the MathILy Record of Mathematics (RoM)

Issue 3: July 14, 2025

AUTHORS: Solon, Benjamin, Jadon, Narnia, Leo, Solon, Patrick, Aidan, Lucy, Shrey, Brandon, Zubi, Jason, Sarah, Miles, Ryan, David, Kyros, J, Brandon, Spark, Andrew, Pavithra, Alex M, Tiger, Clark, Evelyn, Emily, Alex T, Eric, Jaycee

EDITORS: Spark, Thomas, Yoyo COVER BY: Yena (Emily) Choi, Maddie Zhu

Welcome...

...to the 3rd issue of the Record of Mathematics, MathILy 2025! In the five days that MathILy's Week of Chaos offered, we learned what we could about origami constructions, numeric shee-*p*, suspicious voting systems, using primes to make potions, how knitting relates to mathematics, and so much more! What's more, we had some fascinating daily gathers this week, including one from a guest - Pavle - who helped us investigate a way to earn some quick cash. To top the week off, students were set loose in Philly on Saturday! As we pass the halfway point of MathILy, we prepare to use the knowledge that we've built up in Root and Week of Chaos to explore each of three mysterious branch classes: Murderly Mystery, Eelology, and Space Goat Bubbles.

Read more in WoC Classes (Morning Time Slots)!

In this issue...

2.1 How to Fit a Message in a Bottle (Nadav) BY Solon Xia

How much information is necessary to transmit a message (say, in a bottle)? We started by encoding letters into strings consisting of 0, 1 bits, while making sure they were uniquely decodable. We used the expected length (in bits) of a letter to measure the efficiency of our encoding, given its probability of appearing. Eventually, we were able to find and prove an algorithm that generates the most efficient encoding.
Next, we looked at how much information we get from certain events happening. We realized that less likely events give us more information, and found a formula for the amount of information an event contains given its probability.
Lastly, we developed methods to extract uniform randomness out of a biased string of 0's and 1's.

2.2 Game Theory (Natasha) BY Narnia Poddar, Patrick Du

Josh wants some recommendation letters from Josh, but Josh just won't give Josh recommendations without playing a game 999, 999, 999 times: Each Josh picks a number, 1 or 2, to get two numbers x_p and x_d. If $x_p + x_d$ is odd, then Josh has to write Josh $x_p + x_d$ rec letters, but if $x_p + x_d$ is even, then Josh has to get $x_p + x_d$ of his friends to write Josh positive ratings. What should Josh's strategy be?

We made a table to represent the different possible choices and outcomes, and assigned variables for the probabilities of each Josh picking 1 and 2. Eventually, we defined what it means for a strategy to be the best for each Josh, and found that the best strategy for both Joshes leaves Josh writing $\frac{1}{12}$ recommendations for Josh on average. We then generalized this to all games of the same format ("competitive games"), and represented the expected value of the game in terms of the Joshes' strategies (atoms $\hat{p}$ and $\hat{d}$), and the game (fishbox G).

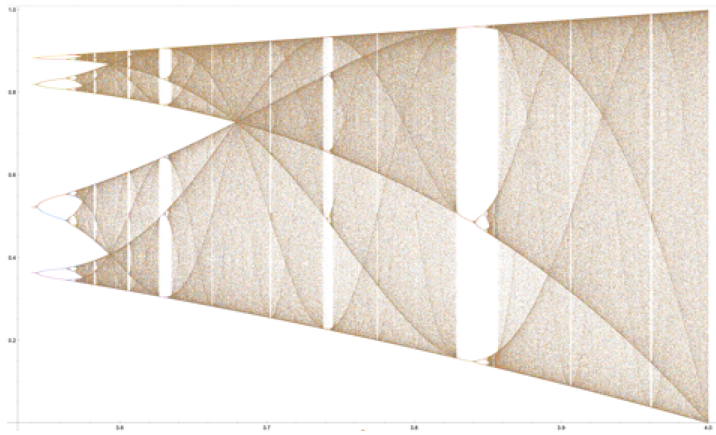
After Josh got tired of writing his 83 million recommendation letters to Josh, they decided to head to the Haverford Duck Pond together, to see some ducks. It turns out that this was also a game! Being quiet together let them see more ducks, being loud together drove ducks away, and being the only loud Josh (and driving quiet Josh away) let you see the most of all. We realized that these "duck games", where each Josh was playing for himself and not against Josh, couldn't be represented in the same way as competitive games, which led to using 2 bubble wraps, G_p and G_d. We also had to redefine what "best strategies" were, which we decided to be a pair of strategies where neither Josh wanted to switch.

We conjectured (! × 96) that for any game, there exists $\hat{p}$ and $\hat{d}$ that we consider best strategies. We found a way to narrow down checking if $\hat{p}$ and $\hat{d}$ were best strategies by only considering switching to Super Simple Salmon Points, and then used one of Berit's shuttles to finish proving our conjecture!

2.3 Origametry (Tom) BY Aidan Bai, Lucy Borunda

In this class, we explored the art of folding paper.

More specifically, we began by learning the terminology for basic origami, like mountain and valley folds. We used these basics to create some modular origami, meaning we slotted smaller pieces together to create larger structures. We then built off of this in order to create different shapes from just a few creases. For example, we discovered multiple ways to construct an equilateral triangle out of a square piece of paper.



But within the chaos we found countless intricate patterns. We also played the hot new game, Teleporting Turnip! The teleporting turnip generated beautiful self-repeating images, while simultaneously burning our computers alive.

2.13 Breaking News: MathILy Solves Democracy! (Vera, Jan) BY J Li

VERY IMPORTANT AND URGENT QUESTION: Who is the most GOATED staff member? Moreover, how do we determine a result from a set of ranked ballots? We started off by defining our ideal voting system and quickly decided it must have some obvious properties:

- like if candidate $x > y$ on every ballot, then $x > y$ in the output (Total Domination)
- or if x is more goated than y and some people change their ballots, but keep x and y in the same relative order, then x is still more goated than y (Who Asked?)
- and also permuting the order of the ballots shouldn't change the result! (Equality)
- and more…

We then came up with possible voting systems, and checked which ones satisfied which properties, tabulating our results in a beautiful table. One system, Vera Domination, was particularly subversive, since it had a One True Queen (Vera), who's ballot solely influenced the result. Subtle foreshadowing: we did not like this.

Using our table, we conjectured and then proved relationships between various properties (such as mutual exclusion), forcing us to narrow down our definition of our ideal system. At long last, we came to a big result: all ideal voting systems are Vera Domination? Goodbye!

2.14 Don't Solve for x: Generating Functions (Berit) BY Brandon Furman

On Monday, Berit made us reimagine our feet. We learned about generating feet, which have infinitely many toes of course. We used our newfound feet to solve numerous problems in combinatorics. For example, how many ways are there to spend 99 cents if you have pennies, nickels, dimes, and quarters? After answering problems like this all week, we made discoveries in many different fields of math. For example, we proved claims about Fibonacci numbers, derangements, and combinatoric identities. On the final day, we found out an awesome new fact about e, and increased our knowledge and happiness meters.

The ultimate conclusion of the class was that solving for x is illegal, and big X is trying to spread corporate propaganda. Don't solve for x. Don't do it.

2.15 Not-Quite-Real Numbers (Jack) BY Spark Shulman

In the beginning, everything was void, and J. H. W. H. Conway began to create numbers. There are two rules: certain things are numbers, and certain numbers might be ≤ other numbers. Specifically:

1. Each number corresponds to two sets of pre-existing numbers, with such that each element of the left set is less than or equal to each element of the right set.

2. $a \leq b$ if and only if no element in the left set of a is greater than or equal to b, and no element in the right set of b is less than or equal to a.

There's a secret third rule: everything is induction. *Everything.*

Fortunately, $0 \equiv 0$, and $-1 < 0 < 1$, and other nice things like that, although they each took large amounts of time, chalk, and induction. Transitivity was key, and all the different 2s did seem to be the same 2. And yet! As it happens, counting to and then past infinity has some remarkable side effects. I believe we all did find ourselves, but we *also* found a lot of numbers. *So many* numbers. And Conway saw that it was good.

2.16 Polynomial Counting (Natasha) BY Andrew Chai, Patrick Du, Sarah Shan

How many outputs of a polynomial are needed to determine its coefficients? To answer this, we first used systems of equations in a fishbox to tackle it for polynomials in one variable. We then conjectured about the two-variable case that given $(k+1)(\ell+1)$ outputs of the polynomial, we could find the coefficient of $x^k y^\ell$. Patrick found a way to magically factor sums to prove this, and we generalized this to $\propto$ (fish) variables with a horrific combination of sigma symbols and subscripts.

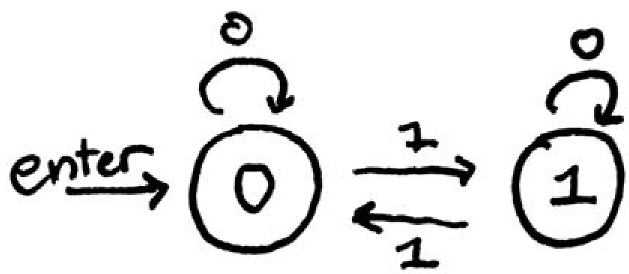
On the final day, we came across problems involving things like sets of sums and choosing numbers on 52-gons. But, we realized that by encoding things using roots and counting degrees of polynomials, we could actually use our conjecture above to count some coefficients in two different ways to prove things about these objects!

5.1 Law-Abiding Hydras (Nadav) BY Mihir Khurana

Nadav started with an introduction to his friendly Hydras: born with 1 teal head (denoted as T), two more are grown back when one is cut off. For colors teal, purple (P) and magenta (M), Nadav defined the following function for hydra-decapitating:

$$\begin{cases} \text{swing}(T) = TP \\ \text{swing}(P) = MT \\ \text{swing}(M) = MP, \end{cases}$$

which is applied to the growing string of Hydra heads. The Hydra starts with one T head, grows to TP, then $TPMT$, $TPMTPMTP$, and so on. By assigning T to 1, and M and P to 0, Nadav's Hydras have generated the binary string 10010010... for us! More than just Hydras, Nadav showed us his nifty Gadgets, which take in a number's binary representation and return either a 0 or 1. He gave this example:



As an example, we considered 10 (1010) in binary, and followed the path of the arrows 1, 0, 1, and 0, which gave us the resulting digit of $0 \to 1 \to 1 \to 0 \to 0$. If we place this digit at the corresponding index, we get a sequence 01101001..., the Thue-Morse sequence! After showing us a plethora of sequences and their respective generating Hydras and Gadgets, Nadav asked us: which sequences come from Hydras? Gadgets? Both? Students worked to find Hydras and Gadgets that created all of Nadav's sequences, and discovered that they were in fact the same! Just as the Thue-Morse sequence can come from a handy Gadget, the rule-following Hydras get to share in the fun. Starting with one head of color A (for auburn?), assigning A to 0, B (blue?) to 1, and given the function:

$$\begin{cases} \text{swing(A)} = AB \\ \text{swing(B)} = BA, \end{cases}$$

the same sequence is generated! Nadav helped us understand this similarity through more helpful notation, and concluded that these Hydras and Gadgets are an example of how different perspectives on the same sequence, or problem are powerful when doing math, for their ability to reveal new patterns. Also, lots of fractals can be described by Hydras!