Research Statement

AJ Bu

1 Introduction

Experimental mathematics is not a specific field of math, but an approach which can be used to research various areas of math. My research focuses on the application of experimental mathematics to a number of problems, demonstrating its methodology, which is described in Section 2 below. In particular, I have worked on problems in combinatorics, with a focus on Dyck and Motzkin paths. In Section 3, I describe my research on Dyck paths, Motzkin paths, and similar paths. These projects include enumerating restricted Dyck paths (see [5]), enumerating restricted Motzkin paths (see [3]), and studying the area under generalized Dyck paths (see [4] and [6]). In Section 4, I describe my paper on the Groebner bases of ideals generated by elementary symmetric functions (see [2]).

Programming and developing algorithms are important tools in generating ideas and tackling potential problems. In general, as I pursue other topics, I plan to continue to develop efficient algorithms and symbolic programming to aid in my research, thereby aiding further investigations and contributing to computer algebra. Both computer algebra and discrete math are fundamental in mathematics, cryptography, science, and technology, including computer technology and the internet. Beyond concrete outcomes, this work will contribute to the collaboration between mathematicians and computers. While mathematical investigation is traditionally done by hand, the methodology of experimental mathematics has shown itself to be invaluable. Computers can provide mathematicians with a vast amount of data used to form new conjectures, which would be extremely tedious – if not impossible – to obtain by hand. As computers become increasingly powerful and allow more flexibility in programming, mathematicians need to adapt and develop new research methods. By designing algorithms, mathematicians can "teach" computers to form and rigorously prove conjectures, unlocking their untapped potential in furthering mathematical studies.

2 The Methodology: Experimental Mathematics

In the most general sense, experimental mathematics is the methodology of using computation and algorithms to study mathematical objects, typically involving computer-assisted proving. One form involves using a computer to investigate mathematical objects and then using this data to form conjectures on their general properties, which are easier to prove *a posteriori*. Here, a mathematician may write a code to generate sufficiently many examples, study these examples, and identify (or even program the computer to guess) patterns among them. I use this approach in my joint paper with Robert Dougherty-Bliss, "Enumerating Dyck Paths with Context-Free Grammars," [5] and my paper "A Combinatorial Approach to the Groebner Bases for Ideals Generated by Elementary Symmetric Functions" [2].

Alternatively, if the mathematician already has a conjecture, they can program the computer to either prove or disprove it. For example, say the mathematician can prove that, if a counterexample exists, then the minimal counterexample must have a given form. The computer can then either show that such a minimal counterexample cannot exist – which is famously done in the Appel-Haken proof of the Four Color Theorem [1] – or find a counterexample.

Finally, the mathematician may program the computer to automatically form and prove a conjecture. In the code, the mathematician provides the general form of the desired solution and outline of the proof. The computer can then try to identify patterns and form rigorous proofs by following the outline provided by the mathematician. For example, the mathematician may translate certain properties of a mathematical object into a system of polynomial equations. Such polynomials generate an ideal. Since any basis will give the same set of solutions, it can be advantageous to change basis. To form and prove conjectures, the computer may for instance use Buchberger's algorithm to compute a Groebner basis, making it easier to manipulate and solve the system of equations. This approach is used in the papers "Automated Counting of Restricted Motzkin Paths" [3] and "Using Symbolic Computation to Explore Generalized Dyck Paths and Their Areas" [6].

2.1 Developing Algorithms and Programming using Groebner bases

Multiple Maple packages that I have written have implemented Buchberger's algorithm and Groebner bases to form and prove conjectures. Designing such algorithms that use Groebner bases for efficient computations is a key problem in computer algebra. For potential readers unfamiliar with Groebner bases, I will briefly elaborate on their general use in proofs.

Definition 1. A *Groebner basis* of an ideal $I \subset k[x_1, ..., x_n]$ (with respect to a monomial order $>$) is a finite subset $G = \{g_1, ..., g_t\}$ of I such that, for every nonzero polynomial f in I, f is divisible by the leading term of g_i for some i.

Definition 2. A Groebner basis G is *reduced* if, for every element $g \in G$,

1. the leading coefficient of g is 1, and
2. no monomial in g is in $\langle LT(G - g)\rangle$, the ideal generated by the leading terms of the other elements in G.

It is known that every nonzero polynomial ideal I has a unique reduced Groebner basis. In general, the Groebner basis makes it easier to interpret the properties and structure of the ideal. It simplifies solving the ideal membership problem and finding solutions to a system of polynomial equations. A polynomial f lies in the ideal $I \subset k[x_1, ..., x_n]$ with Groebner basis G if and only if the remainder on division of f by G is zero.

In forming conjectures, however, our problems will not already be stated as polynomials. By letting variables represent certain properties, we can translate various structures into polynomials, as was done in `MotzkinClever.txt`, the Maple package that I wrote to accompany [3]. Thus, if we let I be the ideal generated by the polynomials describing the properties of the given object, and f be the polynomial representing some claim about the mathematical object, then this claim is true if and only if f is in I. Therefore, the computer can prove or disprove the claim algorithmically, using Buchberger's algorithm to find a Groebner basis and then applying the division algorithm. A more in depth explanation of the Groebner basis method as well as examples can be found in [10] and [11].

3 Research on Dyck, Motzkin, and Similar Paths

3.1 Enumerating Restricted Dyck Paths Using Context-Free Grammars

In my joint paper with Dougherty-Bliss [5], we used Zeilberger's package `DyckClever.txt` from [7] to find the bivariate polynomial $F(x, P)$ such that $F(x, f(x)) = 0$, where $f(x)$ is the generating function for the sequence enumerating Dyck paths with given sets of restrictions.

Definition 3. A *Dyck path* of semi-length n is a walk in the $xy-$plane from the origin $(0,0)$ to $(2n,0)$, consisting of n up-steps $U := (1,1)$ and n down-steps $D := (1,-1)$, that never goes below the $x-$axis.

The restrictions considered in both the maple package and my joint paper are defined as follows.

Definition 4. A *peak* on a Dyck path is the bigram UD. The *height* of this peak is given by the $y-$coordinate of the Dyck path after the step U.

Definition 5. A *valley* on a Dyck path is the bigram DU, and its height is given by the $y-$coordinate after the step D.

Definition 6. A Dyck path has an *upward-run of length* n if there are n consecutive up-steps that are not directly followed by nor directly follow an up-step.

Definition 7. A Dyck path has a *downward-run of length* n if there are n consecutive down-steps that are not directly followed by nor directly follow a down-step.

Let A, B, C, and D be arbitrary sets of positive integers – either finite sets, infinite sets defined by arithmetical progressions, or the finite union of such sets. `DyckClever.txt` includes algorithms which directly compute the equation satisfied by the generating function for the sequence of the number of Dyck paths which avoid the following:

- peak heights in A,
- valley heights in B,
- upward-runs with lengths in C, and
- downward-runs with lengths in D.

In doing so, the procedures supply algebraic proofs of these identities.

While these algorithms can furnish the desired equations satisfied by the generating functions for given sets A, B, C, and D, they cannot produce identities for infinite families, or arbitrary sets of a given form. For example, running the appropriate procedure delivers the desired equation for the sequence of Dyck paths avoiding upward-runs of length $ar+b$ for *given* non-negative integers a and b (e.g. $5r+2$), where r is a variable ranging over the non-negative integers. By running the procedure for various values a and b, I was able to extend these findings and form conjectures on the infinite family of Dyck paths avoiding upward-runs of length $ar+b$ for *arbitrary* non-negative integers a and b. Using this approach, I formed conjectures on other infinite families of restricted Dyck paths. In "Enumerating Dyck Paths with Context-Free Grammars," I present these identities and prove that certain infinite families have an explicit context-free grammar which yields the equation satisfied by the generation function [5]. In doing so, I use recursive rules to show that any path with such restrictions can be rewritten in one of finitely many forms concatenating certain steps with other paths in a specific order. Using the principle of inclusion-exclusion, I resolve any issues of over-counting or unwanted generated paths. "Grammatical proofs" or "proving grammatically" refer to such proofs. These grammatical proofs yield a lot of insight by providing structural information about the restricted Dyck paths not given by the computer-automated, algebraic proofs.

3.2 Enumerating Restricted Motzkin Paths

In my paper [3], I generalized Zeilberger's method for automatic counting of restricted Dyck paths [7] to the Motzkin paths.

Definition 8. A *Motzkin path of length* n is a walk in the $xy-$plane from the origin $(0, 0)$ to $(n, 0)$ with atomic steps $U := (1, 1)$, $D := (1, -1)$, and $F := (1, 0)$ that never goes below the $x-$axis.

Definition 9. A *peak* on a Motzkin path is the sequence of steps UF^kD for $k \geq 0$, where F^k denotes k consecutive F steps. The *height* of this peak is given by the $y-$coordinate of the Motzkin path after the step U.

Definition 10. A *valley* on a Motzkin path is the sequence of steps DF^kU for $k \geq 0$. Its height is given by the $y-$coordinate after the step D.

Definition 11. *Upward-runs* and *downward-runs* are defined the same way as in Dyck paths.

Similarly, we say that a Motzkin path has a *flat-run of length* n if there are n consecutive flat-steps F that are not directly followed by nor directly follow a flat-step.

Let A, B, C, D, and E be arbitrary sets of positive integers – either finite sets, infinite sets defined by arithmetical progressions, or the union of such sets. I wrote Maple packages `Motzkin.txt` and `MotzkinClever.txt`, which include programs which find the polynomial $F(x, P)$ that is zero when P is set as the generating function for the sequence counting Motzkin paths of length n avoiding:

- peak heights in A,
- valley heights in B,
- upward-runs with lengths in C,
- downward-runs with lengths in D, and
- flat-runs with lengths in E.

`Motzkin.txt` uses numeric dynamic programming to generate sufficiently many terms of the sequence enumerating Motzkin paths with the desired restrictions, and then guesses the recurrence. `MotzkinClever.txt` generates a finite system of algebraic equations by using symbolic dynamic programming and then solves the system to get the algebraic equation satisfied by the generating function directly. More specifically, I express recurrences for restricted Motzkin paths as polynomials by translating each set of restrictions into a distinct variable. Then, the procedure efficiently finds the reduced Groebner basis to get the desired equation for the sequence of Motzkin paths with that set of restrictions.

Both Dyck paths and Motzkin paths have a specific set of atomic steps: $\{(1, 1), (1, -1)\}$ and $\{(1, 1), (1, 0), (1, -1)\}$, respectively. For future study, we can extend my findings on Dyck and Motzkin paths to similar paths which allow an arbitrary set of atomic steps. We can try to do this by following the same approach as I used with Motzkin paths: identify recursive relations for the set of possible paths and then find an equality solved by the generating function for the number of such paths of length n. While the same approach should work, finding and justifying the necessary recursive relations is very non-trivial. Afterwards, we can experiment with various sets of steps to form conjectures on infinite families, which we will then work to prove.

3.3 The Area Under Generalized Dyck Paths

The bivariate weight enumerator for Motzkin paths with length n and area m satisfies the following functional equation

$$M(x,q) = 1 + xM(x,q) + x^2qM(qx,q)M(x,q).$$

Definition 12. *Generalized Dyck paths* are walks on the $xy-$plane from the origin $(0,0)$ to $(n,0)$ with an arbitrary set of atomic steps and that never go below the $x-$axis.

In my joint paper [6] with Doron Zeilberger, we use symbolic dynamical programming to automatically generate algebraic equations satisfied by the generating functions enumerating generalized Dyck paths. Using calculus, we can then compute generating functions for the sum of the areas under such paths as well as sum of a given power of the areas. These methods are fully automated in the accompanying Maple package `GDW.txt`.

In the paper [4] , I demonstrate how to use dynamical programming to find the weight enumerator for the area paths of length n with steps in a given set S that start and end at height 0 and never have negative height. I also describe how to find the weight enumerator for such paths when, instead of a set of steps S, we are given bivariate polynomials $P(x,q)$, $Q(x,q)$, and $R(x,q)$ such that the weight enumerator $f(x,q)$ satisfies

$$f(x,q) = P(x,q) + Q(x,q)f(x,q) + R(x,q)f(x,q)f(xq,q).$$

I then present a method for finding $f^{(k)}(x,1) := \frac{d^k}{dq^k}[f(x,q)]\big|_{q=1}$. Rather than outputting algebraic equations as seen in [6], this procedure produces closed-form expressions in terms of radicals.

These methods are fully automated in the accompanying Maple package `qEW.txt`, displaying how the power of computer algebra and using calculus allows us to generate quite a few moments. In the paper, I demonstrate these methods with the bivariate weight enumerators for both Motzkin paths and Dyck paths with length n and area m. Moreover, I show how these procedures can be used to produce the Maclaurin series of $\frac{d^k}{dq^k}[f(x,q)]\big|_{q=1}$, allowing us to find the generating function for the total area under such paths of length n as well as for the sum of a given power of the areas.

For further study, we can use the methods in both papers to study the statistical information about the area under a random generalized Dyck path. Given a family of paths, let $a_0(n)$ be the number of such paths of length n, $a_1(n)$ be the total area under such paths of length n, and $a_2(n)$ be the sum of the squares of the areas under such paths of length n. Using the accompanying Maple package `qEW.txt`, we can generate 10,000 (or more) terms of the sequences of the average areas $\left\{\frac{a_1(n)}{a_0(n)}\right\}$ and the variances $\left\{\frac{a_2(n)}{a_0(n)} - \left(\frac{a_1(n)}{a_0(n)}\right)^2\right\}$ and use numerics for the asymptotics.

4 Groebner Bases for Ideals Generated by Elementary Symmetric Functions

In [9], Mora and Sala provide the reduced Groebner basis of the ideal formed by the elementary symmetric polynomials in n variables of degrees $k = 1,\ldots,n$, $\langle e_{1,n}(x),\ldots,e_{n,n}(x)\rangle$ [9]. Haglund, Rhoades, and Shimonozo expand upon this, finding the reduced Groebner basis of the ideal of elementary symmetric polynomials in n variables of degree d for $d = n-k+1,\ldots,n$ for $k \leq n$ [8].

Definition 13. Let k and n be natural numbers. The *elementary symmetric polynomial* of degree k in n variables $x_1,\ldots,x_n$ is

$$e_{k,n}(x) = \sum_{1\leq i_1<\cdots<i_k\leq n} x_{i_1}\ldots x_{i_k}.$$

Definition 14. The *homogeneous symmetric polynomial* of degree k in n variables $x_1, \ldots, x_n$ is

$$h_{k,n}(x) = \sum_{1 \le i_1 \le \cdots \le i_k \le n} x_{i_1} \ldots x_{i_k}.$$

Mora and Sala prove that $\{h_{1,n}(x), h_{2,n-1}(x), \ldots, h_{n,1}(x)\}$ is a Groebner basis of the ideal $\langle e_{1,n}(x), \ldots, e_{n,n}(x)\rangle$ [9]. In my paper [2], I use the accompanying Maple package that I wrote with Doron Zeilberger, `Solomon.txt`, to efficiently generate the reduced Groebner bases of many *specific* ideals using symbolic computation and extend their findings. I first use experimental methods to deduce a pattern for the reduced Groebner bases of the ideals $\langle e_{1,n}(x), \ldots, e_{k,n}(x)\rangle$ and $\langle e_{1,n}(x), e_{k,n}(x)\rangle$ for arbitrary $k \le n$, and prove them by combinatorial means. I then investigate other cases to expand upon my results to the ideal $\langle e_{k_1,n}(x), \ldots, e_{k_m,n}(x)\rangle$. I find a basis for this general case, proving that it generates the ideal, and show empirically that it is a Groebner basis.

One direction for further research is to formally prove that the basis we have found for the general case is the reduced Groebner basis. We can also try to find similar identities for other ideals, such as those generated by various power sum symmetric polynomials or homogeneous symmetric polynomials of arbitrary degrees.

References

[1] K. Appel and W. Haken, Every planar map is four colorable, *Contemporary Mathematics* **98** (1989).

[2] AJ Bu, "A Combinatorial Approach to the Groebner Bases for Ideals Generated by Elementary Symmetric Functions," Electron. J. Combin. 29 (3) (2022).

[3] AJ Bu, "Automated Counting of Restricted Motzkin Paths", ECA 1 (2) (2021), Article S2R12.

[4] AJ Bu, "Explicit Generating Functions for the Sum of the Areas Under Dyck and Motzkin Paths (and for Their Powers)," (2023).

[5] AJ Bu and Robert Dougherty-Bliss, "Enumerating Restricted Dyck Paths with Context-Free Grammars", Integers 21 (2021).

[6] AJ Bu and Doron Zeilberger, "Using Symbolic Computation to Explore Generalized Dyck Paths and Their Areas," arXiv preprint arXiv:2305.09030 (2023).

[7] S.B. Ekhad and D. Zeilberger, Automatic Counting of Restricted Dyck Paths via (Numeric and Symbolic) Dynamic Programming, *Preprint* arXiv:2006.01961 (2020).

[8] J. Haglund, B. Rhoades, and M. Shimozono, *Ordered set partitions, generalized coinvariant algebras, and the delta conjecture*, Adv. Math. **329** (2018), 851–915.

[9] T. Mora and M. Sala, *On the Gröbner bases of some symmetric systems and their application to coding theory*, J. Symbolic Comput. **35** (2003) no. 2, 177-194.

[10] D. Petrović, Automated proving in Geometry using Gróbner bases in Isabelle/HOL, (2011).

[11] F. Winkler, Gróbner bases in geometry theorem proving and simplest degeneracy conditions, *Mathematica Pannonica* **1** (1990), 15-32.