L System user notes

(Lindenmayer Systems)

Written by

Paul Bourke

Version 2.5

June 1991

Introduction

This program implements some of the L-Systems discussed in "Lecture Notes in Biomathematics" by Przemyslaw Prusinkiewcz and James Hanan. A brief description of an 0L system will be presented here but for a more complete description the user should consult the literature.

This program may be freely copied for interested users. If a copy is kept then it can be registered by sending NZ$80 to the address shown in the about dialog.

The application was initially written to investigate methods of incorporating objects with a large number of drawing elements (lines, polygons) into a CAD package. L-Systems is one way for example of "generating" trees at the rendering stage but not during the editing stage where the image complexity will slow down the response time.

Developed using Symantec Think C version 4.0. Full, source code available for NZ$500.

Simpleminded example of 0L system

A string of characters (symbols) is rewritten on each iteration according to some replacement rules. Consider an initial string (axiom)

F+F+F+F

and a rewriting rule

F --> F+F-F-FF+F+F-F

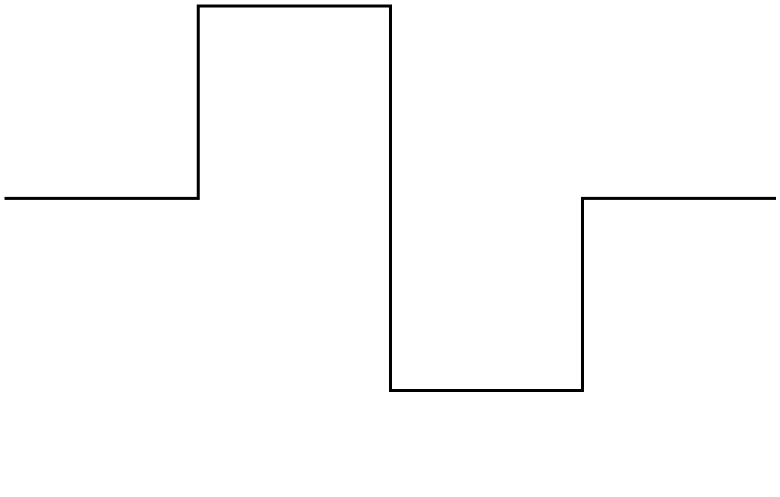
After one iteration the following string would result

F+F-F-FF+F+F-F + F+F-F-FF+F+F-F + F+F-F-FF+F+F-F + F+F-F-FF+F+F-F

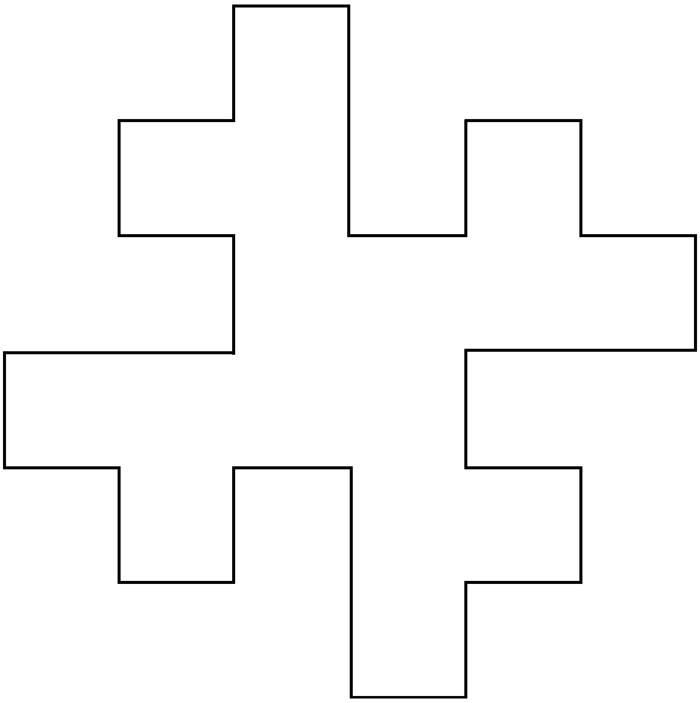
For the next iteration the same rule is applied but now to the string resulting from the last iteration
F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-FF+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-FF+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-FF+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-FF+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F

Some symbols are now given a graphical meaning, for example, F means move forward drawing a line, + means turn right by some predefined angle (90 degrees in this case), - means turn left.

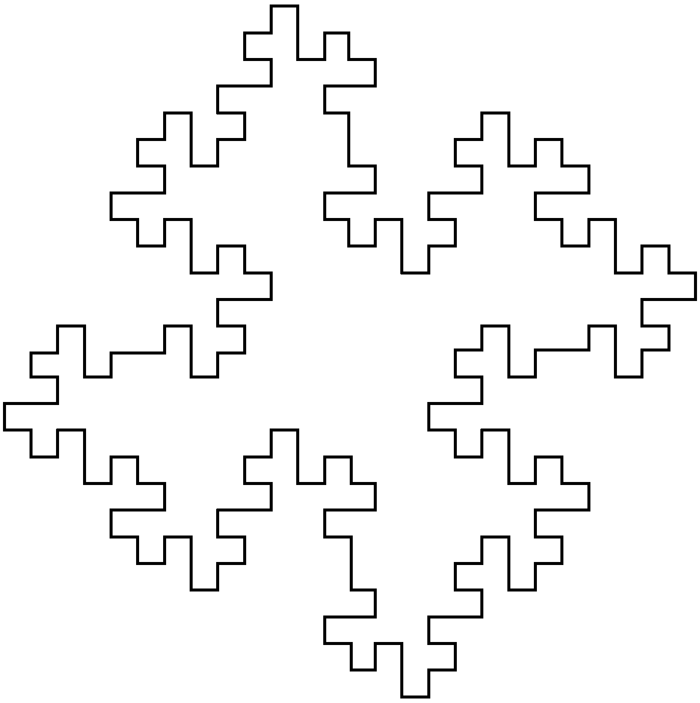
Using these symbols the initial string F+F+F+F is just a rectangle (ø = 90). The replacement rule F --> F+F-F-FF+F+F-F replaces each forward movement by the following figure



The first iteration interpreted graphically is



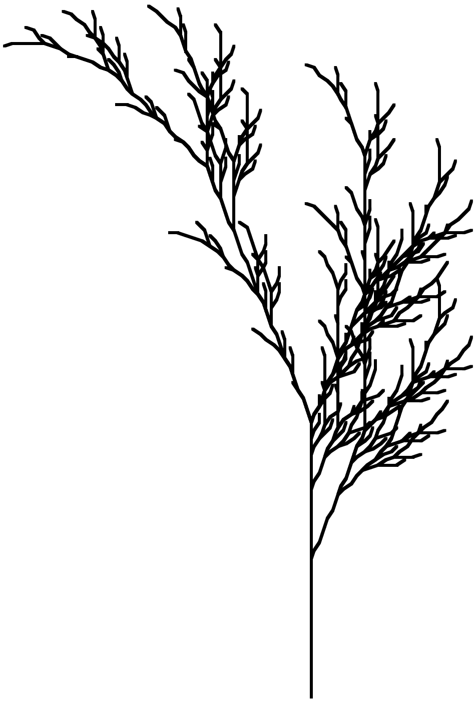
The next iteration interpreted graphically is:



and so on.

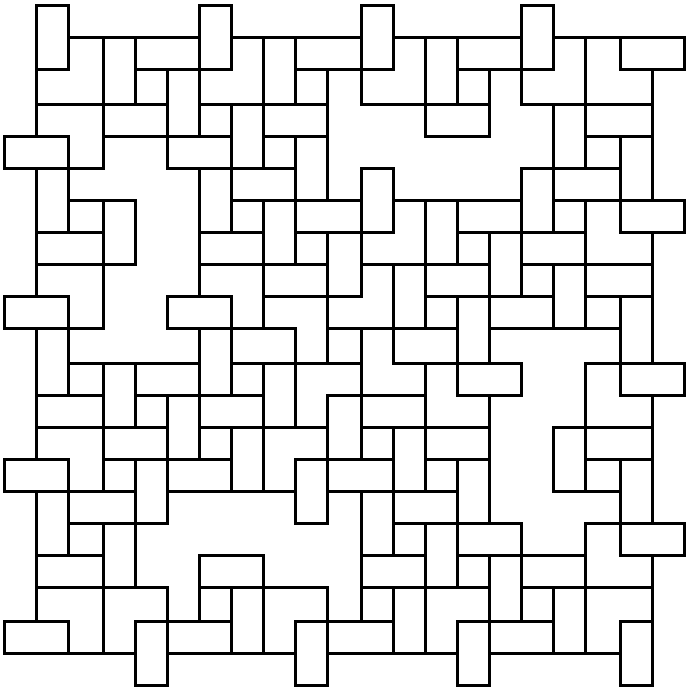
Example:

Axiom	X
F -->	FF
X -->	F-[[X]+X]+F[+FX]-X
ø =	22.5



Example:

Axiom	F+F+F+F
F -->	FF+F-F+F+FF
ø =	90



Symbols

The following characters have a geometric interpretation.

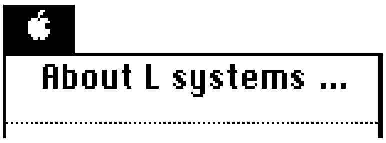
Character	Meaning
F	Move forward by line length drawing a line
f	Move forward by line length without drawing a line
+	Turn left by turning angle
-	Turn right by turning angle
|	Reverse direction (ie: turn by 180 degrees)
[	Push current drawing state onto stack
]	Pop current drawing state from the stack
#	Increment the line width by line width increment
!	Decrement the line width by line width increment
@	Draw a dot with line width radius
{	Open a polygon
}	Close a polygon and fill it with fill colour
>	Multiply the line length by the line length scale factor
<	Divide the line length by the line length scale factor
&	Swap the meaning of + and -
(	Decrement turning angle by turning angle increment
)	Increment turning angle by turning angle increment

When drawing the graphical representation of the L string all other characters are ignored.

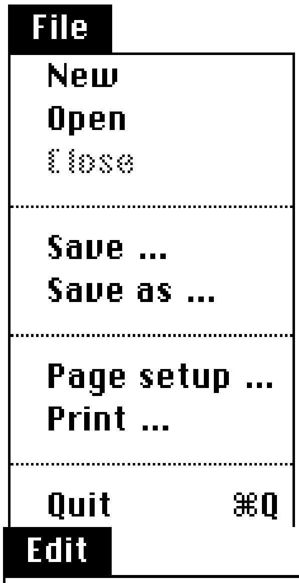
The user may choose and use any other single printable characters for the replacement rules except, note: this excludes "white" characters such as spaces and tabs. For context sensitive L systems the * character is used to represent any match.

See the SYMBOL menu item for a summary of the reserved symbols.

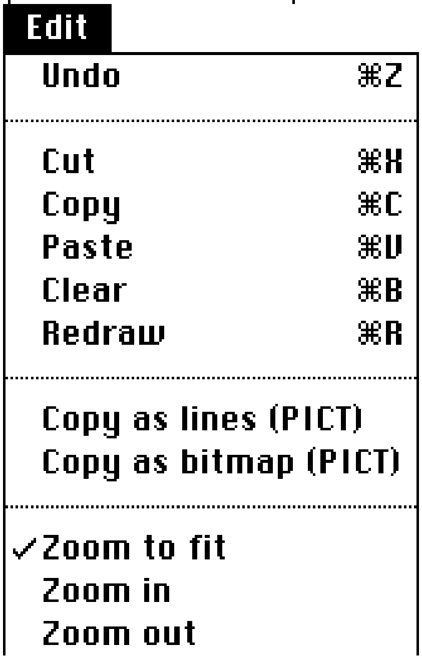
Menus



As with most applications the ABOUT menu item located in the Apple menu before the desk accessories displays information such as the version number and contact address of the developer.



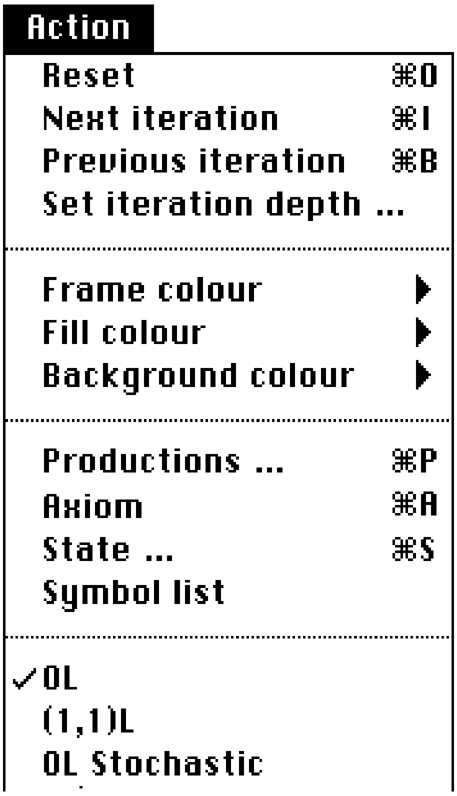
This is the standard FILE menu although some of the items are not implemented.
NEW resets all the variables including the rules to their default values.
OPEN to load a rule description file previously saved or possibly created with a text editor.
SAVE and SAVE AS create rule description files.
PRINT and PAGESETUP control direct printing.
QUIT to exit from the program when finished.



The standard EDIT menu. The first 5 items are only included for compatibility reasons and do not apply to this application. (they are used with desk accesories in some cases)

The two special copy items allow images to be transfered to other applications depending on the prefered data type. Line drawings should be copied for CAD packages and bitmaps for painting programs.

Normally images will be scaled to fit the window, it is possible to zoom in or out of areas with the last two items.



This is the main menu for this application.

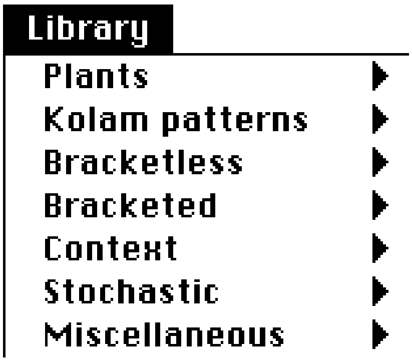
REDRAW updates the current display, for example, if the drawing was canceled with clover-. (period)

Menu items 2 through 4 control the iteration depth. RESET sets it to 0, the number of iterations to compute may be incremented, decremented, or set to any particular number.

FRAME, FILL and BACKGROUND are popup menus that allow the colour of the lines, polygon fills and background to be set.

PRODUCTIONS allows the current rules to be altered. AXIOM menu item sets the initial string. STATE menu item varies environment variables. SYMBOL LIST displays the meaning given to various predefined and reserved symbols.

The type of system is selectable from the last 3 items.



This contains a number of predefined image specifications grouped together in categories given by the popup menu names.

Users are encouraged to submit further examples that can be added to this library.

Advanced features/comments

• Always ensure that the '[' and ']' brackets match in number, ie: there should always be the same number of each type. If not a stack overflow will quickly occur, a message displayed, and the iteration depth decremented. Note: non matching brackets are not tested before the string in interpreted, a problem only occurs when a stack overflow situation is encountered.

• Polygons (using "{" and "}") cannot be nested, ie: there can only be one open at a time. If a polygon open symbol "{" is encountered while another polygon is still open it will be ignored. If a polygon is left open when the end of the string is reached it will be automatically closed.

• For a complete list of predefined symbol meanings look at the SYMBOL list under the ACTION menu.

• Best quality hardcopy line drawing images can be obtained by using MacDraw or Claris CAD. In both cases copy the image as a line drawing, paste it into MacDraw say, set the lines to black and line width 0.1mm, then print on a LaserWriter. Note: the number of lines can quickly approach the thousands and even tens of thousands, don't expect the printing to be fast.

• Time consuming drawing may be cancelled by typing clover-. (period)

• The clover key equivalents to the Cut/Copy/Paste menu items can be used within dialogs for transfering data between fields.

• The maximum length of the strings generated is 300,000 symbols. Attempting to exceed this will give an error message and automatic decrementing of the iteration count.

• The maximum stack depth is 100. This means that a bracketed string cannot have brackets nested more than 100 deep.

```
ie:  [  [  [  ]  ]  [  [  [  [  ]  ]  [  ]  ]  ]  ]
     1  2  3  2  1  2  3  4  5  4  3  4  3  2  1  0  is nested 5 deep
```

If the maximum stack depth is exceeded an appropriate error message will be displayed and the iteration depth decremented until the stack no longer overflows.

- The graphical representation of the L system is automatically scaled to fit within the L-System window.

- When copying to the clipboard as lines make the drawing window as large as possible for maximum resolution.

- Read this at your own risk! While the program was being developed if proved useful to be able to spy on what was happening. To do this hold the option and clover keys down simultaneously while selecting the ABOUT item in the APPLE menu. This will result in a debug window which among other things will show the current state of some useful variables, for example, the current L-String. (This is more useful if there are two monitors attached to the Mac II) Features in this window can also be copied to the clipboard, useful maybe for documenting the progress of a particular L system.

References

Heinz-Otto and Deitmar Saupe
The Science of Fractal Images
Springer-Verlag

Przemyslaw Prusinkiewicz
Application of L-Systems to Computer Imagery
Lecture Notes in Computer Science #291, Pages 534-548

Przemyslaw Prusinkiewicz, James Hanan
Lecture Notes in Biomathematics #79

Prusinkiewicz, P
Graphical Applications of L-Systems
Proc. of Graphics Interface 1986 - Vision Interface, 1986, Pages 247-253.

Smith, A.R.
Plants, Fractals, and Formal Languages
Computer Graphics, 18, 3, 1984, Pages 1-10

Prusinkiewicz, P and Lindenmayer A.
The Algorithmic Beauty of Plants
Springer Verleg, 1990, Pages 40-50