RESEARCH STATEMENT
JENNIFER ELDER

I have worked in several different research areas throughout my academic career: knot theory, combinatorial game theory, algebra, and enumerative combinatorics. Currently, my research interests are related to computational combinatorics, which involves a mix of computer programming and rigorous proofs.

My current collaboration groups include undergraduate and graduate students, faculty members at universities across the country, as well as non-academic professionals. As of February 2023, among these groups we have two papers accepted for publication, another submitted and awaiting reviewer approval, and we have five more papers in progress.

As I discuss my present and past research projects, I will highlight any publications and preprints that have been produced, as well as list some of the additional research topics that I intend to pursue in the future. Potential student projects will be clearly labeled.

1. Current Research

1.1. **Dynamical Algebraic Combinatorics.** In August 2021, I was fortunate to be able to participate in the workshop Research Community in Algebraic Combinatorics through ICERM. Since then, I have been working with a small research group in dynamical algebraic combinatorics, headed up by Jessica Striker.

Dynamics is the study of systems that evolve over time. Combinatorics studies well defined discrete objects such as posets, permutations, and other families of graphs and diagrams. We are asking the question "how do these well studied discrete objects behave under a new action?"

We have completed our first paper together, titled Homomesies on Permutations. Homomesy is an important phenomenon in dynamical algebraic combinatorics that occurs when the average value of some statistic (for example, a meaningful map $g : \mathfrak{S}_n \to Z$) is the same over each orbit of the action. Homomesy occurs in many contexts, notably that of rowmotion on order ideals of certain families of posets and promotion on various sets of tableaux. More information about homomesy can be found in [5].

In our paper, we perform a systematic study of permutation statistics and bijective maps on permutations in which we identify and prove 122 instances of the homomesy phenomenon. The maps we investigate include the Lehmer code rotation, the reverse, the complement, the Foata bijection, and the Kreweras complement. The statistics studied relate to familiar notions such as inversions, descents, and permutation patterns, and also more obscure constructs. Beside the many new homomesy results, we discuss our research method, in which we used SageMath to search the FindStat combinatorial statistics database to identify potential homomesies. Our paper has been accepted at Mathematics of Computation, and is posted to the arXiv.

We are currently finishing a paper on Rowmotion of Interval Closed Sets on Posets. This paper generalizes work from another paper by Striker, and includes several directions for future study in addition to the Sage code required to complete these investigations. We anticipate submission of this paper in the summer of 2023.

A variety of other projects should keep us active through at least 2024, with journal submissions planned as we finish each project. We were approved for a Collaborate@ICERM opportunity for January 2023 to be able to work together in person. The week was extremely productive, and allowed us to make significant headway on two papers.

Future Work:

- **Cyclic Sieving Phenomenon:** Traditionally, Homomesy and Cyclic Sieving have been viewed as interconnected properties. In a new project, we offer up counterexamples for this idea by exploring permutation maps and statistics that exhibit a lot of homomesy, and almost no cyclic sieving.
- **The Dynamics of Knot Diagrams:** After the ideals project, we intend to work on the dynamical algebraic combinatorics of knot diagrams. This project may get its start in January 2023, if we finish the ideals project early.
- **Dynamical Algebraic Combinatorics for other Combinatorial Objects: (Suitable for Students)** Over the course of the 2021-2022 academic year, I worked with several groups that were interested in statistics related to combinatorial objects. I would like to look at using the methods in my Homomesies on Permutations paper to work on the dynamics of other combinatorial statistics. This would require the student to work with Sage/Python to accomplish these goals, as well as writing proofs for the results.
- **Further Permutation Properties:** Our study of permutation statistics revealed certain permutation map properties that are interesting in their own right outside of the study of dynamics. Currently, I have a student working on a permutation properties project viewed through the lens of linear algebra to try and broaden this field of study. Further student projects in this area are of particular interest to me.

1.2. **Properties of Families of Parking Functions.** Parking Function of size n is a sequence $(a_1, \ldots, a_n)$ of positive integers such that its weakly increasing rearrangement $a'_1 \leq a'_2 \leq \ldots \leq a'_n$

satisfies $a'_i \leq i$. Any permutation on $[n]$ is also a parking function, but parking functions are allowed to have repeated entries.

During the summer of 2022, I was a research teaching assistant for Summer@ICERM with a group who worked on flattened parking functions under the supervision of Pamela Harris. The concept of flattenedness originally came from a command in Mathematica, and has since been used to study permutations and partitions. Our group extended the idea further to parking functions.

A run in a parking function is a generalization of the idea of a run from a permutation. If we have $\omega = \omega_1 \ldots \omega_n$, a run is determined by a sequence of indices $i, i+1, \ldots, i+\ell$ such that

$$\omega_{i-1} > \omega_i \leq \omega_{i+1} \leq \cdots \leq \omega_{i+\ell} > \omega_{i+\ell+1}.$$

We call the start of a run a leading term. If all the leading terms in a parking function form a weakly increasing sequence from left to right, we call that a flattened parking function.

Over the course of the summer, my group studied run distributions in this class of parking functions. We were able to produce a variety of different bijections, recursions, and generating functions results for these parking functions. Our paper can be found on the arXiv, and has been accepted at the Journal of Integer Sequences.

I am currently working with a group of researchers on further study of flattened objects. This group includes two UW Milwaukee graduate students, faculty at UW Milwaukee, and a non-academic collaborator in Indiana. We are working on two papers at the moment, with plans for more. This has including the successful application for an AWM Mentorship travel grant.

In a separate group involving undergraduate students at several universities, I am working on papers on statistics for parking functions. We have submitted one paper, which is in the process of a double blind review. I am happy to answer questions about this project, but for the purpose of that review I will not write further about it here.

We have several other papers in the works to further establish statistics of parking functions, and look at other families of generalized permutations.

Future work:

- **Sage Code for Multiple Types of Objects: (Suitable for Students)** While working with my Summer@ICERM group, we discovered that if we want to work solely with parking functions, permutations, or partitions, Sage has many built in commands to chose from. However, if we wish to start with a permutation and look at partition properties or parking function properties, Sage will not allow us to use code for one object on another. I would like to work with a student on considering other areas of overlapping property questions, and then writing code that will allow us to deal with these issues.

- **Flattened Parking Functions: (Suitable for Students)** There are a variety of spin off projects that I would like to consider from the paper the Summer@ICERM group has written. I would continue to coordinate with the group to see which topics were available, and which ones are being studied by new groups of students. These potential projects have a wide range of levels. Some could be suitable for an undergraduate capstone project, while others could be developed into a graduate thesis.
- **Choose Your Own Adventure: (Suitable for Students)** Carlson et. al. published the paper Parking Functions: Choose Your Own Adventure, [3], on how to generate and work on research problems for parking functions. I would like to be able to use this paper, and the existing literature, to work with other students in the future.

2. Past Research

2.1. **Reduced Words.** For my doctoral dissertation, I studied properties of permutations through the lens of reduced words. We generate our standard finite symmetric group $\mathfrak{S}_n$ with the functions $s_i = (i\ i+1)$, which only swaps two consecutive elements for $1 \leq i \leq n-1$. Every permutation σ can be decomposed into products of these s_i's, and we consider reduced decompositions, written using a minimal number of these s_i's. For example: $\sigma = [3\ 2\ 1] = s_4 s_1 s_2 s_1 s_4$ is not reduced, while $\sigma = s_1 s_2 s_1$ is reduced. We can also write these products of transpositions as words using the indices. In this example, σ can be represented by the word 121.

From there, we study all possible ways of writing our fixed permutation as reduced decompositions and words. To do this, we consider the following relations among the s_i's:

$$s_i^2 = 1 \tag{1}$$

$$s_i s_j = s_j s_i, \text{ for } |i-j| > 1 \tag{2}$$

$$s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1}, \text{ for } |i-j| = 1 \tag{3}$$

The second relation is called a commutation move, while the third relation is called a braid move.

We use these relations to generate all possible reduced decompositions for a permutation, and we denote the related set of reduced words as $R(\sigma)$. For example, for $\sigma = [3\ 2\ 1]$, the set of reduced words is $R(\sigma) = \{121, 212\}$.

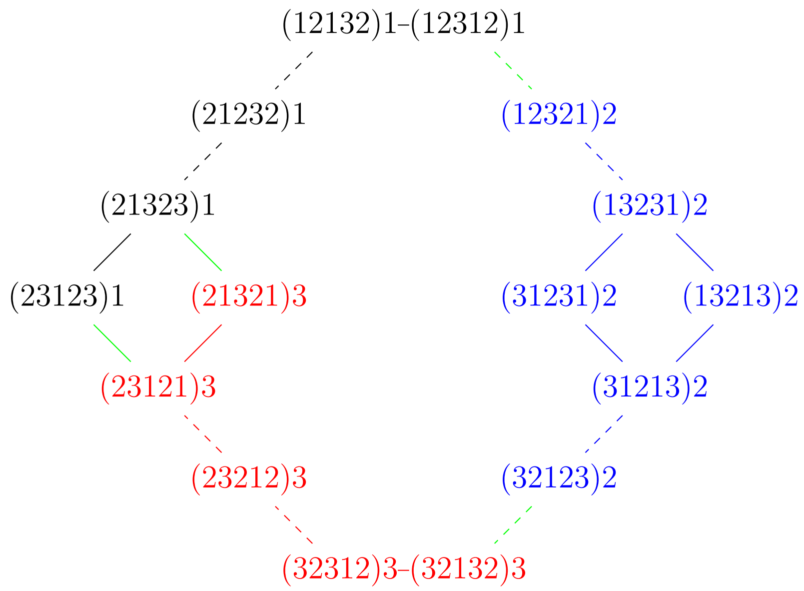
The generating relations can be used to produce partial order called the weak order on $\mathfrak{S}_n$, denoted $W(\mathfrak{S}_n)$. The cover relations in $W(\mathfrak{S}_n)$ come from the addition or removal of a single s_i, dependent on the descent set for the permutation, denoted $D(\sigma)$.

As I began to tackle problems in this field, I was interested in being able to use recursive relations to break the problems into smaller pieces. Studying σs_i can be easier than studying

σ itself. Unfortunately, there are few published results that relate $R(\sigma)$ to $R(\sigma s_i)$ for arbitrary permutations. This gap in the literature has motivated most of my research.

2.1.1. *Graphs of sets of Reduced Words.* I have also been focused on using the partition of the elements in $R(\sigma)$ to study subgraph structures related to this set. From the elements in $R(\sigma)$, we produce a graph with the following requirements: if two words are associated by commutation moves, we give a double edge in the graph, and if they are associated by braid moves, we give a single edge. We call this graph $G(\sigma)$.

Consider $w_0 \in \mathfrak{S}_4$, the longest permutation in $\mathfrak{S}_4$. The graph $G(w_0)$ has been drawn below, with the induced subgraphs highlighted in different colors: $G(w_0 s_1)$ is drawn black, $G(w_0 s_2)$ in blue, and $G(w_0 s_3)$ in red. There are also four edges in green that only appear in $G(w_0)$. Dashed lines are for braid relation edges and solid lines are for commutation relation edges.



From this partition of vertices, I have been able to produce results on induced subgraphs, as well as arriving at recursive formulas for the edges in $G(\sigma)$.

Additionally, I have produced another family of subgraphs that will partition the vertices in a different way, which has produced results for studying the average braid degree in $G(\sigma)$.

Currently, I am in the process of rewriting and expanding these results for journal submission. A preliminary draft has been posted to the arXiv.

Future Work:

- **Graphs of Sets of Reduced Words: (Suitable for Students)** When we study graphs of reduced words, we do not often ask standard graph theory questions. While we may care about the diameter of a graph, we tend not to ask about planarity, or vertex degree, cliques or co-cliques, etc.

- **Braid Groups and Reduced Words: (Suitable for Graduate Students)** In knot theory, it is known that every knot can be drawn as a braid diagram. These braid diagrams can also be written as permutations using the same reduced decompositions I have been studying. I am interested in tying sets of reduced words back to knot theory, and considering what new results for each field could be generated by considering each family of problems from a different angle.

2.1.2. *Ratios and Sizes of sets of Reduced Words.* From the partition of the set of reduced words used for calculations of $|R(\sigma)|$, I have tried to find a good lower bound for the ratios $\frac{|R(\sigma s_i)|}{|R(\sigma)|}$. This led me to studying what other results are known for calculating $|R(\sigma)|$.

Some families of permutations have structures that allow us to easily calculate $|R(\sigma)|$. For example, we study certain families of pattern avoiding permutations. That is, permutations that when written in one line notation do not contain any subsequences of a particular form. The permutation $\sigma = [51324]$ contains patterns of the form 312, but does not contain the pattern 2143. This particular permutation is called vexillary, since it is 2143 pattern avoiding. Vexillary permutations have a relationship between its set of reduced words and Young tableaux: there is a $\lambda \vdash l(\sigma)$ such that $|R(\sigma)| = f^{\lambda}$. In addition to vexillary permutations, I have tackled calculations for Grassmannian permutations that have a unique descent, and Fully Commutative permutations which are 321 pattern avoiding, finding lower bounds for the desired ratios for both of these families.

Future work:

- **Improved Bounds for these Ratios:** I have been able to find a lower bound that works for all permutations, and I'm exploring methods of improving the lower bound before final journal submission.

2.2. **The Futurama Theorem.** For my master's thesis, I created generalizations of The Futurama Theorem, or Keeler's Theorem. Created for an episode of the TV show Futurama by Ken Keeler, this is a permutation problem disguised as Science Fiction. In *The Prisoner of Benda*, two characters build a two seat brain swapping machine, but upon using it they discover that the machine will not swap the same two people back. More and more characters use the machine, and hilarity ensues.

In other words, you have a permutation as a product of two cycles that cannot be reused in the inverse function. How do you get everyone back where they belong? Introduce two extra people who can be used in a particular manner to swap everyone back. Translated into the language of

$\mathfrak{S}_n$, the solution to the problem in the episode can be written as follows:

$$\begin{aligned}((1\ 2)(3\ 4\ 5\ 6\ 7\ 8\ 9))^{-1} &= (y\ 6)(y\ 7)(y\ 8)(y\ 9)\\ &\quad (x\ 3)(x\ 4)(x\ 5)(y\ 3)(x\ 6)\\ &\quad (y\ 2)(x\ 1)(y\ 1)(x\ 2)\end{aligned}$$

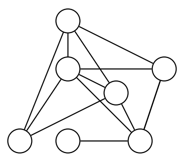
For my thesis, I imagined dealing with larger machines, created rules for these machines to perform swaps, and then created the inverse functions that would follow these rules.

Let us say we want to study an n person brain swapping machine. We need to be able to write the function σ created by the machine, so that we know where everyone's brains end up. We restricted the machine to bumping each brain one person to the right. Additionally, we needed a restriction for who could sit in the machine. The original problem said that if two people sat in the machine with each other they could not use it again at the same time. We translated this to mean that the same n-people could not arrange themselves in the σ^{-1} configuration, or in any other configuration that would appear in $\langle\sigma\rangle$.

From this last condition, I studied machines with a prime number of seats $p > 2$, since those generated subgroups all contain cycles of length p. By the time I defended, I also had the inverses functions for machines with an even number of seats, and an inkling of how to solve the problem for any odd number. My advisor and I finished the problem, and published our results in [4].

- **The Futurama Theorem: (Suitable for Students)** Making simple changes to the phrasing of the question and the requirements of the fake "machine" would prove interesting. Additionally, translating the problem into graphs, and studying paths that do not contain the same vertices. This also has the potential for some additional graph theory study on cycles in a graph, and there could be some applications to computer programming.

2.3. **Graphs and Games.** GRIM is a two player game on finite graphs, similar to NIM. For example, we present two players with the graph below.



The players then take turns deleting vertices, and the edges that are incident to those vertices until there are no more moves to make. The last person to play on the graph wins. Created by my advisor and two classmates, I was the second of three research groups to contribute to our initial findings. Our work was published in [2].

The basic question for the game is, who can win a game on a particular graph? The first person to play, or the second? For a graph like the one above, that does not appear to fall into a familiar family of graphs, the answer might be difficult.

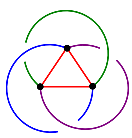
For certain families of graphs the answer is simple. For example, a complete graph on n vertices means that every choice of vertex is identical, so the answer only depends on n. Other families could be deceptively hard. It is easy to prove that player one has a winning strategy on all paths of odd vertex length, but there was no universal strategy for paths of even vertex length.

While I worked on the project, I contributed to results on complete multi-partite graphs. I studied the vertex-weighted graph version of the game, showing that it was also just an n-partite graph problem. And I studied graph automorphisms, and helped classify which families of graphs would be won by which player using automorphisms.

- **Combinatorial Games: (Suitable for Students)** Combinatorial games are a great place for students to explore research. Whether starting with an existing game, or creating a brand new combinatorial game. At outreach events, I run a popular activity on the card game SET, which has a lot of combinatorial structure to it, and could be another jumping off point. Additionally, the AWM has just released another combinatorial card game, EvenQuads, that would make for an interesting student project.

2.4. **Knot Theory:** Knot theory at a basic level is focused on the following problem: if we have two three-dimensional knots that don't look alike, how do we prove that they are in fact distinct, and we cannot simply manipulate one to match the other? This is often done by projecting the knots into a two dimensional plane, with crossing information encoded, and then relating the images to invariants of certain kinds. Once you have a two dimensional picture and an invariant to use, you start resolving crossings in some manner until you have a collection of circles, and perhaps a polynomial in several of variables. If two knots produce two different polynomials, you know they cannot be the same three dimensional object.

The focus of my research was spatial graphs: knots with traditional crossings and certain fussed points that resembled vertices in a graph, as illustrated below.



The goal was to use known knot invariants to create new spatial graph invariants. My research partner and I developed these invariants by splitting the vertices apart in a similar manner to traditional crossing resolutions in knot theory.

We worked mainly with regular spatial graphs, both oriented and unoriented. That is, every vertex had the same number of edges incident to it in our graph. This led to questions of how many knot diagrams would we produce as we split our vertices apart? We ended up using basic counting techniques from combinatorics to produce $\binom{n}{2}^k$ for our unoriented graphs, where n referred to the degree of our vertices, and k to the number of vertices in our graph. These splittings produced a multiset of knots, which were then assigned invariants and summed together to create an invariant for the original graph. A similar approach was used for oriented graphs with balanced vertex in and out degrees.

In addition to this work, and proving that it was a better invariant than similar existing work, we also considered the contraction deletion move from graph theory to produce similar invariants, but in a smaller degree space. Our results were published in [1].

References

[1] E. Aceves, J. Elder. Invariants for Spatial Graphs, The Rose Hulman Undergraduate Mathematics Journal Vol. 16, Issue 2, 2015.

[2] R. Adams, J. Dixon, J. Elder, J. Peabody, O. Vega, K. Willis. Combinatorial Analysis of a Subtraction Game on Graphs, International Journal of Combinatorics vol. (2016), Article ID 1476359, 8 pages.

[3] J. Carlson, A. Christensen, P. E. Harris, Z. Jones A. R. Rodríguez. Parking Functions: Choose Your Own Adventure, The College Mathematics Journal, (2021) 52:4, 254-264, DOI: 10.1080/07468342.2021.1943115

[4] J. Elder, O. Vega. A Note on the Factorization of Permutations into Cycles, International Electronic Journal of Algebra 22 (2017), 125-132.

[5] T. Roby. Dynamical algebraic combinatorics and the homomesy phenomenon. In Recent Trends in Combinatorics, (2016), pages 619–652, Cham.