



PascGalois JE User's Manual

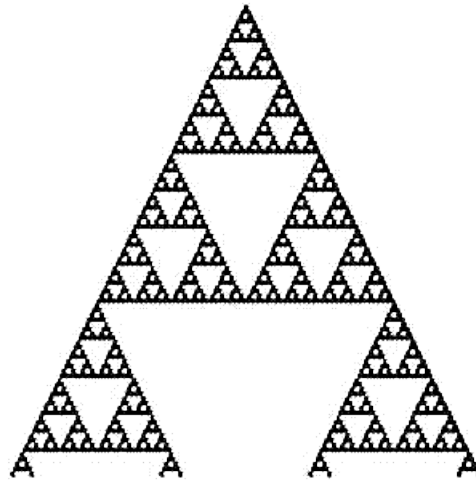
by Don Spickler

Contents

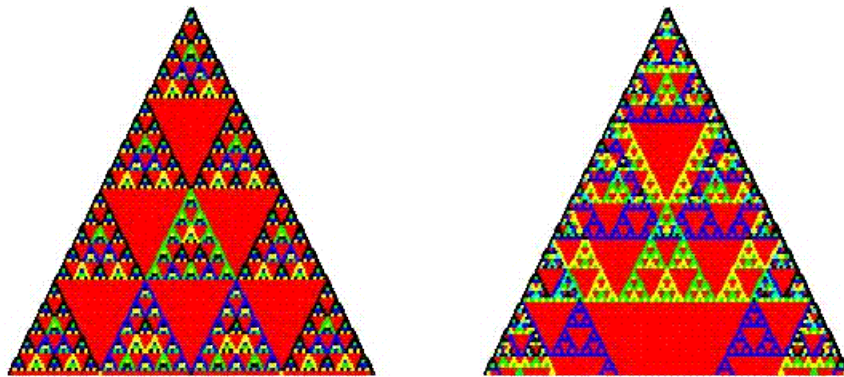
Introduction to the PascGalois Project	1
Introduction to the PascGalois JE Program	4
The Main Window	4
The Child Windows	4
License	13
Group Overview	14
Selecting a Group	14
User-Defined Structures	15
Advanced Mode	17
Supported Group Structures & Syntax	19
Supported Group Structures	19
Advanced Input Syntax	23
Cartesian Products	23
Quotients	23
1-D Cellular Automata	25
Overview	25
The Options Tab	29
The Seed Tab	31
The Update Rule Tab	32
The Image Tab	36
The Counts/Period Tab	46
2-D Cellular Automata	51
Overview	51
The Options Tab	53
The Seed Tab	56
The Update Rule Tab	57
The Image Tab	58
The Counts/Period Tab	68
3-D Viewers	73
POV-Ray Exporter	77
Group Calculator	84

Introduction to the PascGalois Project

PascGalois is a project devoted to the visualization of concepts in Abstract Algebra. Through the use of automata, students are able to visualize some of the more difficult concepts in group theory. The images produced by this program and the recoloring schemes that are possible allow the student to visualize group properties as well as structure. For example, the images can clearly display a group's subgroups, quotient groups and with a little effort even subnormal series. The idea behind the images is an old concept with a new twist. Most mathematics students have seen Pascal's triangle and possibly even Pascal's triangle (mod n) where the entries from 0 to $n - 1$ are all colored a different color. For example, Pascal's triangle mod 2 looks like

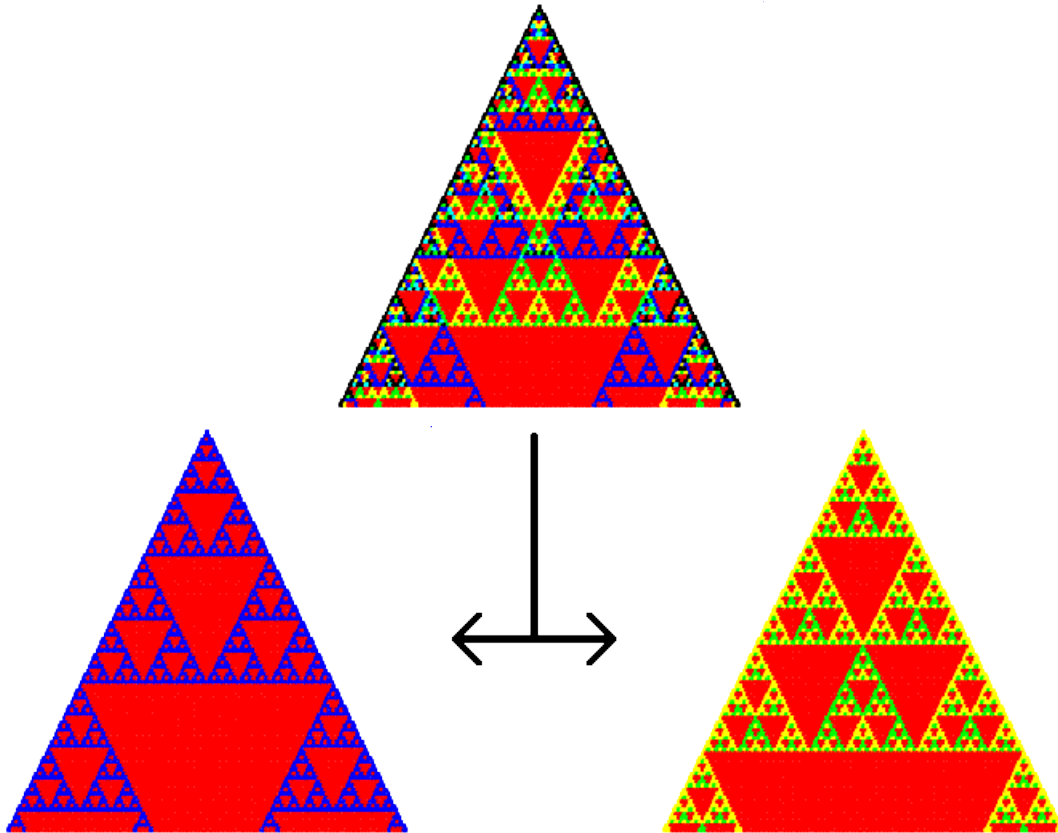


and the Pascal triangles mod 5 and mod 6 are, respectively,

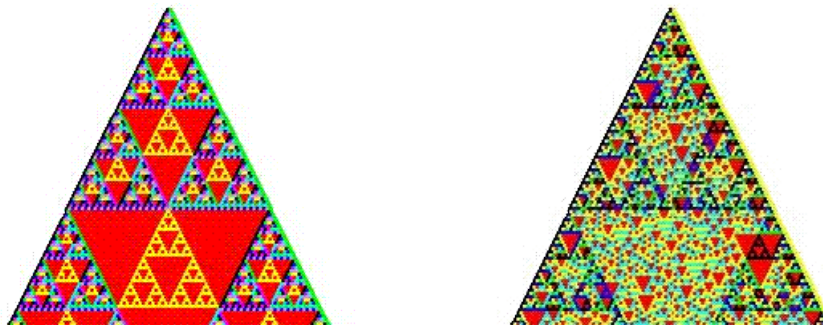


Notice in the triangle mod 6 there appears to be 2 superimposed images, one which is the mod 2 triangle and one that is the mod 3 triangle. These are, of course, the prime factors of 6 but they also

represent the non-trivial subgroups of Z_6 . Thinking in terms of subgroups, if we draw the mod 2 and mod 3 triangles we see that inside the structure of Z_6 we have things that look like Z_2 and Z_3 . Hence Z_6 has subgroups that are isomorphic to the groups Z_2 and Z_3 .



This is where the new twist comes in. Why not try the same investigation with other finite groups? Take $Z_n \times Z_m$, D_n or S_n , start with a seed row of elements from the group and construct the triangle in the same way as Pascal's triangle, except use the group's operation instead of addition. For example, triangles using the Quaternions and S_3 are below.



The current program is capable of graphing both one and two dimensional automata over finite group structures. It uses a group operation engine capable of calculating arbitrary products and

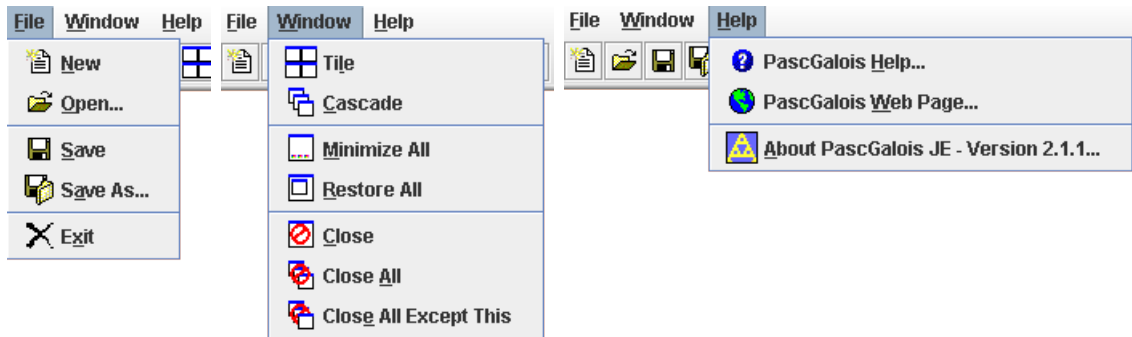
quotients of Z_n , Z_n (under multiplication), D_n , S_n , the Quaternions, Q_n (Generalized Quaternions) and C_n (Dicyclic groups). It also has the capability of fully integrating user-defined structures. The interface was built to facilitate zooming and positioning, coloring of subsets, subgroups and cosets in order to extract more information about the group's structure quickly. It also has facilities for saving and loading workspace files as well as the saving of images and color/element correspondences. The program supports drag and drop of colors between two or more windows and or running applications to facilitate the investigation of isomorphisms. This version also has a group calculator that allows the user to calculate single results as well as generate subgroups and cosets. This program is an ongoing project and will evolve as the research in abstract algebra visualization evolves. The current version was written in Java so that it is platform independent.

Introduction to the PascGalois JE Program

The PascGalois JE program has a multiple document interface with standard file and window operation facilities. Each of the internal (or child) windows contains all of the information and facilities needed to explore an automaton, as well as some other group theoretic concepts. Each of the options and facilities will be discussed in more detail in their own section. We will simply hit the highlights here to give you a feel for the program's capabilities.

The Main Window

The main window has very little functionality, its main purpose is to simply hold the child windows. There are three menu items in the main window, File, Window and Help. Each of the options in these submenus have a corresponding tool button on the toolbar.



The File menu has options to create a new child window, open a previously saved automaton, save the current automaton information to a file and to exit the program. Note that when you select Save or Save As the only automaton that is saved is the one in the currently selected child window.

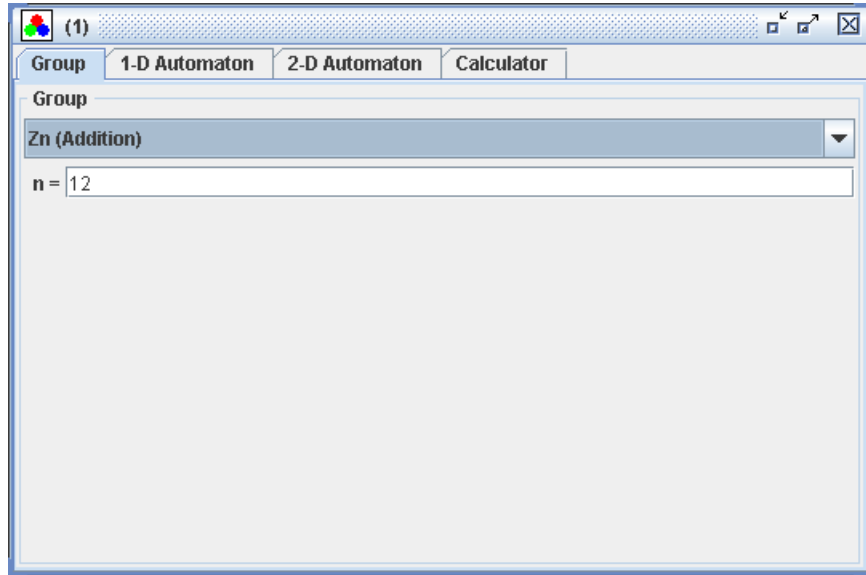
The Window menu has tools for tiling, cascading, minimizing and restoring child windows as well as closing the currently selected window, closing all of windows and closing all of the windows except the currently selected window.

The Help menu has options to open the help system and go to the PascGalois home page at (www.pascgalois.org) as well as an About screen. Note that the option to go to the PascGalois home page is only available if you are running this program on a Windows system, and it will attempt to open the default browser on your computer and connect to the Internet.

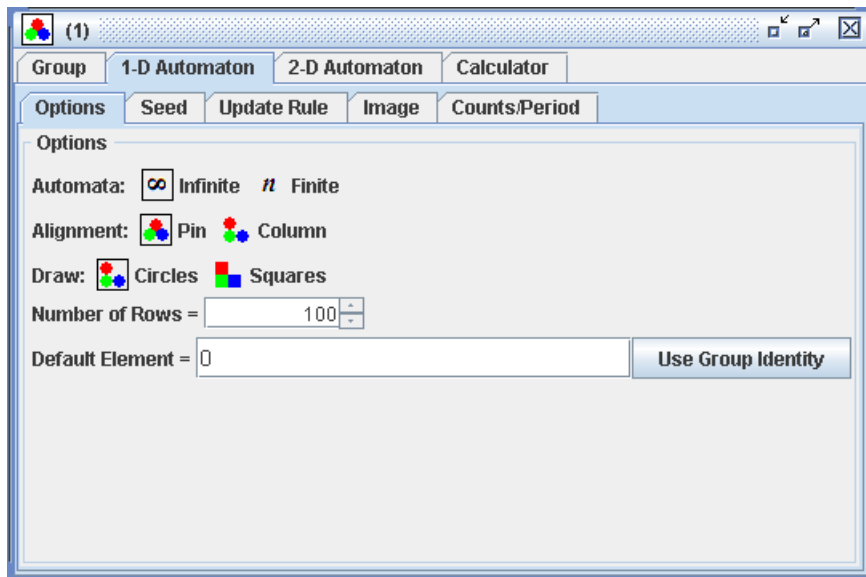
The Child Windows

The child window has four tabs: Group, 1-D Automata, 2-D Automata, and Calculator. The Group tab gives the user a simple but flexible way to select the group structure they wish to work with. The current program has the capability of working with Z_n , Z_n (under multiplication), D_n , S_n , the

Quaternions, Q_n (Generalized Quaternions) and C_n (Dicyclic groups). The user can also input their own structure by specifying an operation table. Furthermore, there is an advanced input option that allows the user to use arbitrary Cartesian products and quotient groups of any of the above structures. Details about group and element syntax can be found in the Supported Group Structures & Syntax section.

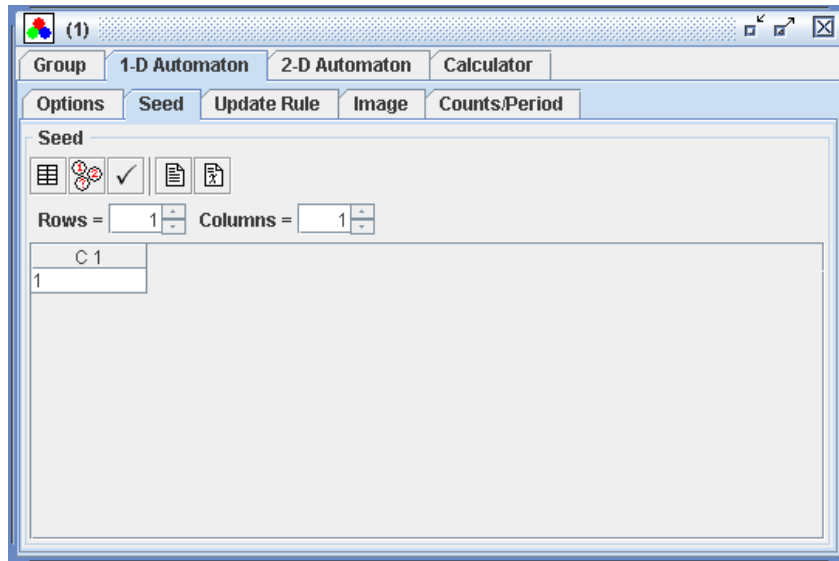


Under the 1-D Automata tab there are currently five other tabs which allow the user to input all of the data necessary to create and explore both finite and infinite one-dimensional automata. The Options tab allows the user to select between a finite or infinite automata, alignment of the image, the number of rows, the wrapping style and whether it draws circles or squares.

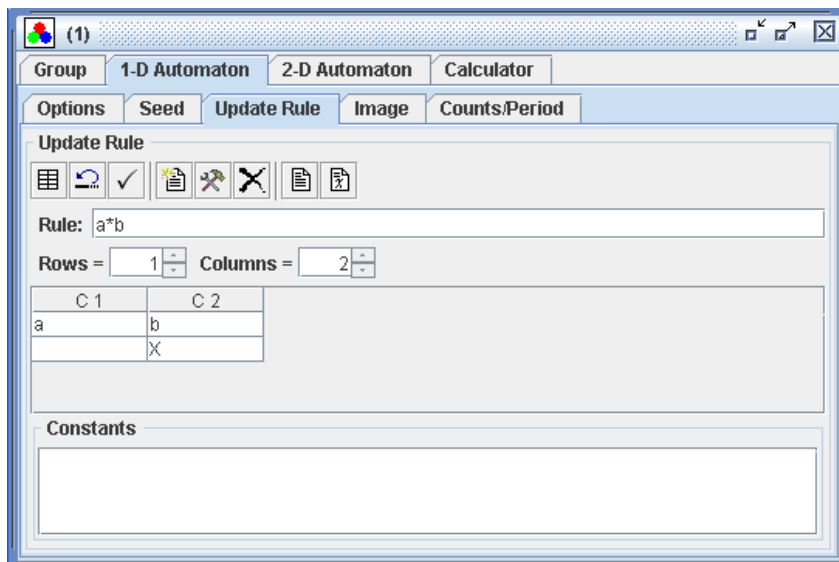


The Seed tab allows the user to input the seed (or starting point) of the automaton. The seed can consist of up to 20 rows (or time-steps) and 100 columns. There are facilities for clearing the seed grid,

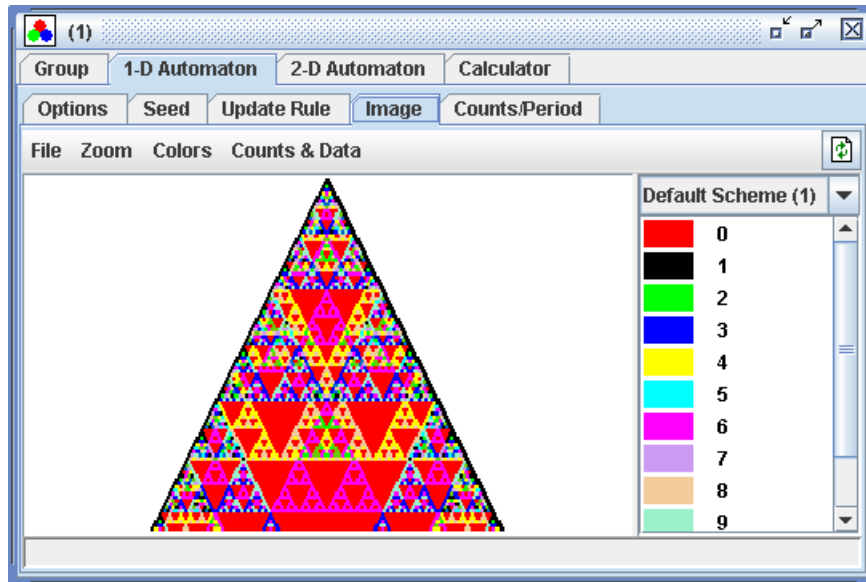
filling the grid with random group elements, checking the syntax of the elements in the grid and copying the grid elements to the clipboard in either a text format or LaTeX code.



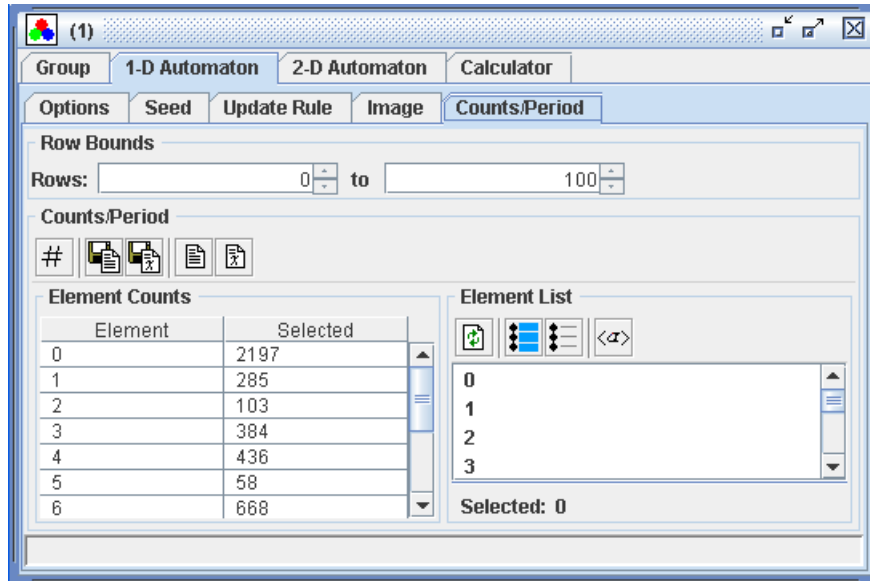
The Update Rule tab allows the user to alter the way each time-step is created from the previous time-steps. The program automatically sets the default update rule to that of Pascal's Triangle, that is, adding the two above to get the next cell value. So if you never wish to use an update rule other than Pascal's Triangle you will never need to alter the information on this tab. On the other hand, if you wish to do some experimentation this tab allows you to input a general update formula that uses the group operation "*" and powers "^" that are integer values. You link the automaton position with the update formula through variable names in the grid. You may also use constants in the formula via a constant name that you specify in the constants list at the bottom. The program has facilities for clearing the grid, resetting the rule to Pascal's Triangle, checking the syntax of the rule, creating, editing and deleting constants, and copying the update rule to the clipboard as either text or LaTeX code.



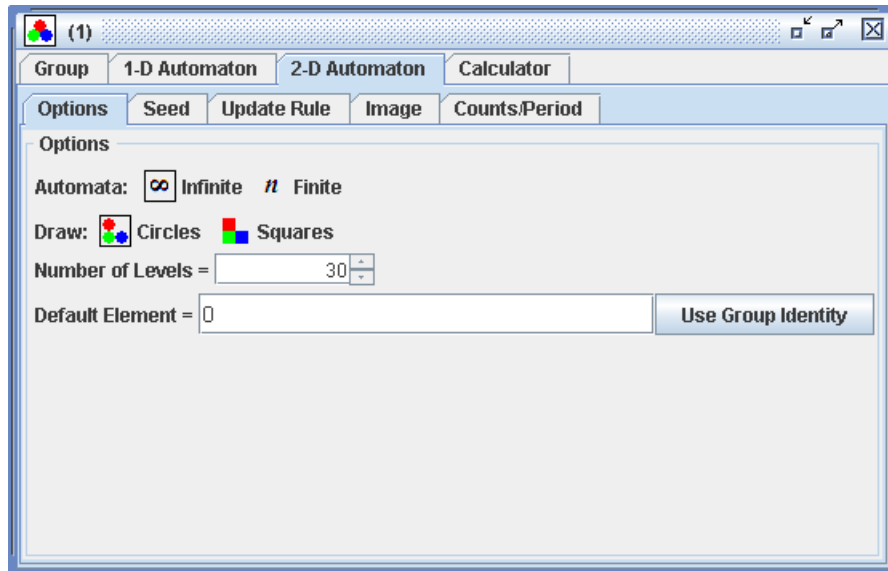
The Image tab is the place you will do most of your exploration. It consists of an image window to the left and a color scheme to the right. The program allows the user to change element colors, group element colors by sets, subgroups, and cosets. There are facilities for zooming in and out of the automaton. The user can copy images and color correspondence to the clipboard or save them as image files. You can have an unlimited number of different color schemes loaded with the capability of saving them to files and loading in from files. There are also facilities for selecting regions of the automaton and displaying element counts within the region as well as copying or saving these counts or the grid of data as either text or LaTeX code. For finite automata there is also a facility for determining the period of the automaton, within the graphed rows. The zooming, color manipulation and region selection all have undo/redo capabilities.



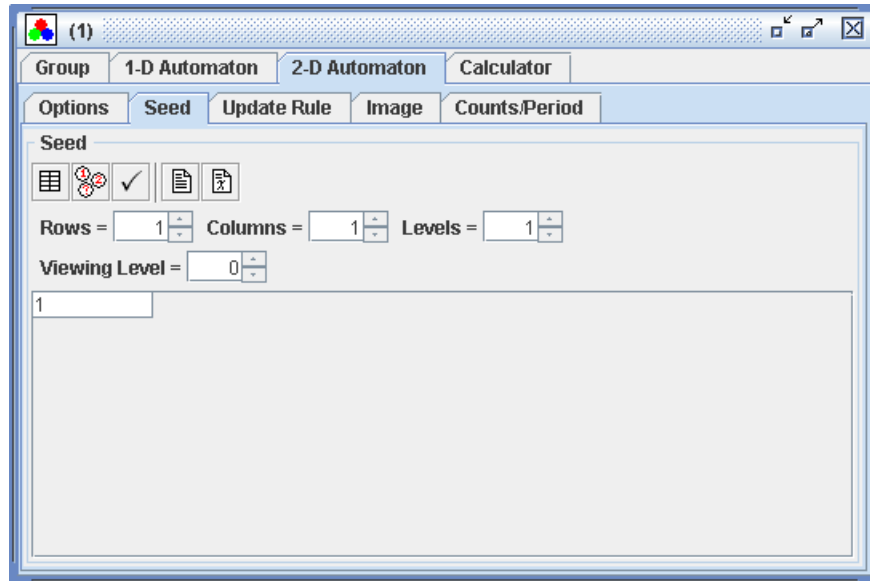
The Counts/Period tab has some more advanced methods for finding the period or death of a finite automaton, it also allows the user to do element counts for row ranges without the need to graph and select the region. When you just want element counts on rows this is a much faster and memory reducing way to get them. The program has facilities for displaying, copying and saving count data as either text or LaTeX code. There is also an element list to the right that will sum all of the selected element counts with the ability of automatically selecting subgroups.



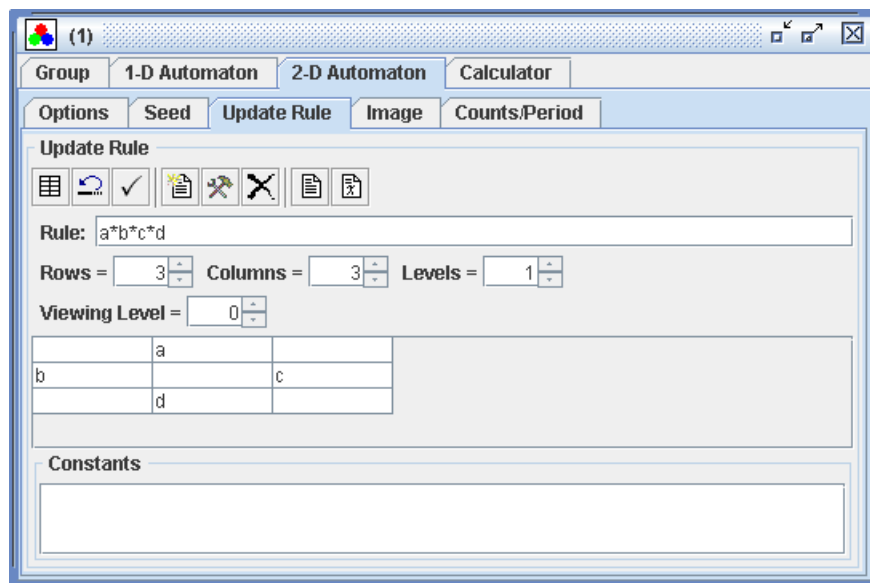
Under the 2-D Automata tab there are currently five other tabs which allow the user to input all of the data necessary to create and explore both finite and infinite two-dimensional automata. The Options tab allows the user to select between a finite or infinite automata, the number of levels graphed, the wrapping style, seed placement, grid size and whether it draws circles or squares.



The Seed tab allows the user to input the seed (or starting point) of the automaton. The seed can consist of up to 20 levels (or time-steps) and a maximum grid size of 100 X 100. There are facilities for clearing the seed grid, filling the grid with random group elements, checking the syntax of the elements in the grid and copying the grid elements to the clipboard in either a text format or LaTeX code.

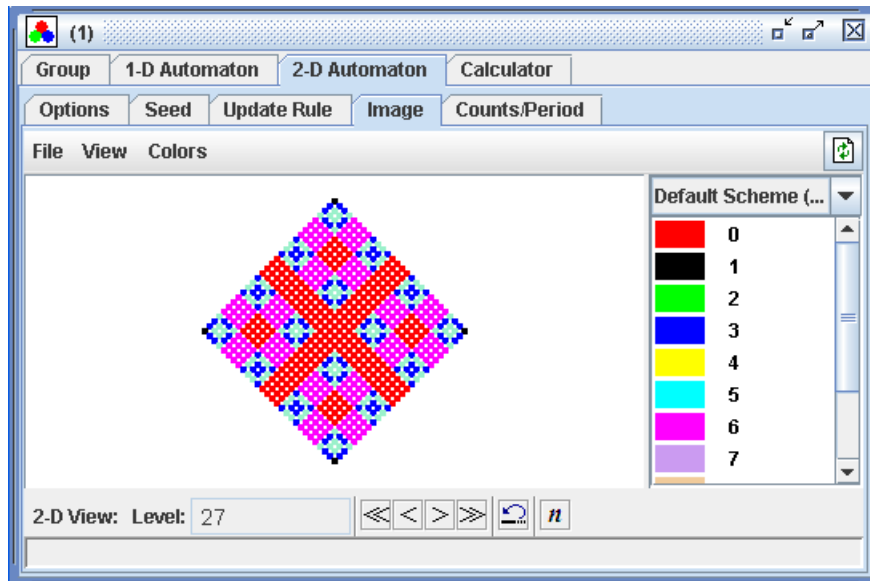


The Update Rule tab allows the user to alter the way each time-step is created from the previous time-steps. The program automatically sets the default update rule to that of one of the standard four-rules. The program offers 11 commonly used default rules as well as the ability to alter these into any update rule desired within a 100 X 100 grid and a depth of 20 time-steps. This tab also allows you to input a general update formula that uses the group operation "*" and powers "^" that are integer values. You link the automaton position with the update formula through variable names in the grid. You may also use constants in the formula via a constant name that you specify in the constants list at the bottom. The program has facilities for clearing the grid, resetting the rule to any one the 11 built-in presets, checking the syntax of the rule, creating, editing and deleting constants, and copying the update rule to the clipboard as either text or LaTeX code.

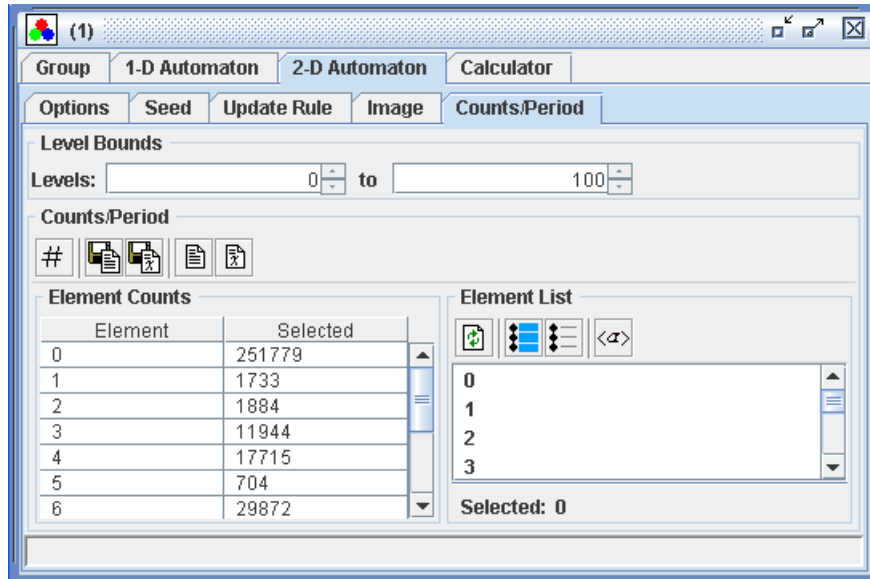


The image tab is the place you will do most of your visual exploration. It consists of an image window to the left, a color scheme to the right and a level navigation bar at the bottom. The program allows the user to change element colors, and group element colors by sets, subgroups, and cosets. The navigation bar allows the user to select which time-step to view. It has buttons to move to the next and previous time-steps as well as a 10 step jump, reset and a facility to move to any level by specifying its number. The user can copy images and color correspondence to the clipboard or save them as image files. You can have an unlimited number of different color schemes loaded with the capability of saving them to files and loading in from files. You can also copy or save the elements that make up the level or set of levels as either text or LaTeX code.

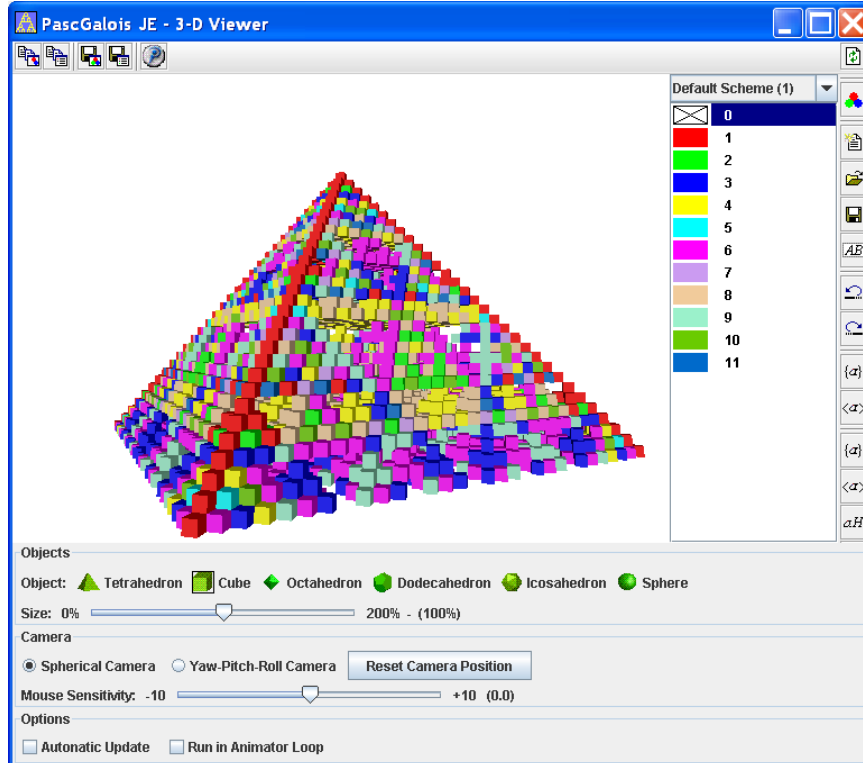
The program offers two facilities for viewing the automaton in three dimensions. First, there is a three-dimensional viewing window that will allow you to rotate, shift and zoom in and out on the automaton. It also has color scheme, clipboard and file transfer capabilities. Second, the program has a POV-Ray exporter that allows you to output the automaton as a POV-Ray file and use the freeware ray tracer POV-Ray to render the image. No knowledge of POV-Ray programming is necessary.



The Counts/Period tab has some more advanced methods for finding the period and death of a finite automaton, it also allows the user to do element counts for level ranges. The program has facilities for displaying, copying and saving count data as either text or LaTeX code. There is also an element list to the right that will sum all of the selected element counts with the ability of automatically selecting subgroups.



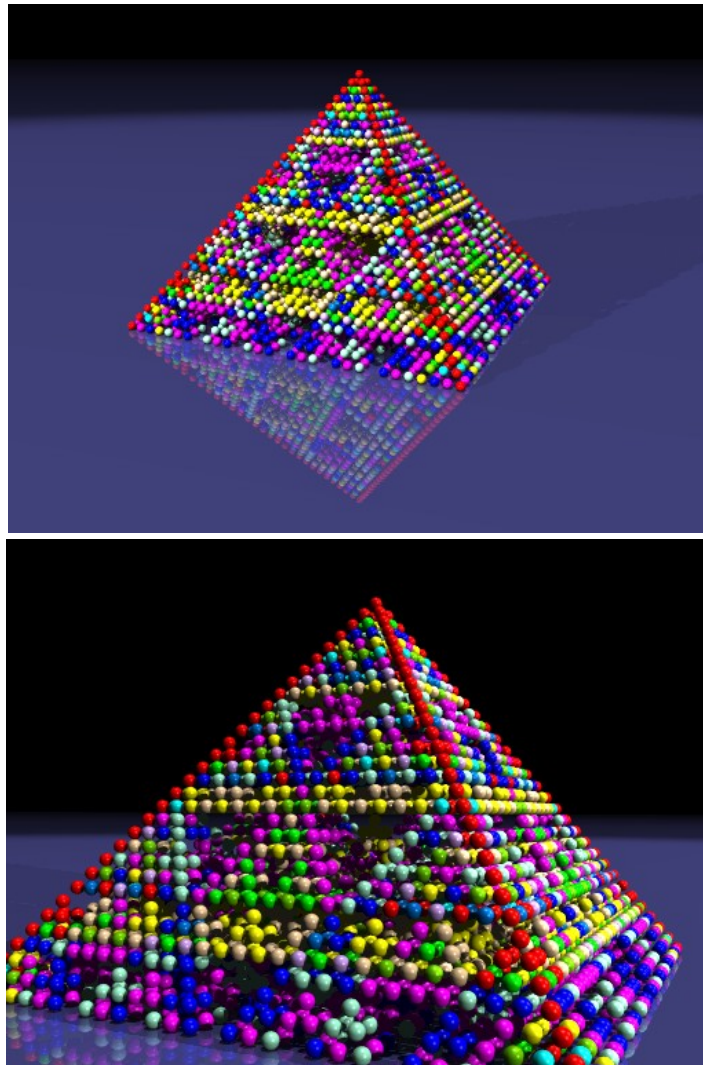
The 3-D viewer is set up in a similar manner as the viewing window. The image is to the left and the color scheme is to the right. Color schemes can be saved and loaded into this window as well. At the bottom are a few options for changing the size of the graphed objects as well as what type of object is graphed and what type of camera interface is used. Along the right side are the same color manipulation options as found in both the one and two dimensional viewing windows. At the top are clipboard and file transfer options as well as the same POV-Ray exporter that is in the two-dimensional viewing tab.



Selecting the POV-Ray output option from either the 2D or 3D viewers will create two files, a pov

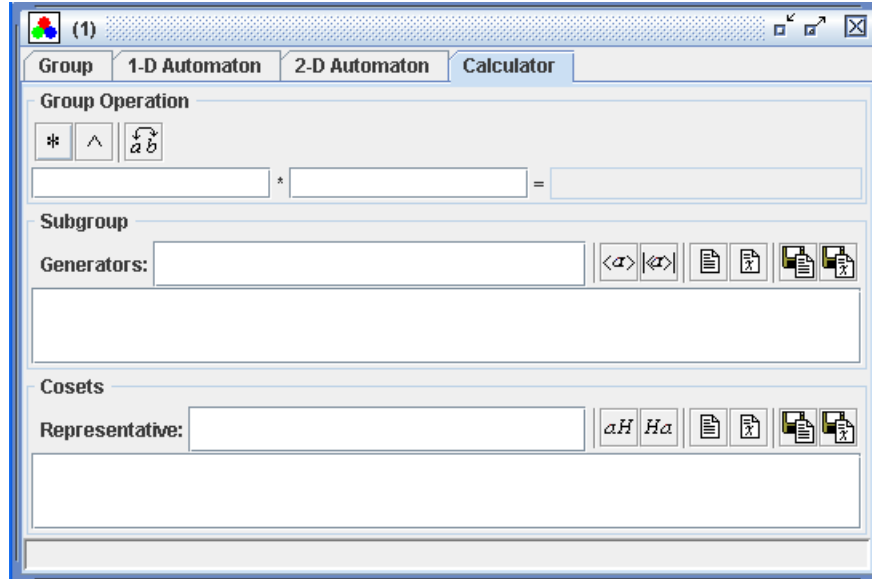
file and an inc file. The pov file contains the setup for the image and the inc file contains a model for the automaton. The inc file never needs to be opened and in many cases it will contain thousands of lines. Use POV-Ray to open the pov file and select run from the toolbar in POV-Ray. It will then render the image and save it as a bitmap file in the same folder as your pov file. If you are on a non-windows system you may need to use a command-line command to render the image. Please consult a POV-Ray manual for details on your implementation.

The exporter also allows the user to create a POV-Ray animation. One type of animation is a color animation that will produce an animation of the automaton as it shifts from one color scheme to another. Another type is a rotation animation that will rotate the camera around the automaton and you can also combine the two. For the animation files the program will also produce an ini file. Load the ini file into POV-Ray and select run. This will produce a series of images on your hard drive and not a single movie file. You will need another program to paste these images onto a movie file. There are several freeware programs that will do this and they can be found for most platforms.



The Calculator tab allows the user to do single group operations or take elements to powers. It also

has the ability to switch the order of the elements so the user can check commutativity quickly. There are also facilities for generating subgroups and cosets given subgroup generators and coset representatives. Furthermore, both the subgroups and cosets can be copied and saved as either text or LaTeX code.



License

This program is freeware intended for educational purposes and is offered “as is” under the following stipulations:

1. The copyright for the program remains the property of Dr. Don Spickler.
2. License is granted for one user and/or one computer. To obtain additional copies for another user on a different computer please revisit our web site at <http://pascgalois.org/>.
3. If you are able to use the program for anything productive (i.e. student exercises, obtaining mathematical results for publication, etc.) Please e-mail us at despickler@salisbury.edu and let us know.
4. Any publications resulting in part from use of this program should acknowledge the program and its developers: Drs. Don Spickler, Kathleen M. Shannon and Michael J. Bardzell.

Copyright © by Dr. Don Spickler (2004-2007)

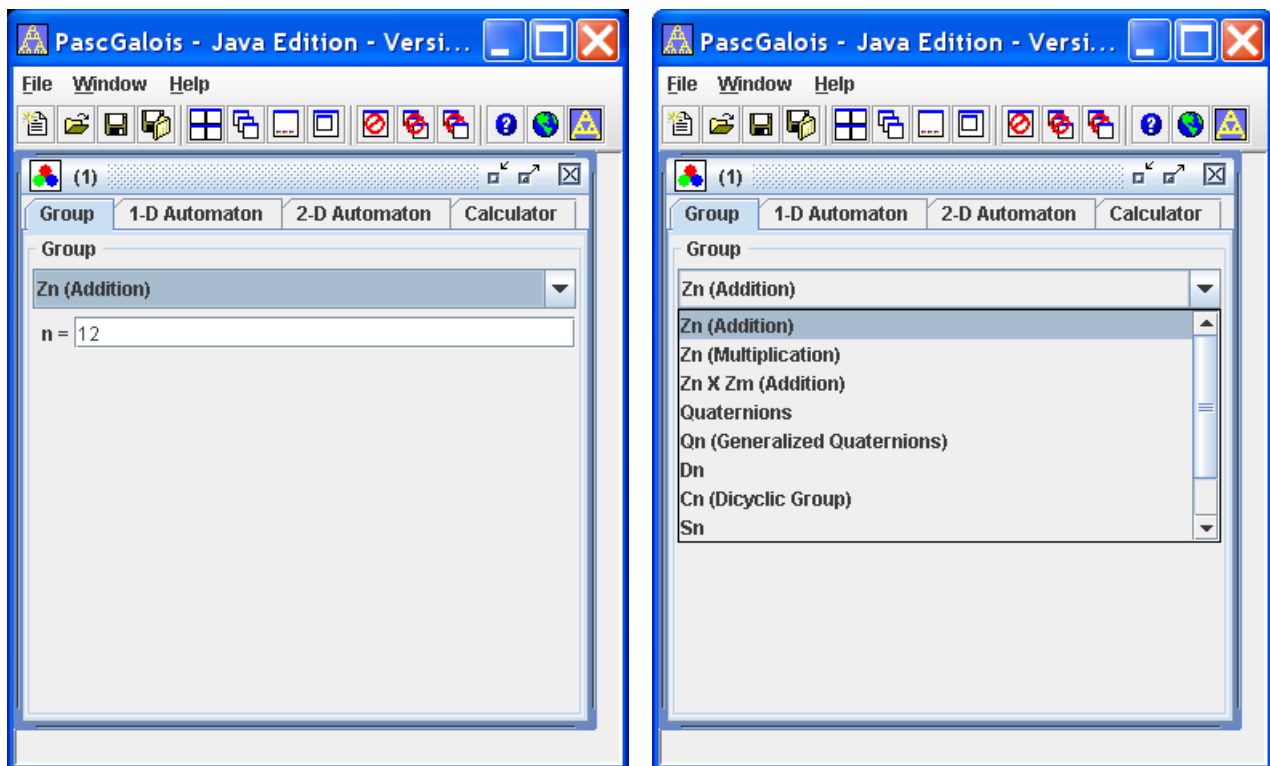
The creation of this program was made possible through a grant from the National Science Foundation (NSF #DUE-0339477).

Group Overview

As you read in the program overview the Group tab is for selecting the group (or structure) you intend on working with. Although this tab has quite a few options and facilities, most of them are for user-defined structures, which most people will not be using at the beginning. We will first discuss the built-in group structures and then worry about the user-defined ones.

Selecting a Group

When selecting a group structure first select the class of group using the drop-down menu at the top of the tab.



The supported group structures are

- Z_n : Integers under addition mod n .
- Z_n : Integers under multiplication mod n .
- $Z_n \times Z_m$: The Cartesian product of Z_n and Z_m , both under addition.
- Q_n : Generalized Quaternion groups.
- D_n : Symmetry group for a regular n -gon.
- C_n : Dicyclic group.

- S_n : Group of permutations on n letters.
- G_n : User-Defined Groups.
- Q : The Quaternions.
- Advanced Mode: Any Cartesian product or quotient of the above structures.

For a detailed description of these structures and their syntax please see the Supported Structures & Syntax section. Once the group structure type has been set in most cases you need to supply one or two integers. Do this in the box(es) below the drop-down selector. For example, if you want to work with Z_{12} , integers mod 12 under addition, you would select Zn (Addition) from the drop-down and type in 12 into the $n=$ box.

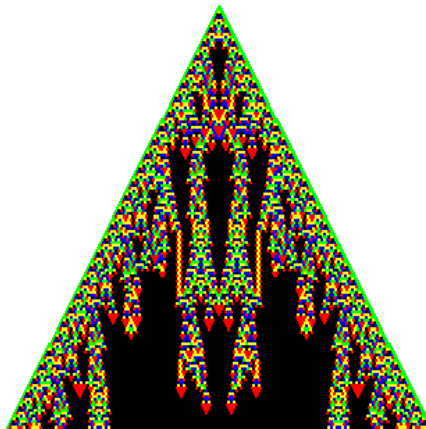


There is really no restriction on n . Theoretically, you can work with S_{10} or even S_{100} for that matter. From a programming point of view n (and m) are long integers in Java so they must be less than $2^{63}-1$. Of course, as the values of n and m grow the program will consume more and more memory. Possibly to the point where all of the computer's memory is used. So from a practical point of view the values of n and m should be reasonable.

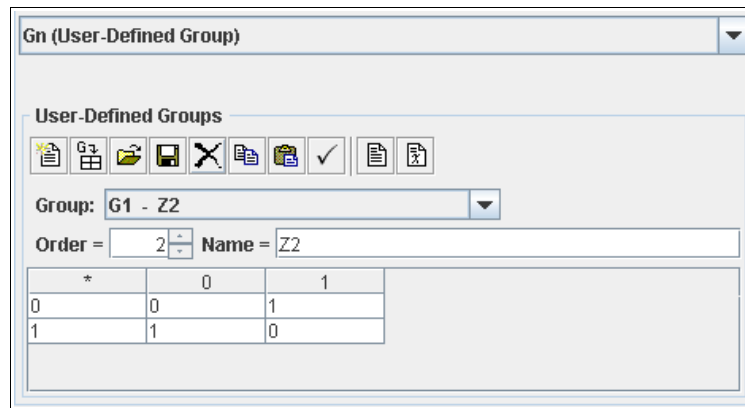
The only groups that do not need an associated number are the Quaternions (Q), User-Defined structures and the Advanced Mode. The Quaternions are the noncommutative group of order 8 consisting of $\{1, -1, i, -i, j, -j, k, -k\}$ with the group operation, $i*j = k, j*i = -k, j*k = i, k*j = -i, \dots$ User-defined structures use the operation table that is currently being displayed and the Advanced Mode uses a textual input of the group. We will discuss the syntax of the Advanced Mode later on this page.

User-Defined Structures

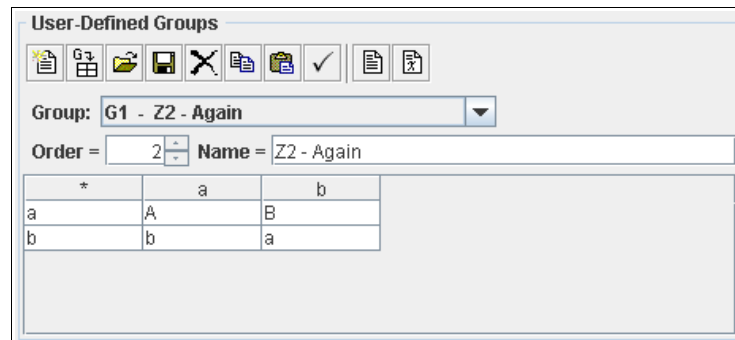
User-Defined structures are given by an operation table. User-defined structures need not be groups. In fact, they need not have any group properties at all. It seems that these automata are not widely studied but they can create some very interesting and beautiful images. For example, one small alteration to the operation table of Z_5 produces.



To use a User-Defined structure select the G_n (User-Defined Group) from the drop-down box and then create the operation table for the structure at the bottom of the tab.

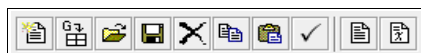


To change the group's order simply use the Order spinner. User-Defined structures can up to 200 elements. The structure must also have a name. The name is not used in any calculation it is simply to help organize the structures. The name can contain spaces and special characters, except for the comma. The elements can be any strings that contain letters, numbers or an underscore. You may not use spaces or special characters. Also, the names are not case sensitive. That is a and A produce the same element. The first column of the table represents the elements. As you input the entries in this first column you will see them appear along the top of the table as well. For example, the following table is also a representation of Z_2 .



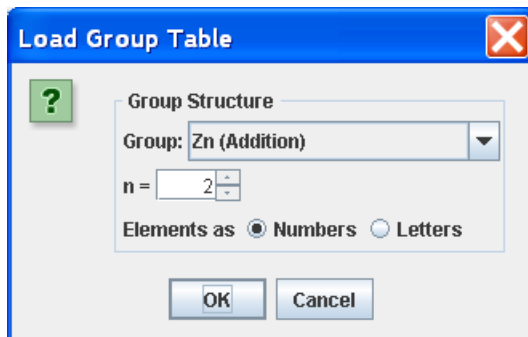
Note that as you change the name of the structure it also changes in the User-Defined drop-down box as well. You use this drop-down selector to switch between user-defined structures. The group tab can contain any number of user-defined structures. Remember that if you have the G_n (User-Defined Group) selected the structure that is used is the one currently being viewed.

The toolbar contains several facilities to make constructing your table a little easier.



- The first tool creates a new group (a copy of Z_2) and loads it into the tab. You can then alter the size, name and table to your liking.
- The second tool loads the table of a supported group into the table. When you select this tool

another dialog box comes up. Select the group type, type in n (or n and m, if necessary), the output as letters or numbers and click OK. At this point the operation table of that group will be loaded into the table. You can, of course, edit it to your liking.



- The third and fourth tools load and save user-defined structures respectively.
- The fifth tool will delete the current table.
- The sixth and seventh tools copy and paste user-defined structures to the and from the clipboard. This creates an easy way to copy a user-defined structure from one child window to another.
- The eighth tool (the check mark) checks that syntax of the current user-defined structure. It does not check to see if the structure is in fact a group. It simply checks to see that all of the elements are strings with letters, numbers and underscores only as well as make sure that the structure is closed (i.e. * is a binary operation).
- The last two tools copy the current user-defined structure to the clipboard in either a text format or as LaTeX code.

Advanced Mode

Advanced Mode allows you to use arbitrary Cartesian products and quotients of any of the supported groups. To use advanced mode simply select Advanced from the group drop-down box and type in the group you want to work with. For example, in the screen shot below we will be using the group Z_2 .



The syntax for the group structure is fairly easy. We will start with the supported group structures. The Supported Structures & Syntax section will get into more detail on the element syntax of these structures. The only change between the advanced input and the drop-down is that the integers mod n under multiplication is represented by U_n . Be a little careful here. In some abstract algebra texts U_n is the set of all integers between 1 and n-1 that are relatively prime to n, and the operation is multiplication mod n. The program does not restrict the numbers to be relatively prime to n.

- Z_n : Integers under addition mod n.
- U_n : Integers under multiplication mod n.

- Dn: Symmetry group for a regular n-gon.
- Qn: Generalized Quaternion groups.
- Cn: Dicyclic group.
- Sn: Group of permutations on n letters.
- Gn: User-Defined Groups.
- Q: The Quaternions.

The syntax for these structures is simple, it is the letter followed by the number desired for n. For example,

- Z23 is written Z23.
- U17 is written U17.
- D5 is written D5.
- Q4 is written Q4.
- C8 is written C8.
- S7 is written S7.
- G2 is written G2 and is the second user-defined group in the list.
- Q is written Q.

The program is not case sensitive so z5 and Z5 are identical.

The program can also take arbitrary products and quotients of the supported groups. Products are represented by "X" and quotients are represented by either "/" or "%". The "/" will use left cosets for the calculations and "%" will use right cosets. For example,

- Z4 X Z5 is written Z4 X Z5.
- Z3 X D23 X S7 X Q is written Z3 X D23 X S7 X Q.

Another thing to note is that spaces are ignored, so Z3 X D23 X S7 X Q produces the same group as Z3XD23XS7XQ. When modding out by a subgroup you need to enclose the subgroup generators in < >. When using more than one group generator separate the generators with a single comma. For example,

- Z10/<5> is written Z10/<5>.
- S7/<(1 2 3)(4 7), (3 5 7)> is written S7/<(1 2 3)(4 7), (3 5 7)>.

Supported Group Structures & Syntax

Supported Group Structures

- Z_n : Integers under addition mod n .
- U_n : Integers under multiplication mod n .
- D_n : Symmetry group for a regular n -gon.
- Q_n : Generalized Quaternion groups.
- C_n : Dicyclic group.
- S_n : Group of permutations on n letters.
- Q : The Quaternions.
- G_n : User-Defined Groups.

We will discuss each of these in turn in an order that one would probably see them in an abstract algebra course.

Z_n : Integers under addition mod n .

Z_n is the set of elements $\{0, 1, 2, \dots, n-1\}$ under the operation of addition mod n . For example, in Z_5 , $4+3 = 2$. The program syntax for elements in this group are simply non-negative integers. Any input greater than or equal to n is replaced with itself mod n . So if you are working with Z_5 and use some input of 12 it will be converted to 2.

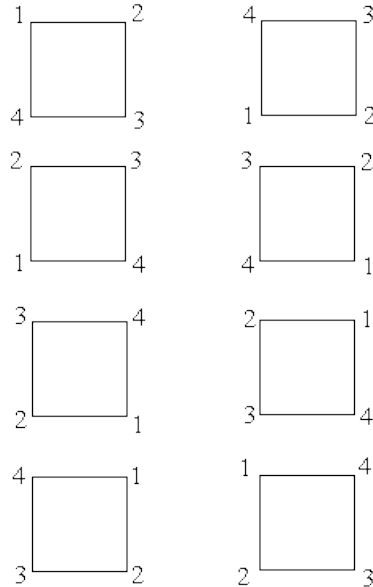
U_n : Integers under multiplication mod n .

U_n is the set of elements $\{0, 1, 2, \dots, n-1\}$ under the operation of multiplication mod n . Note that the elements used in the program need not be relatively prime to n . For example, in U_6 , $2*3 = 0$. The program syntax for elements in this group are simply non-negative integers. Any input greater than or equal to n is replaced with itself mod n . So if you are working with U_5 and use some input of 12 it will be converted to 2.

D_n : Symmetry group for a regular n -gon.

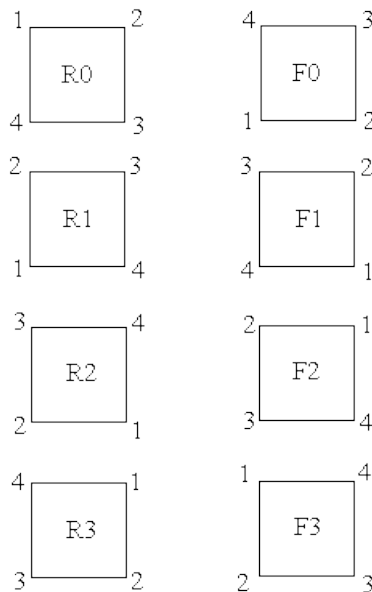
D_n is the set of all symmetry transformations of a regular n -gon, also known as the Dihedral group. Hence D_n contains $2n$ elements, n rotations and n reflections or flips. The operation is composition of transformations. For example, D_4 is the symmetry group of the square. There are four rotations (0, 90, 180, and 270 degrees) and four flips (one over the horizontal, one over $y=x$, one over the vertical and

one over $y=-x$). Graphically we can represent the elements of this group by the following figure. The original square is in the upper left. The rotations are down the first column and the flips are down the second column.



When representing these elements in the computer we can not easily use degree or radian measurement for rotations, especially when the angle does not divide evenly into 360 (for example, with D_7). So we have adopted the following scheme. R_0 represents rotation by 0 degrees, that is, the identity element of D_n . R_1 represents the first rotation that is a symmetry. So in D_4 , R_1 represents rotation by 90 degrees, in D_5 R_1 represents rotation by 72 degrees and in D_6 R_1 represents rotation by 60 degrees. In general, in D_n R_1 represents rotation by $360/n$ degrees. R_2 is the second such rotation, that is, rotation by $360*2/n$ degrees and so on up to $R_{(n-1)}$. The flips are represented by $F_0, F_1, \dots, F_{(n-1)}$. F_0 is the flip over the horizontal. F_1 is the first possible flip that is a symmetry. In D_4 , F_1 represents the flip over the line that is 45 degrees from the horizontal, in D_5 F_1 represents the flip over the line that is 36 degrees from the horizontal, and in D_6 F_1 the flip over the line that is 30 degrees from the horizontal. And so on. All of our angle measurements are counterclockwise, of course.

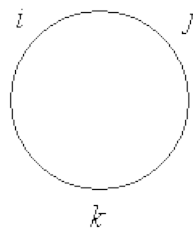
So from our above example of D_4 , we have,



The operation for this group is composition of functions. We use the right to left convention of composing functions. For example, $R1 * F2 = F3$ and $F2 * R1 = F1$.

Q: The Quaternions.

The Quaternions is a noncommutative group of order 8 consisting of the elements $\{1, -1, i, -i, j, -j, k, -k\}$ with $i^2 = j^2 = k^2 = -1$ and a circular definition for multiplying $i, j,$ and k . If we consider the following circle with $i, j,$ and k on it. To find the product of $i * j$ we start at i and go around the circle to j (using the smallest arc) then we keep going to the next letter, which is of course k . Now if we moved in a clockwise direction we use a positive k and if we had to go counterclockwise we use a $-k$.



So we see that, $i*j = k, j*i = -k, j*k = i, k*j = -i, \dots$ Multiplication by 1 and -1 is the same as with integers, $-1 * i = -i$ and so on.

The Quaternions can also be defined as the group of order 8 having two generators x and y satisfying the relations $x^2 = y^2$ and $xyx = y$. Since the first representation is more common in introductory abstract algebra classes we use it in this program. The element syntax for the program is simply 1, -1, i, -i, j, -j, k, -k, where $i, j,$ and k can be lower or upper case.

S_n : Group of permutations on n letters.

S_n is the group of permutations on n letters, also called the symmetric group on n letters. There are several ways to represent the elements of this group but we have adopted the cycle notation since it is easier to input from a keyboard and is a little faster to do the necessary calculations. For example, if we are working in S_7 the permutation (2 3 5 4 7) means that 2 is sent to 3, 3 is sent to 5, 5 is sent to 4, 4 is sent to 7, 7 is sent to 2, and both 1 and 6 are fixed. The identity element is represented by (1). The operation for this group is composition and as with the Dihedral groups we do composition from right to left. For example,

$$(2\ 3\ 5\ 4\ 7)(1\ 2\ 3\ 5\ 6) = (1\ 3\ 4\ 7\ 2\ 5\ 6) \text{ and } (1\ 2\ 3\ 5\ 6)(2\ 3\ 5\ 4\ 7) = (1\ 2\ 5\ 4\ 7\ 3\ 6)$$

The program syntax for S_n is just how it is written mathematically. A cycle must start with a (have numbers between 1 and n separated by at least one space and ending with a). For permutations that are composed of two or more disjoint cycles we simply use juxtaposition. For example, (1 3 2)(4 6 5). The program outputs cycles in a standard form with the smallest numerical entry in the cycle at the beginning. As a user, you do not need to input cycles in that form. For example, (1 3 2) can be input as (3 2 1) or as (2 1 3).

Q_n : Generalized Quaternion groups.

Q_n is the generalized quaternion group, $n > 2$. It is a group of order 2^n generated by two elements a and b under the following relations.

$$a^{2^{n-1}} = 1, \text{ } bab^{-1} = a^{-1}, \text{ and } b^2 = a^{2^{n-2}}.$$

The element syntax for this group is any string of a and b to positive powers. For example, the user can input aba, babbab, a^2b^3 , b^2a^3 , and so on. Since every element of the generalized quaternions can be rewritten in the form a^tb^r with $0 \leq t \leq 2^{n-1}$ and $0 \leq r \leq 1$ the program will simplify every element to this form.

C_n : Dicyclic group.

C_n is the dicyclic group, $n > 1$. It is a group of order $4n$ generated by two elements a and b under the following relations.

$$a^{2n} = 1, \text{ } b^{-1}ab = a^{-1}, \text{ and } b^2 = a^n.$$

The element syntax for this group is any string of a and b to positive powers. For example, the user can input aba, babbab, a^2b^3 , b^2a^3 , and so on. Since every element of the dicyclic group can be rewritten in the form a^tb^r with $0 \leq t \leq 2n-1$ and $0 \leq r \leq 1$ the program will simplify every element to this form. One thing to note is that when n is a power of 2 the dicyclic group is isomorphic to a generalized quaternion group. More specifically, C_{2^n} is isomorphic to Q_{n+2} and C_2 is isomorphic to the quaternion group.

G_n : User-Defined Groups.

User-defined structures use the element names that you supply as the element syntax. Remember that these are not case sensitive.

Advanced Input Syntax

For any of the built-in group structures the syntax is simply the letter designation of the class of groups followed by the number n (or m) you would have input for that group. For example, the symmetry group of the square (D_4) would be $D4$, the integers mod 32 would be $Z32$ and the symmetric group on 5 letters would be $S5$. For a user-defined group we use a G followed by the number of the group in the list. So $G3$ would be the third user-defined group in our list of user defined groups. This is also found in the user-defined group drop-down box. In the example above you see that the user-defined group is $G2$. The power of the advanced input option is in the ability to work with arbitrary products and quotients of the supported group structures and the user-defined structures. We discuss the product and quotient syntax below.

Cartesian Products

To take the Cartesian product of groups we simply put an X between the structures we wish to product. So for example, if we want to work with $S_3 \times D_5$ we would use the syntax $S3 \times D5$. We can product more than 2 groups together as well. So to work with $D_7 \times Q \times S_4 \times Z_{32}$ we would use the syntax $D7 \times Q \times S4 \times Z32$.

As for the element syntax with these groups we simply use ordered tuple notation. For example, if your group is $Z3 \times Z5$ then an element would look like $(2, 3)$. If your group is $Z3 \times D_{23} \times S_7 \times Q$ then an element would look like $(2, F10, (1\ 3\ 5)(2\ 4), -J)$. As with the group syntax, the spaces are not important, except for the elements of S_n where at least one space is needed between each of the numbers in a cycle. A tuple must begin with a $($, end with a $)$ and have a single comma between each of the components. One note of caution with products is that tuple simplification is done automatically. That is, if our group is input as $Z_{24} \times (S_7 \times D_9)$ the program will view it as $Z_{24} \times S_7 \times D_9$ and hence elements should be input in the form $(19, (1\ 2\ 3), R5)$ and not $(19, ((1\ 2\ 3), R5))$.

Quotients

The syntax for quotients is also much like we would write it mathematically. You simply input the group structure followed by a $/$ followed by the subgroup you wish to mod out by. The subgroup is designated by placing the generators for the subgroup between angled brackets $\langle \rangle$. So for example, if we want to create automata over the group $Z_{10}/\langle 5 \rangle$ we would use the syntax $Z10/\langle 5 \rangle$. If the subgroup has multiple generators we put all of the generators between the angles brackets and separate them with a comma. For example, if we wanted to use S_5 modded out by the subgroup generated by $(1\ 2)$ and $(3\ 4)$ we would use the syntax $S5/\langle (1\ 2), (3\ 4) \rangle$. Note that the subgroup need not be a normal subgroup, the program will simply use the structure of left cosets. If you would prefer to use right cosets instead of left cosets replace the $/$ with a $\%$. So the structure of right cosets of $\langle (1\ 2), (3\ 4) \rangle$ in S_5 would be

represented as $S5\%(1\ 2), (3\ 4)\%$.

For elements in quotient groups we simply ignore the modded subgroup in the element syntax. For example, if our group is $Z_{10}/\langle 5 \rangle$ an element would be represented as 3 and not as $3 + \langle 5 \rangle$. There is a printing option for cosets like $3 + \langle 5 \rangle$ but the input parser would not understand the syntax. Similarly, if our group is $Z_{10}/\langle 5 \rangle \times Q$ an element would look like (3, J). One final example, say our group was $(Z_{10}/\langle 5 \rangle \times Q/\langle K \rangle \times D_{10})/\langle (3, J, F2), (2, -I, R3) \rangle$ then an element would look like (1, J, R1).

1-D Cellular Automata

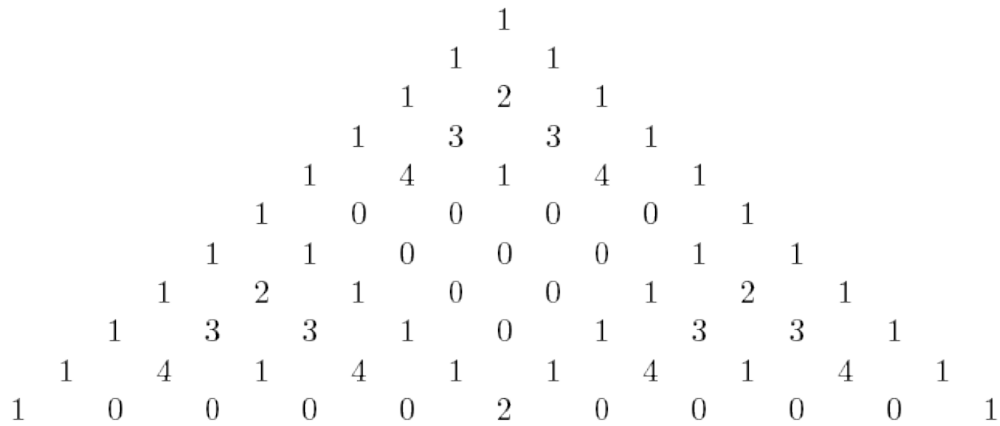
Overview

One-dimensional cellular automata comes in two flavors, finite and infinite. Both work in a similar manner. Both start with a seed, which is one or more rows of elements thought of as sitting on a line. This line (or row) is considered to be a time-step, specifically the initial time-step. Every time-step (or row) after the initial one is created by applying some update rule to the previous time-step or time-steps. This process is continued indefinitely. With this terminology we can consider Pascal's Triangle as a one-dimensional cellular automaton. Below are the first six rows of Pascal's Triangle. Think of the first row (just the 1) as the seed.

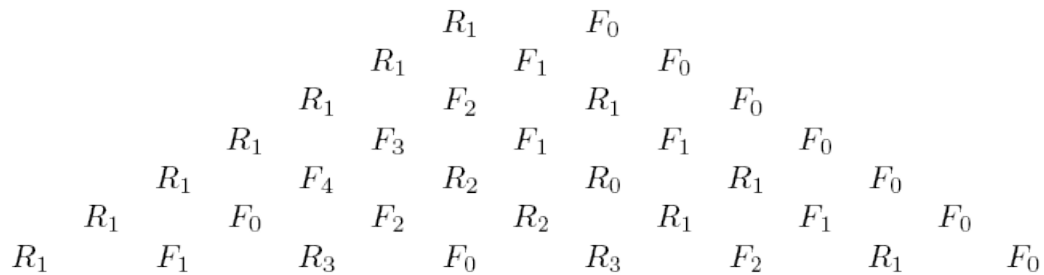
				1						
				1		1				
			1		2		1			
		1		3		3		1		
	1		4		6		4		1	
1		5		10		10		5		1

The second row is now generated from the first by setting each entry in the second row to the sum of the two elements directly above it. If there is no element in one of the two above positions we simply use 0. The rule where we take the sum of the two elements directly above each entry is called the update rule. The third row is generated from the second in the same manner as is the fourth from the third, and so on. In the theoretical Pascal's Triangle this process continues indefinitely.

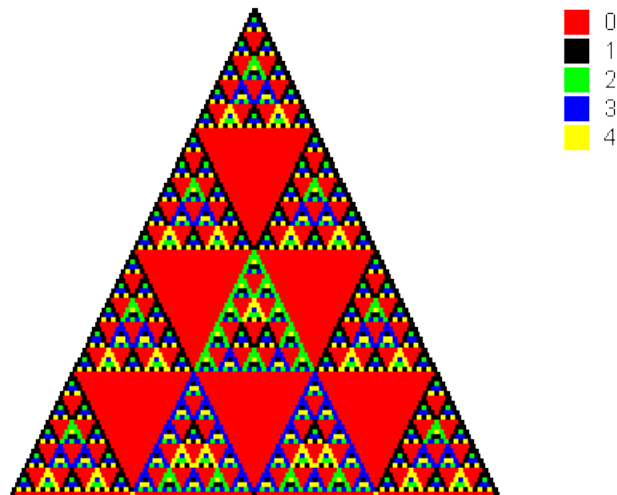
The main feature of the PascGalois JE program is the creation of these automata. Since this is a computer program with limited recourses trying to work with infinite groups can pose several difficulties. So instead of doing Pascal's Triangle on the integers we do Pascal's Triangle on finite group structures. The update rule of adding the two numbers to get the next entry is replaced by doing the group operation on the two above group elements. For example, if we are working in Z_5 the first 11 rows would look like.



If we are working in D_5 using a seed of R_1 and F_0 the first 7 rows would look like.



The program goes one step further by applying a coloring scheme to the elements of the group to help students see algebraic structure. In the Z_5 example if we continue the process for 100 rows and apply the coloring scheme we see,



Which clearly has some nice structural properties.

The program is also capable of accepting seeds with more than one row and update rules that span more than just the previous row, or time-step. That is you can update an element by reaching back

further than just the previous time-step. Furthermore, the program is capable of using a general update rule formula. You will learn more about these features on their specific pages.

Finite Automata

Finite automata have a finite fixed width and an update that either wraps around or does not wrap around (sometimes called null-bounded). For example, let's say we are working in Z_5 with a seed of

$$1 \ 1 \ 1 \ 1.$$

The update rule will still be that of Pascal's Triangle but since we tend to stack the entries of finite automata in columns the update rule is more like adding the element above and the one above and one to the left. If we do not wrap the update rule we would get.

$$\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \end{array}$$

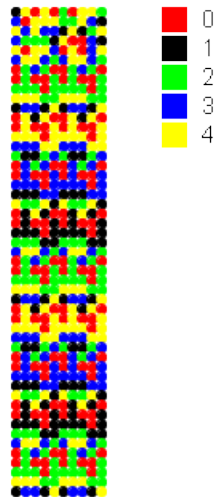
The three 2's on the second row are simply the sum of the entries above and above to the left. The first entry of the second row is the sum of the element above it (the 1) and the element above to the left, which does not exist and hence is taken as 0. On the other hand if we wrap the update rule we get the following.

$$\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array}$$

The only difference is that the first entry of the second row is now a 2. This entry is gotten from the element directly above and above left. Now the above left element is wrapped around to the last element of the first row (another 1). You can think of wrapped update rules as being drawn on a cylinder. On a cylinder moving to the left (or right) automatically wraps around the cylinder. For example, let's stick with Z_5 and take the random seed of,

$$1 \ 4 \ 0 \ 4 \ 0 \ 2 \ 0 \ 4 \ 4 \ 2$$

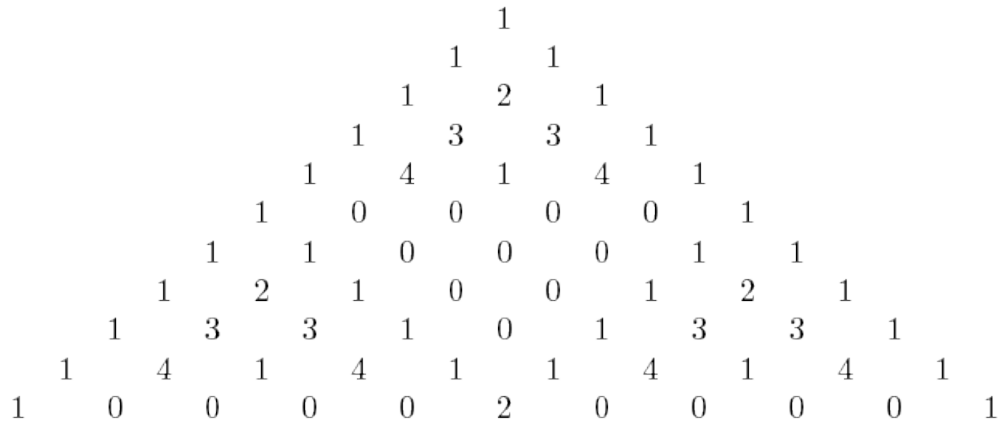
The first 50 rows of this finite automata (with wrapping) looks like,



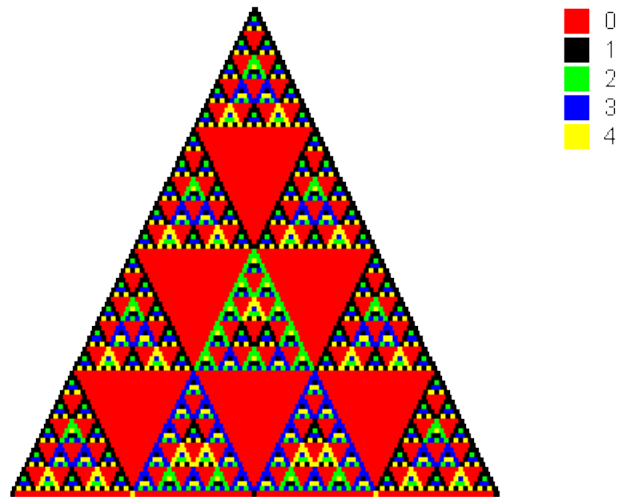
On closer inspection you will see that this image eventually repeats every 20 rows, hence the period of this automaton is 20.

Infinite Automata

An infinite automaton is not fixed in width nor is it wrapped. Like in the Pascal's Triangle examples we imagine that the seed is padded indefinitely on both sides by the group identity. So for Pascal's Triangle the seed row of 1 is really ... 0 0 0 1 0 0 0 ..., hence the term infinite. So the examples at the beginning of this page are all infinite automata. For example, if we are working in Z_5 with seed 1 and Pascal's update rule the first 11 rows would look like.



and applying the coloring scheme to the first 100 rows looks like,



The Options Tab

The Options tab for the infinite 1-D Automata looks like,

Automata:	<input checked="" type="radio"/> Infinite	<input type="radio"/> Finite
Alignment:	<input checked="" type="radio"/> Pin	<input type="radio"/> Column
Draw:	<input checked="" type="radio"/> Circles	<input type="radio"/> Squares
Number of Rows =	<input type="text" value="100"/>	
Default Element =	<input type="text" value="0"/>	<input type="button" value="Use Group Identity"/>

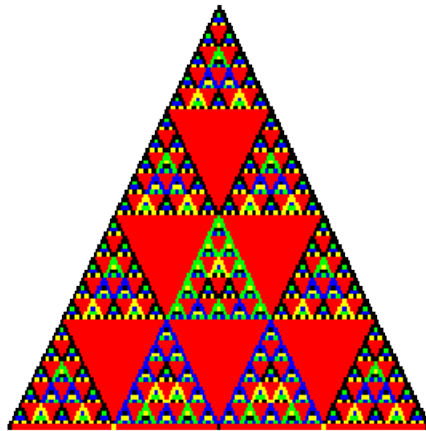
and the Options tab for the finite 1-D Automata looks like,

Automata:	<input type="radio"/> Infinite	<input checked="" type="radio"/> Finite
Update:	<input checked="" type="radio"/> Wrap	<input type="radio"/> Do Not Wrap
Draw:	<input checked="" type="radio"/> Circles	<input type="radio"/> Squares
Number of Rows =	<input type="text" value="100"/>	
Default Element =	<input type="text" value="0"/>	<input type="button" value="Use Group Identity"/>

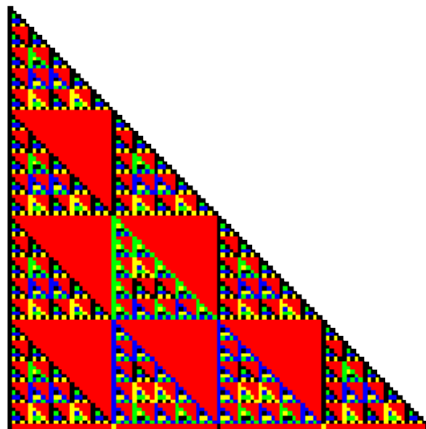
We will discuss each of the options in the order they appear.

Automata: This is simply a selection between generating a finite or infinite automaton.

Alignment: If you are creating an infinite automaton you can set the way the data is arranged. The pin alignment will arrange the data in a manner similar to the way we usually write Pascal's Triangle. For example,



The same data aligned in a column alignment would look like,



Update: If you are creating a finite automaton you can set the update rule to either wrap or not wrap. The process of wrapping is discussed in the Overview of 1-D Cellular Automata.

Draw: Simply sets the drawing mode to circles or squares.

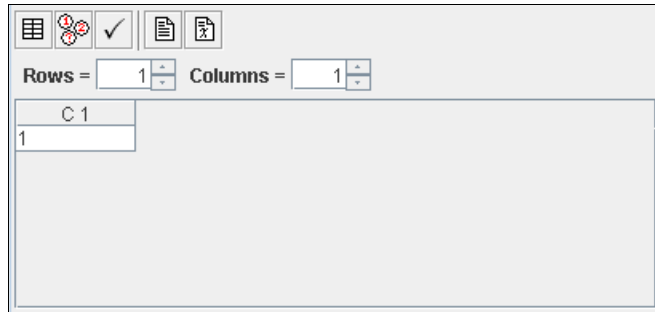
Number of Rows: Sets the number of rows that are created past the seed rows. So if you have one seed row and the number of rows is set to 100 you will graph a total of 101 rows.

Default Element: In the Overview of 1-D Cellular Automata you saw that whenever the update rule went “outside” the automaton it used the group identity in the update calculation. Furthermore, as you can see on the page discussing the seed options, if there is a seed entry that is left blank some element must be placed there as well. This program allows the user to select the element that is to be used “outside” the automaton and in the blank spaces of the seed entries. This is the Default element. In most cases you will use the group identity but the program does not force this choice. Simply input the element you wish to use as the default. If you want to use the group's identity simply click the Use Group Identity button and the identity element will be loaded into the text box. If you are working with a user-defined structure it is possible that there does not exist a unique identity element. In that case a message box will appear telling you so.

The Seed Tab

Setting the Seed

The Seed tab looks like,

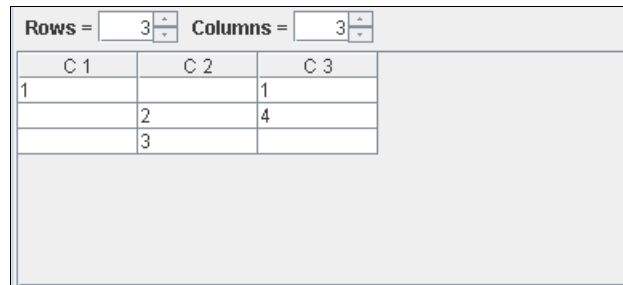


Each row of the seed represents a time-step of the automaton. If your update rule goes back only one time-step then you will need only one row in your seed. If you use more than one previous time-step in the update you may want to have more than one seed row. Recall that if the update rule must go outside the automaton (including the seed) it will use the default element.

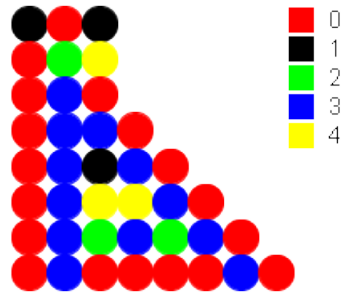
You can set the number of rows and columns by the spinners above the grid. The program can take a maximum of 20 rows and 100 columns. Once the number of rows and columns are selected simply fill in the grid with the seed elements you wish to use. For example, say we are working in Z_5 , infinite automaton, our alignment is set to column, our default element is the identity and we want the following seed.

C 1	C 2	C 3
1		1
	2	4
	3	

We would change the number of rows and columns of the seed table to three each and input the data so that our table looks like,



Once we graphed this we get the following. Note that the blank cells in the seed table have been changed to 0, the identity element in Z_5 .

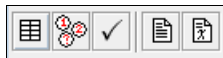


If we change the alignment to pin and regraph the automaton we get the following. It looks the same except that in the pin alignment the top is pushed over a bit.



The Seed Toolbar

There are five tools for the seed tab.

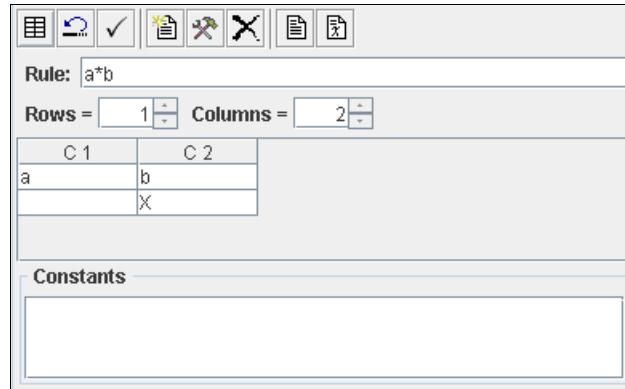


- The first simply clears the seed grid.
- The second fills the seed grid with random group elements.
- The third checks the syntax of the elements in the seed grid.
- Tools four and five copy the grid contents to the clipboard as either text or LaTeX code.

The Update Rule Tab

Setting the Update Rule

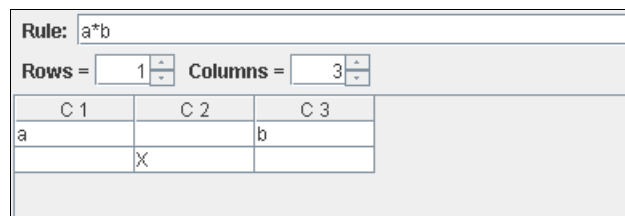
The Update Rule tab looks like,



The update rule is set, by default, to the Pascal's Triangle update rule. So if you never use an update rule other than Pascal's Triangle you have no need to alter this information. On the other hand, if you wish to experiment with other update rules here is how the updating system is set up.

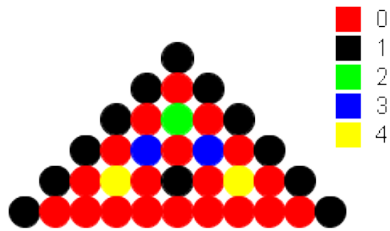
The rule is any valid group theoretic expression that uses variable names for the group elements and integers for the exponents. Juxtaposition for the group operation is not recognized, you must use a * to denote a group operation. Powers are denoted by ^ and any negative powers must be put in parentheses. Variable names can be any string of letters. Numbers, underscores and special characters are not permitted. For example, the following are legitimate update rule expressions, $a*b$, a^2*b , $b*a*b^{(-1)}$, $a*b*a^{(-1)}*b^{(-1)}$, and $a^3*b^{(-4)}$. The following are not legitimate expressions, a^2b and $a*b*a^{-1}*b^{-1}$.

The update grid is how you tell the program where to get the elements that are to be used in the update formula. There must be one and only one X in the entire grid and it must be on the last row of the grid, but it may be in any column. The position of the X tells the program that this is the cell to be updated. The other table entries are names of variables used in the update rule. When the program generates its data it will start at the X and substitute the respective cell entries in for the corresponding variables in the formula to determine the value of the new cell. For example, say we are working with Z_5 , infinite automaton, using a column alignment, a seed of just a single 1 and the following update rule.



From the grid, when we are updating a cell we will take the element from the last row and one cell to the left and substitute that element in for a in the update formula and we will take the element from the last row and one cell to the right and substitute that element in for b in the update formula. We then evaluate the formula and place the result in the X cell.

Graphing the first several rows of this automaton gives the following image.

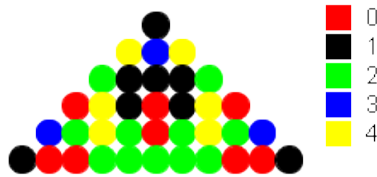


Note that each cell is the sum (mod 5) of the elements to its NW and NE.

The update rule also allows for the inclusion of constants. Constants must be group elements and are designated in the constants list at the bottom of the tab, not in the update rule expression. To create, delete and edit a constant you use the toolbar buttons, discussed below. For now, let's assume that we have Z_5 , infinite automaton, using a column alignment, a seed of just a single 1 and the following update rule.

Rule: $a*b*c$		
Rows =	<input type="text" value="1"/>	Columns = <input type="text" value="3"/>
C 1	C 2	C 3
a		b
	x	
Constants		
c = 3		

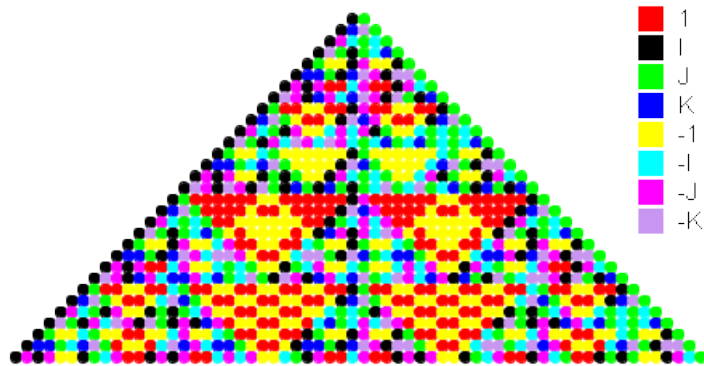
Just as in the last example, each cell is the sum (mod 5) of the elements to its NW and NE but this time we add 3 (mod 5) to that result.



As a last example, say we are working with the Quaternions, default element is the identity, seed is "i j" and has the following update rule.

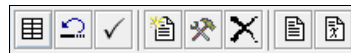
Rule: $a*b*c^{(-1)}$		
Rows =	<input type="text" value="1"/>	Columns = <input type="text" value="3"/>
C 1	C 2	C 3
a	c	b
	x	

When graphed, we have the following image.

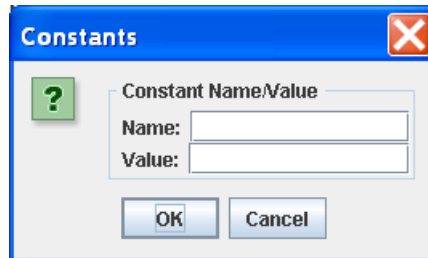


The Update Rule Toolbar

There are eight tools for the update rule tab.



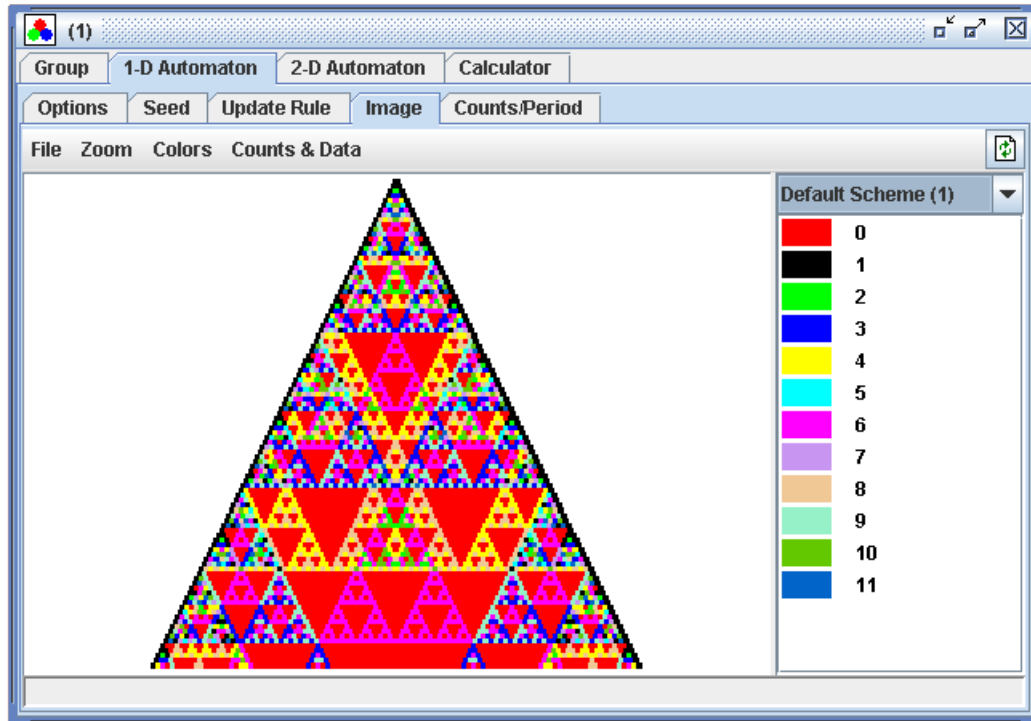
- The first simply clears the update rule grid.
- The second resets the update rule grid to the Pascal's Triangle update rule.
- The third checks the syntax of the update rule as well as the syntax of the constants.
- The fourth tool creates a new constant. When you select this tool the following dialog box appears.



- Simply type in a constant name and group element value. The name follows the same restrictions as the variable names, all letters. The value must be a legitimate group element value for whatever group you are currently working with. The syntax of the constant name and the value are not checked when you input or edit a constant. To check the syntax select the syntax checking tool.
- The fifth tool edits a constant. When you select this tool you get the same dialog box as above but the constant name is disabled. Note that you can edit a constant by double-clicking it in the list at the bottom.
- The sixth tool will delete a constant.
- Tools seven and eight copy the update rule formula, grid and constants to the clipboard as either text or LaTeX code.

The Image Tab

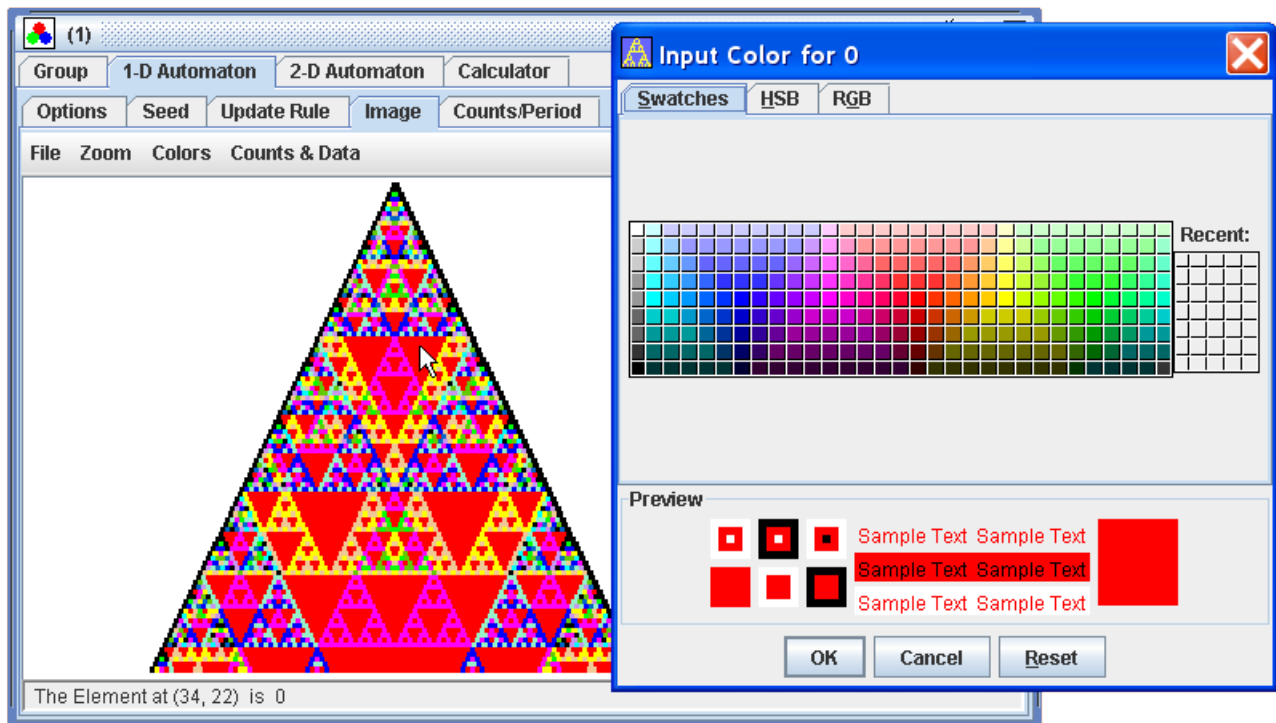
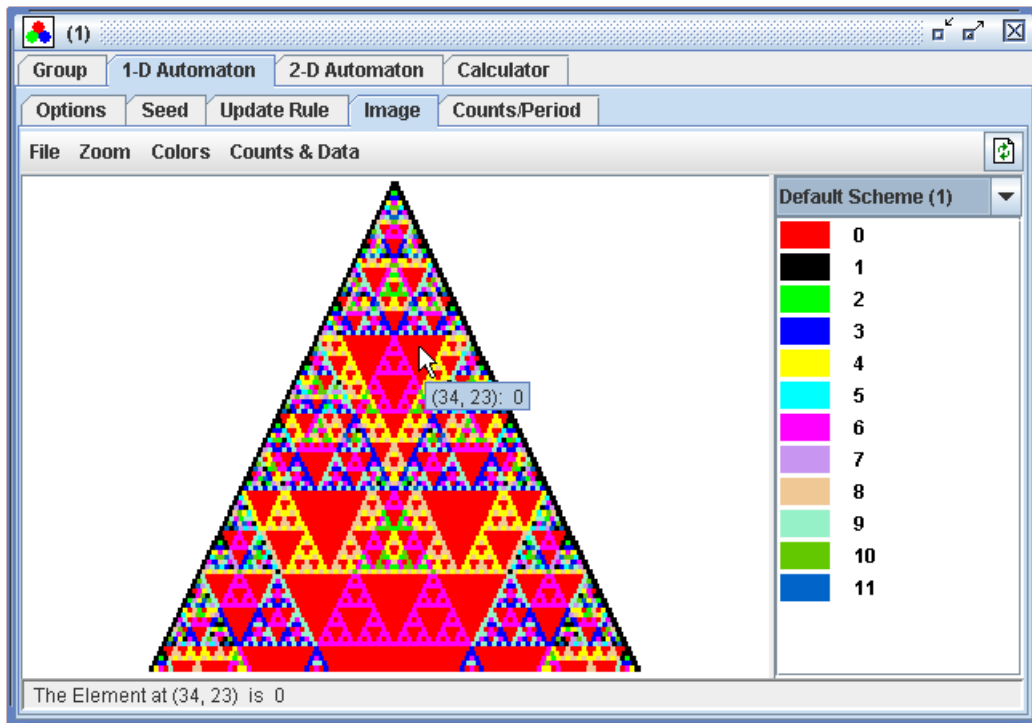
Once all of the options are set you are ready to graph the automaton. Simply select the Image tab and click the refresh button, the one in the upper right of the tab. Once you do you will see something like the following image.



The tab is fairly simple but it has many options in the menu system. We will discuss the layout and mouse options before going into each menu option. When exploring these images the main thing you as a user want to do is alter the color schemes to bring out information on the underlying structure of the group. Since regraphing the image can take some time and you will most likely be doing several color changes to achieve the desired result we do not regraph the image after every change you make. Hence, when dealing with color changes to see the change you will need to click on the refresh button in the upper right. The layout of the tab is simple. The image is to the left and the element/color correspondence is on the right. There is a menu at the top and a status bar at the bottom. Directly above the color correspondence is a color scheme selector drop-down box. Each child window may have as many color schemes as you would like. The divider between the image and the element/color correspondence is movable.

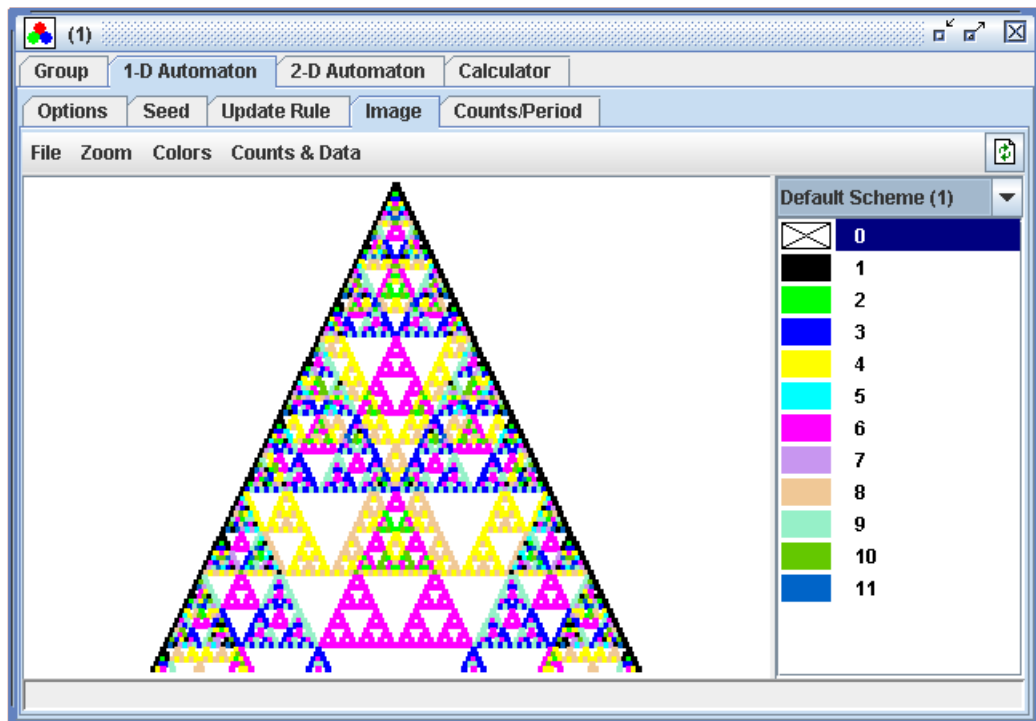
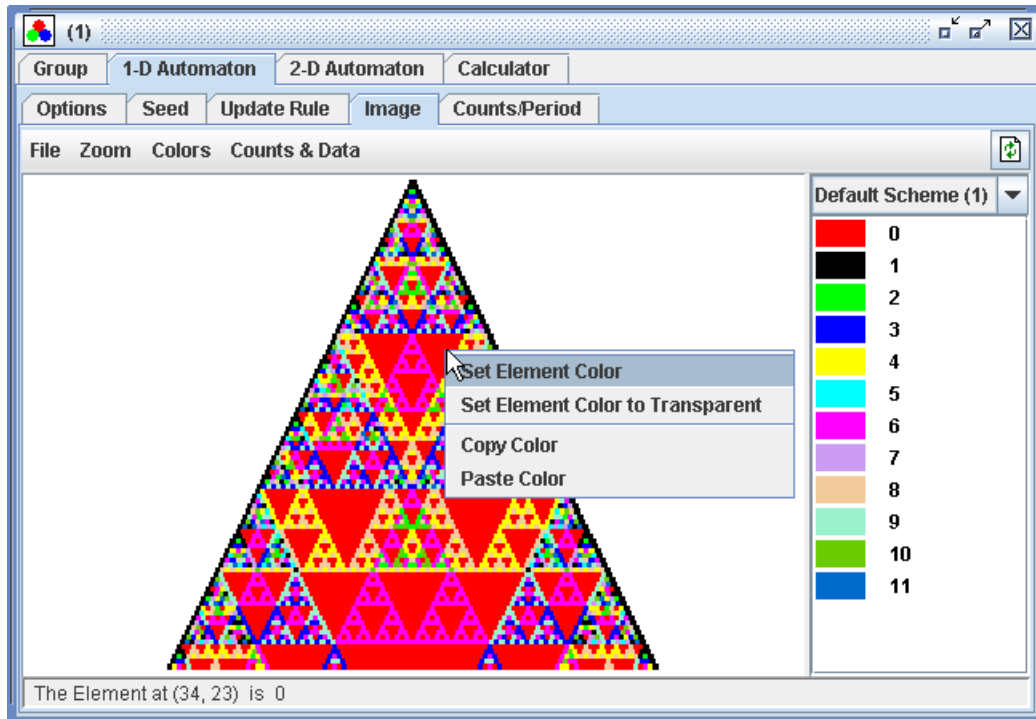
When you move the mouse over the automaton the position you are pointing at and the value of the element at that position are displayed in the status bar and if you hover for a few seconds over the automaton the position and element are also displayed as a tool tip close to the cursor.

If you double-click on an element in the image or on an element in the element list a color chooser dialog box will appear allowing you to select a new color for that element. Remember that you will need to select the refresh button to see the color change.



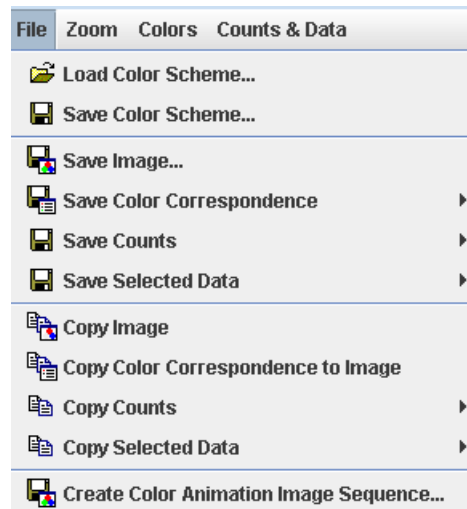
If you right-click on the image or on an element in the element list a popup menu will appear with the options to change the element's color, to make the element transparent and to copy and paste a color. If you select to change the element's color the same color chooser dialog box will come up. If

you select the transparent color option then the color in the element list for the selected element will be a box with an x through it, signifying a transparent color. When the graph is refreshed the background color will show through the transparent elements. Copy and paste will simply copy and paste the color to and from the system clipboard.



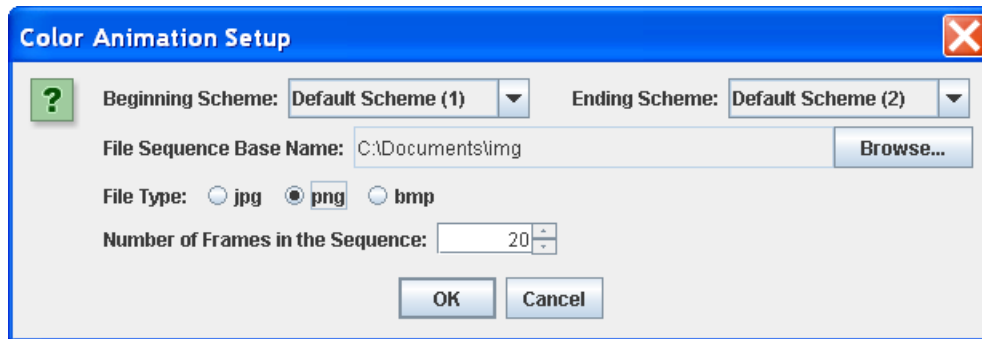
The Image Menu Options

The File Options



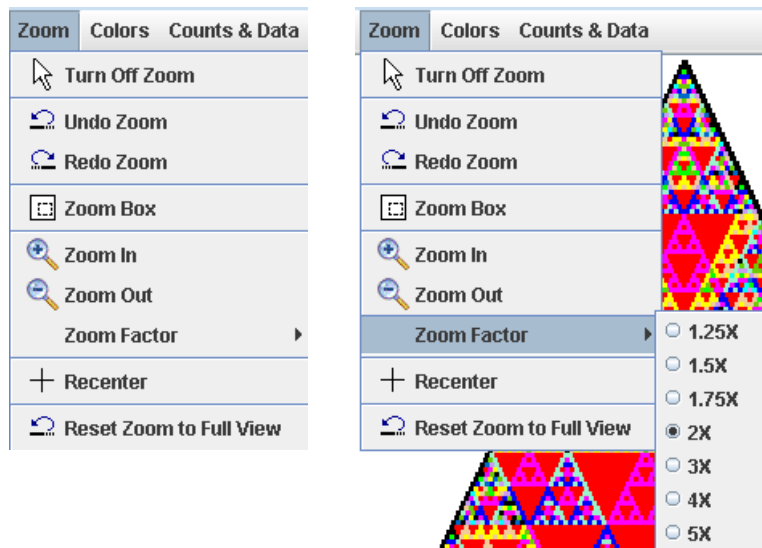
- The load color scheme options will load a previously saved color scheme into the list of current color schemes the user can choose from. If the loaded color scheme has fewer entries than the current number of elements the scheme will be cycled and if the loaded scheme has more entries than the current number of elements the scheme will be truncated. Also if the name of the saved color scheme is the same as the name of any currently loaded color schemes you will be prompted to select a different name.
- The save color scheme option will save the color scheme to a file.
- The save image option will save the current image to either a bitmap, jpeg or png file.
- The save color correspondence option will save the color/element correspondence to either a bitmap, jpeg or png file. There are two sub-options for this menu item, one is to save the correspondence as a single image and the other is to save it as multiple images. If you choose a single image the program will create an image of the color/element data that has 25 elements per column and as many columns as it takes to fit all of the elements. If you choose multiple images the program will save a sequence of image files with 25 elements per file.
- The save counts item will save element counts to either a text file or as a LaTeX table. See the section on counts below.
- The save selected data option will save the data elements in the current selected region to either a text file or a LaTeX array. See the section on counts below for a more detailed description of the process for selecting regions.
- The copy image, color correspondence, counts and data are the same as their saving counterparts except that the information is copied to the clipboard. Also, as for the color correspondence, when copied to the clipboard it is copied as a single image.
- The color animation image sequence option will create a sequence of images (bitmap, jpeg or

png) that go from one color scheme to another. To use this option you need to have at least two color schemes loaded. When you select this option you will get the following dialog.



- Select the beginning and ending color schemes by using the drop-down boxes at the top. Select a base filename by selecting the browse button and typing in the directory and name you want. The actual filenames will be the base name followed by an underscore and then a four digit number, in sequence, of course. Select the file type and the number of frames (that is, files) you want in the sequence and click OK. The program will then create a sequence of image files that smoothly go from the beginning color scheme to the ending color scheme. You can then use a different program to create a video file out of this set of images.

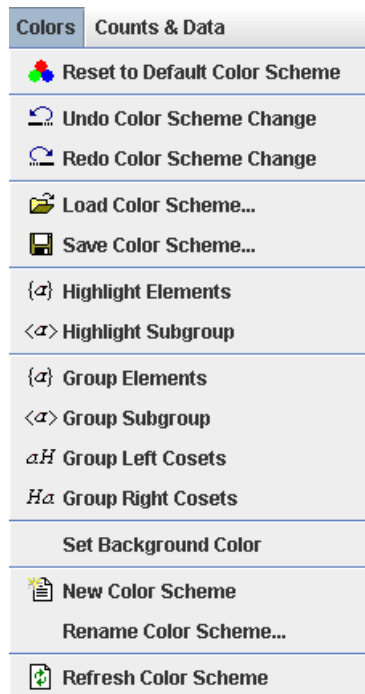
The Zoom Options



- The turn off zoom option sets the mouse actions back to their normal state. Zooming is done through mouse actions on the image and hence when a zoom mode is on the mouse has more functionality than usual. There are times when it is convenient to turn this mode off. This option will also turn off any selection mode, discussed later.
- The undo and redo options will undo and redo zooms. The program will hold a maximum sequence of 50 zooms.

- The zoom box option is one of the three zoom modes. When selected the cursor will change and when you click and drag over the image a shaded box will be drawn. When you release the mouse the program will zoom to the box you selected. Due to having to keep the aspect ratio the same you may get more than what you selected but you will not get any less. To cancel a zoom, click the right mouse button while the left mouse button is still down.
- The zoom in and zoom out options work the same way. When in either of these modes you simply click on the image and the program will zoom the current zoom factor place the position of the click in the center of the zoomed window. The current zoom factor can be set in the zoom factor option.
- The recenter option will simply recenter the image at the position of the click.
- The reset zoom to full view option will zoom out so that you can see the entire automaton image.

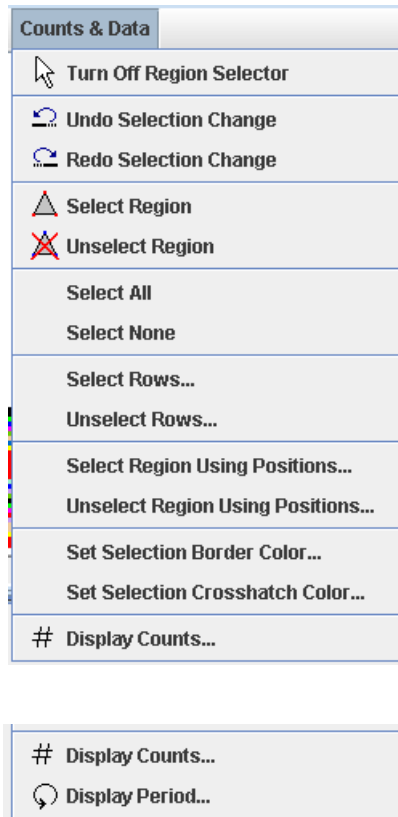
The Colors Options



- The reset to default color scheme option will reset the current color scheme to the default colors. This scheme consists of a cycle of 60 distinct colors that begin with the standard VGA 16 and then continue with combinations of these 16 to form a scheme similar to the SVGA 64 sequence of colors. Most of the options in this menu will change the color scheme and moreover link colors together so that changing one color changes the color of all those elements that are linked to it. This option will not only reset the colors to their original state but it will also break all of the color links, in essence, wiping the slate clean.
- The undo and redo options will undo and redo color scheme changes. The program will hold a maximum sequence of 20 scheme changes.

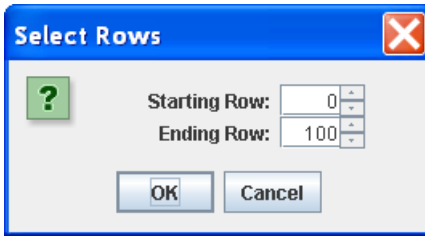
- The load color scheme options will load a previously saved color scheme into the list of current color schemes the user can choose from. If the loaded color scheme has fewer entries than the current number of elements the scheme will be cycled and if the loaded scheme has more entries than the current number of elements the scheme will be truncated. Also if the name of the saved color scheme is the same as the name of any currently loaded color schemes you will be prompted to select a different name.
- The save color scheme option will save the color scheme to a file.
- The highlight elements option will color all of the currently selected elements in the element list one color and all of the other elements another color. Furthermore, these sets of colors are now linked together. So if you change the color of an element with the color chooser all of the elements in that group are also changed to the new color.
- The highlight subgroup option will take all of the selected elements in the element list, derive the subgroup generated by these elements, color all of the subgroup elements one color and all the other elements another color. As with the highlighting elements option the elements are linked as well.
- The group elements option will take the currently selected elements, link (or group) them and color them all the same color. The difference between this and the highlight elements option is that the unselected elements are unaltered.
- The group subgroup option will take the currently selected elements, derive the subgroup generated by these elements, color all of the subgroup elements one color and link them. The difference between this and the highlight subgroup option is that the elements that are not in the subgroup are unaltered.
- The group left cosets and right cosets option will take the currently selected elements, derive the subgroup generated by these elements, color all of the subgroup elements one color and link them. It will then find each coset link and color them.
- The set background color option will invoke a color chooser dialog box to allow the user to select a different background color.
- The new color scheme option will load a new (default) color scheme into the list of available color schemes.
- The rename color scheme option will bring up a dialog box that will allow you to select a new name for the color scheme.
- The refresh color scheme will simply refresh the color/element list, it will not refresh the image. That is the job of the refresh button in the upper right. Theoretically you should never need to use this option since the color correspondence is updated after each change to the color scheme.

The Counts & Data Options

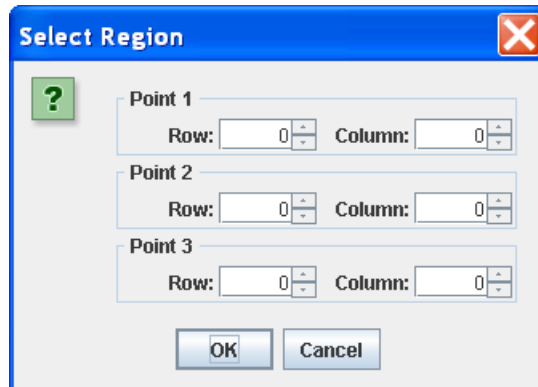


- The turn off region selector option sets the mouse actions back to their normal state. Selecting regions for counts or data extraction is done through mouse actions on the image and hence when a selection mode is on the mouse has more functionality than usual. There are times when it is convenient to turn this mode off. This option will also turn off any zoom mode.
- The undo and redo options will undo and redo selection changes. The program will hold a maximum sequence of 100 selection changes.
- The select region option will turn on the region selection mode. Each region that can be selected is a triangle. To select a region click on a cell entry in the image, move to the second vertex of the desired triangle and click again, finally move to the last vertex and click again. The region will then be highlighted. Multiple regions may be selected at one time. To select a line simply click twice on one endpoint. To select a point click that point three times. When selecting a point two or three times do not click the mouse too fast or the computer will see that as a double-click and think you wish to change the element's color.
- The unselect region option works the same way as the select region except that it unselects the cells in the region.
- Select all and select none options simply selects the entire automaton or unselects the entire automaton.

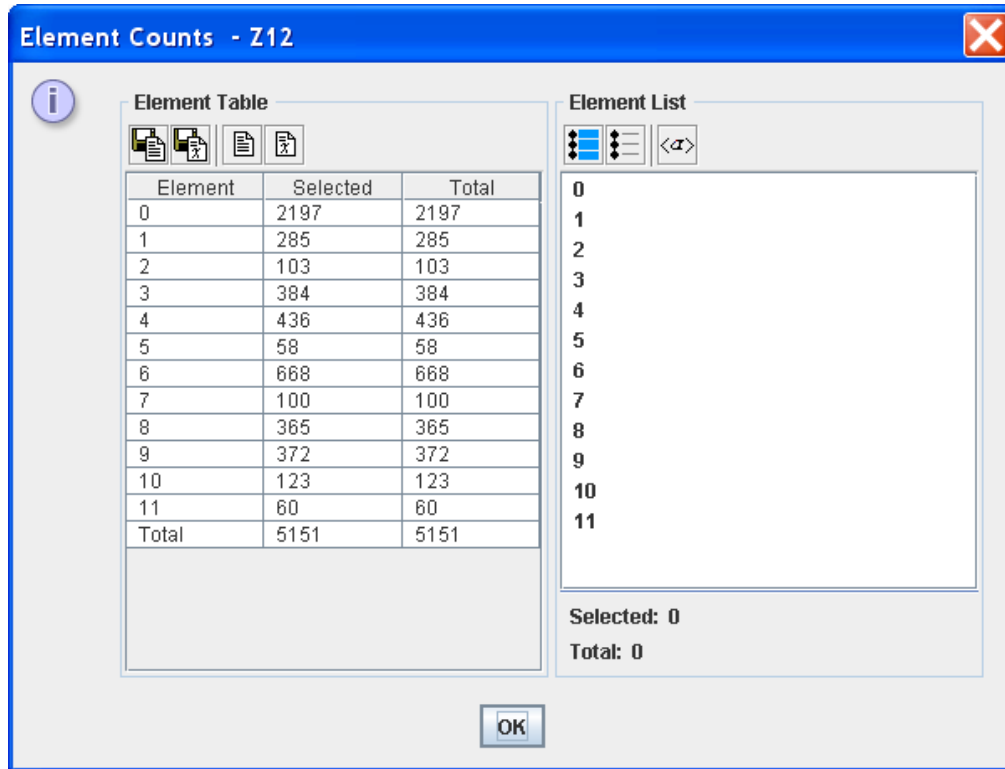
- The select and unselect rows option allows the user to select or unselect a range of rows. When this option is selected you will see a select (or unselect) rows dialog box



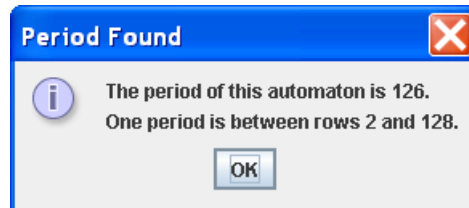
- Simply input the starting and ending rows and click OK. The rows will then be highlighted. Remember that the row numbering starts at 0.
- The select and unselect region using positions allows the user to input the positions of the triangle vertices in a dialog box, instead of by clicking on the image. When selected the following dialog box will appear. Simply put in the row and column positions of the vertices and click OK. Remember that both the row and column numbering start at 0.



- The set selection border color brings up the color chooser dialog box that allows the user to select the color of the border used in the highlighting scheme for selected cells.
- The set selection crosshatch color brings up the color chooser dialog box that allows the user to select the color of the crosshatch used in the highlighting scheme for selected cells.
- The display counts option will count the elements in the selected region as well as count the elements in the entire automaton. These counts are then displayed in the following dialog box.



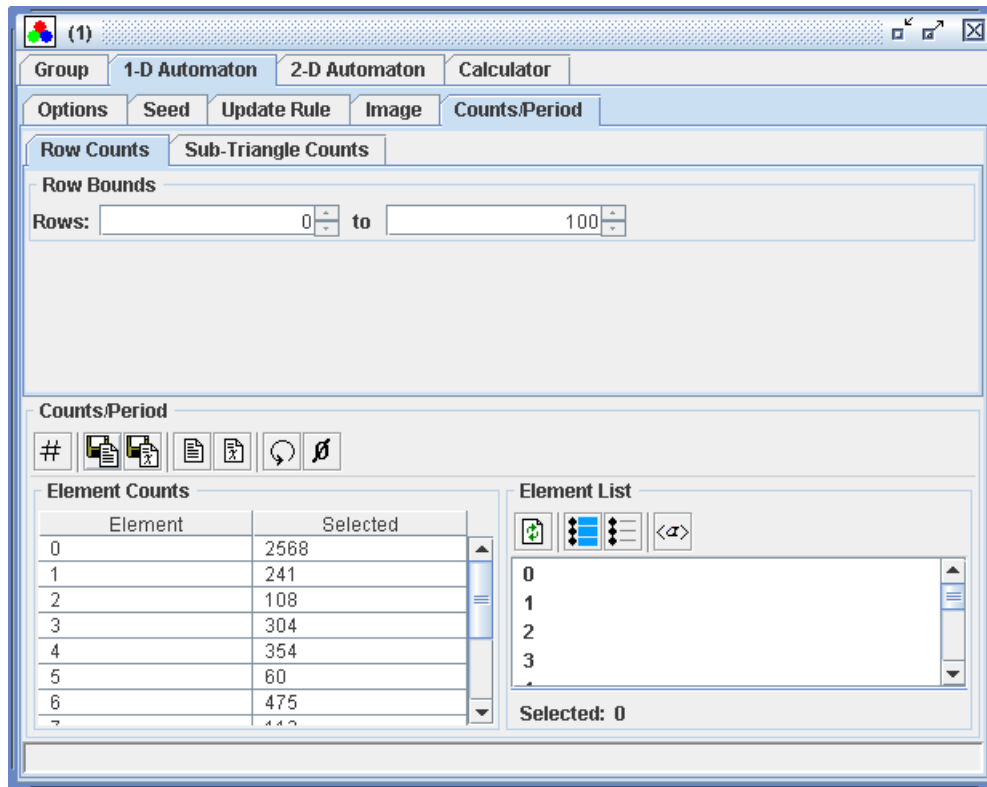
- In the counts display to the left there is a list of all the elements in the automaton and the counts of those elements in the selected region and in the total automaton. The small toolbar at the top has options for saving and copying the table of counts as text or LaTeX. On the right is another element list that can aid in the counting of subsets and subgroups. If you make a selection in this list the total of the selected and total counts is displayed at the bottom. The small toolbar at the top has three simple options. The first selects all of the elements in the list, the second unselects all of the elements in the list and the third selects the subgroup generated by the currently selected elements.
- If you are exploring a finite automaton you will see another option below the display counts, display period. When you select this option the program will determine the period of the automaton, or tell you that it could not find it. If one is found, a message box will appear with the period as well as a range of rows that contain a single period.



The Counts/Period Tab

If you wish to count the number of elements in a sequence of rows, a sub-triangle, a sub-trapezoid, or find the period or death of a finite automaton this tab may be of use to you. The calculations here are independent of those in the image panel so you can count elements in sequences of rows that extend beyond the number of rows you input in the options tab. Furthermore, since the data is regenerated each time a count is done the system does not need to store more rows than what is needed and hence one can do counts on far more rows than would be possible to graph. This tab also has more advanced options for finding the period or death of a finite automaton.

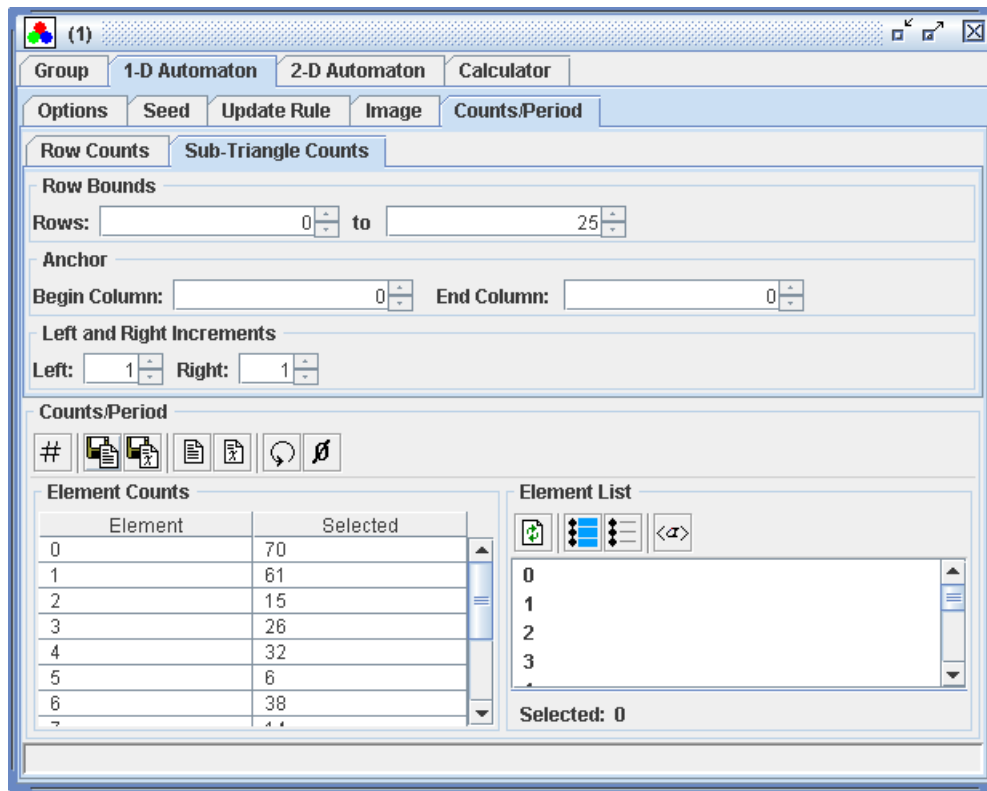
At the top of the tab is another tab system that allows the user to select the mode for counting. One mode is to simply count the elements on a sequence of rows and the other is for counting the number of elements in a sub-triangle or a sub-trapezoid. Directly below that is the counts table and element list, similar to the setup of the counts dialog box that is displayed when the display counts option is invoked from the image tab. The selected column is, of course, the number of the elements counted in the range of rows.



The Counting Mode

There are two counting modes, Row Counts and Sub-Triangle Counts. With the Row Counts mode, pictured above, the user inputs the beginning row and ending row to count, this range is inclusive. When the user selects the # button the program will display the element counts for those rows.

The Sub-Triangle Counts tab is pictured below.



In this mode the user selects an anchor (or starting point) and then tells the program how to increase (or decrease) the number of columns that are in the count on each subsequent row. This mode can be a little tricky and may take some time to get used to. We would suggest that the user try smaller examples and refer to the graph to get a feel for the positioning before doing large counts without the aid of the graph.

We will discuss the options in general and then look at a few examples. The position numbering for this mode is exactly the same as the numbering of the graphed image. Also, the Sub-Triangle title of this mode is really a misnomer, this mode will also count sub-trapezoids.

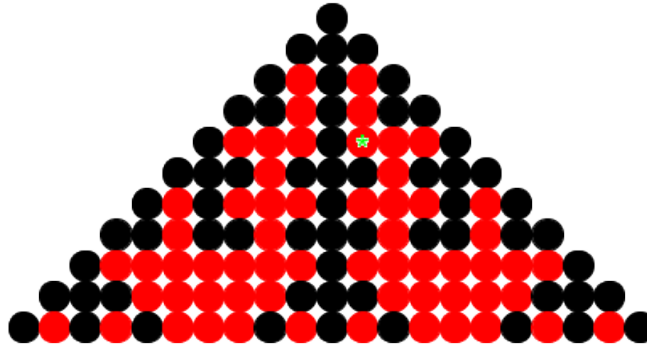
First, the user must select the range of rows. The first row is the anchor row, this is where the triangle or trapezoid starts. Note that the first row is not necessarily the smallest numbered row. If the user inputs rows 5 to 9 then the anchor row is row 5 but if the user inputs the rows 9 to 5 then the anchor row is row 9. Furthermore, in the 5 to 9 case the triangle or trapezoid will grow from row 5 down to row 9 and in the 9 to 5 case the triangle or trapezoid will grow from row 9 up to row 5, that is a triangle pointing down.

Second, the user selects a beginning column and an ending column for the anchor. This makes all of the cells on the beginning row from the beginning column to the ending column the anchor. These cells are also in the counts.

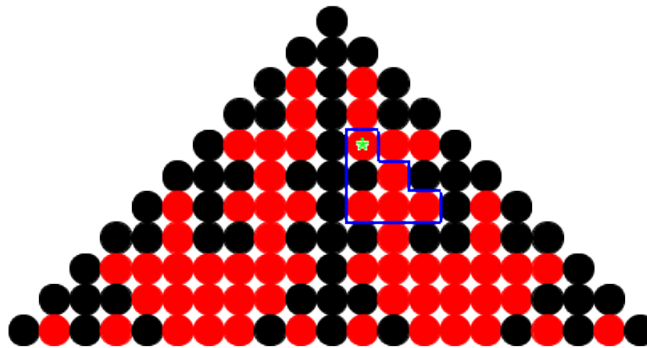
Finally, the user selects how the counts will be done on subsequent lines using the left and right updates. The way this works is that the beginning column of the next row will be the old beginning column moved to the left the number of cells in the left box and the ending column of the next row will

be the old ending column moved to the right the number of cells in the right box. So if the current row is counting from column 2 to 5 and both the left and right are set to 1 then the next row will count cells from 1 to 6 and then the row after that will count columns 0 to 7 and so on. Note that if a cell is to be counted and it is outside the automaton it will simply be ignored.

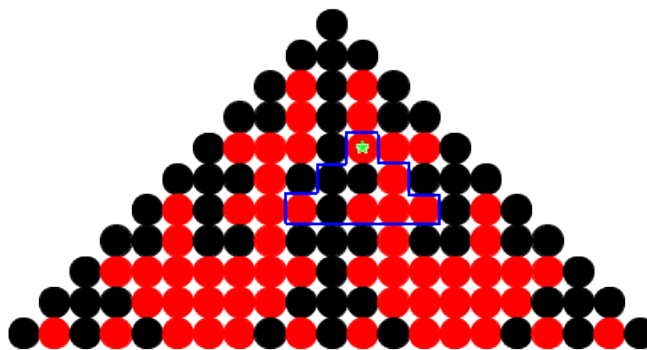
For example, in the image below the cell marked with a green "*" is at row 4 and column 5.



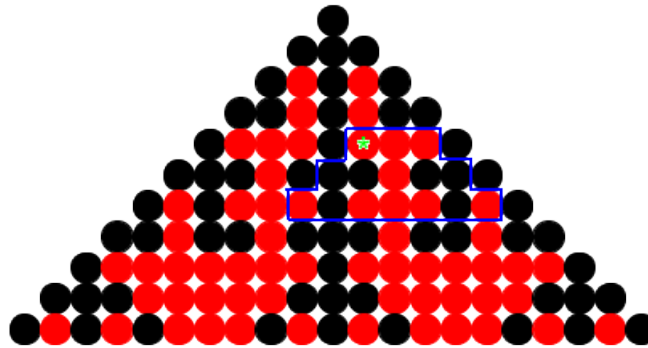
If we set the row range from 4 to 6, the beginning and ending columns to 5 and 5, the left increment to 0 and the right increment to 1 we would count the region in blue below.



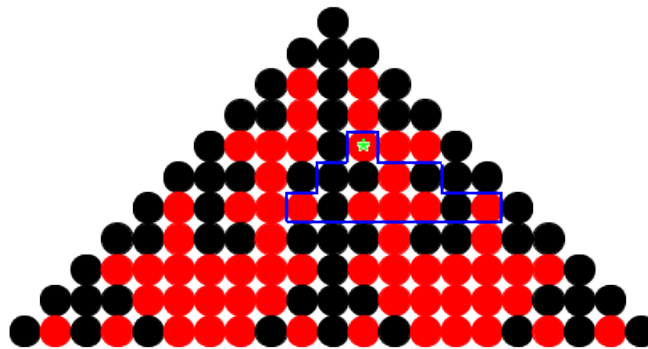
If we set the row range from 4 to 6, the beginning and ending columns to 5 and 5, the left increment to 1 and the right increment to 1 we would count the region in blue below.



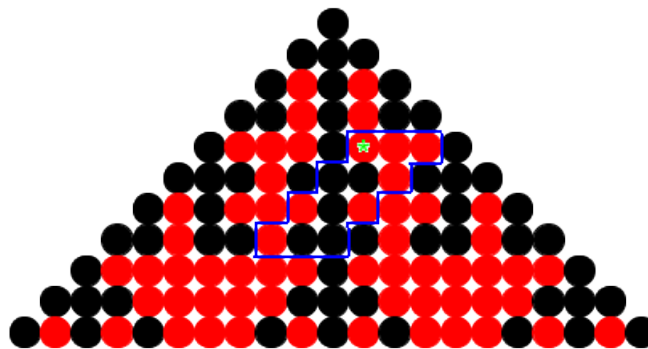
If we set the row range from 4 to 6, the beginning and ending columns to 5 and 7, the left increment to 1 and the right increment to 1 we would count the region in blue below.



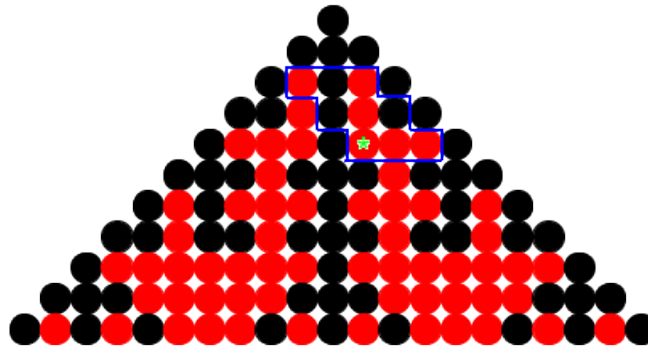
If we set the row range from 4 to 6, the beginning and ending columns to 5 and 5, the left increment to 1 and the right increment to 2 we would count the region in blue below.



Finally, if we set the row range from 4 to 7, the beginning and ending columns to 5 and 7, the left increment to 1 and the right increment to -1 we would count the region in blue below.



Note that you can also have the anchor at the bottom of the region and count up to the top by having the larger row value first. For example, if we set the row range from 4 to 2, the beginning and ending columns to 5 and 7, the left increment to 1 and the right increment to -1 we would count the region in blue below.



The Counts/Period Toolbar

The Counts/Period Toolbar consists of seven options (five for infinite automata).

- The first generates the data and displays the new counts in the table.
- The second and third save the count data to either a text file or a LaTeX file.
- The fourth and fifth copy the counts to the clipboard as either text or LaTeX code.
- When the automaton is finite there will be a sixth option to find the period. The period calculation here is a bit different from the period calculation in the image tab. In the image tab the program will start with row 0 and try to find the period starting at row 0. If this is not found it will try to find the period by starting at row 1, and then row 2 and so on. So if a period is in the graphed set it will be found. In the counts/period tab the period is found a bit differently. The program will try only once starting at the beginning row (the lowest numbered row that is input by the user). So it is possible for the image tab to find the period and the counts tab not to, if the beginning row is not set high enough. What you want to do is set the beginning row fairly large, possibly larger than the number of rows you have set in the options tab and then set the ending row very large.
- When the automaton is finite there will also be a seventh option to find the death of the automaton. The calculation of the automaton death searches for a row (or set of n rows if the depth of the update rule is n) that match the group identity. If a set of identity rows are found the program will tell you where the death starts. That is, the last row number of the first block of identity rows that are the same depth as the update rule.

The Element List Toolbar

The element list toolbar has four simple options.

- The first option refreshes the element list. Theoretically you should never need this option but as with the image tab the calculations are done in separate threads and if the element list for some reason does not appear this option will display it.
- The second option selects all of the elements in the list.
- The third option unselects all of the elements in the list.
- The fourth option selects the subgroup generated by the currently selected elements.

2-D Cellular Automata

Overview

Two-dimensional cellular automata comes in two flavors, finite and infinite. Both work in a similar manner. Both start with a seed, which is one or more levels of elements thought of as sitting on a plane. This level is considered to be a time-step, specifically the initial time-step. Every time-step (or level) after the initial one is created by applying some update rule to the previous time-step or time-steps. This process is continued indefinitely. For example, say we use a seed level that contains just a single one.

$$1$$

For our update rule we will add the North, South, East and West entries of the previous level. If there is no element in one of the positions we simply use 0. The second row is now generated from the first giving.

$$\begin{array}{ccc} & 1 & \\ 1 & & 1 \\ & 1 & \end{array}$$

The third row is generated from the second in the same manner,

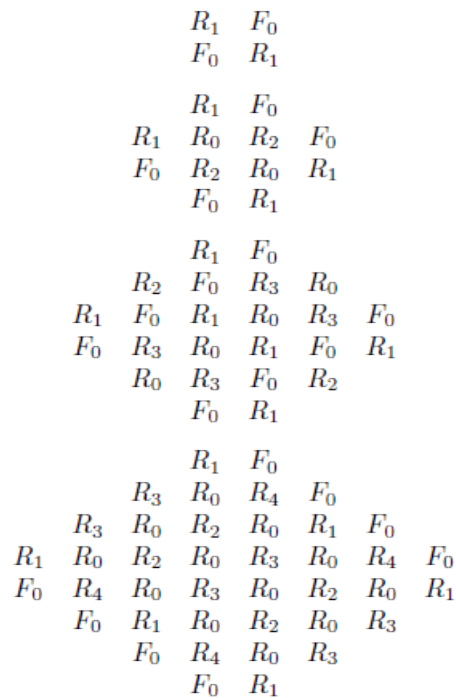
$$\begin{array}{ccc} & 1 & \\ 2 & & 2 \\ 1 & 4 & 1 \\ 2 & & 2 \\ & 1 & \end{array}$$

The fourth row is generated from the third in the same manner.

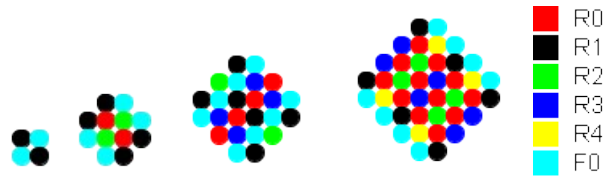
$$\begin{array}{ccc} & 1 & \\ 3 & & 3 \\ 3 & 9 & 3 \\ 1 & 9 & 9 & 1 \\ 3 & 9 & 3 \\ 3 & & 3 \\ & 1 & \end{array}$$

In the theory this process continues indefinitely.

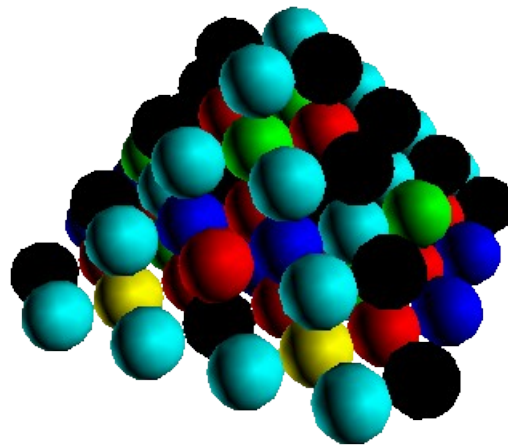
As with the one-dimensional automata the entries in the automaton are elements from a finite group. So if we are working in D_5 using a seed of R_1 and F_0 the first 4 levels would look like.



As with the one-dimensional automata the entries in the automaton are then colored to help the user to see algebraic structure and patterns. The images of the first four levels that were generated above would look something like the following.



If we then stack these levels we see a three-dimensional version of this automaton.



The program is also capable of accepting seeds with more than one level and update rules that span

more than just the previous level, or time-step. That is you can update an element by reaching back further than just the previous time-step. Furthermore, the program is capable of using a general update rule formula. You will learn more about these features on their specific pages.

Finite Automata

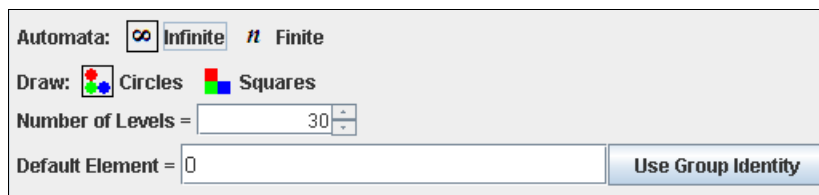
A two-dimensional finite automaton has a finite fixed width and height for each level. The update rule either wraps around or it does not. For a two-dimensional automaton there are two directions that the wrapping can be done, horizontally and vertically. Furthermore, the wrap could be done with a twist. Hence we can graph these automata on objects such as a cylinder, torus, Mobius strip or Klein bottle.

Infinite Automata

An infinite automaton is not fixed in width or height nor is it wrapped. Like in the Pascal's Triangle examples we imagine that the seed is padded indefinitely by the group identity.

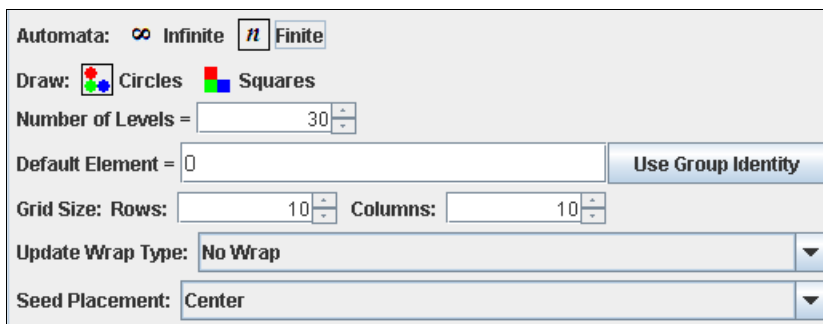
The Options Tab

The Options tab for the infinite 2-D Automata looks like,



The screenshot shows a control panel for an infinite 2-D automaton. It includes a radio button for 'Infinite' (selected) and 'Finite'. There are color-coded icons for 'Circles' (red, green, blue) and 'Squares' (red, blue). A 'Number of Levels' field is set to 30. A 'Default Element' field is empty, with a 'Use Group Identity' button to its right.

and the Options tab for the finite 2-D Automata looks like,



The screenshot shows a control panel for a finite 2-D automaton. It includes a radio button for 'Finite' (selected) and 'Infinite'. There are color-coded icons for 'Circles' (red, green, blue) and 'Squares' (red, blue). A 'Number of Levels' field is set to 30. A 'Default Element' field is empty, with a 'Use Group Identity' button to its right. Below this, there are 'Grid Size' fields for 'Rows' (set to 10) and 'Columns' (set to 10). There are also dropdown menus for 'Update Wrap Type' (set to 'No Wrap') and 'Seed Placement' (set to 'Center').

We will discuss each of the options in the order they appear.

Automata: This is simply a selection between generating a finite or infinite automaton.

Draw: Simply sets the drawing mode to circles or squares.

Number of Levels: Sets the number of levels that are created past the seed levels. So if you have one seed level and the number of levels is set to 30 you will graph a total of 31 levels.

Default Element: In the Overview of 1-D and 2-D Cellular Automata you saw that whenever the

update rule went “outside” the automaton it used the group identity in the update calculation. Furthermore, as you can see on the page discussing the seed options, if there is a seed entry that is left blank some element must be placed there as well. This program allows the user to select the element that is to be used “outside” the automaton and in the blank spaces of the seed entries. This is the Default element. In most cases you will use the group identity but the program does not force this choice. Simply input the element you wish to use as the default. If you want to use the group's identity simply click the Use Group Identity button and the identity element will be loaded into the text box. If you are working with a user-defined structure it is possible that there does not exist a unique identity element. In that case a message box will appear telling you so.

Grid Size: This is the size of the grid used for a finite 2-D automaton.

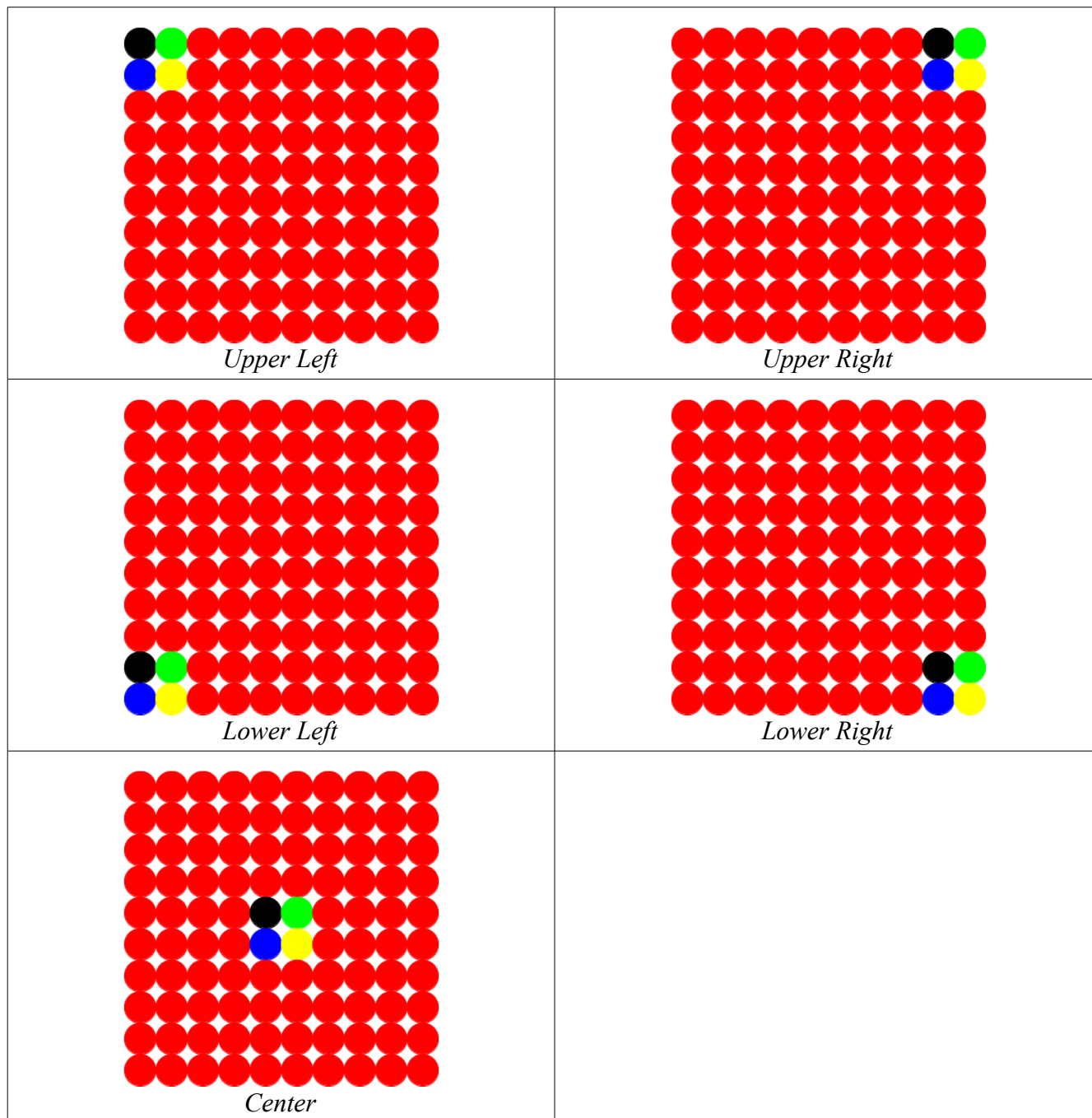
Update Wrap Type: If you are creating a finite automaton you can set the update rule to either wrap or not wrap in either direction and you can set it up to twist the wrap. The options here are

- No Wrap - No wrapping is done either horizontally or vertically.
- Cyl. Hor. / No Vert. - Horizontally wrapping is done and vertically no wrapping is done. In effect, the automaton is graphed on a vertical cylinder.
- Cyl. Vert. / No Hor. - Vertically wrapping is done and horizontally no wrapping is done. In effect, the automaton is graphed on a horizontal cylinder.
- Torus - Wrapping is done both vertically and horizontally. In effect, the automaton is graphed on a torus.
- Mobius Strip Hor. / No Vert. - Horizontally wrapping is done with a single twist and vertically no wrapping is done. In effect, the automaton is graphed on a horizontal Mobius strip.
- Mobius Strip Vert. / No Hor. - Vertically wrapping is done with a single twist and horizontally no wrapping is done. In effect, the automaton is graphed on a vertical Mobius strip.
- Mobius Strip Hor. / Cyl. Vert. - Horizontally wrapping is done with a single twist and vertically wrapping is done.
- Mobius Strip Vert. / Cyl. Hor. - Vertically wrapping is done with a single twist and horizontally wrapping is done.
- Klein Bottle - Wrapping is done with single twists both vertically and horizontally. In effect, the automaton is graphed on a Klein bottle..

Seed Placement: As you will see the seed grid may not be the same size as the grid used for the finite automaton. The placement allows you to automatically position the seed grid within the automaton grid. The options are Upper Left, Upper Right, Lower Left, Lower Right, and Center. For example, say our grid is 10 X 10 and the seed is 2 X 2, below

1	2
3	4

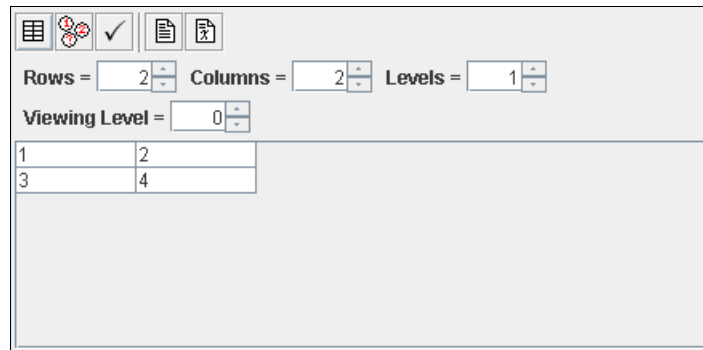
then



The Seed Tab

Setting the Seed

Setting the seed for a 2-D automaton is very similar to setting the seed for a 1-D automaton. The only difference is that each level consists of a grid of values. The Seed tab looks like,

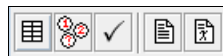


Each level of the seed represents a time-step of the automaton. If your update rule goes back only one time-step then you will need only one level in your seed. If you use more than one previous time-step in the update you may want to have more than one seed level. Recall that if the update rule must go outside the automaton (including the seed) it will use the default element.

You can set the number of rows, columns and levels by the spinners above the grid. Also the Viewing Level spinner tells you what level you are currently looking at. The program can take a maximum of 100 rows, 100 columns and 20 levels. Once the number of rows, columns and levels are selected simply fill in the grid with the seed elements you wish to use.

The Seed Toolbar

There are five tools for the seed tab.

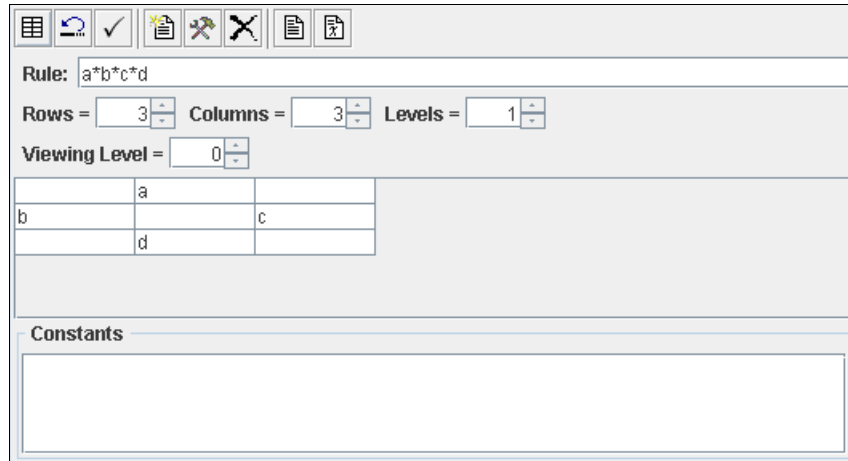


- The first simply clears the seed grid.
- The second fills the seed grid with random group elements.
- The third checks the syntax of the elements in the seed grid.
- Tools four and five copy the grid contents to the clipboard as either text or LaTeX code.

The Update Rule Tab

Setting the Update Rule

Setting the update rule for 2-D automata is similar to that for the 1-D automata. As with the seed the only difference is that you are working with levels of grids instead of rows. The Update Rule tab looks like,

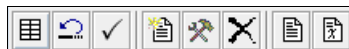


The rule is any valid group theoretic expression that uses variable names for the group elements and integers for the exponents. Juxtaposition for the group operation is not recognized, you must use a * to denote a group operation. Powers are denoted by ^ and any negative powers must be put in parentheses. Variable names can be any string of letters. Numbers, underscores and special characters are not permitted. For example, the following are legitimate update rule expressions, $a*b$, a^2*b , $b*a*b^{(-1)}$, $a*b*a^{(-1)}*b^{(-1)}$, and $a^3*b^{(-4)}$. The following are not legitimate expressions, a^2b and $a*b*a^{-1}*b^{-1}$.

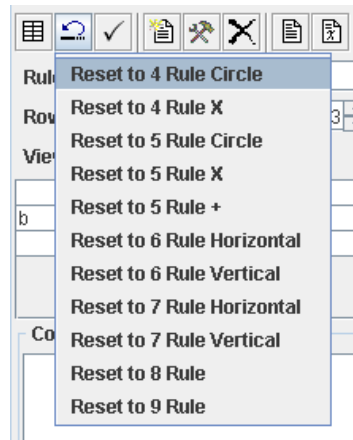
The update grid is how you tell the program where to get the elements that are to be used in the update formula. There must be one and only one X in the entire grid and it must be on the last level of the grid, but it may be in any position on that level. The position of the X tells the program that this is the cell to be updated. The other entries are names of variables used in the update rule. When the program generates its data it will start at the X and substitute the respective cell entries in for the corresponding variables in the formula to determine the value of the new cell.

The Update Rule Toolbar

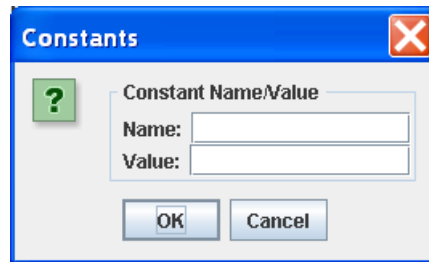
There are eight tools for the update rule tab.



- The first simply clears the update rule grid.
- The second resets the update rule grid to one of the preset update rules.



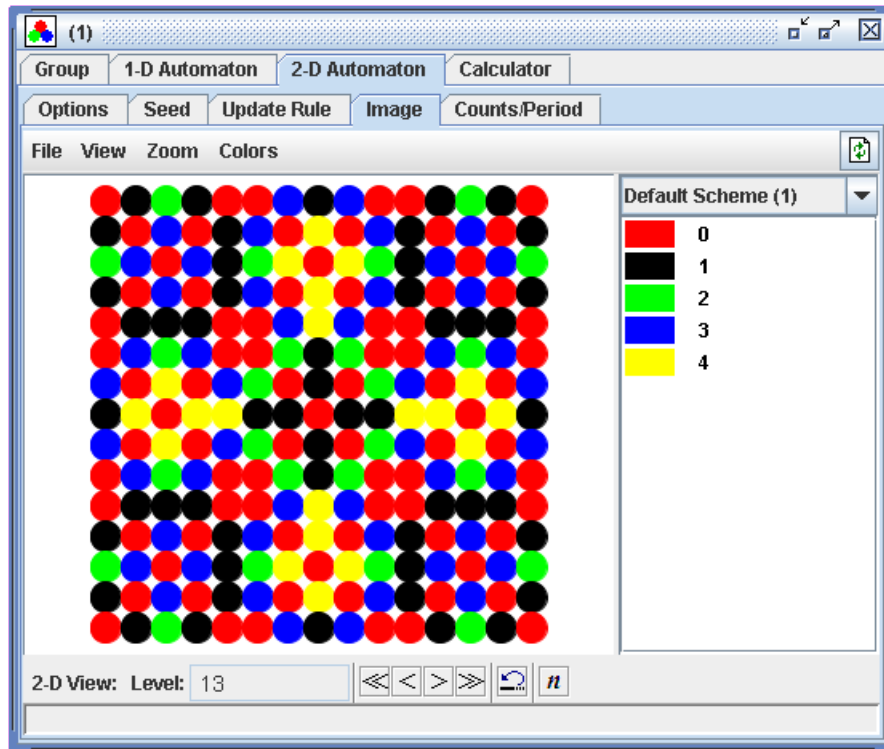
- The third checks the syntax of the update rule as well as the syntax of the constants.
- The fourth tool creates a new constant. When you select this tool the following dialog box appears.



- Simply type in a constant name and group element value. The name follows the same restrictions as the variable names, all letters. The value must be a legitimate group element value for whatever group you are currently working with. The syntax of the constant name and the value are not checked when you input or edit a constant. To check the syntax select the syntax checking tool.
- The fifth tool edits a constant. When you select this tool you get the same dialog box as above but the constant name is disabled. Note that you can edit a constant by double-clicking it in the list at the bottom.
- The sixth tool will delete a constant.
- Tools seven and eight copy the update rule formula, grid and constants to the clipboard as either text or LaTeX code.

The Image Tab

Once all of the options are set you are ready to graph the automaton. Simply select the Image tab and click the refresh button, the one in the upper right of the tab. Once you do you will see something like the following image. Here we are using a finite automaton over Z_5 .



The tab is fairly simple but it has many options in the menu system. We will discuss the layout and mouse options before going into each menu option. When exploring these images the main thing you as a user want to do is alter the color schemes to bring out information on the underlying structure of the group. Since regraphing the image can take some time and you will most likely be doing several color changes to achieve the desired result we do not regraph the image after every change you make. Hence, when dealing with color changes to see the change you will need to click on the refresh button in the upper right. The layout of the tab is simple. The image is to the left and the element/color correspondence is on the right. There is a menu at the top and a status bar at the bottom. Directly above the color correspondence is a color scheme selector drop-down box. Each child window may have as many color schemes as you would like. The divider between the image and the element/color correspondence is movable.

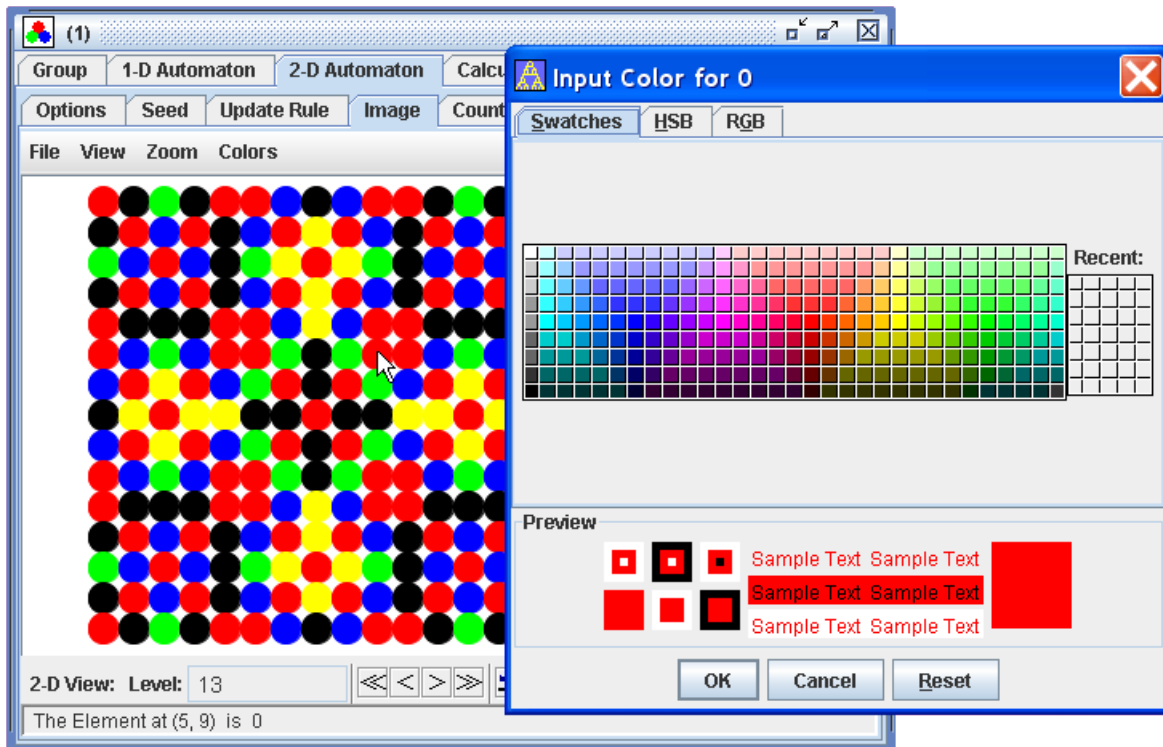
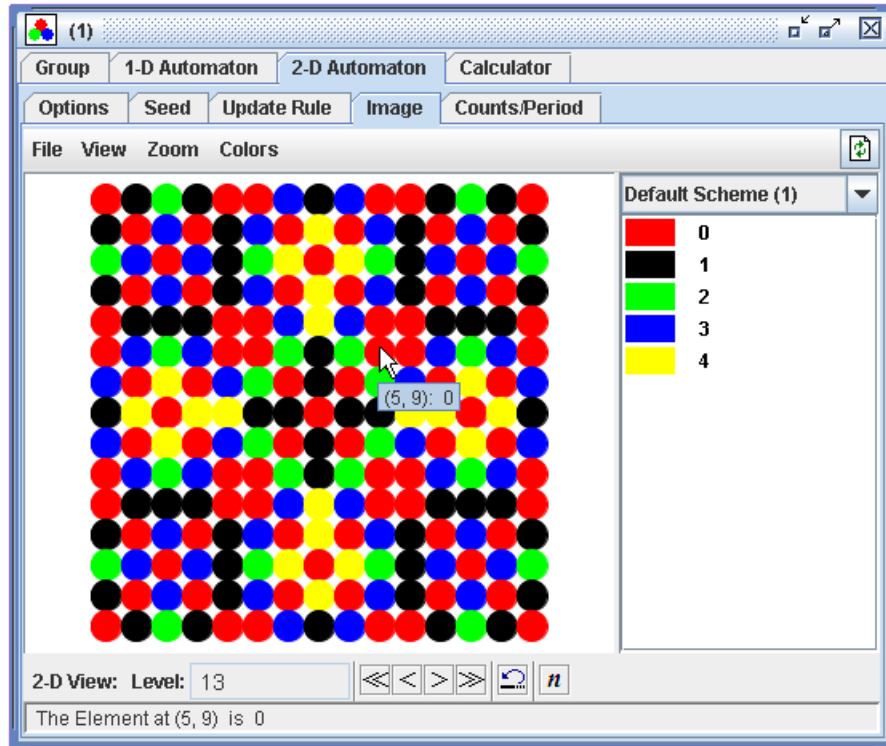
There are three viewing modes for two-dimensional automata. One is a three dimensional rendering of the stacked levels (time-steps) of the automata. This view is done in the external 3-D Viewer. The other two modes are an element viewer for a specific time-step and a probability density viewer which graphs the probability that a cell contains an element from some selected subset of a finite group.

The Time-Step Viewer

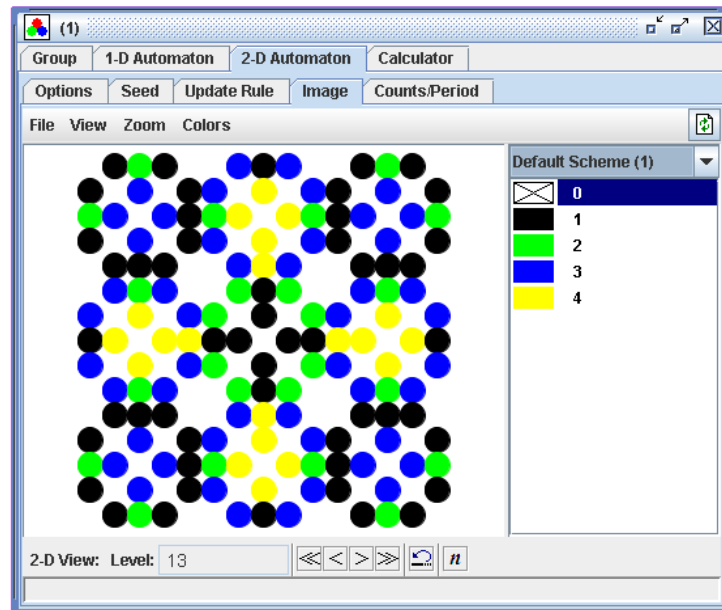
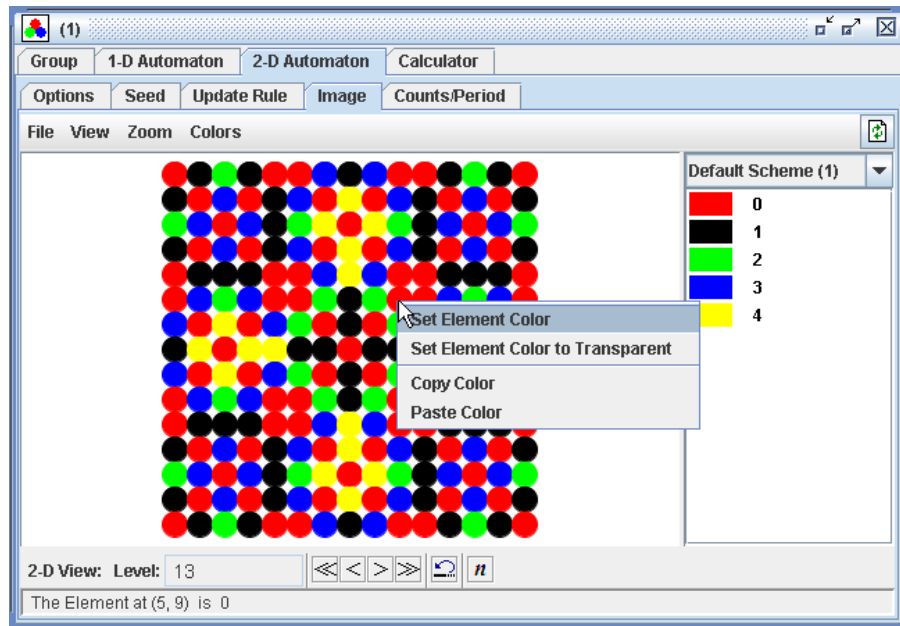
When you move the mouse over the automaton the position you are pointing at and the value of the element at that position are displayed in the status bar and if you hover for a few seconds over the automaton the position and element are also displayed as a tool tip close to the cursor.

If you double-click on an element in the image or on an element in the element list a color chooser dialog box will appear allowing you to select a new color for that element. Remember that you will

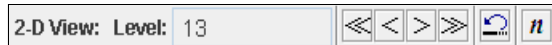
need to select the refresh button to see the color change.



If you right-click on the image or on an element in the element list a popup menu will appear with the options to change the element's color, to make the element transparent and to copy and paste a color. If you select to change the element's color the same color chooser dialog box will come up. If you select the transparent color option then the color in the element list for the selected element will be a box with an x through it, signifying a transparent color. When the graph is refreshed the background color will show through the transparent elements. Copy and paste will simply copy and paste the color to and from the system clipboard.



Before moving onto the menu options, the bottom of the tab contains the level navigation tools that allow you to view the different levels of the automaton.



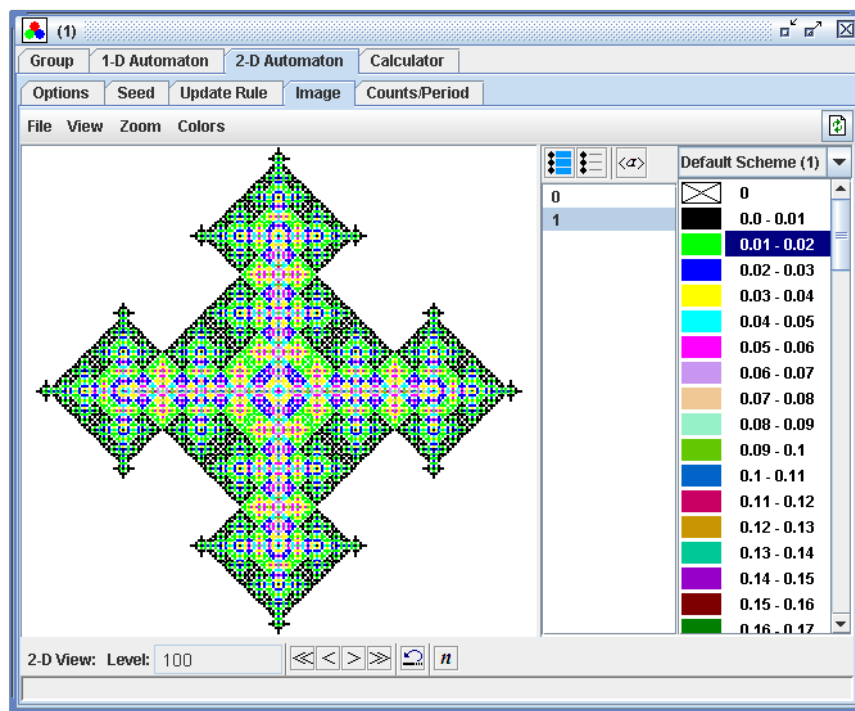
The Level box simply displays the current level that is being viewed. The first four buttons move the viewing level either forward or backward by either one level or 10 levels. The next button moves the viewing level back to the first level and the last button allows you to input the level number to view. Note that this number must be in the range of levels generated.

The Probability Density Viewer

The layout of the probability density viewer is similar to the level viewer except that there are two lists to the right. The color scheme now corresponds to subintervals of $[0, 1]$. The first color corresponds to probability 0 and all other intervals represent of the form $(a, b]$. The list to the left of the color scheme is a list of the group elements. The way this mode works is you select a subset of elements from the element list and refresh the image. For each cell in the automaton the program will calculate the probability that an element from this subset is in that cell, using level 0 to the current level.

When you move the mouse over the automaton the position you are pointing at and the value the probability of that cell are displayed in the status bar and if you hover for a few seconds over the automaton the position and probability are also displayed as a tool tip close to the cursor.

If you double-click on a cell in the image or on a probability range in the list a color chooser dialog box will appear allowing you to select a new color for that range. Remember that you will need to select the refresh button to see the color change.

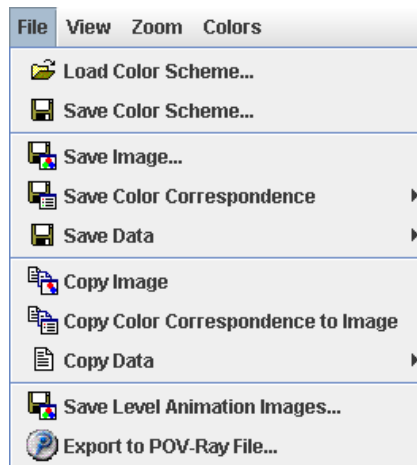


If you right-click on the image or on a probability range in the list a popup menu will appear with the options to change the element's color, to make the element transparent and to copy and paste a color. If you select to change the element's color the same color chooser dialog box will come up. If you select the transparent color option then the displayed color in the selected probability range will be an x, signifying a transparent color. When the graph is refreshed the background color will show through the transparent cells. Copy and paste will simply copy and paste the color to and from the system clipboard.

The Level box works the same way as with the level viewer. In this mode the image displayed will be the probability density from level 0 to the current level.

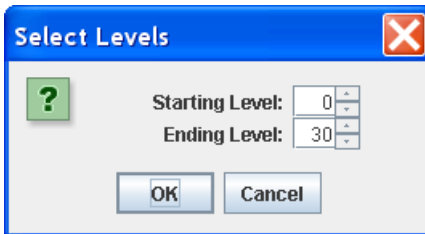
The Image Menu Options

The File Options



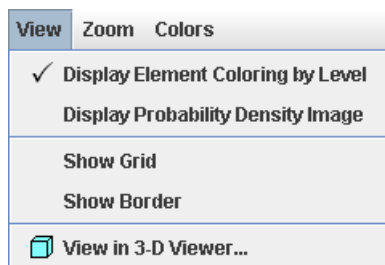
- The Load Color Scheme option will load a previously saved color scheme into the list of current color schemes the user can choose from. If the loaded color scheme has fewer entries than the current number of elements the scheme will be cycled and if the loaded scheme has more entries than the current number of elements the scheme will be truncated. Also if the name of the saved color scheme is the same as the name of any currently loaded color schemes you will be prompted to select a different name.
- The Save Color Scheme option will save the color scheme to a file.
- The Save Image option will save the current image to either a bitmap, jpeg or png file.
- The Save Color Correspondence option will save the color/element correspondence to either a bitmap, jpeg or png file. There are two sub-options for this menu item, one is to save the correspondence as a single image and the other is to save it as multiple images. If you choose a single image the program will create an image of the color/element data that has 25 elements per column and as many columns as it takes to fit all of the elements. If you choose multiple images the program will save a sequence of image files with 25 elements per file.
- The Save Data option has a submenu consisting of Current Level and Levels each of these has a

submenu allowing you to select between saving the data as a text file and a LaTeX file. Saving the current will simply save the data in the level you are currently viewing. The levels option will allow you to select a range of levels for saving.

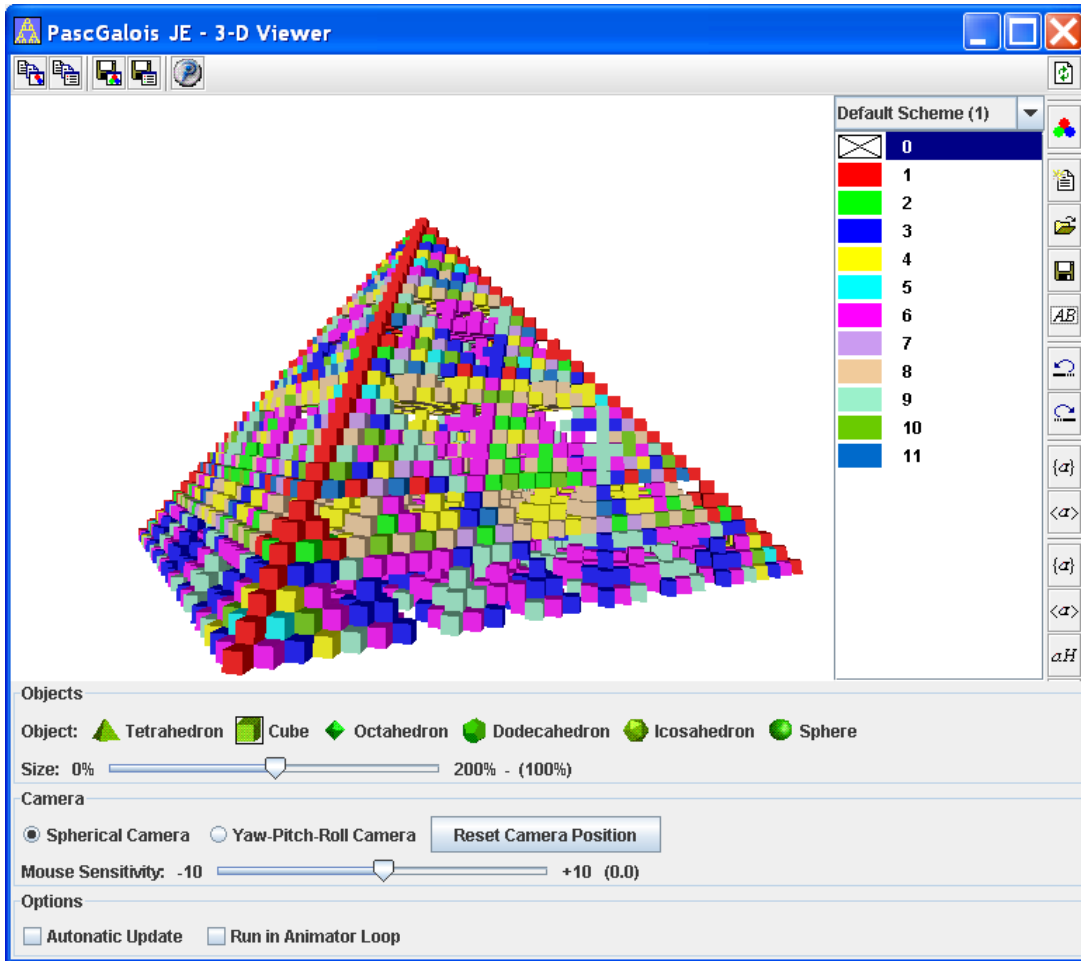


- The copy image, color correspondence, and data are the same as their saving counterparts except that the information is copied to the clipboard. Also, as for the color correspondence, when copied to the clipboard it is copied as a single image.
- The Save Level Animation Images option will render each level (time-step) of the automaton as a set of image files that can be made into an animation using an external animator. When you select this option a dialog box will appear allowing you to input a base name of the images and the type of graphics you want to save the images to (jpg, png or bmp). The image filenames will be the base image followed by an underscore and then followed by the level number.
- The Export to POV-Ray option will save the automaton as a set of files that can be read and rendered by POV-Ray. POV-Ray is a freeware ray tracer which is available on all major platforms. In Windows, there is a very nice and easy to use user interface. Most Linux implementations have a command line version that is included in the distribution. For details on the options of the POV-Ray exporter see the POV-Ray Exporter page.

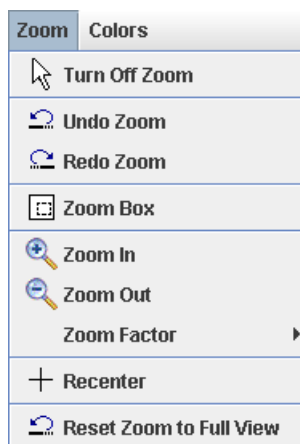
The View Options



- The first two options toggle the viewing mode between the level viewer and the probability density viewer.
- The Show Grid and Show Border options simply draw a grid or border on the automaton level.
- The View in 3-D Viewer option displays the automaton of density plot in three dimensions. For details on the options of the 3-D Viewers see the 3-D Viewers page.



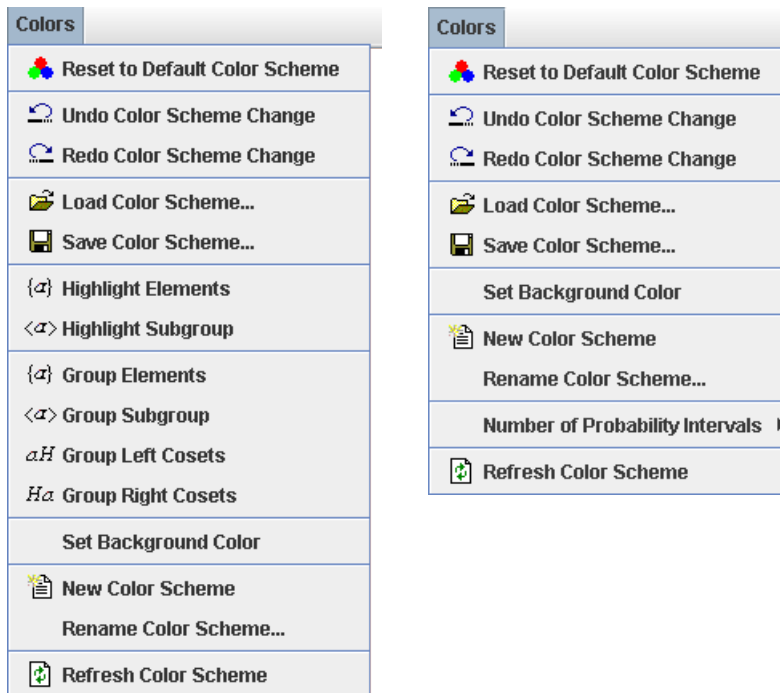
The Zoom Options



- The turn off zoom option sets the mouse actions back to their normal state. Zooming is done through mouse actions on the image and hence when a zoom mode is on the mouse has more functionality than usual. There are times when it is convenient to turn this mode off.

- The undo and redo options will undo and redo zooms. The program will hold a maximum sequence of 50 zooms.
- The zoom box option is one of the three zoom modes. When selected the cursor will change and when you click and drag over the image a shaded box will be drawn. When you release the mouse the program will zoom to the box you selected. Due to having to keep the aspect ratio the same you may get more than what you selected but you will not get any less. To cancel a zoom, click the right mouse button while the left mouse button is still down.
- The zoom in and zoom out options work the same way. When in either of these modes you simply click on the image and the program will zoom the current zoom factor place the position of the click in the center of the zoomed window. The current zoom factor can be set in the zoom factor option.
- The recenter option will simply recenter the image at the position of the click.
- The reset zoom to full view option will zoom out so that you can see the entire automaton image.

The Colors Options



The menu on the left is the one for the level viewing mode and the one on the right is for the probability density viewing mode.

- The reset to default color scheme option will reset the current color scheme to the default colors. This scheme consists of a cycle of 60 distinct colors that begin with the standard VGA 16 and then continue with combinations of these 16 to form a scheme similar to the SVGA 64 sequence of colors. Most of the options in this menu will change the color scheme and moreover link colors

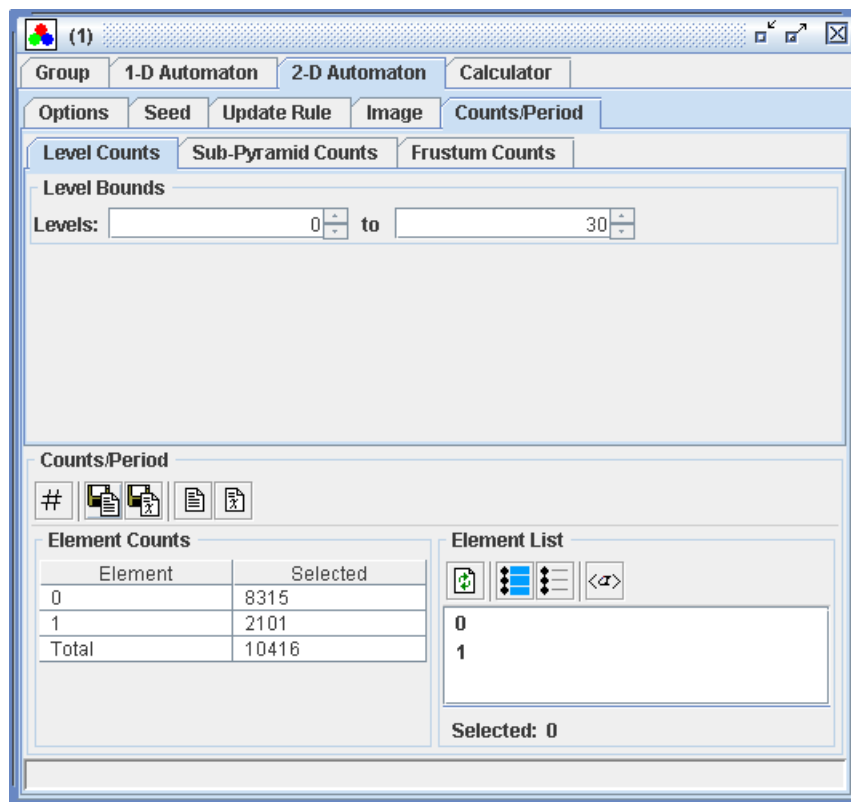
together so that changing one color changes the color of all those elements that are linked to it. This option will not only reset the colors to their original state but it will also break all of the color links, in essence, wiping the slate clean.

- The undo and redo options will undo and redo color scheme changes. The program will hold a maximum sequence of 20 scheme changes.
- The Load Color Scheme option will load a previously saved color scheme into the list of current color schemes the user can choose from. If the loaded color scheme has fewer entries than the current number of elements the scheme will be cycled and if the loaded scheme has more entries than the current number of elements the scheme will be truncated. Also if the name of the saved color scheme is the same as the name of any currently loaded color schemes you will be prompted to select a different name.
- The Save Color Scheme option will save the color scheme to a file.
- The highlight elements option will color all of the currently selected elements in the element list one color and all of the other elements another color. Furthermore, these sets of colors are now linked together. So if you change the color of an element with the color chooser all of the elements in that group are also changed to the new color.
- The highlight subgroup option will take all of the selected elements in the element list, derive the subgroup generated by these elements, color all of the subgroup elements one color and all the other elements another color. As with the highlighting elements option the elements are linked as well.
- The group elements option will take the currently selected elements, link (or group) them and color them all the same color. The difference between this and the highlight elements option is that the unselected elements are unaltered.
- The group subgroup option will take the currently selected elements, derive the subgroup generated by these elements, color all of the subgroup elements one color and link them. The difference between this and the highlight subgroup option is that the elements that are not in the subgroup are unaltered.
- The group left cosets and right cosets option will take the currently selected elements, derive the subgroup generated by these elements, color all of the subgroup elements one color and link them. It will then find each coset link and color them.
- The set background color option will invoke a color chooser dialog box to allow the user to select a different background color.
- The new color scheme option will load a new (default) color scheme into the list of available color schemes.
- The rename color scheme option will bring up a dialog box that will allow you to select a new name for the color scheme.
- The number of probability intervals lets the user select the number of different colors to use in the probability density viewing mode. The interval $[0, 1]$ is divided evenly into this number of subintervals. A cell is then colored by the interval color where the cell probability lies.
- The refresh color scheme will simply refresh the color/element list, it will not refresh the image.

That is the job of the refresh button in the upper right. Theoretically you should never need to use this option since the color correspondence is updated after each change to the color scheme.

The Counts/Period Tab

If you wish to count the number of elements in a sequence of levels, the number of elements in a sub-pyramid or frustum, or find the period or death of a finite automaton this tab may be of use to you. The calculations here are independent of those in the image panel so you can count elements in sequences of levels that extend beyond the number of levels you input in the options tab. Furthermore, since the data is regenerated each time a count is done the system does not need to store more levels than what is needed and hence one can do counts on far more levels than would be possible to graph. This tab also has more advanced options for finding the period or death of a finite automaton.

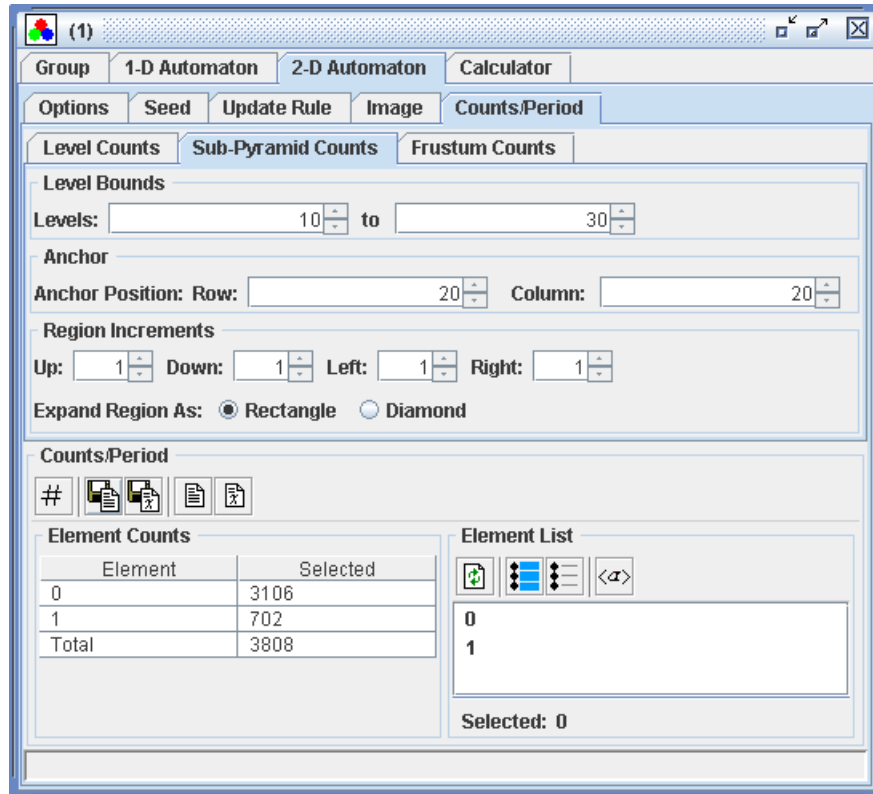


At the top you will see a tab system that allows the user to select a counting mode. The modes are level counting, sub-pyramid counting or frustum counting.

The Counting Modes

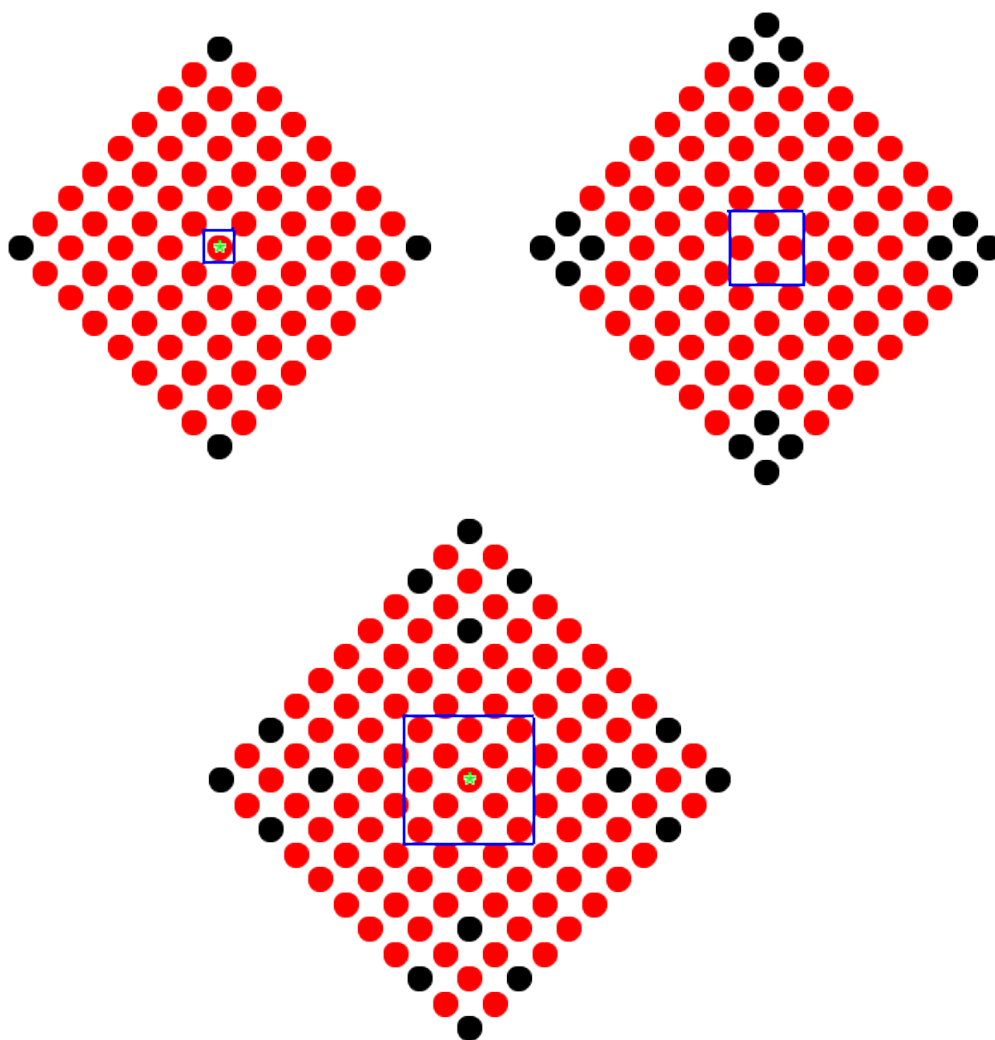
In the level counting mode at the top of the tab is the range of levels to be used. This range is inclusive. In this mode the program will count all of the elements on each level in the range.

In the sub-pyramid counting mode, pictured below, the user selects the range of levels, an anchor position and the way that the counting region is increased as you move to the next level.

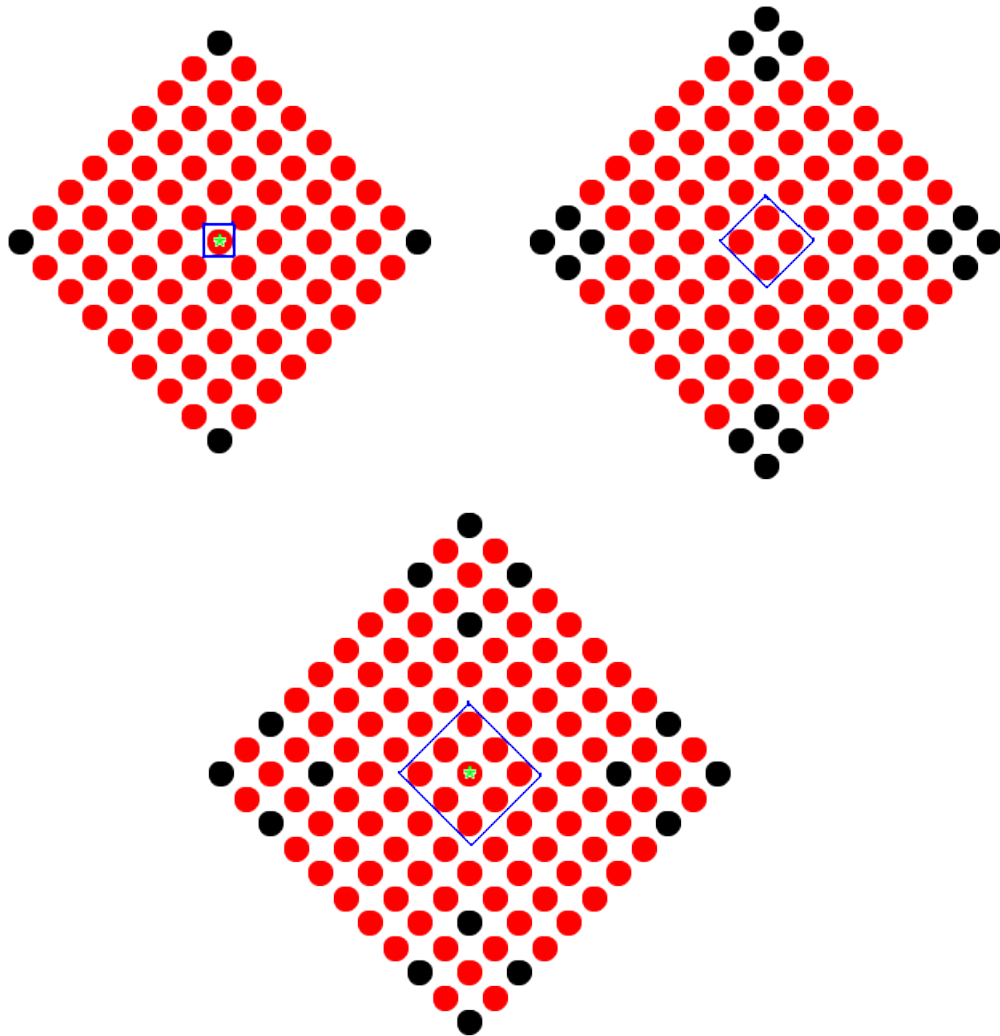


The first entry in the level range is the anchor level, this is where the sub-pyramid will start. As with the counting of the one-dimensional automata if the beginning level is larger than the ending level the sub-pyramid will be pointing downward. The user then selects the position of the "top" of the sub-pyramid. This can be a little tricky because if you change the level range the position numbering of a point may also change. For example, use the default 4-rule with a single seed element and graph 20 levels. If you place the mouse over the element on the first level in the image tab you will see that the position is row 20 and column 20. Now if you change the number of levels to 30 and regraph the automaton the position of this "top" point is row 30 and column 30. So care must be taken when selecting this position. The position numbering system that is used in the counting is the same as the numbering system used in the graphing routines. Furthermore, the numbering that is used is the same as with the image of the largest level in the range of levels that are counted. That is, if you are counting a sub-pyramid that is between levels 23 and 47 the numbering that is used is exactly the same as if you were to graph the first 47 levels of the automaton. The user should use the graphs to get a feel for the numbering system before trying to count a large pyramid without the aid of a graph.

The region increments allow the user to specify how the counting region is increased as you progress to the next level. The region can also be expanded as either a rectangle or a diamond. For example, say that the green "*" is at row 10 and column 10. If we set all increments to 1 and use the rectangular region the counting regions will be in blue.

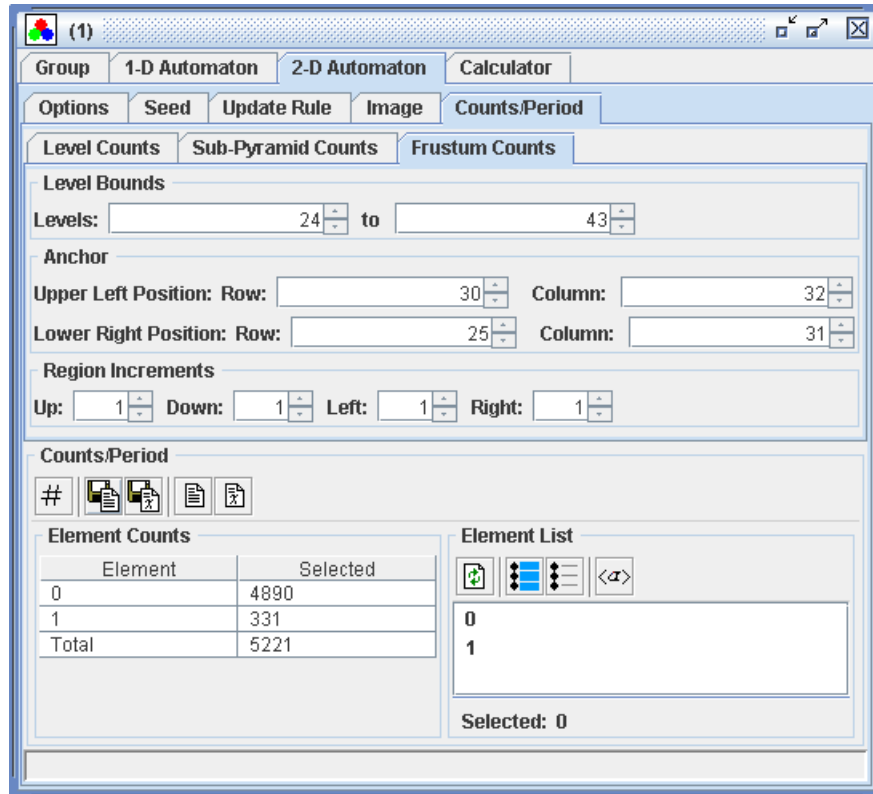


Under the same setup but using the diamond regions our counting regions would be.



As with the one-dimensional automaton counts the increments can be 0 and negative as well. For example, if you set the down to -1, keep the rest of the increments as 1 and use a rectangular region you will count the elements on one face of the pyramid.

In the frustum counting mode, pictured below, the user selects the range of levels, an anchor rectangle and the way that the counting region is increased as you move to the next level.



This mode works the same way that the sub-pyramid counting mode works except that the increase in region is always rectangular and the anchor is a rectangular region instead of a single cell. The rectangular region is selected by specifying the upper left position and lower right position of the rectangle.

The Counts/Period Toolbar

The Counts/Period Toolbar consists of seven options (five for infinite automata).

- The first generates the data and displays the new counts in the table.
- The second and third save the count data to either a text file or a LaTeX file.
- The fourth and fifth copy the counts to the clipboard as either text or LaTeX code.
- When the automaton is finite there will be a sixth option to find the period. The program will start at the beginning level, memorize a sequence of levels (with depth the same as the depth of the update rule) and search until the ending level for that sequence of levels. If found, the program will return the period and if not it will notify you that a period could not be found. In general, what you want to do is set the beginning row fairly large, and then set the ending row very large. This should increase your chances of finding the period.
- When the automaton is finite there will also be a seventh option to find the death of the automaton. The calculation of the automaton death searches for a level (or set of n levels if the depth of the update rule is n) that match the group identity. If a set of identity levels are found the program

will tell you where the death starts. That is, the last level number of the first block of identity levels that are the same depth as the update rule.

The Element List Toolbar

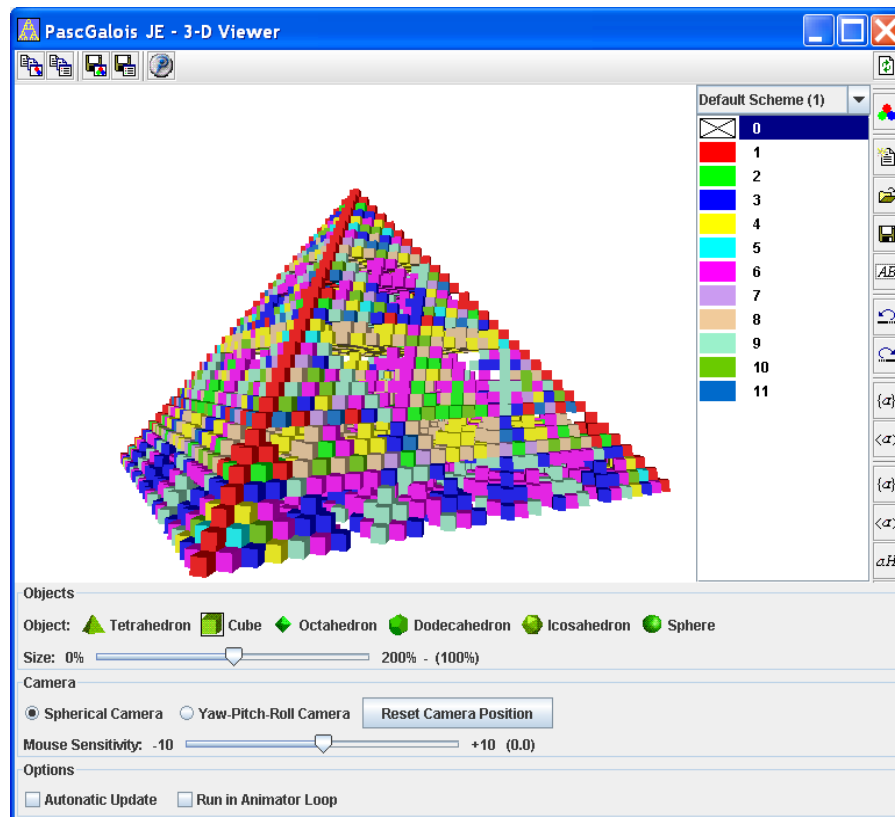
The element list toolbar has four simple options.

- The first option refreshes the element list. Theoretically you should never need this option but as with the image tab the calculations are done in separate threads and if the element list for some reason does not appear this option will display it.
- The second option selects all of the elements in the list.
- The third option unselects all of the elements in the list.
- The fourth option selects the subgroup generated by the currently selected elements.

3-D Viewers

PascGalois JE has two three-dimensional viewers for the two-dimensional automata. The first takes each level (or time-step) of the automaton and stacks them into a "pyramid". The second is invoked when you are viewing a density plot, it will graph the density plot as a three-dimensional bar chart.

If you invoke either of the 3-D Viewers a new window will appear displaying the current automaton using the current color scheme. The level grapher is pictured below.



The toolbar at the top of the window has options to copy the image to the clipboard, copy the color correspondence to the clipboard, save the image and save the color correspondence respectively. The last option is to export the image as a POV-Ray file.

The Export to POV-Ray option will save the automaton as a set of files that can be read and rendered by POV-Ray. POV-Ray is a freeware ray tracer which is available on all major platforms. In Windows, there is a very nice and easy to use user interface. Most Linux implementations have a command line version that is included in the distribution. For details on the options of the POV-Ray exporter see the POV-Ray Exporter page.

The tool button in the upper right hand corner is the refresh button that will apply all of the changes to the automaton and regraph the image. The toolbar down the right hand side of the window are the color correspondence tools, which are.

- The reset to default color scheme option will reset the current color scheme to the default colors. This scheme consists of a cycle of 60 distinct colors that begin with the standard VGA 16 and then continue with combinations of these 16 to form a scheme similar to the SVGA 64 sequence of colors. Most of the options in this menu will change the color scheme and moreover link colors together so that changing one color changes the color of all those elements that are linked to it. This option will not only reset the colors to their original state but it will also break all of the color links, in essence, wiping the slate clean.
- The new color scheme option will load a new (default) color scheme into the list of available color schemes.
- The Load Color Scheme option will load a previously saved color scheme into the list of current color schemes the user can choose from. If the loaded color scheme has fewer entries than the current number of elements the scheme will be cycled and if the loaded scheme has more entries than the current number of elements the scheme will be truncated. Also if the name of the saved color scheme is the same as the name of any currently loaded color schemes you will be prompted to select a different name.
- The Save Color Scheme option will save the color scheme to a file.
- The rename color scheme option will bring up a dialog box that will allow you to select a new name for the color scheme.
- The undo and redo options will undo and redo color scheme changes. The program will hold a maximum sequence of 20 scheme changes.
- The highlight elements option will color all of the currently selected elements in the element list one color and all of the other elements another color. Furthermore, these sets of colors are now linked together. So if you change the color of an element with the color chooser all of the elements in that group are also changed to the new color.
- The highlight subgroup option will take all of the selected elements in the element list, derive the subgroup generated by these elements, color all of the subgroup elements one color and all the other elements another color. As with the highlighting elements option the elements are linked as well.
- The group elements option will take the currently selected elements, link (or group) them and color them all the same color. The difference between this and the highlight elements option is that

the unselected elements are unaltered.

- The group subgroup option will take the currently selected elements, derive the subgroup generated by these elements, color all of the subgroup elements one color and link them. The difference between this and the highlight subgroup option is that the elements that are not in the subgroup are unaltered.
- The group left cosets and right cosets option will take the currently selected elements, derive the subgroup generated by these elements, color all of the subgroup elements one color and link them. It will then find each coset link and color them.
- The set background color option will invoke a color chooser dialog box to allow the user to select a different background color.
- The refresh color scheme will simply refresh the color/element list, it will not refresh the image. That is the job of the refresh button in the upper right. Theoretically you should never need to use this option since the color correspondence is updated after each change to the color scheme.

The color correspondence list has the same right-click popup menu as in the 1-D and 2-D Image tabs and double-clicking the color will bring up the color selection box.

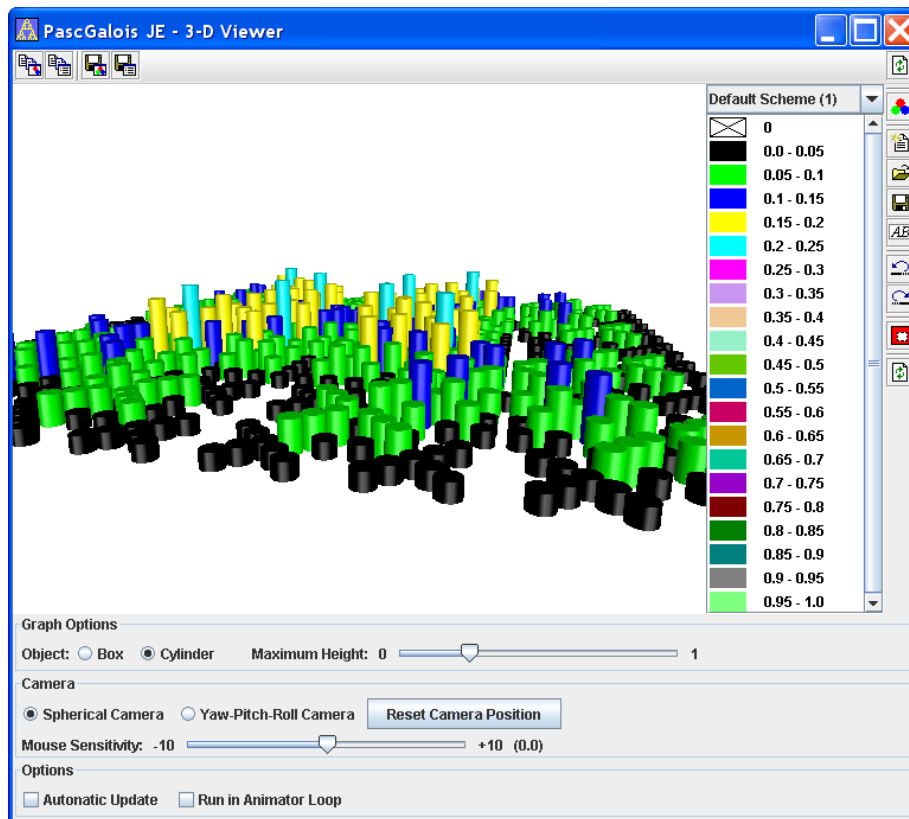
The options at the bottom control the objects that are graphed and the camera interface. The objects are the solids that are graphed at each of the automaton cells. You can select between any of the five Platonic solids or a sphere. Note that if you are graphing a substantial number of levels the interface may become a bit sluggish, in which case you may not want to use the sphere option since this is most computationally expensive object in the list. There is also a slider that allows you to change the size of the object. Note that the default size of the cube is just big enough to fit exactly in the space of the cell. The default size of the sphere is also the largest sphere that can fit inside the box for a cell. The other solids were made slightly smaller than the maximum that could fit in the box for a cell.

There are two independent camera interfaces, the spherical camera and the Yaw-Pitch-Roll camera. The spherical camera is the easier of the two to use but if you want to navigate inside the automaton you will find the Yaw-Pitch-Roll camera more versatile. The spherical camera always points toward the center of the automaton and sits on a sphere around that center. A click and drag of the mouse will change the camera position on the viewing sphere and a click and drag while the Control (Ctrl) key is down will move the camera closer to the center or further away from the center. If you are primarily looking at the automaton from the outside then this camera interface is probably the one you will find most convenient.

The Yaw-Pitch-Roll camera is a little more difficult to use but it has the advantage of being able to position itself anywhere and point anywhere. So if you are moving around inside the automaton this interface will be more useful. Yaw is left and right angular rotation of the camera at the current camera position, Pitch is up and down angular rotation of the camera at the current camera position, and Roll is an angular rotation about the viewing axis. These rotations do not change the camera position they simply change the viewing vector (and the up vector). In this interface a click and drag will change the pitch and the yaw, and a click and drag with the Shift key down will change the roll. A click and drag with the Control (Ctrl) key down will move the camera along the viewing vector (forward and backward) and a click and drag with the Alt key down will move the camera in the plane perpendicular to the viewing vector.

There are also options to reset the camera position and to adjust the mouse sensitivity. There are two final option at the bottom of the window. The first is to turn on the automatic update. Normally, when a change is made you need to hit the refresh button for the changes to take place. If you are working with a reasonable number of levels the updates are fairly fast. Selecting the automatic update option will update the image after any change is made. The second is to run the graphics system through the animator loop. This option does not animate the automaton, it was put in because of some difficulties we have experienced with some graphics cards. On some machines the image does not update correctly when the window is resized or when the window is hidden and then restored. If you experience any of these problems try turning on the animation loop to see if this solves the problem. If you do not have any of these problems you should leave this option off since it will use up CPU cycles.

The density grapher is pictured below.

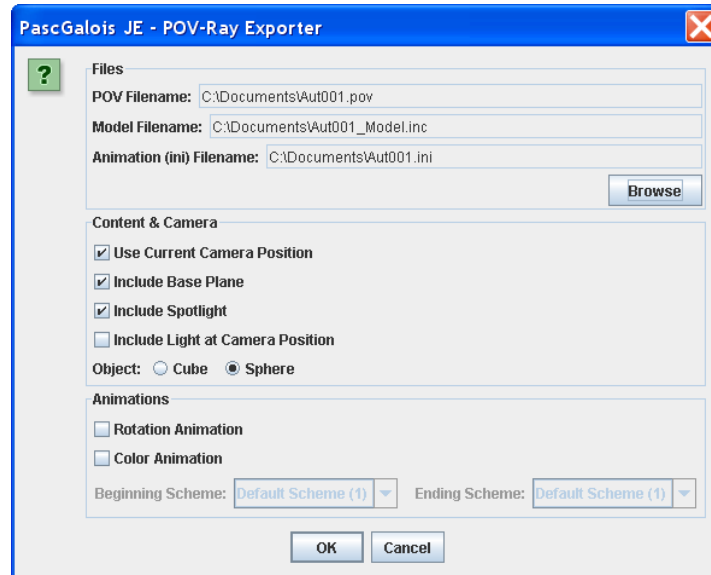


As with the level grapher, the toolbar at the top of the window has options to copy the image to the clipboard, copy the color correspondence to the clipboard, save the image and save the color correspondence respectively. Similarly, the color scheme options on the right are a subset of those from the level grapher but their function is identical, as are the camera and options at the bottom of the window.

The only other options are the object type and the maximum height of the bar chart. The object that is used as the bar can either be a rectangular solid (Box) or a cylinder. The maximum height is the height of the bars that represent the highest probability in the graph.

POV-Ray Exporter

The Export to POV-Ray option will save the automaton as a set of files that can be read and rendered by POV-Ray. POV-Ray is a freeware ray tracer which is available on all major platforms. In Windows, there is a very nice and easy to use user interface. Most Linux implementations have a command line version that is included in the distribution. When you select the POV-Ray Exporter option from either the 2-D or 3-D viewers the following dialog box will appear.

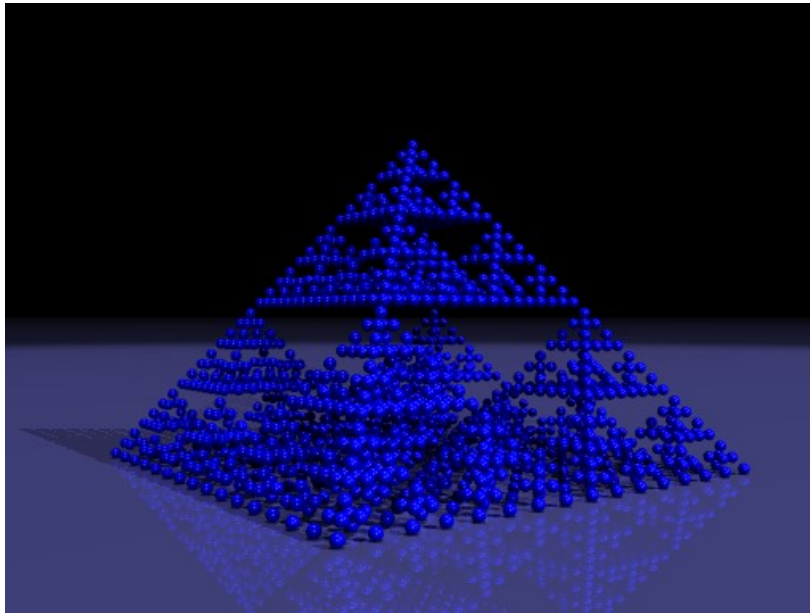


The POV-Ray Files

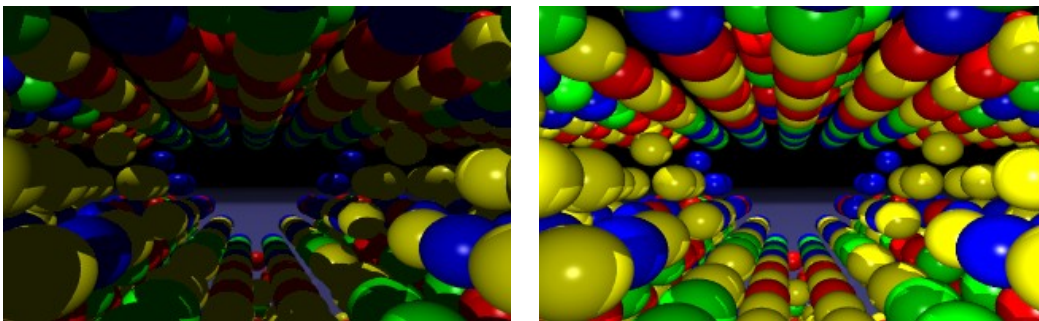
First, select the output filenames by clicking Browse, navigate to the directory in which you wish to save the files and then type in the filename of the pov file. The program will take the name and create three filenames. The main pov file will simply have the .pov extension, the model file which will be the input name with _Model.inc at the end and finally the .ini file. The pov file will contain the scene layout, the inc file will contain the automaton information and the ini file will contain the animation information. Note that the ini file is created only if you select one of the animation options.

The Content & Camera Options

- Use Current Camera Position - The Use Current Camera Position option is only available if you invoke the POV-Ray Exporter from the 3-D Viewer. As its name suggests, if this option is selected the POV-Ray scene will use the current viewing position and viewing direction. If this option is not selected the POV-Ray scene will use a spherical type camera with the position set outside the automaton.
- Include Base Plane - The base plane is simply a plane that the automaton sits on, as can be seen in the image below. If you are viewing the automaton from below you may wish to exclude the base plane since if you are below the base plane you will not see any of the automaton.



- Include Spotlight - There are several light sources in this scene. Two point lights are always used and two lights are optional. One of the optional lights is the spotlight, which creates the curved light edge that can be seen in the background of the above image.
- Include Light at Camera Position - If you are viewing the automaton from the outside you probably do not need a light source at the camera but if you are viewing inside the automaton you may find the image too dark if you do not include the light source. In the 3-D Viewer the image is graphed without using any shadows. So light passes through all of the objects and illuminates each object equally. So a cube or sphere on the inside is given the same amount of light as a cube or sphere on the outer edge. In POV-Ray, on the other hand, shadows are drawn. So the light inside the automaton is significantly less than the light on the outer edge. If you want to illustrate something that is inside turning this light on will place a point light source at the camera position. An example of not using this option and using this option is below.



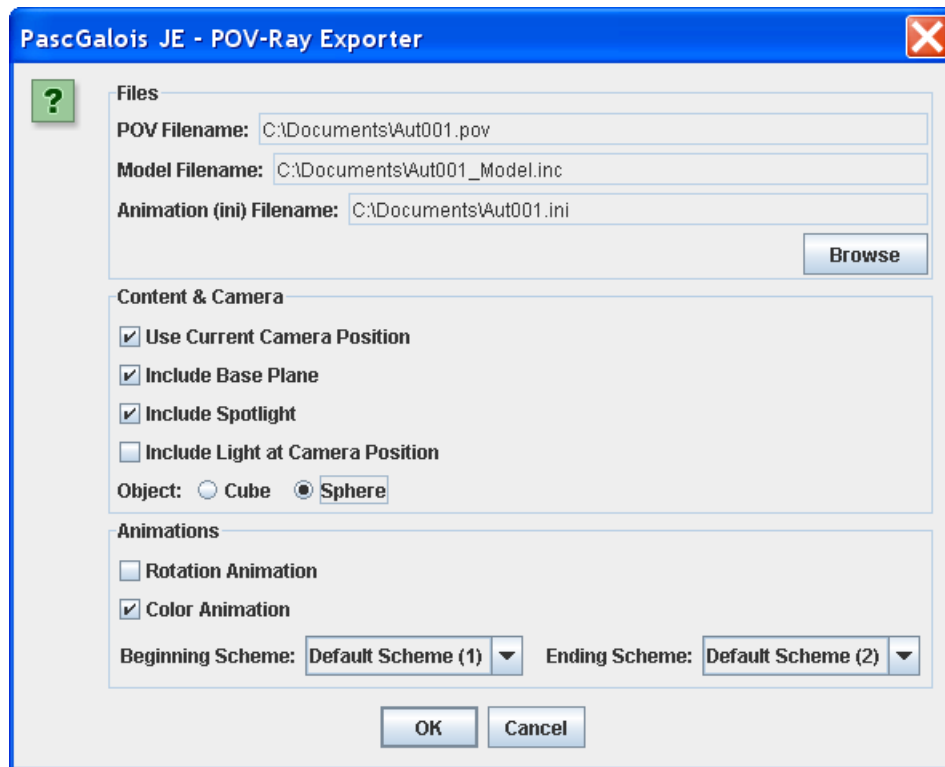
- Object - The objects that can be drawn are either cubes or spheres.

The Animation Options

When either of the animation options are selected the POV-Ray Exporter will also produce an ini

file. To render an animation you will render the ini file and not the pov file. We will discuss rendering and animation options later.

- **Rotation Animation** - A rotation animation can only be used if the current camera is the spherical camera. If this option is selected the exporter will set up the scene so that POV-Ray will create a sequence of images that rotate the camera position in the horizontal plane containing the camera through 360 degrees. The point the camera is looking at is held constant.
- **Color Animation** - A color animation is when the automaton fades from one color scheme smoothly to another color scheme. When this option is selected the beginning color scheme and ending color scheme drop-down selectors will be enabled.



- Simply select the color scheme you wish to start with and the one you wish to end with. Note that if you do not have at least two color schemes loaded when you select the POV-Ray Exporter this option will not be available to you.

Rendering the POV-Ray Files

To render the POV-Ray image you will, of course, need the POV-Ray application. A command-line version of POV-Ray is normally included in most Linux distributions and the most current versions for all major platforms can be found at the POV-Ray web site at <http://www.povray.org/>. The Windows version is very easy to use and includes a nice user interface with menus and toolbars.

On a Windows system simply load in the pov file (or ini file if doing an animation) and click on the Run button. Once you do POV-Ray will render the image and save the image as a bitmap file in the same folder as your pov file. If you are running an animation then POV-Ray will save a sequence of

bitmap files, one for each frame in the animation. Note that POV-Ray will not create a movie file of the animation, you will need to paste the images together with another program.

On a Unix or Linux system using a command-line version you can get a list of flags by simply typing `povray` into the console or by checking the man pages. For example, if the pov file is `Aut.pov` you can render a 640X480 image using antialiasing by

```
povray -W640 -H480 -A0.3 Aut.pov
```

On a Unix or Linux system POV-Ray will save the image as a png file instead of a bitmap file.

The POV-Ray Code

Although you do not need to know anything about POV-Ray programming to use this feature, we have set up the POV-Ray files so that they are easy to manipulate. For example, the pov file looks something like the following,

```
// POV-Ray output from PascGalois JE

// Constants

#declare PI = 3.14159265358979323846;
#declare PI_DIV_180 = 0.017453292519943296;
#declare deg = PI_DIV_180;
background { <0,0,0> }

// User Editable Values

#declare theta = 81.22905;
#declare phi = -0.61676615;
#declare r = 2.073058068752289;

#declare CamPosX = r*cos(phi*deg)*sin(theta*deg);
#declare CamPosY = r*sin(phi*deg);
#declare CamPosZ = r*cos(phi*deg)*cos(theta*deg);

#declare LookAtX = 0.0;
#declare LookAtY = 0.0;
#declare LookAtZ = 0.0;

#declare UpX = 0.0;
#declare UpY = 1.0;
#declare UpZ = 0.0;

#declare L1_theta = 20;
#declare L1_phi = 10;
#declare L1_r = 20;
```

```
#declare L2_theta = 70;
#declare L2_phi = 30;
#declare L2_r = 20;

#declare Spot_theta = 30;
#declare Spot_phi = 80;
#declare Spot_r = 40;

#declare SpotPoint_theta = 0;
#declare SpotPoint_phi = 0;
#declare SpotPoint_r = 0;

#declare SpotRadius = 10;
#declare SpotFalloff = 15;

// Scene

camera {
    location <CamPosX, CamPosY, CamPosZ>
    angle 50
    sky <UpX, UpY, UpZ>
    look_at <LookAtX, LookAtY, LookAtZ>
}

light_source { <L1_r*cos(L1_phi*deg)*cos(L1_theta*deg),
    L1_r*sin(L1_phi*deg),
    L1_r*cos(L1_phi*deg)*sin(L1_theta*deg)>
    color <1,1,1>
}

light_source { <L2_r*cos(L2_phi*deg)*cos(L2_theta*deg),
    L2_r*sin(L2_phi*deg),
    L2_r*cos(L2_phi*deg)*sin(L2_theta*deg)>
    color <1,1,1>
}

light_source { <Spot_r*cos(Spot_phi*deg)*cos(Spot_theta*deg),
    Spot_r*sin(Spot_phi*deg),
    Spot_r*cos(Spot_phi*deg)*sin(Spot_theta*deg)>
    color <1,1,1>
    spotlight
    radius SpotRadius
    falloff SpotFalloff
    point_at
```

```
<SpotPoint_r*cos(SpotPoint_phi*deg)*cos(SpotPoint_theta*deg),
    SpotPoint_r*sin(SpotPoint_phi*deg),
    SpotPoint_r*cos(SpotPoint_phi*deg)*sin(SpotPoint_theta*deg)>
}

light_source { <CamPosX, CamPosY, CamPosZ>
  color <1,1,1>
}

plane { <0, 1, 0>, -0.375
  pigment { color rgb <.2,.2,.4> }
  finish { reflection {0.25} phong 0.4 }
}

#include "Aut_Model.inc"
object { aut_output }
```

If you are using a spherical coordinate camera then the first three lines under the User Editable Values are the spherical coordinates of the camera.

```
// User Editable Values

#declare theta = 81.22905;
#declare phi = -0.61676615;
#declare r = 2.073058068752289;
```

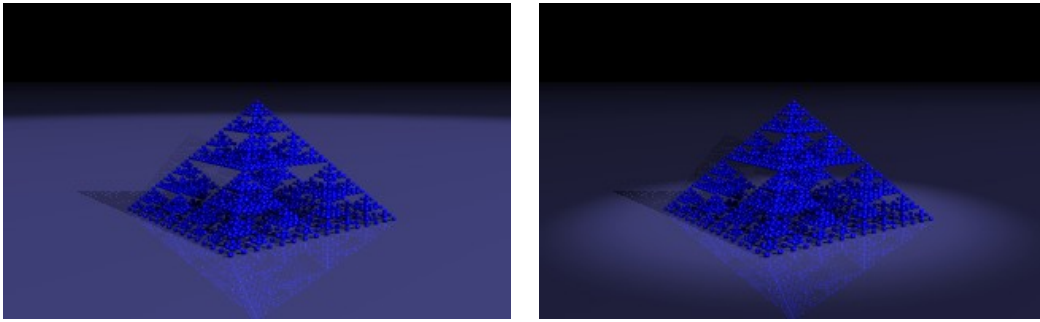
On the other hand if you are using a Yaw-Pitch-Roll camera the position will be given in rectangular coordinates.

```
#declare CamPosX = 0.28307894;
#declare CamPosY = -0.0908582;
#declare CamPosZ = -0.07512487;
```

Keep in mind that POV-Ray uses a left-hand coordinate system with the positive y-axis in the up direction and the positive x-axis moving off to the right. The three lines declaring variables that start with L1 are spherical coordinates of light number 1 and the three starting with L2 are spherical coordinates of light number 2. If you included the spotlight then the rest of the declaration statements are the spherical coordinates of the position and the point-at point. The final two are the radius and fall-off of the spotlight. You may edit any of these values to change the camera and light positions and if you want to narrow the spotlight simply decrease the values of

```
#declare SpotRadius = 10;
#declare SpotFalloff = 15;
```

For example, in the images below we used the default values for the first image and then changed these values to 1 and 2 respectively for the second image.



POV-Ray uses the same syntax as C for comments, so anything following a `//` is not rendered and anything between `/*` and `*/` is not rendered.

The Scene section of the pov file should, for the most part, not be edited, since most of it simply changes spherical coordinates to rectangular coordinates. There are however a couple numbers that you may wish to change. For example, you can comment out an entire object (like a `light_source`) by using the comment tags discussed above. In the camera section the `angle 50` sets the angle of view of the camera to 50 degrees, you may wish to increase or decrease this number. You will also note that the colors for the lights are `color <1, 1, 1>`, which is bright white. The syntax for the colors is `color <r, g, b>` where r, g and b are the red, green and blue components of the color. Each of r, g and b should be numbers between 0 and 1. So if you would prefer a red light you can edit the color to `color <1, 0, 0>`. You can also change the intensity of a color by multiplying the vector by a constant. For example, `color 0.7*<1, 0, 0>` will produce a white light at 70% intensity. Finally, if you included the base plane the values

```
reflection {0.25} phong 0.4
```

control the amount of reflection and shine the surface has. You can set these to any number between 0 and 1.

The ini files are very easy to update. If you do an animation then the ini file should look something like this.

```
Input_File_Name="Aut.pov"  
Initial_Frame = 1  
Final_Frame = 360
```

This simply creates 360 frames in the animation. To change the number of animation frames just change the value of `Final_Frame`.

The final file is the model file, which contains the model of the automaton, the color scheme and some variables that are used in an animation. Of these the only one you may wish to change is the `aut_phong` value. This controls the amount of shine that is on the object.

```
// POV-Ray output from PascGalois JE  
  
// Constants and Animation Variables
```

```
#local PI = 3.14159265358979323846;
#local r = 0.0125;
#local dr = cos(clock*PI/2)*r;
#local ir = cos(PI/2-clock*PI/2)*r;

#local aut_phong = 0.5;
#local fin = finish { phong aut_phong };
#local inc_aut_phong = clock*aut_phong;
#local ifin = finish { phong inc_aut_phong };
#local dec_aut_phong = (1-clock)*aut_phong;
#local dfin = finish { phong dec_aut_phong };

// Colors

#local c1 = pigment { color rgb <1.0, 0.0, 0.0> };
#local c2 = pigment { color rgb <0.0, 1.0, 0.0> };
#local c3 = pigment { color rgb <0.0, 0.0, 1.0> };
#local c4 = pigment { color rgb <1.0, 1.0, 0.0> };

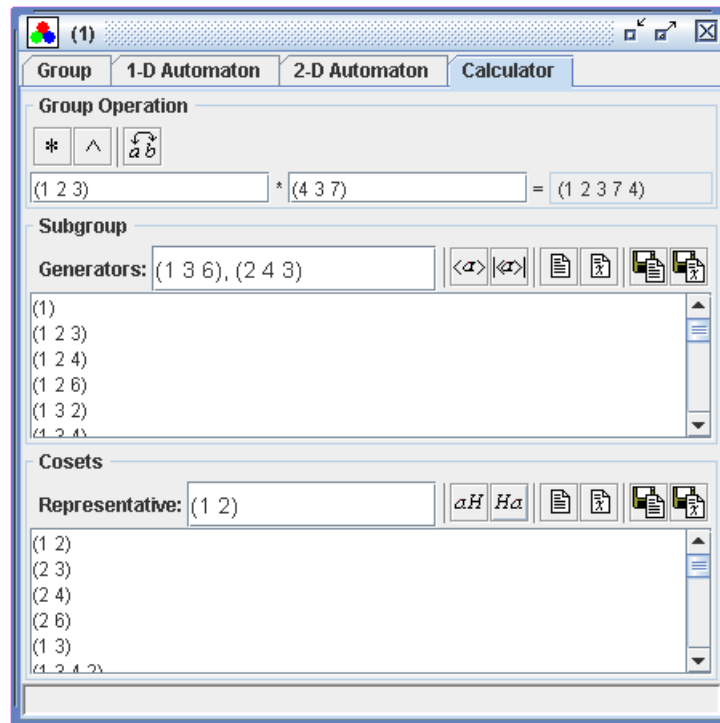
// Automaton Model

#declare aut_output = union{
  sphere {<0.0, 0.3875, 0.0>, r texture{c1 fin}}
  sphere {<-0.025, 0.36250000000000004, 0.0>, r texture{c1 fin}}
  sphere {<0.0, 0.36250000000000004, -0.025>, r texture{c1 fin}}
  ...
}
```

Two final notes about the POV-Ray files. First, the image you get from POV-Ray may not be exactly what you see in the 3-D viewer. In the 3-D Viewer if the camera is too close to an object the object is not graphed, but this is not done in POV-Ray. In effect, the camera could really be inside a sphere but if it is close to the edge, the 3-D viewer will see through it. On the other hand when you render the image in POV-Ray all you see is the inside of a sphere. Also, there is a noticeable difference in the field of view even though both systems are set to 50 degrees. Second, the saving of the POV-Ray file is done in a separate thread of execution. What this means is that you can keep working while the program saves the file in the background, which is good. On the other hand, if you close the program too quickly after you save the POV-Ray file and the automaton is very large it is possible that you may truncate the model file. Under most circumstances this will not happen but if you are saving an exceptionally large automaton you may want to use a file browser to make sure the model file is not increasing in size before you close the program.

Group Calculator

The group calculator is a simple facility to help the user do group operations. While you can probably do calculations in Z_n faster in your head than type them into this calculator you may find it handy when working with a dihedral or permutation group.



The calculator is broken down into three parts. The top is for operating on two group elements and for taking powers of a single element. There is also a feature for interchanging the two entries for a quick check of commutativity. The middle contains a facility for generating a subgroup. The bottom section allows the user to generate a coset of the subgroup generated in the middle.

The Group Operation Section

To do a group operation on two group elements simply type the two elements into the text boxes and click on the "*" tool. The result of the operation will be displayed in the disabled text box after the equal sign. To take an element to a power type the element in the first box and the power in the second and select the "^" tool. In this case the second text box must contain an integer. In the event that an element does not have an inverse, which could happen if a user-defined structure is involved or if you are using Z_n under multiplication, an error message will be displayed. The final tool simply interchanges the two text boxes.

The Subgroup Section

To generate a subgroup simply type in the elements you wish to use as generators with a comma between each and then click the generate subgroup tool. You may use $\langle \rangle$ around the element list but it is not necessary.

The toolbar consists of six options.

- The first generates the subgroup.
- The second generates the subgroup and displays the order of the subgroup.

- The third and fourth copy the subgroup elements to the clipboard either as text or as LaTeX code.
- The fifth and sixth save the subgroup elements to either a text file or LaTeX file.

The Cosets Section

To generate a coset of the above subgroup simply type in a coset representative and click either the left coset button or the right coset button. If the subgroup has not been generated the coset generator will generate the subgroup first and then the coset. Note that when generating a subgroup the elements of the subgroup will be sorted. As for the coset, we do not sort the elements but leave them in the order calculated. This way the user can see the correspondence between the subgroup list and the coset list.

The element list toolbar has six options.

- The first option generates a left coset.
- The second option generates a right coset.
- The third and fourth copy the coset elements to the clipboard either as text or as LaTeX code.
- The fifth and sixth save the coset elements to either a text file or LaTeX file.