

Segundo Exercício Programa

Prof. Alair — MAC122

Um compilador de fractais -II

Resumo: Neste exercício programa, cada aluno deverá melhorar o compilador de fractais implementado no EP1. Estas melhorias incluirão melhorias que confirmam um poder maior ao compilador de fractais bem como melhorias que facilitem a vida do usuário. Um terceiro tipo de melhoria, melhorias de ordem técnica que permitam um melhor uso dos recursos computacionais, também serão cobradas.

Mais poder à recorrência que define as curvas fractais

No EP1, vimos definido um compilador de fractais. Neste compilador, era particularmente importante a recorrência que permitia definir uma curva fractal. Assim, por exemplo, a curva de Koch era definida pelo string (que chamáramos de recorrência) $F-F++F-F$. Este string substituiu uma ocorrência de F . Há curvas fractais porém que não podem ser descritas por recorrências desta forma, mas admitem uma descrição semelhante que faça uso de três recorrências, uma que substitui a string F , chamada de *recorrência* F , uma que substitui uma ocorrência de X , chamada de *recorrência* X , e outra que substitui a string Y , chamada de *recorrência* Y . Cada uma destas três recorrências é simplesmente chamada de *recorrência*. Para simplificar a descrição, também o axioma será interpretado como sendo uma recorrência, associado no caso à letra A .

Ao nível de recursão 0, apenas os símbolos F , $+$ e $-$ produzem alguma mudança nos estados. (Por exemplo, o símbolo F produz a execução do método *forward* como antes.) Os demais símbolos simplesmente não produzem qualquer alteração num estado.

Tomemos como exemplo a curva de Hilbert. Neste curva, o axioma é X , e as três respectivas recorrências são F , $-YF+XFX+FY-$ e $+XF-YFY-FX+$. O

ângulo β é $\pi/2$. Assim sendo, temos as primeiras curvas de Hilbert 0 na Figura 1. Uma particularidade interessante das curvas de Hilbert que vale

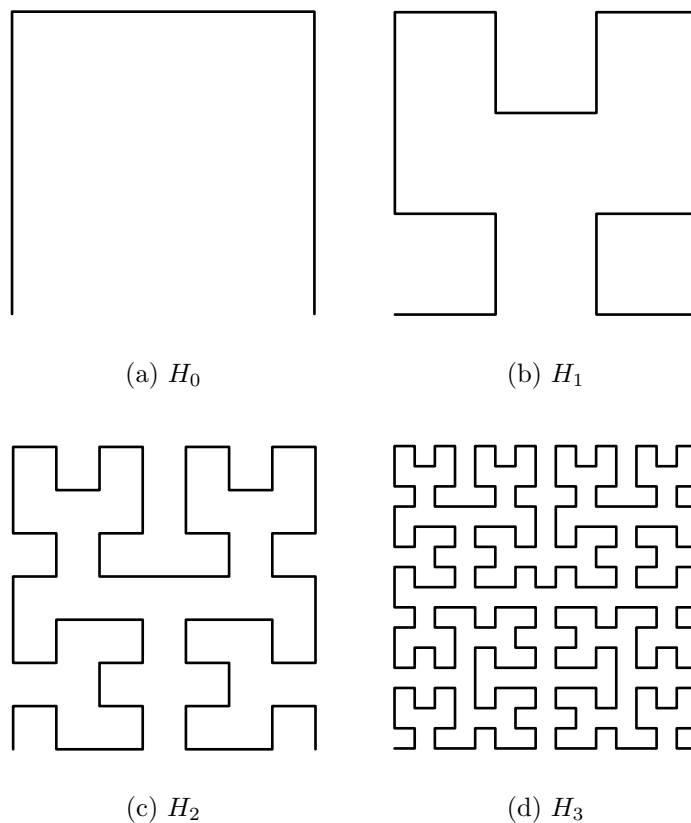


Figura 1: Primeiras curvas de Hilbert

a pena observar é a propriedade de que ela “preenche o plano”. De forma mais precisa, dado um ponto qualquer no quadrado por ela delimitado e uma distância $\epsilon > 0$ qualquer, existe um índice de recorrência N tal que a distância deste ponto a H_n seja menor que ϵ para todo $n > N$.

Na Tabela 1 temos exemplos de parâmetros usados por curvas mais complexas que aquelas que foram vistas no EP1.

Nome	β	Axioma / RecorrênciaF	RecorrênciaX / RecorrênciaY
hilbert	$\pi/2$	X F	-YF+XFX+FY- +XF-YFY-FX+
moore	$\pi/2$	XFX+F+XFX F	-YF+XFX+FY- +XF-YFY-FX+
hilbert2	$\pi/2$	Y F	XFYFX+F+YFXFY-F-XFYFX YFXFY-F-XFYFX+F+YFXFY
gosper	$\pi/3$	XF F	X+YF++YF-FX-FXFX-YF+ -FX+YFYF++YF+FX-FX-Y
dragon	$\pi/2$	X F	X+YF+ -FX-Y
gasket	$\pi/3$	XF-FF-FF-F FF	-FXF++FXF++FXF-
arrowhead	$\pi/3$	YF F	YF+XF+Y XF-YF-X
peano	$\pi/2$	F F+F-F-F-F+F+F-F	
square	$\pi/2$	XF+F+XF+F+	XF-F+F-XF+F+XF-F+F-X
sq	$\pi/2$	X+F+X+F+	X-F+F-X+F+X-F+F-X

Tabela 1: Algumas curvas fractais mais complexas

Tabela de parâmetros

Para facilitar a vida do usuário do programa gerador de fractais, o usuário poderá optar por fornecer na linha de comando apenas o nível de recursão desejada e o nome da curva fractal em questão. O programa deverá então ser capaz de identificar os parâmetros adequados que estarão sendo usados por este fractal.

Desta forma, o programa gerador de fractais deverá guardar uma tabela que associe a cada nome de uma curva fractal, os respectivos parâmetros que serão usados no desenho da referida curva. Uma possível implementação, é a de um vetor de objetos de uma classe (ou structs). Neste caso um dos campos da classe é o nome do fractal. Os demais campos devem formar o conjunto adequado de parâmetros que deve ser associado ao nome em questão. As informações presentes na Tabela 1 deste enunciado, bem como a Tabela 1 do EP1, devem ser incluídas nesta tabela. Suporemos valores razoáveis para os parâmetros não descritos nas tabelas. Assim o ponto do estado inicial deve ser $(0, 0)$ e o ângulo α deve ser também 0. O parâmetro u não é guardado nesta tabela.

Ao se dar o nome do fractal em questão, o programa deverá buscar neste vetor o objeto que possui o nome procurado e deve usar o conjunto de parâmetros correspondente. Por simplicidade, adote-se parâmetro $u = 1$ nestes

casos.

Melhorias Técnicas

Nalguns algoritmos a recursão é um problema e pode interessar-nos um uso melhor dos recursos através da eliminação da recursão. Neste EP, as recursões devem ser eliminadas através do uso de uma pilha (implementada através de lista ligada). Na pilha, deverão ser empilhados pares da forma (s, m) , onde m é um nível de recursão e s é um símbolo dentre **A**, **F**, **X**, **Y**, **+** e **-**. (O símbolo **A** representa um axioma.) No início, empilha-se o par $(A, n + 1)$, onde n é o parâmetro que designa o nível de recursão da curva fractal. Para cada par (S, m) que for desempilhado, deve-se executar uma mudança de estado adequada se $m = 0$ ou se S for um dos símbolos **+** ou **-**. Ademais, usando-se a recorrência correspondente, deve-se empilhar a seqüência de pares $(S, m - 1)$ que dão origem ao desenho da curva S_m . Por exemplo, no caso da curva de Hilbert, ao ser desempilhado o par $(A, 4)$, deve-se empilhar a seqüência de pares (de comprimento 1): $(X, 3)$. Se fosse o para desempilhado, deveríamos empilhar os pares: $(-, 2)$, $(Y, 2)$, $(F, 2)$, $(+, 2)$, $(X, 2)$, $(F, 2)$, $(X, 2)$, $(+, 2)$, $(F, 2)$, $(Y, 2)$, $(-, 2)$.

O que deve ser feito no EP

O exercício-programa a ser entregue é individual e deve ser feito usando-se as linguagens de programação: java, python, ou C. Pode-se entregar em pascal desde que seja compilado com compilador a ser combinado com a monitora. A grosso modo, ele deve ser o exercício programa EP1 com as alterações necessárias às implementações das modificações acima descritas. Daremos alguns detalhes a mais.

Ele deve receber alguns parâmetros da linha de comando (de forma semelhante ao à receita já dada para o EP1) e imprimir os vértices que definem uma seqüência contígua de segmentos de reta (curva) que desenham o fractal relativo aos parâmetros fornecidos.

O programa deve receber os seguintes parâmetros ordenados na linha de comando:

1. o nível n de recursão da curva fractal;
2. o quociente π/β ;

3. o string RF correspondente à recorrência F ;
4. o string RX correspondente à recorrência X ;
5. o string RY correspondente à recorrência Y ;
6. o string A correspondente ao axioma;
7. o comprimento u ;
8. a coordenada x do estado inicial;
9. a coordenada y do estado inicial;
10. o quociente π/α , onde α é a direção do estado inicial.

Com os parâmetros 2 e 10 o que se deseja, de fato, é poder saber quais são os parâmetros β e α . Um parâmetro que é a string : representa a string vazia.

Alternativamente, o programa poderá receber os seguintes parâmetros na linha de comando:

1. o nível n de recursão da curva fractal;
2. o nome N da curva fractal.

Tanto numa hipótese quanto noutra, a lista de parâmetros poderá ser precedida pela opção de linha de comando `-d`, de fato uma string `-d`, que produzirá a impressão de algumas linhas com informações úteis à depuração. Estas impressões devem ser formadas pela impressão da tabela descrita na seção . Também deve ser impresso o conjunto dos parâmetros que de fato estarão sendo utilizados. Por fim, deve ser impresso cada par empilhado ou desempilhado da pilha.

O exercício-programa *não deve* fazer uso de recursividade para o desenho dos fractais e a pilha utilizada deverá fazer uso de listas ligadas.

Informações adicionais

Informações adicionais e alguns exemplos de execução de um protótipo sobre o EP2 serão encontradas na URL:

<http://www.ime.usp.br/~alair/mac122-04/ep2/>