible columns that could appear. Our state machine will use 15 symbols.

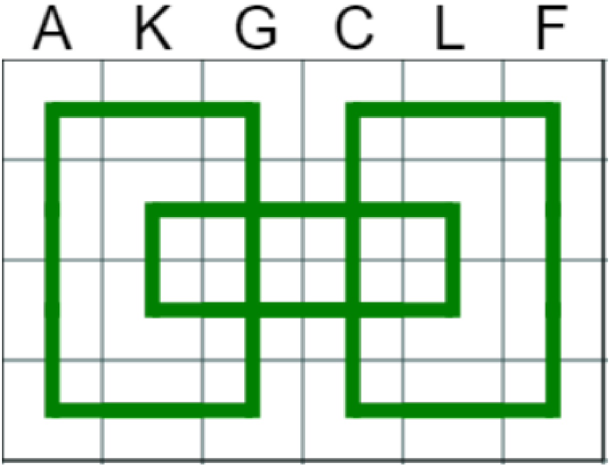
What are the states?

- All we need to keep track of is the locations of each rectangle that is in progress.
- A set of disjoint subsets of the rows, where each subset is of size 2.
- A present subset indicates a rectangle that is currently using those 2 rows.
- The empty set is the start state and the only accept state!

State Machine for $r = 4$



AKGCLF



The solution corresponding the sequence of symbols AKGCLF

Results for the 4 row case:

- The first few terms are 0, 2, 1, 8, 12, 45, 98, 292, ...
- The generating function is:

$$\frac{(1+x)(1-2x)(1-2x-x^2)}{(1-3x-3x^2+10x^3+3x^4-5x^5-x^6)}$$

Other values of r

- We were able to compute the generating functions up to $r = 8$.
- For $r = 2$, we get the beloved fibonacci sequence!

Gerrymander Sequence

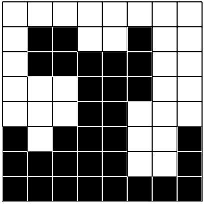
This chapter is joint work with Manuel Kauers and Christoph Koutschan.

Voting Districts

- How many ways are there to divide the $r \times c$ chessboard into two connected regions of equal area?
- This question was motivated by the number of ways to gerrymander voting districts, illustrating the extent of the problem.

Example Division

Below is a valid division of the 8×8 chessboard.



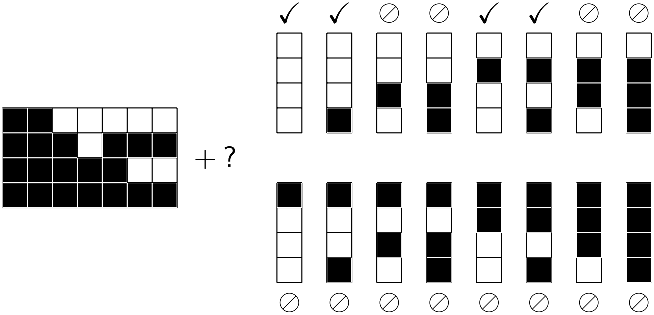
Note: Simply connected is not required.

Most wanted number in the OEIS

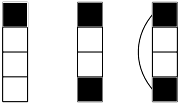
- A348456 is the OEIS entry for the number of arrangements for the $2n \times 2n$ chessboard.
- Neil Sloane gave a guest lecture in our Experimental Mathematics class on April 28, 2022, posing the computation of $A(4)$ as a challenge.

Columns

The possible columns for r rows are binary strings of length r.



Keeping track of connectivity



We must store whether rows are currently connected in the state.

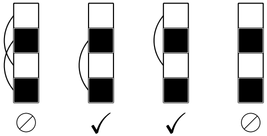
The above diagram shows 3 example states that the machine could be in.

Keeping track of connectivity

- A state is defined by a set partition
- The rows are partitioned into components which are currently connected.
- We also store a bit for each component to indicate which region it belongs to.

Removing Invalid States

- Key to the successful computation of the 8×8 case was reducing the size of the matrix.
- States can be impossible to resolve in the past
- States can be impossible to resolve in the future



Keeping track of area

- The regions are required to be of equal size.
- Use a weight enumerator variable, x, when constructing the transfer matrix.
- Let the entry corresponding to a transition that added k white squares be x^k.
- The result of our computation for the 8×8 case will now be a polynomial in x of degree 64.
- We look specifically at the coefficient of x^{32} to get the answer.

Final Answer

7157114189

Do It Yourself Guide

In which we show how YOU can use this work!

Abstract the code!

- Any computer scientist will tell you, never write the same code more than once!
- All the problems so far have been similar in nature.
- I have created a stencil code file which is easily adaptable to variations on the problem.

What needs to be changed

Suppose the avid listener has an idea for a type of rectangular arrangement. There are 4 functions that they will need to implement.

1. gen-symbols: What are the entries that appear in our arrangment?
2. gen-states(r): Given the number of rows, r, produce a list of states with the start state listed first.
3. is-final(s): Is the state s an accept state? Return true/false
4. valid-trans(s1,s2): Is there a valid transition from state s1 to state s2? Return true/false

Example!

Count the number of matrices with entries $\in \{0, 1, 2\}$ such any two adjacent entries are distinct.

Example

1. gen-symbols: Return the set $\{0, 1, 2\}$
2. gen-states(r): Use the exact contents of the previous column as a state. This is streamlined in the stencil code, simply return gen-columns(r,symbols). The set of states is exactly the set of columns.
3. is-final(s): Just return true! Do all of the checking in the transition function
4. valid-trans(s1,s2): s1 is the previous column, s2 is the proposed next column. Loop over each entry in s2 to make sure the rules aren't broken.

Great Work

- The function comp-seq(r,n) now computes the first n terms of the sequence. comp-seq(3,10) gives

  $12, 54, 246, 1122, 5118, 23346, 106494, 485778, 2215902, 10107954$

- gen-fun(r) now gives the generating function for r rows. gen-fun(3) gives

$$\frac{-4x^2 + 7x + 1}{2x^2 - 5x + 1}$$

Future work

Submit all these sequences and more to the OEIS. The possibilities are endless, and we have generating functions to go with them!

The End

Thanks for listening!