Enumerating Solutions to Grid-Based Puzzles with a Fixed Number of Rows

George Spahn

Received: 9 March 2022 / Revised: 14 April 2022 / Accepted: 29 April 2022
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2022

Abstract In this paper we demonstrate a method for counting the number of solutions to various logic puzzles. Specifically, we remove all of the “clues” from the puzzle which help the solver to a unique solution, and instead start from an empty grid. We then count the number of ways to fill in this empty grid to a valid solution. We fix the number of rows $k$, vary the number of columns $n$, and then compute the sequence $A_k(n)$, which gives the number of solutions on an empty grid of size $k \times n$.

Mathematics Subject Classification 05-08

1 Ring-Ring

The New York Times has recently been publishing Ring-Ring puzzles. A solution consists of drawing rectangles so that no grid square remains empty. Rectangles are not allowed to share a side or a corner, however they may overlap. See Fig. 1 for an example of a completed puzzle. A natural question to ask: How many solutions are there on an empty grid (no black squares) of size $k \times n$?

We consider fixing $k$, and define $A_k(n)$ to be the sequence that gives the number of Ring-Ring solutions on an empty grid of size $k \times n$. This paper gives a method for finding the generating function of this sequence, and gives an explicit formula for the generating function for some small values of $k$.

1.1 The 4 Row Case

First consider the case $k = 4$. Suppose you have a completed solution of an $n \times 4$ grid. Look at one particular column. It is not too difficult to show that it must look like one of the following 15 possibilities.

If we call each of the above possibilities a symbol, we have that a solution to the grid must consist of some sequence of symbols. What determines whether such a sequence is legal? We need to ensure that if a symbol leaves some rectangles in progress, that those rectangles are continued in the next symbol. The information of what rectangles are currently in progress can be thought of as a state. When no rectangles are in progress, as is the case...
for the leftmost column of the grid, the only legal symbols are A and B. However if there is currently a rectangle in progress that occupies rows 1 and 2 (the top two rows), then the legal symbols are J and O. A state can be described as a set of disjoint subsets of the rows, where each subset is of size 2. These subsets specify the rows in which a rectangle is currently in progress. Each rectangle occupies exactly two rows, one for the top edge, and one for the bottom edge.

We can model this situation nicely using a finite state machine. See [1] for an introduction on finite state machines. The machine in Fig. 3 computes whether a string of symbols gives a legal solution for $k = 4$. Any legal solution must start in the state corresponding to the empty set (no rectangles in progress, labelled with START), and proceed to trace a path in the state machine, eventually returning to the empty state. For example, suppose we proceed from the empty state, to $\{1, 4\}$ and then to $\{1, 4\}, \{2, 3\}$ and then to $\{2, 3\}$ and then to $\{1, 4\}, \{2, 3\}$ and then to $\{1, 4\}$ and then to the empty state. This gives the sequence of symbols AKGCLF, which gives the $4 \times 6$ solution that is pictured in Fig. 4.

To count the number of solutions on an $4 \times n$ grid, we need to compute the number of paths of length $n$ that both start and end on the empty state. We can solve for the generating function of this sequence directly by solving a linear system of equations. Let $F_i$ be the generating function such that the coefficient of $x^n$ counts the number of
paths of length $n$ from the start state that end up at state $i$. Then

$$F_i = x \cdot \sum_j F_j$$

where the sum is taken over all states $j$ that have a directed edge to $i$. In words, the number of paths to state $i$ of length $n$ is equal to the sum of the number of paths of length $n - 1$ to states that have an edge to state $i$. We now must modify the equation slightly for the start state itself, to account for the fact that there is 1 way to get to the start state of length 0. So we have the above equations for $i \neq 0$, and that

$$F_0 = 1 + x \cdot \sum_j F_j$$

For $k = 4$, we can solve this system of equations to get

$$F_0 = \frac{(1 + x)(1 - 2x)(1 - 2x - x^2)}{(1 - 3x - 3x^2 + 10x^3 + 3x^4 - 5x^5 - x^6)}$$

1.2 Summary of Results

The approach works for any $k$, and the author has Maple code that can output the corresponding generating function when given $k$ as input. The states are automatically generated by constructing all possible partitions of \{1, 2, \ldots, k\}, into sets of size at most 2. The sets of size exactly 2 indicate the rows at which rectangles are currently in progress. The edges of the state machine are constructed by ensuring the rules of the puzzle are followed. Maple’s built-in
linear algebra package can solve the corresponding system of equations. Here is the output for some small values of $k$:

$k = 2$:

$$F = \frac{1 - x}{1 - x - x^2}$$

As described in [2], we can get a recurrence for the sequence by reading off the coefficients of the denominator of $F$.  

$$A_2(n) - A_2(n - 1) - A_2(n - 2) = 0$$

This is the recurrence for the beloved Fibonacci sequence, and indeed if we start from $n = 1$ we have

$$A_2(n) = 0, 1, 1, 2, 3, 5, 8, 13, \ldots$$

There is also a nice combinatorial explanation for why we get the Fibonacci recurrence. Consider the first two columns of a 2 by $n$ solution. If it contains a 2 by 2 rectangle, than the number of ways to fill out the rest of the solution is equal to the number of 2 by ($n - 2$) solutions. Otherwise the first column must open a rectangle and the second column must continue it. We can now remove the second column to obtain a solution of length ($n - 1$).

$k = 3$:

$$F = \frac{1}{1 - x^2}$$

$$A_3(n) = 0, 2, 1, 8, 12, 45, 98, 292, \ldots$$

The only way to fill out a 3 by $n$ solution is lining up 3 by 2 solutions next to each other. We also note that whenever $k$ is odd, we will obtain an even function for $F(x)$. This is because there are no solutions if the total number of grid squares available is odd.

$k = 4$:

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

$$A(n) = 0, 2, 1, 8, 12, 45, 98, 292, \ldots$$

$k = 5$:

$$F(x) = \frac{-(2x^2 - 1)(x^4 - 3x^2 + 1)}{x^8 - 14x^6 + 19x^4 - 8x^2 + 1}$$

$$A(n) = 0, 3, 0, 12, 0, 51, 0, 221, \ldots$$

$k = 6$:

$$F(x) = \frac{-(64x^{23} + 518x^{22} - 660x^{21} - \ldots)}{(x + 1)(68x^{24} + 496x^{23} - 1685x^{22} - \ldots)}$$

$$A(n) = 0, 5, 1, 45, 51, 573, 1365, 8995, \ldots$$

$k = 7$:

$$F(x) = \frac{-(672x^{90} - 177832x^{88} + \ldots)}{(144x^{92} - 55476x^{90} + \ldots)}$$

$$A(n) = 0, 8, 0, 98, 0, 1365, 0, 19982, \ldots$$

$k = 8$:

$$F(x) = \frac{-(41419800576x^{201} + \ldots)}{(507706343424x^{202} + \ldots)}$$

$$A(n) = 0, 13, 1, 292, 221, 8995, 19982, 346281, \ldots$$
Beyond $k = 8$, the Maple code runs into computational limits. In general, the degree of the denominator is bounded by the number of states, which is equal to the number of partitions of $\{1, 2, \ldots, k\}$, into sets of size at most 2. This is A000085 in the OEIS [3], which exhibits super-exponential growth.

Since the generating functions are rational polynomials, we can get formulae for the corresponding sequences by computing the roots of the denominators. In general, we have

$$A_k(n) = \sum_r \frac{p_r(n)}{r^n}$$

where the sum is taken over all the roots of the denominator of $F$, and the $p_r$ are complex polynomials in $n$. See [2] for more details on this. Let $R$ be the root of the denominator with the smallest absolute value, and let $s$ be its multiplicity. Using big O notation, we have $A_k(n) = O(n^{s-1} \cdot |1/R|^n)$. If all the roots of the denominator are distinct, then $p_R(n)$ will be a constant, so we have $A_k(n) = O(|1/R|^n)$. In practice, this always was the case. Below is a table of $|1/R|$ from $k = 2$ to $k = 8$.

| $k$ | $|1/R|$ |
|-----|--------|
| 2   | 1.618  |
| 3   | 1      |
| 4   | 2.667  |
| 5   | 2.093  |
| 6   | 4.431  |
| 7   | 3.908  |
| 8   | 7.392  |

2 2 Not Touch

We can apply the techniques of the previous section to count the number of 0–1 matrices with fixed column sums and various local constraints. The constraints are inspired by the New York Times’ “2 not touch” puzzles, in which you must place stars so that they are not adjacent horizontally, vertically, or diagonally, and so that the each row and column have exactly 2 stars. Additionally, there must be exactly 2 stars in each region. See Fig. 6 for a completed solution. For the purposes of this paper, we remove the 2 stars per region requirement, and count the number of ways to place stars on an empty grid. Let $a(n)$ count the number of ways to solve it with rows of length $n$. Let $a(n)$ count the number of ways to solve it with rows of length $n$. Since we are placing $2n$ stars on a $k$ by $n$ grid, once $n > k$ we will have more than $2k$ stars, so it will not be possible to satisfy the 2 stars per row requirement. Thus we consider the slightly different problem where the row sum requirement is removed.

Here the symbols are again the possibilities for each column, and again, we can make a finite state machine to describe which symbols can come after each other in a valid solution. We have code to solve various different versions of the problem, using the method described in the previous section. Let $a_{k,r}(n)$ be the number of $k \times n$ 0–1 matrices with the sum of each column equal to $r$. We now add the restrictions:

(i) No two consecutive 1s in a column
(ii) No two consecutive 1s in a row
(iii) No two consecutive 1s diagonally

For each subset $S$ of {{i),(ii),(iii)} we have code to find the generating function of $a_{k,r,S}(n)$. If $S$ is the empty set, this corresponds to the problem without any of the restrictions.

For example, the Maple command `gen_fun(5,2,[1,1,0])` counts the number of solutions with 5 rows, column sum equal to 2, and no two consecutive 1s in a row or column. It produces the output

$$-x^3 - 8x^2 - 6x - 1$$
$$x^3 + 4x^2 - 1$$

corresponding to the sequence (starting from $n = 0$): 1, 6, 12, 26, 54, 116, 242, 518, 1084, 2314.
3 Circuit Board

Lastly, we consider the puzzle “Circuit Board.” Here there exist dots at the center of each grid square. We are allowed to connect a dot to any subset of its four immediate neighbors (up, down, left, right). The goal is to connect all the dots so that each dot has either one edge or three edges emanating from it, without forming any closed loops. See Fig. 7 for an example puzzle, where the squares in the bottom right are blocked out and will not be used. We assume that there are no blocked out squares and count solutions on a $k \times n$ grid.

Transitioning to the language of graph theory, we are looking for spanning trees of the grid graph where each vertex has degree 1 or 3. To check whether solutions with $k$ rows are valid, we can easily ensure that each vertex has the appropriate degree as we read the columns. However we must also ensure that the graph is connected and that there are no cycles. To do this, our state also keeps track of which subsets of the current column are connected. Thus a state is essentially a set-partition of the rows, each partition indicating locations of the most recent column that are connected using the columns that have been read so far.

The final state consists of set partition with only one sub-partition since everything must be connected. In the start state, the set partition is a group of singletons because nothing is connected yet. The number of states is a function of the number of set partitions on a set with $k$ elements. This count grows super-exponentially, and is given by the Bell numbers.
The above has all been implemented in Maple, and we can solve for the generating functions using the same technique as described in Sect. 1. The Maple code was able to complete for \( k \leq 6 \), but for \( k = 7 \) ran into computational limits.

For \( k = 4 \), it turns out there are never any solutions, regardless of the number of columns. Intrigued by this, I found that the number of vertices must be congruent to 2 mod 4 for there to be solutions. Any solution can be constructed by starting with 2 vertices that are connected, and then repeatedly expanding degree 1 vertices to be degree 3. If we think of the vertices on a chess board, then some vertices lie on black squares while others lie on white squares. Expanding a vertex either adds 2 black squares or 2 white squares. Thus our final state must have an even number of vertices, and therefore has the same number of white and black squares covered (since it forms a rectangular grid). Because we started with 2 squares covered, the total amount of squares covered must be 2 mod 4.

For \( k = 3 \) the first 10 terms of the sequence (starting from \( n = 1 \)) are given by: 0, 1, 0, 0, 0, 10, 0, 0, 0, 36

The corresponding generating function is

\[
x^2 \left(4x^8 - 6x^4 - 1\right) \quad \frac{4x^4 - 1}{(4x^4 - 1)}
\]

Using this, we can show that after the first 10 terms, our sequence satisfies the recurrence \( A_3(n) = 4 \cdot A_3(n - 4) \). Adding 4 more columns gives a factor of 4 more solutions! In this case, there are four distinct \( 3 \times 4 \) puzzle pieces that can be sequentially laid down after the first two columns to produce solutions. Also note that the generating function has no constant term, reflecting the fact that the start state is not an accept state.

For \( k = 5 \) the generating function has approximate degree 50, and for \( k = 6 \) it has approximate degree 100. Download the code to try it yourself.

4. Code

My code for the above computations is all available [https://github.com/DarthCalculus/Puzzle-Combinatorics](https://github.com/DarthCalculus/Puzzle-Combinatorics) here [4]. It is written in Maple, and makes use of Maple’s built-in linear algebra package to solve very large systems of equations.

References

1. Sipser, M.: Introduction to the Theory of Computation. Thomson Course Technology, Boston (2006)
2. Kauers, M., Paule, P.: The Concrete Tetrahedron. Asymptotic Estimates. Symbolic Sums, Recurrence Equations, Generation Functions. Springer, New York (2011)
3. [http://oeis.org/](http://oeis.org/). The On-Line Encyclopedia of Integer Sequences, founded in 1964 by N. J. A. Sloane
4. [https://github.com/DarthCalculus/Puzzle-Combinatorics](https://github.com/DarthCalculus/Puzzle-Combinatorics). Accessed 14 April 2022

Publisher’s Note  Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.