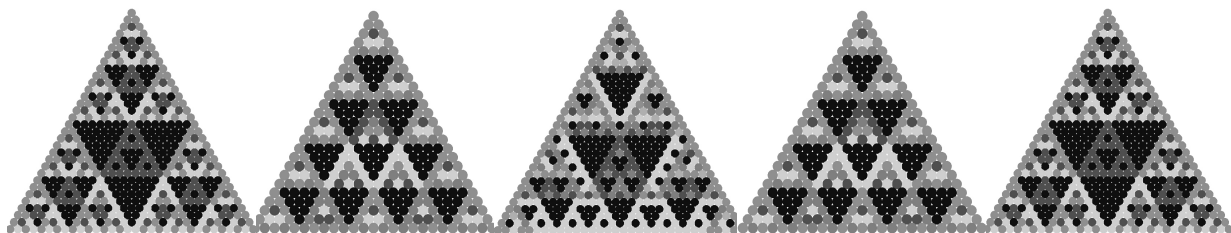


Visualization Projects for Abstract Algebra

From the PascGalois Project
by Michael J. Bardzell
and Kathleen M. Shannon
Salisbury University

Including PascalGT for Windows

Support provided by the National Science Foundation award #DUE-0087644
and by the Richard A. Henson endowment for the Richard A. Henson School
of Science and Technology at Salisbury University.



Contents

Introduction to the student

PascGalois Project 1: Pattern recognition and Pascal's Triangle Modulo n

PascGalois Project 2: Pattern Recognition Using Dihedral Groups

PascGalois Project 3: Building A Group with PascGalois Triangles

PascGalois Project 4: Direct Products and Automorphisms

PascGalois Project 5: Quotient Groups 1

PascGalois Project 6: Quotient Groups 2

PascGalois Enhancement Projects:

Self-Similarity and the Klein-4 Group

Rings and Fields

Two Dimensional Automata

Appendix: Brief Introduction to the PascalGT Software

Support provided by the National Science Foundation award #DUE-0087644 and by the Richard A. Henson endowment for the Richard A. Henson School of Science and Technology at Salisbury University.

Introduction to the Student

“... A mathematician, like a painter or poet, is a maker of patterns. If his patterns are more permanent than theirs, it is because they are made with ideas.” G.H. Hardy

Welcome to the PascGalois Project. The materials included are designed as supplements to a modern algebra course. Algebra is regarded by many students as the least visual course in the undergraduate mathematics curriculum. The first goal of these activities is to help you visualize many of the important concepts from a first semester abstract algebra course, particularly group theory. These activities are not intended to replace a textbook, nor the standard homework problems that are found in any traditional text. Rather, they should serve as a supplement allowing you to attempt different types of problems that rely more on exploration and visualization using computer graphics. It is our hope that approaching algebra from multiple perspectives will help create a deeper understanding of the subject. We also hope that you will see connections with other areas of mathematics. The structures studied in these projects have applications with other disciplines including dynamical systems and number theory. However, no prior knowledge of those subjects is assumed to use these materials.

A second goal of these materials is to allow you opportunities to examine more “open-ended” problems. For example, many of the exercises ask you to look for a pattern in a given image. Actually, these images contain numerous patterns. You may find one pattern whereas a classmate finds a completely different one. However, the algebraic structures under consideration play a crucial role in producing the patterns found in these exercises. You may even find a pattern that nobody else has noticed yet! In fact, these projects can serve as a good starting point for an undergraduate research project. Although the exercises are designed for course work, they do have an exploratory flavor that makes them amenable to further research.

One of your tasks will be to make connections between the appearance of the computer generated images and the underlying concepts from abstract algebra. Even once these connections have been found, you will see how difficult it is to describe the relationships using precise terminology while writing up solutions. Mathematics majors certainly need to learn how to solve problems. However, it is equally important that they learn how to put their solutions clearly and accurately in writing so that others can read and understand their work. Avoiding vague and undefined terms is crucial to this effort. As you write solutions to these projects ask yourself whether the reader should already know the terms you are using. Or, have you defined them sufficiently in the write-up so that the reader has a precise understanding of what you are trying to convey?

The PascalGT software (included) is used for all of the projects. Documentation is included. The best way to get started with PascalGT is just to sit down and “play” with it for awhile. As you work the projects feel free to experiment a bit along the way. Ask your own questions and follow up with an investigation. You may very well soon start making your own discoveries!

PascGalois Project 1

Pattern Recognition and Pascal's Triangle Modulo n

You may be familiar with Pascal's triangle and its construction using the binomial coefficients $\binom{n}{k} = \frac{n!}{k!(n-k)!}$:

$$\begin{array}{ccccccc} & & & & \binom{0}{0} & & \\ & & & & \binom{1}{0} & & \binom{1}{1} \\ & & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} \\ \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} \\ & & \vdots & & \vdots & & \\ & & & & & & \end{array}$$

Using the fact that $\binom{n}{0} = \binom{n}{n} = 1$ for all $n \geq 0$, we see that Pascal's triangle has 1's down both sides of the triangle. Upon computing the rest of the entries we obtain the following:

$$\begin{array}{ccccccc} & & & & 1 & & \\ & & & & 1 & & 1 \\ & & & 1 & 2 & & 1 \\ & & 1 & 3 & 3 & & 1 \\ 1 & & 4 & 6 & 4 & & 1 \\ & & \vdots & & \vdots & & \end{array}$$

It is clear that any entry in the interior of the triangle is the sum of the two entries above it. This follows from the identity $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$. You may also have noticed that each row is symmetric, that is,

a palindrome. This follows from the fact that $\binom{n}{k} = \binom{n}{n-k}$. The binomial theorem also states that the n^{th} row of Pascal's triangle gives the coefficients of the expansion for $(a+b)^n$:

$$(a+b)^n = \binom{n}{0} a^n + \binom{n}{1} a^{n-1} b + \binom{n}{2} a^{n-2} b^2 + \cdots + \binom{n}{n-1} a b^{n-1} + \binom{n}{n} b^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k}.$$

In this exercise we will study a modified version of this triangle called Pascal's triangle modulo n . Pick an integer $n \geq 2$. Then reduce each entry of the triangle modulo n . Note that any entry in the interior of this new triangle is the sum mod n of the two entries directly above it. Also note that some very interesting patterns emerge in this revised triangle. Of course these patterns depend upon which positive integer you picked initially. Since this construction can be time consuming to do by hand, use the program PascalGT to do it for you. PascalGT assigns to each of the values $0, 1, \dots, n-1$ a unique color. See the PascalGT section in the Appendix for instructions on how to draw these triangles.

Exercises:

1. Construct Pascal's triangle modulo 2. What general pattern do you observe?
2. Repeat this for triangles modulo n , $n = 3, 4$, and 5. Compare all four triangles constructed thus far. Does any one of them have a quality or qualities that seem different from the other 3?
3. Experiment with Pascal's triangle mod n for $n = 6, 7, 8, 9$, and 10. How does n being prime or composite affect the appearance of the corresponding triangle?

Now we are going to interpret these triangles in terms of abstract algebra using the machinery of group theory. Recall that addition mod n is the group multiplication for the cyclic group Z_n . Think about the subgroups of Z_n , in particular for the values of n listed above. How many subgroups does Z_n have? List them for $n = 2, 3, \dots, 10$.

You may also be familiar with the *order* of a group element. If G is a finite group and $a \in G$, then the order of a is the least positive integer n such that

$a^n = e$, where e is the identity of G . What can you say about the orders of elements from Z_n ? List all the possible orders for $n = 2, 3, \dots, 10$.

4. Is there any correlation between the lattice of subgroups of Z_n and the complexity of the corresponding mod n triangle? How does this relate to the factorization of n into prime factors? (*Hint*: Consider the number of relatively prime factors for each factorization).

5. By now you have studied the concept of *closure* and how each subgroup of a given group is a closed subset of that group under the group operation. Look for regions in the mod n triangles that only contain elements from a given subgroup. What is the shape of these regions? How does the concept of closure explain this shape?

6. Consider the triangles for Z_p where $p < 10$ is a prime. What can you say about the orders of the nonzero elements in these groups? Notice that the corresponding triangles have an outward growing self-similarity analogous to the inward self-similarity found in many fractals. How many rows must you “zoom out” to see this self-similarity in each triangle? Can you make a conjecture relating this row scaling factor to the orders of the non-zero group elements?

Note: If you are unfamiliar with fractals and the concept of self-similarity, your instructor may provide a brief overview or appropriate references.

PascGalois Project 2

Pattern Recognition Using Dihedral Groups

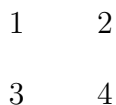
We are now ready to generalize the construction for Pascal’s triangle mod n . Motivated by the fact that addition mod n is the group multiplication for the cyclic group Z_n , we now take any finite group G and let $a, b \in G$. A *PascGalois triangle* is formed by placing a down the left side of an equilateral triangle and b down the right. An entry in the interior of the triangle is determined by multiplying the two entries above it using the group multiplication. Of course, if G is nonabelian then one must specify a left or right multiplication. We denote this PascGalois triangle by (P_G, a, b) . When a and b are clear from context we denote it simply by P_G . Note that Pascal’s triangle mod n is $(P_{Z_n}, 1, 1)$. We will denote this simply P_{Z_n} . The construction of (P_G, a, b) is the following:

$$\begin{array}{ccccccc}
 & & & & a & & & & \\
 & & & & a & & b & & \\
 & & & a & ab & & b & & \\
 & a & & a^2b & ab^2 & & b & & \\
 a & & a^3b & & a^2bab^2 & & ab^3 & & b \\
 & & \vdots & & & & \vdots & &
 \end{array}$$

Note that the a on top of the triangle is arbitrary and only appears for aesthetics. As a mathematical structure, P_G really begins at the second row. The word PascGalois comes from “splicing” Pascal with Galois. Galois (1811-1832) pioneered the study of groups while working with the theory of equations. Like Pascal’s triangle mod n , PascGalois triangles can have many interesting patterns and self-similar properties. One of the goal’s of this project, and the subsequent ones as well, is to understand these patterns in terms of group structure. Hopefully, studying the plethora of patterns within these triangles will provide you with a mechanism to visualize many of the fundamental concepts from abstract algebra.

In this project we will consider the dihedral groups D_n , $n \geq 3$ and introduce the concept of a p -group. Dihedral groups are the symmetry groups

of regular polygons. D_3 is the symmetry group of an equilateral triangle, D_4 is the symmetry group of a square, D_5 is the symmetry group of a regular pentagon, and so on. If M is an n sided regular polygon, then M has n rotational symmetries and n reflectional symmetries. Hence the corresponding symmetry group has $2n$ elements. For example, let us consider the symmetry group of a square with corners labeled 1,2,3 and 4:



By turning the square counterclockwise, we see that there are 4 rotational symmetries: 0° , 90° , 180° , and 270° . Of course 360° , 450° , ... are also symmetries, but they are equivalent to one of the four already listed. Note that we could also have done our rotations clockwise. We will use the convention that all the rotations will be counterclockwise since each clockwise rotation is equivalent to a counterclockwise rotation. Can you see why?

We also see that there are four axes for reflectional symmetries. There is a vertical axis that bisects the square. Reflecting about this axis interchanges corners 1 and 2 and also 3 and 4. There is an analogous horizontal axis. Reflection about it interchanges 1 and 3 and also 2 and 4. Finally, there are two axes passing through the diagonals of the square. One passed through corners 1 and 4. Reflection about this axis leaves corners 1 and 4 fixed but interchanges 2 and 3. You should be able to describe the last axis at this point.

We say a group G is a **p -group** if $|G| = p^n$ for some prime p . An equivalent definition is that G is a p -group if, given any $a \in G$, $o(a) = p^m$ for some nonnegative integer m . Here p is a fixed prime. For example the cyclic groups Z_{17} , Z_4 , Z_{27} , and Z_{125} are all p -groups. However, Z_6 is not a p -group. Why? Before continuing to the problems below answer the following:

Question: D_n is a p -group if and only if _____.

Exercises:

For the first few problems we will consider the dihedral groups D_3 and D_4 .

1. Using PascalGT, draw the first 16 and 32 rows of (P_{D_3}, μ, σ) where σ is the 120° rotation and μ is a reflection. Note that this is the default setting in PascalGT (this corresponds to generators 1 and 3) so you do not need to change the generators once you have chosen D_3 as your group. Do you see any patterns? How do these images compare with the Pascal's triangle mod n pictures?
2. Now draw the first 64, 128, 256, and 512 rows of the same triangle. As you add more rows, what happens to the corresponding image? Are certain group elements "clumping together" in the triangles. If so, how does this relate to group structure? closure? If you have trouble seeing any patterns, try changing the color scheme (see PascalGT instructions - Color Schemes 5 and 14 may be good choices). Note: Later when we study quotient groups, it will be much easier to understand this triangle.
3. Repeat Exercises 1 and 2 for the D_4 triangle. Compare and contrast the images you see with the D_3 images. Can you make a conjecture regarding the differences between these two images?
4. Draw the first 64 and 128 rows of the D_5 , D_6 , D_7 , and D_8 triangles. Compare and contrast the qualitative properties of these images. Does your conjecture in the previous exercise still seem to be true?
5. Earlier you should have given a necessary and sufficient condition for D_n to be a p -group. Which of the dihedral groups D_n , $3 \leq n \leq 8$, are p -groups? Does this seem to affect the appearance of the corresponding triangles?
6. Recall that two integers r and s are relatively prime if $\gcd(r, s) = 1$. Which of the dihedral groups in Exercise 5 have at least one pair of non-identity elements whose orders are relatively prime and which do not? Look at the subgroup triangles for these six PascGalois triangles. Is there any correlation between the existence of group elements of relatively prime order and the appearance of the subgroup triangles?

PascGalois Project 3

Building A Group With PascGalois Triangles

In this project you will build a group out of the PascGalois triangles. So far these triangles have been complicated structures with a plethora of internal detail. However, for the first three exercises you will think of each triangle as a mere “point” in a more abstract structure. The addition of these triangles will then be visualized using the superimpose option of PascalGT. This project should help you see that mathematical structures can often have many “layers”. That is, a structure can have intricate internal detail when viewed from one perspective, but then appear point-like or atomic when viewed from another. Although PascGalois triangles are complex when viewed internally, they will appear to be point-like in the group you are about to form from them. Understanding the different layers of abstraction for various structures is one of the key goals of mathematics. It can also be a very powerful tool. This way of thinking is essential for every mathematics major.

Let G be any group and consider $\mathcal{G} = \{P_{a,b} \mid a, b \in G\}$. \mathcal{G} can be thought of as the set of all possible PascGalois triangles that can be generated by the group G . In this lesson we will see that \mathcal{G} can be given an algebraic structure of its own. If $P_{a,b}, P_{c,d} \in \mathcal{G}$ we can “multiply” $P_{a,b}$ and $P_{c,d}$ by defining $P_{a,b} * P_{c,d} \triangleq P_{ac,bd}$ where ac and bd are computed using the group operation of G .

Exercises:

1. Prove that under the operation $*$ defined above, \mathcal{G} forms a group. If G is abelian, show that \mathcal{G} is also. You need to check that:
 - i) $*$ is a binary operation on \mathcal{G}
 - ii) $*$ is associative
 - iii) there is an identity element for \mathcal{G} under $*$
 - iv) given any $P_{a,b} \in \mathcal{G}$, there exists an inverse $(P_{a,b})^{-1} \in \mathcal{G}$
 - v) $P_{a,b} * P_{c,d} = P_{c,d} * P_{a,b}$ for all $a, b, c, d \in G$, when G is abelian.
2. If G is finite, what can you say about the order of \mathcal{G} ?
3. Prove that $\mathcal{G} \cong G \times G$.

4. How does the internal structure of each triangle (for example, self-similarity, level of organization, etc.) affect your solutions for Exercises 1, 2, and 3?

5. We have defined our operation on \mathcal{G} in terms of the generators down the sides of the triangle. Now we will investigate what would happen if we defined a superposition operation \otimes where $P_{a,b} \otimes P_{c,d}$ is defined to be the triangle obtained by multiplying corresponding entries of $P_{a,b}$ and $P_{c,d}$ using the group operation of G . That is, if $g \in G$ is in the i^{th} row and j^{th} column of $P_{a,b}$ and $h \in G$ is in the i^{th} row and j^{th} column of $P_{c,d}$, then $gh \in G$ is in the i^{th} row and j^{th} column of $P_{a,b} \otimes P_{c,d}$. We need to be careful with this definition. First of all, is \otimes well-defined? In other words, is $P_{a,b} \otimes P_{c,d}$ necessarily an element of \mathcal{G} ? If not, then we do not have a binary operation. Even if \otimes is a binary operation, is it the same as $*$? The purpose of the next three exercises is to answer these questions.

Draw $P_{1,2}$ and $P_{2,4}$ using Z_5 group multiplication. Use the Superimpose option in PascalGT to add mod 5 corresponding entries of the two triangles and then draw the result. Now, under our definition of addition for these triangles, $P_{1,2} * P_{2,4} = P_{3,1}$. Next have the program draw $P_{3,1}$. You should see that the $P_{3,1}$ triangle is the same as the triangle you obtained by superimposing the $P_{1,2}$ and $P_{2,4}$ triangles. Try this for other combinations as well. That is, superimpose other PascGalois triangles constructed over Z_5 .

6. Prove that what occurred in the previous exercise happens in general for any abelian group G and $a, b, c, d \in G$. That is, show that by taking the n^{th} row of $P_{a,b}$ to the n^{th} row of $P_{c,d}$ produces the n^{th} row of $P_{ac,bd}$. What can you say about $*$ and \otimes in this case?

Hint: For any abelian group G and $a, b \in G$, the n^{th} row of (P_G, a, b) is

$${}_a \binom{n}{1} {}_b \binom{n}{0} \quad {}_a \binom{n}{2} {}_b \binom{n}{1} \quad \dots \quad {}_a \binom{n}{n-1} {}_b \binom{n}{n-2} \quad {}_a \binom{n}{n} {}_b \binom{n}{n-1} \quad {}_b \cdot$$

7. Now draw the D_4 triangles with generators 1,4 and then 2,5. Note that 1 and 2 (as PascalGT labels them) are rotations of 90 and 180 degrees, respectively. Use the Superimpose option in PascalGT to superimpose these two triangles. Is the superimposed triangle the same as $(P_{D_4}, 1 \circ 2, 4 \circ 5)$?

If the two triangles are the same, explain why. If they are different, explain why superposition for D_4 triangles does not work like it does for triangles generated by Z_n . What can you say about \otimes in this case?

PascGalois Project 4

Direct Products and Automorphisms

Consider Pascal's triangle mod n . We have already observed that it is symmetric. This follows from the fact that Pascal's triangle is symmetric.

Question: What property of binomial coefficients causes this symmetry?

Another way we can interpret this symmetry is that reflecting the triangle about its central vertical axis induces the identity map $\iota : Z_n \rightarrow Z_n$ given by $\iota(m) = m$ for every $m \in Z_n$. That is, this reflection takes every color in the triangle to itself. Note that the identity map gives a (trivial) automorphism of Z_n . More generally, let G be *any* abelian group and $a, b \in G$. If we place a down the left side of the triangle and b down the right, then the group multiplication induces the following:

$$\begin{array}{ccccccc}
 & & & & a & & \\
 & & & & a & & b \\
 & & & & a & & ab & & b \\
 & & & & a & & a^2b & & ab^2 & & b \\
 & & & & \vdots & & & & & & \\
 a & a & \binom{n}{1}_b & \binom{n}{0} & a & \binom{n}{2}_b & \binom{n}{1} & \dots & a & \binom{n}{n-1}_b & \binom{n}{n-2} & a & \binom{n}{n}_b & \binom{n}{n-1} & b \\
 & & & & \vdots & & & & & & & & & & \\
 & & & & \vdots & & & & & & & & & &
 \end{array}$$

Let V denote the central vertical axis that bisects P_G . If we reflect P_G about V then $a \longleftrightarrow b$ and

$$\begin{array}{ccc}
 \binom{n}{1}_b \binom{n}{0} & \longleftrightarrow & \binom{n}{n}_a \binom{n}{n-1} \\
 \binom{n}{2}_b \binom{n}{1} & \longleftrightarrow & \binom{n}{n-1}_a \binom{n}{n-2} \\
 \vdots & & \vdots \\
 \binom{n}{j}_b \binom{n}{j-1} & \longleftrightarrow & \binom{n}{n-j+1}_a \binom{n}{j}
 \end{array}$$

Using the fact that $\binom{n}{k} = \binom{n}{n-k}$, we have $a \binom{n}{n-j+1} = a \binom{n}{j-1}$.

So in general $a \binom{n}{j} \binom{n}{j-1} \longleftrightarrow a \binom{n}{j-1} \binom{n}{j}$. In other words, reflection about V just interchanges the exponents of a and b .

In this lesson we will investigate what happens when $(P_{Z_n \times Z_m}, a, b)$, $a = (0, 1)$ and $b = (1, 0)$, is reflected about its central axis. Note that $Z_n \times Z_m$ is abelian so we can use the results from the previous paragraph.

Exercises:

1. Consider $(P_{Z_2 \times Z_2}, a, b)$, $a = (0, 1)$ and $b = (1, 0)$. Does reflecting this triangle induce a function $\phi : Z_2 \times Z_2 \rightarrow Z_2 \times Z_2$? That is, can one color in two different locations of the triangle ever be sent to two different colors? (Recall that a function must have a unique output for a given input). If a function is not induced can you say why? If a function is induced, is there anything special about this map (think in terms of preserving algebraic structure)?

2. Consider $(P_{Z_2 \times Z_3}, a, b)$, $a = (0, 1)$ and $b = (1, 0)$. Does reflecting this triangle induce a function $\phi : Z_2 \times Z_3 \rightarrow Z_2 \times Z_3$? If a function is not induced can you say why? If a function is induced, is there anything special about this map?

3. Now let us generalize the previous Exercises to $G = Z_n \times Z_m$, where $n, m \geq 2$. Reflect $(P_G, (0, 1), (1, 0))$ about its central axis for various values of m and n . Try several combinations of n and m until you understand how these values affect the triangle. Give a conjecture as to when this reflection induces a function $\phi : Z_n \times Z_m \rightarrow Z_n \times Z_m$? Note your answer will depend on the choices for n and m . When a set map is obtained, what special properties do you think it has?

4. (Optional) Give a proof for your conjecture(s) in the previous exercise.

You may want to use the result discussed above that $a \binom{n}{j} \binom{n}{j-1} \longleftrightarrow a \binom{n}{j-1} \binom{n}{j}$ for any a, b from an abelian group G .

PascGalois Project 5

Quotient Groups 1

One of the most important results often covered in a first semester undergraduate algebra course is the Fundamental Homomorphism Theorem which establishes the relationship between homomorphic images and quotient groups. To understand this result, you must be comfortable with both homomorphic images and quotient groups. However, quotient groups can often be a difficult concept to understand. Building groups out of cosets (which are equivalence classes defined using a normal subgroup) can be unsettling when first studied. The purpose of this project is to visualize the coset identifications created when constructing a quotient group. Hopefully seeing graphical representations of these identifications will make it easier for you to see these equivalence classes as points in a quotient group. In Project 3 you treated a certain set of PascGalois triangles as points in a group. The internal detail of the triangles was ignored for this level of abstraction. The same idea applies to quotient groups. Even though each coset has internal structure (recall that cosets are subsets that partition the group), it is treated as a point in the quotient group.

Exercises:

1. Construct Pascal's triangle mod 10 with 64 rows, 125 rows, and 128 rows. Study the patterns you see very carefully. Notice that there seem to be "competing themes" within the triangle. That is, there appear to be two distinct patterns, one superimposed on top of the other. Based on the colors you see and the group elements to which they correspond, can you predict what may be causing this visual effect?

Note that $\{[0],[5]\}$ is a subgroup of Z_{10} . Since every subgroup of an abelian group is normal, we know that coset addition is well-defined for $Z_{10}/\{[0],[5]\}$ and, hence, it is a group. Certainly $Z_{10}/\{[0],[5]\} \cong Z_5$. The cosets are $\{[0],[5]\}$, $\{[1],[6]\}$, $\{[2],[7]\}$, $\{[3],[8]\}$, and $\{[4],[9]\}$. Which coset acts as the identity for the quotient group? Which coset(s) generate the quotient group?

2. Go back to your computer drawing of Pascal's triangle mod 10. Using the color subsets options in PascalGT, re-color the elements of Z_{10} by identifying elements in a common coset with the same color. You will need 5 distinct

colors – the first color for $\{[0], [5]\}$, the second for $\{[1], [6]\}$, and soon. Re-draw the triangle with these color identifications. What do you see? Can you explain the observed pattern based on group structure and quotient groups?

3. Repeat the previous exercise with the subgroup $\{[0], [2], [4], [6], [8]\}$. You should determine what the cosets are and what the quotient group is isomorphic to. Then repeat the color identification by cosets. Re-draw the triangle and then answer the questions from the last exercise using your new image.

4. Using your answers for the previous exercises, explain how the subgroup lattice of Z_{10} relates to the appearance of Pascal's triangle mod 10.

PascGalois Project 6

Quotient Groups 2

This lesson is a continuation of the work you did with quotient groups in Project 5. There you considered chains of subgroups of the form $G \geq H \geq \{e\}$, where e is the identity of G . Note that chains of this form have length 2. You may wonder what happens if one considers longer chains of subgroups of the form $G \geq H_1 \geq H_2 \geq \cdots \geq H_n \geq \{e\}$. We will consider such chains in this lesson. Note that if H_{i+1} is a normal subgroup of H_i for $i = 1, \dots, n - 1$ and H_1 is a normal subgroup of G , then we may construct quotient groups $G/H_1, H_1/H_2, \dots, H_{n-1}/H_n$, and $H_n/\{e\} \cong H_n$. In this case the chain is called a *subnormal series* and written $G \triangleright H_1 \triangleright H_2 \triangleright \cdots \triangleright H_n \triangleright \{e\}$ (in general, $H \triangleleft G$ means H is a normal subgroup of G).

For our first example, we will consider the subnormal series $Z_8 \triangleright \{0, 2, 4, 6\} \triangleright \{0, 4\} \triangleright \{0\}$ (since Z_8 is abelian, we know that each subgroup in the chain is a normal subgroup of the proceeding group).

Exercises:

1. Using PascalGT draw the first 64 rows of Pascal's triangle mod 8. Look at the image you just created. Find the $\{0, 2, 4, 6\}$ subgroup triangles in your image. Next consider the quotient group $Z_8/\{0, 2, 4, 6\}$. Determine the cosets of this quotient. Using the color options in PascalGT, re-color the elements of Z_8 by identifying elements in a common coset with the same color. You will need a distinct color for each coset. Redraw the triangle and describe the picture that you obtain.

2. Go back to the $\{0, 2, 4, 6\}$ subgroup triangles that you identified in Exercise 1. Identify the $\{0, 4\}$ subgroup triangles *within* the $\{0, 2, 4, 6\}$ subgroup triangles. Determine the cosets for $\{0, 2, 4, 6\}/\{0, 4\}$. Using the *Color Subsets* options in PascalGT, re-color the elements of a $\{0, 2, 4, 6\}$ subgroup triangle by identifying elements in a common coset with the same color. Re-draw the subgroup triangle and describe what you see.

The next example we will consider is the dihedral group D_4 , the symmetry group of a square. You may want to review Project 2 regarding dihedral groups. Let us denote the corners of the square as follows:

1

2

3

4

Then the elements of D_4 in cycle notation are as follows:

$$r_0 = (1) \quad r_1 = (1342) \quad r_2 = (14)(23) \quad r_3 = (1243)$$

$$\mu_1 = (12)(34) \quad \mu_2 = (13)(24) \quad \mu_3 = (23) \quad \mu_4 = (14)$$

Certainly $r_0, r_1, r_2,$ and r_3 are counterclockwise rotations of $0^\circ, 90^\circ, 180^\circ,$ and $270^\circ,$ respectively. Then r_0 is the identity of D_4 . The last four elements correspond to the reflectional symmetries of the square.

3. Using PascalGT draw the first 64 rows of $(P_{D_4}, 4, 1)$. Note that the program labels the rotations (from smallest to largest) as 0, 1, 2, and 3. Likewise the reflections are 4, 5, 6, and 7. Now look at the image you just created. Find the rotational subgroup triangles. Caution: It may appear that the subgroup triangles only contain 2 of the 4 rotations. However, look more closely. There should be some subgroup triangles containing all 4 rotations. Describe how each of the 4 rotations is distributed within the subgroup triangle.

4. Note that $H_1 = \{r_0, r_1, r_2, r_3\}$ is a normal subgroup of D_4 (why?). So D_4/H_1 is a quotient group. What is the order of this group? What are the cosets? Which coset acts as the identity? Using the color options in PascalGT, re-color the elements of D_4 by identifying elements in a common coset with the same color. You will need a distinct color for each coset. Redraw the triangle and describe the picture that you obtain. Have you seen it before?

5. Explain why the chain $D_4 \triangleright H_1 \triangleright H_2 \triangleright \{r_0\}$, where $H_2 = \{r_0, r_2\}$, is a subnormal series. We have already considered the quotient D_4/H_1 in the previous exercise. Go back to the H_1 -subgroup triangles that you identified in Exercise 1. Identify the H_2 -subgroup triangles *within* the H_1 -subgroup triangles. Determine the cosets for H_1/H_2 . Using the color options in PascalGT, re-color the elements of H_1 by identifying elements in a common coset

with the same color. Re-draw the triangle and describe what you see. Does anything occur with this subnormal series and corresponding triangle that did not occur with the Z_8 case from Exercises 1 and 2?

6. Consider the subgroup $H = \{r_0, r_2\}$ of D_4 . You should check that this is indeed a normal subgroup and that $D_4 \triangleright H \triangleright \{r_0\}$ is a subnormal series (note that $H = \{0, 2\}$ in the notation of the program). Determine the cosets of D_4/H . This quotient is isomorphic to a group you have seen before. Which group is it? Using the color options in PascalGT, re-color the elements of D_4 by identifying elements in a common coset with the same color. Re-draw the triangle and describe what you see. Is this a triangle you have seen before?

Now consider D_8 , the symmetry group of a regular octagon.

7. Find a subnormal series $D_8 \triangleright H_1 \triangleright H_2 \triangleright H_3 \triangleright \{r_0\}$ where $|H_1| = 8$, $|H_2| = 4$, and $|H_3| = 2$. Perform the same type of analysis that you did for D_4 in Exercises 3, 4 and 5 on this subnormal series. Do you obtain similar results?

8. Can you find a normal subgroup $H \triangleleft D_8$ such that $D_8/H \cong Z_2 \times Z_2$? If so, redraw the D_8 triangle by identifying colors according to cosets so that your image looks like the Klien-4 triangle.

PascGalois Project

Self-Similarity and the Klein-4 Group

In this section we will characterize the self-similar growth structure of $(P_{Z_2 \times Z_2}, a, b)$ where $a = (0, 1)$ is placed down the left side of the triangle and $b = (1, 0)$ is placed down the right. The self-similarity studied in this and the earlier lessons is different from that of fractal geometry. Instead of “zooming in” to see copies of a pattern, we “zoom out”. Now, you may have already studied $Z_2 \times Z_2$ under a different name - the Klein-4 group. Both groups are non-cyclic of order 4 and, hence, are isomorphic. Before starting the exercises you should construct the group table for the Klein-4 group and write down its subgroup lattice.

Remark: *PascalGT* uses the notation $0 = (0,0)$, $1 = (1,0)$, $2 = (0,1)$, and $3 = (1,1)$ to represent $Z_2 \times Z_2$.

Exercises:

1. Construct $(P_{Z_2 \times Z_2}, a, b)$ as described above. Try the first 4, 8, 16, 32, and 64 rows of the triangle. What is the dominant pattern emerging? What sub-triangles of the triangle are repeated at different scales?
2. Give a description of the first 2^n rows of $P_{Z_2 \times Z_2}$ that applies for *any* $n \geq 3$. Draw a diagram that provides the overall structure as part of your answer. This diagram is called a *growth* triangle - it reveals the self-similar structure of the PascGalois triangle. How does the group structure determine this growth triangle? Note: You should pay close attention to the boundaries of the sub-triangles as you try to understand the growth triangle in terms of the structure of the Klein-4 group. It may also help to write out the first 16 rows by hand to understand how the geometry of the growth triangle is determined by the group multiplication.
3. Determine the growth triangles for P_{Z_n} for $n = 2, 3, 4, 5$ and 6. Compare the patterns you see with the subgroup lattice of each group. Then compare the orders of the group elements for each group versus the number of rows needed to repeat the self-similar structure. Does the value of n , whether it is prime or composite, seem to affect the corresponding PascGalois triangle?

What about the number of relatively prime factors? (The answers to these last two questions are addressed in more detail in the projects on quotient groups).

PascGalois Project

Rings and Fields

So far we have focused on groups where only a single binary operation is used. However, there is a second classic algebraic structure that has two binary operations (usually denoted $+$ and \cdot) - namely a **ring** R . When just considering the first operation, $(R, +)$ is an abelian group. We always denote the identity element under addition by 0 . We also require that the ring multiplication \cdot is associative and R satisfies the distributive laws:

$$a(b + c) = ab + ac \qquad (a + b)c = ac + bc$$

for all $a, b, c \in R$. Examples of rings include the integers and real numbers (both under the standard addition and multiplication) and the set of all 2×2 matrices with integer entries (under matrix addition and matrix multiplication). Of course, these three examples are all infinite rings.

To see a finite example of a ring, consider the integers modulo n , i.e. $Z_n = \{0, 1, \dots, n - 1\}$ for some $n \geq 2$. We already know that (Z_n, \oplus) is an abelian group, where \oplus denotes addition mod n . To obtain a ring structure, we need a second binary operation. Let \odot denote multiplication mod n . For example, if $n = 12$ then $3 \odot 8 = 0$ and $2 \odot 9 = 6$. Since \odot is associative and the distributive laws hold for \oplus and \odot , it follows that (Z_n, \oplus, \odot) , which we will denote by Z_n , is a ring.

Like groups, there are special classes of rings that mathematicians are particularly interested in. For instance, a ring R is *commutative* if $a \cdot b = b \cdot a$ for all $a, b \in R$. Certainly the integers, real numbers, and Z_n all form commutative rings. The ring of 2×2 matrices under matrix addition and matrix multiplication does not form a commutative ring. Can you see why? A ring R is said to have unity if there exists some $1_R \in R$ such that $1_R \cdot a = a \cdot 1_R$ for all $a \in R$. Note that 1_R , if it exists, is the multiplicative identity element of R . To introduce the next class of rings we first need a definition.

A nonzero element $a \in R$, where R is commutative, is called a *zero divisor* for R if there exists some nonzero $b \in R$ satisfying $ab = 0$.

A commutative ring R with unity is called an *integral domain* if it contains no zero divisors. Note that both the integers and real numbers are integral domains. However, even though Z_n is a commutative ring with unity, it is an integral domain only for certain values of n .

Question: For which values of n does Z_n have zero divisors. (Note: this will determine precisely when Z_n is an integral domain).

Finally, we define a *field* to be a commutative ring where the nonzero elements form a group with respect to multiplication. Note that a field necessarily has a unity element 1_R . In addition, if a is a nonzero element from a field R , then a must have a multiplicative inverse $a^{-1} \in R$ that satisfies $a \cdot a^{-1} = a^{-1} \cdot a = 1_R$. Note that every field is an integral domain. Also, every finite integral domain is a field (both of these results should be in your text as theorems and/or exercises).

PascalGT allows the user to experiment with Z_n ring multiplication and other binary operations on a sufficiently small finite set. The Z_n ring multiplication triangles are drawn by placing two elements a and b of $Z_n \setminus \{0\}$ down the sides of the triangle (what will happen if 0 is placed down one of the sides?).

Exercises:

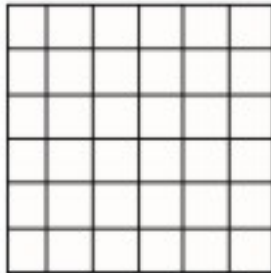
1. Construct triangles (P_{Z_n}, a, b) using various nonzero ring values $a, b \in Z_n$. Note that the operation here is multiplication mod n rather than addition mod n . Try $n = 3, 4, 5, 6, 7, 8$, and 9. Record what happens for each value of n and each combination $a, b \in Z_n$ that you try.
2. Which of your pictures contain elements that are zero divisors? How does the presence of zero divisors affect the corresponding triangles?
3. Describe what your triangles look like when n is prime versus when n is composite. Does this relate to the presence of zero divisors?
4. Do any of the pictures you obtain look similar to Pascal's triangle mod n ? Based on this, can you make a conjecture regarding which values of n make Z_n a field?
5. Construct a PascGalois triangle using Z_{15} ring multiplication with 2 down the left side of the triangle and 3 down the right. Give a description of

the resulting triangle. Does this example violate your conjecture from the previous exercise? Why or why not?

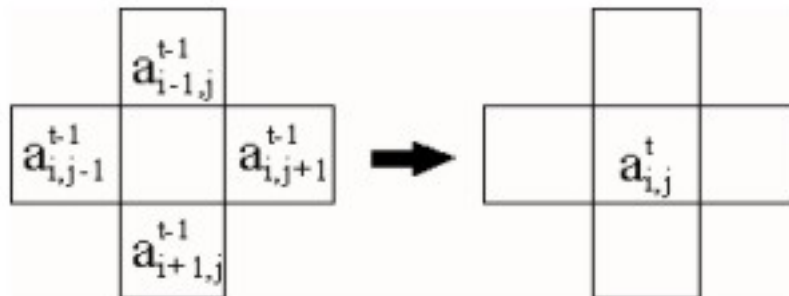
PascGalois Project

Two Dimensional Automata

In this project we will consider a finite two dimensional dynamical system generated using group multiplication from a finite group $(G, *)$. We begin with a finite $n \times n$ checkerboard grid:

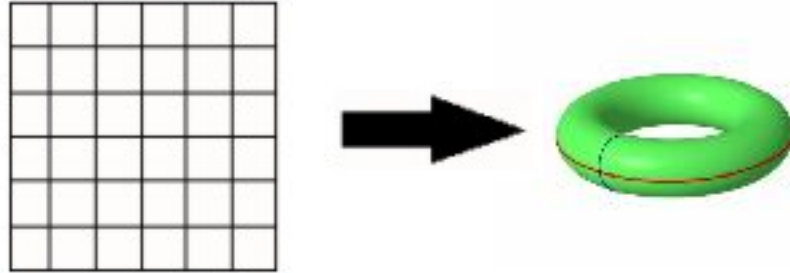


Each of the n^2 squares can be thought of as “cells” that, at any given time t , can hold some value from G . Let $a_{i,j}^t$ denote the contents of the cell in the i^{th} row and j^{th} column at time t . So $a_{i,j}^t \in G$. At time $t = 0$ the arrangement of cell values is considered the initial state (or initial condition) of the system. The grid is then updated in discrete time steps according to the local rule $a_{i,j}^t = a_{i,j-1}^{t-1} * a_{i-1,j}^{t-1} * a_{i,j+1}^{t-1} * a_{i+1,j}^{t-1}$:



The resulting dynamical system is an example of what is called a 2-D cellular automata. Note that this definition makes sense for the interior of the grid. However, there is a problem with the cells adjacent to the boundaries since those cells are not surrounded on all sides by neighboring cells. To fix this, we identify the top and bottom boundaries. Likewise we identify the two side boundaries. This causes all four corners to be identified to a single point.

The square grid is now wrapped into the surface of a torus so that every cell is surrounded by the same number of neighboring cells:



PascalGT will let you choose an initial state and then animate the time evolution of the system. In this project you will investigate how the choice of group, initial state, and grid size affects this evolution. Some of the exercises below are intentionally open-ended. Hopefully you will explore many examples and make some new conjectures of your own!

Exercises:

1. Using PascalGT select Z_4 as the group and then construct a 2-D automata using a 64×64 grid with initial state a one point square. This will place a 1 in one cell of the grid and zeros in all other cells. Animate the system until you understand the long-term behavior of the automata. Do all four group elements appear as cell values during the process? Does the automata ever enter a sequence of time frames where only a subset (or subgroup) of Z_4 appears as cell values?
2. Repeat Exercise 1 using a 65×65 grid. What are the similarities between the two evolutions? What are the differences? Try other square grid sizes of side length 32, 34, 128, and 130. Can you make any conjectures based on these animations? Do they tell you anything about sensitivity to initial conditions?
3. In the previous two exercises, cell values were taken from abelian groups. Now let us try a nonabelian example. Repeat Exercises 1 and 2 using the dihedral group D_4 . As before, note any similarities and differences that you find.
4. Experiment with different groups and grid sizes of your choice. You could try (but not necessarily) $Z_2 \times Z_2$, $Z_2 \times Z_3$, the quaternion group, D_5 , Z_6 , etc.

Make sure to use groups that have a variety of different structural properties. See what interesting conjectures you can find of your own. Can you prove (find a counter example) for any of them? Note: This is a great way to start an undergraduate research project!

Appendix: Brief Introduction to the PascalGT Software:

Welcome to PascalGT, a Windows-based program for drawing PascGalois triangles and hexagons, 1-D and 2-D automata and more. In this program, the cell values for automata generally correspond to group elements. Let's take a quick walk through the program and explore the basic features (this seven step tour can also be found in the quick start screen in the program's help files. That screen includes color images of what you should see.):

- 1) When you first start the program, you are in the Enter Data Window. Choose how you would like to enter the numbers into the group Multiplication Table using the Enter Group/Set Information in the Menu. Z_n , a cyclic group, with order of 12, for example, is a good start.
- 2) Choose Identity Element (any element added/multiplied times this number is itself -if you choose a preprogrammed group like Z_n then the identity element is 0) and Generators (Numbers down the side of a PascGalois Triangle)---Try 1,2. Note that in order to change the generators you must highlight the number in the box and type over it. Then click continue.
- 3) Now you are in the Image/Draw Window. To see the Triangle you have made, click Triangle from the PascGalois Menu button. To see what number each circle represents, Go into the Setup Conditions, move into the Choose color options list and choose Display current colors $n \leq 16$. (We choose this instead of $n > 16$ because our group order is 12 from above).
- 4) This triangle doesn't tell us too much. Let's look at it with more rows. Under Setup Conditions, Click on Choose number of rows. The default for changing is 125 rows which should work for us. (Watch out, the larger the rows, the longer it takes to draw it)
- 5) Click again on Triangle to display the larger triangle (Don't worry, the screen automatically clears). Do you see any patterns inside this triangle? If not, Click on Choose Color Scheme(Rotate 16 colors) inside Choose Color Options under Setup Conditions to rotate the colors in the triangle. Then redraw the triangle with the new colors. Try different color schemes numbers and watch how some patterns jump out at you.
- 6) Okay, you still can't see some patterns, but there are some tricks to help you. You can white out the infrequent Colors. Click on Reset Popular Color Values inside Choose Color Options. There will be a prompt to enter a 1 if you want to white out the other values: Enter a 1. Now, click on Triangle to see your new triangle. It may look blurry, so increase the number of rows to 513 . (Warning, this may take a long time on older computers.)
- 7) We've finally drawn a nice pretty triangle. Why don't we add a caption to our triangle using the save menu. Be sure the Point-Up triangle option is clicked. Let's

save it as a .bmp or .jpg using the save menu.

Congratulations, you have successfully used the PascalGT program. There are a number of other functions which the program can perform but not all of them will be necessary to complete the laboratory exercises you will be using this semester. For the remainder of this section, we will go over the things you will need to use for the laboratory exercises to follow. You are in the interesting situation of trying to learn to use a package that supports inquiry into a field in mathematics at the same time you are learning the terminology of that field. Therefore the first time you read through this document, many of the terms may be unfamiliar to you. Don't worry! You will know what they mean by the time you need to use them. Refer back to this section as needed throughout the term while you are completing the labs.

The *Enter Data* Window:

Choosing a group or set: In the *Enter Data* window you will generally be choosing from the sub-menu items under "*Enter Group/Set Information*" one of the preprogrammed groups. Those you will need are:

- Z_n : The cyclic group of order n . *This is also the choice you make in the first project for modulo n addition.*
- D_n : The dihedral group of order $2n$, the symmetry group of a regular n -gon.
- $Z_n \times Z_m$: The cross product of the cyclic group of order n with the cyclic group of order m .
- Z_n *Ring multiplication*: You will need this option for the *Rings and Fields* Project. As a set Z_n is the integers mod n . But this option uses multiplication mod n rather than addition as the operation. This option isn't a group multiplication (why?).

Group element identification: Although when you are introduced to these groups in class, the group elements may have various names and representations, for the purposes of the program, each group element is assigned an integer starting at 0 and continuing to one less than the order of the group. Thus the assignments for Z_n are 0, 1, 2, 3, . . . , $n-1$ (which does follow traditional notation). The assignments for D_n are 0, 1, 2, . . . , $2n-1$ and the assignments for $Z_n \times Z_m$ are 0, 1, 2, . . . , $nm-1$. Once you select a group from this list a multiplication table is displayed. If the order for your group is 16 or less the entire table is displayed; if the order of your group is more than 16 only part of the table is displayed. A good exercise in understanding what is going on would be to identify the symbols you are using to represent the elements of these groups with the numbers used by the program by examining the multiplication table. In all of the preprogrammed **groups** the number 0 corresponds to the identity.

System Generators and Identity: The last thing you need to do before clicking on continue to move to the *Draw Screen* is to identify the generators and the identity. For all of the preprogrammed groups the identity will be correctly entered for you (although the program will not keep you from changing this and some strange things may occur in some of your drawings if you do so). There are also default choices for the generators.

For the first two projects use the default settings. For Z_n the default setting is for both generators to be set to 1. For D_n , the default generators are a rotation through $360/n$ degrees and a reflection. For $Z_n \times Z_m$ the default generators are (1,0) and (0,1) which are numbered 1 and n in the program. For some of your labs you may wish to change these; you may do so either by entering the number corresponding to the elements you wish to use instead in the identity and generators frame on the enter data screen or by choosing the *Change Generators* option under *Setup Conditions* on the *Draw Screen*. These generators, referred to as “system generators” will be of interest to you primarily because the PascGalois triangles that will be drawn by the program are:

- $P_G(\text{generator2}, \text{generator1})$ or $P_G(a,b)$.

Once you are satisfied with the generators, click on continue or press <ctrl>G to load the draw window.

The Draw Window

The draw window is where you will produce images. There are four menu headings with which you will need to be concerned. They are:

Setup Conditions, PascGalois, Color Subsets and Save.

Setup Conditions: Under Setup Conditions the options you will need are found under:

- “Choose number of rows” this will change the number of rows that you will draw when you select from the PascGalois menu. You can also use the shortcut ctrl <R> to access this item. A dialogue box will appear where you can enter the number of rows, referred to hereafter as the *system number of rows*. Remember that drawing a large number of rows (more than 500) may take some time. If you do not exercise this option, the program defaults to 25 rows.
- “Choose color options.”
 - The most important item in this submenu for you is the “*Display Current Colors, n<=16*)” option which can also be accessed through the shortcut ctrl<D>. This will draw circles on the screen with the colors corresponding to each of the group elements, so long as the order of your group is ≤ 16 . Next to each circle will be a number. The number is the one which corresponds to the same group element as the color. The numbers used are the same as the numbers used in the enter data window for the multiplication table. If you want to look at a group with more than 16 elements use the “*Show Current Colors (n>16)*” or ctrl<S> option.
 - For the first five labs, it is not necessary that you use any of the options that change color assignments. However, if you are having difficulty “seeing” what you are supposed to see, it is possible that a change of color scheme might help. The easiest way to do this is to select the “Choose Color Scheme (rotate 16 colors)” option from this sub menu. Simply enter a number between 0 (default colors) and 15. This will change the assignments.
 - Eventually you may want to choose colors specifically. The easiest way to do this is by choosing the “set colors manually” or ctrl<C> option. When you choose this, a grid with 16 text boxes and an ok button will

appear next to 16 circles of different colors. Simply enter the number of the group elements next to the colors to which you wish them to correspond. Any elements left out will be colored white and will disappear against the white background.

- There are many other options for changing colors, feel free to play with them at your leisure to draw pleasing pictures. Consult the system help for descriptions of these options.
- Change Generators: Choose this option if you want to draw images using different generators than those you previously entered in the enter data window. You will be prompted for the group elements you want to use as the new generators. You must use the numbers corresponding to the group elements for the program to recognize them. This changes the “system generators.”
- Display group order: If you forget which group you are using or want to check on its order this will write to the screen the number of elements in the group (entered in the enter data screen) the program is currently using.

PascGalois: Under PascGalois you find the options that actually draw the images. *{If, when you choose from this menu, nothing gets drawn there are two possibilities. Most likely the program is taking a while because you have a large number of rows, in which case you cannot use any other menu options because the program is busy. The other possibility is that you have changed the colors so that everything is white. If nothing gets drawn and the program isn't still busy, try resetting to system colors from the menu: “Setup Conditions”, “Choose Color Options”, “reset to system colors” or by pressing the F1 key.}* For the labs you will need to use:

- *Triangle* $PG(a,b)$, which can also be activated by $\text{ctrl}<T>$, draws the PascGalois Triangle $P_G(a,b)$ where a and b are the generators you either entered on the enter data screen or changed to in the setup conditions. If your group is Z_n and you use the default generators, then this is Pascal's triangle mod n with each number represented by a color. Pressing $\text{ctrl}<D>$ will identify the color-number correspondences for you. The triangle has either 25 rows or the number of rows you entered in the setup options. Once the program has drawn the triangle you can use the mouse and click on a position in it and the program will display the row and column number of the position.
- *Superimposed Triangles:* This option, needed for the *Building a Group with PascGalois Triangles* project, prompts you for the numbers corresponding to two group elements you wish to use as a second set of generators. It then draws three triangles, one PascGalois Triangle using the system generators on the left, one PascGalois Triangle using the generators you just supplied on the right, and finally, below them, a triangle where each position is the color corresponding to the group product of the elements in the same position in the two triangles above it.

• In the course of your investigations you may also wish to look more closely at a portion of a triangle and thus use the PascGalois options:

- “*Bottom of Triangle*” prompts you for a row to begin on and draws the bottom portion of the triangle starting at the row you entered and ending at the row number of the current system number of rows.
- “*Subtriangle*” displays a dialogue box in which you enter the row and column number of the top vertex of the desired triangle and the row number in the PascGalois triangle that you wish to draw to (this needs to be larger than the row number of the vertex). When you press ok, if the numbers entered are legitimate, the program draws the portion of the PascGalois triangle that corresponds to the point up Subtriangle beginning at the vertex entered and ending at the row entered.
- “*Point Down Subtriangle*” prompts you for the beginning row and leftmost column of the triangle then prompts you for the width of the triangle. You must choose a width of greater than three and less than the difference between the beginning row and the beginning column so you need to choose a beginning column at least three less than your beginning row. Then the program draws the portion of the PascGalois triangle that constitutes the point down triangle starting at the given point with the given width.

Color Subsets: The items on the color subsets menu allow you to identify a number of group elements and color them the same or to select only some elements to color at all. These items alter the color identification that the program is using. It is best to reset to system colors (press F1 or see above) after using one of these to produce an image. Very strange colorings can occur if you use different options in this menu in sequence without resetting.

- *Select a few elements to color:* If you choose this option you will be prompted for the number of elements you want to color and the numbers corresponding to those elements. The system will then assign a color of white to all other elements so that when you draw images they will not show up.
- *Color a set of elements the same:* This allows you to assign to a group of elements the same color. It prompts you first for the number of elements you want to identify and then for the elements. You can repeat this if, for example you want to color 0, 2, and 4 one color and 1, 3, and 5 a different. You would choose this option, tell it you want 3 elements, then tell it the elements are 0,2, and 4. Then you would choose the option again and tell it you want 3 elements; then tell it the elements are 1,3, and 5. Try this with Z_6 . You can also use the shortcut $\text{ctrl}<I>$ to access this item.
- *D_n subsets:* If the group you are using is a dihedral group there are some preprogrammed identifications available to you. They should be self-explanatory from the submenu under D_n subsets.
- *$Z_n \times Z_m$ subsets:* There are also some preprogrammed identifications for $Z_n \times Z_m$ which should be self explanatory.

In order to **change the group** you are working with you need to exit the draw screen and choose new group from the menu which appears.

To **exit the program** you can either choose exit from the draw screen and from the new group screen or you can simply click on the x at the top right of the screen.

If your instructor assigns the *Two Dimensional Automata* Project, then there will also be some items from the “*2-D Automata*” menu which you will need to use. These options draw and animate cellular automata (c.a.’s) which are defined in the project. Since this project is more open ended than many of the others, descriptions for all of the options in this menu are included below, even though you may not need to use them all:

Enter Starting Values:

For a one point square: This will prompt you for the side length n of the c.a. and will then internally define an $n \times n$ c.a. with cell values of 0 (or the identity) for all but one center cell whose value will be 1 (or the group element that corresponds to 1 in the program).

For a five point square: This will prompt you for the side length n of the c.a. and will then internally define an $n \times n$ c.a. with cell values of 0 (or the identity) for all but one center cell whose value will be the right generator (1 or b) defined on the enter data screen and the four corner cells whose value will be the left generator (2 or a). Note that as the sides are linked, the four corners are actually all neighbors.

Randomly: This will prompt you for the number of columns (width) m and the number of rows (height) n of the c.a. and will then internally define and randomly assign cell values to an $m \times n$ c.a.

Enter non-zero values: This will prompt you for the number of columns (width) m and the number of rows (height) n of the c.a. and will then prompt you for the number of non-zero (or non-identity) entries, their positions and their values. As always you must use the integer identification that the program uses to assign group element values to the cells.

By Hand: This will prompt you for the number of columns (width) m and the number of rows (height) n of the c.a. and will then ask you for the value for each cell. This is recommended only for small c.a.’s. As always you must use the integer identification that the program uses to assign group element values to the cells.

From file: This option can be used to reset the automata to a state that you had previously saved.

Animate

Define Rule for Automata: The program defaults to the rule described in the project. You can, however, define other rules for advancing the c.a. This option will display a 3×3 grid of buttons representing a cell (in the center) and all of its neighbors. You can click on the buttons in the order in which you would want to multiply the cell values to update the automata and thus define the rule. If you

want to return to the default rule, press the reset button. Click done when you have finished.

Animate Step by step: This will draw the automata and display a button to advance it. Every time you click on the button, the program will advance the automata and draw the new state. Use this when you want to carefully watch the steps.

Animate n Times: This will prompt you for a value of n and compute the states as the automata advances n times and display the result.

Animate continuously: This activates three buttons labeled *stop it*, *faster* and *slower* and will display, update and display the automata continuously until you click on the *stop it* button. You can speed up or slow down the display by clicking on the *faster* or *slower* buttons.

Make a 3-D file: This will save the automata states to a file which could then be used as data for a program that you write to do a 3 dimensional version of the cellular automata with states piled up vertically on top of one another. This option is not required for the project.

Save the current state to file: This will save the current state of the automata to a file which you can then use to reset the automata using the **Enter Starting Values, By Hand** option above.

Draw and Save: This will prompt you for a number of times n to animate. Then it will draw n iterations of the automata and save each as a bitmap titled ani01.bmp , ani02.bmp . . . anin.bmp in the directory where the program was last pointing. In other words, it will save to the directory where the program itself is located unless you have saved or opened a file somewhere else, in which case it will save to that directory. The best thing to do is to save an image using the save options from the main menu before using this option. You can create animations using these images and a package like GIF Construction Set, which creates animated gifs or by writing a program to display them.

Counting and Shading: These menu options do not work very well but they shade the cells of the automata based on how often the cells take on values from the elements you identify.

Searching for steady state: Since the c.a.'s used in the program are finite, there are only finitely many possible states so that eventually the states must repeat and the c.a. will become periodic. These menu items are available to help you identify the periodic, steady state in which the c.a. eventually lands, the transient phase which occurs before the periodic cycle is reached, and their lengths in terms of the number of states included in each. (Note that for some automata, the steady state will simply be the state where all values are the identity or some

other constant state; in this case the length of the periodic cycle is one.):

Just Iterate: This option iterates the automata. It is used to try to get beyond the transient phase.

Find Period: This option saves the current state and iterates the automata until that state repeats. It will then report the length of the periodic cycle. If it fails, it will report that either the period is too long or the transient state had not been passed. A second try of this option will generally be successful if the period is not too long.

Save a State: Unlike the *Save the current state to file* option from the menu above, this option saves the current state internally and you can then reset to it using the next option or you can check to see if it recurs using the following option.

Reset to a Saved State: Reset to the state you last saved using the option above.

Check for recurrence of a saved state: If you have saved a state using the option before last, you can check to see if it recurs using this option. You will be prompted for the maximum number of iterations the program should perform if the state has not recurred. If the state recurs within that number of iterations, it will report back which iteration the state was found on. This is useful for finding long periods.

Looking for the transient? Use this option when you know the length of the periodic cycle to which the c.a. eventually evolves. You will be prompted for the length of the periodic cycle and then the program will test to see how long the c.a. iterates before reaching that cycle.