

RICARDO RONALD EBERSON

**UM ESTUDO SOBRE A CONSTRUÇÃO DE FRACTAIS EM
AMBIENTES COMPUTACIONAIS E SUAS RELAÇÕES
COM TRANSFORMAÇÕES GEOMÉTRICAS NO PLANO**

MESTRADO EM EDUCAÇÃO MATEMÁTICA

**PUC/SP
São Paulo
2004**

RICARDO RONALD EBERSON

**UM ESTUDO SOBRE A CONSTRUÇÃO DE FRACTAIS EM
AMBIENTES COMPUTACIONAIS E SUAS RELAÇÕES
COM TRANSFORMAÇÕES GEOMÉTRICAS NO PLANO**

*Dissertação apresentada à Banca Examinadora da Pontifícia Universidade Católica de São Paulo, como exigência parcial para obtenção do título de **MESTRE EM EDUCAÇÃO MATEMÁTICA**, sob a orientação da **Profa. Dra. Ana Paula Jahn**.*

PUC/SP
São Paulo
2004

Banca Examinadora

Autorizo, exclusivamente para fins acadêmicos e científicos, a reprodução total ou parcial desta Dissertação por processos de fotocopiadoras ou eletrônicos.

Assinatura: _____ **Local e Data:** _____

AGRADECIMENTOS

A *Deus*, que em Sua infinita bondade, deu-me a força, a perseverança e a luz necessárias à conclusão deste trabalho, pois, com toda a certeza, sem isso não chegaria até aqui.

À *Professora Doutora Ana Paula Jahn*, que tanto me honrou por aceitar o convite como orientadora e pela confiança e entusiasmo dedicados a este trabalho.

Às *Professoras Doutoras Lulu Healy e Miriam Godoy Penteado*, pelas valiosas críticas e sugestões na banca de qualificação, em especial, à *Profa. Dra. Lulu Healy*, por tantas colaborações decisivas ao trabalho.

A todos os *Professores do Programa de Estudos Pós-Graduados em Educação Matemática* da Pontifícia Universidade Católica de São Paulo, pelo seu apoio e profissionalismo, em especial, ao *Prof. Dr. Benedito Antônio da Silva* pela dedicação e competência.

Aos grandes amigos *Cecília e Claudinei*, que tanto me apoiaram e foram solidários nos momentos difíceis.

Aos queridos *colegas de mestrado*, em especial, *Luciane, Benedito, Rose e Agnaldo*, pelo companheirismo e incentivo.

Aos *funcionários da Biblioteca do Centro de Exatas da PUC / SP*, por toda a ajuda prestada e colaboração.

Aos queridos amigos, quase irmãos, *Marlene Lima Pires Corrêa e Ricardo Garanhani Neto*, que tanto me ouviram e incentivaram, estando sempre dispostos a me ceder um ombro amigo para dividir os bons e maus momentos da vida.

A meus queridos pais, *Henrique Ronald Eberson e Elisa Ignez de Campos Eberson*, e meu irmão, *Roberto*, por estarem em todos os momentos ao meu lado, me apoiando e incentivando, registro minha eterna e mais sincera gratidão.

Enfim, a *todos* aqueles que, direta ou indiretamente, contribuíram para que este trabalho se concretizasse.

O Autor.

RESUMO

Esta pesquisa insere-se no quadro do ensino e da aprendizagem de Matemática integrando ferramentas computacionais, em particular, no que se refere à construção de objetos fractais. O objetivo principal deste trabalho é contribuir para uma análise, em termos da *transposição informática* (Balacheff, 1994), da Geometria Fractal em quatro ambientes computacionais de aprendizagem humana. Fundamentado nessa noção, o estudo concentra-se na análise da natureza dos significados que um dado software educativo permite construir para uma determinada noção matemática, isto é, em seu “*domínio de validade epistemológica*”. Além disso, utiliza a idéia de *micro-mundo* no que se refere à ambientes de “*Geometria Dinâmica*” e aqueles baseados na “*Geometria da Tartaruga*”. Tal abordagem teórica justifica e orienta o desenvolvimento metodológico, voltado à análise qualitativa do conjunto de ambientes informáticos escolhidos. As análises realizadas mostram que a construção de fractais em dois desses ambientes relacionam-se à concepção do *Jogo do Caos*. Em ambos os casos, as estruturas formais destes softwares apresentam ferramentas baseadas em transformações geométricas no plano, embora com tipos de controle distintos, em particular, no que se refere à explicitação da relação entre as ferramentas e o conceito envolvido. Os resultados também permitem propor situações de ensino envolvendo a construção de fractais nesses ambientes, visando contribuir para a contextualização de noções relacionadas às transformações geométricas no plano, com a exploração de suas representações geométricas e algébricas.

Palavras-Chave: geometria fractal; ambientes informáticos de aprendizagem; transposição informática; transformações geométricas.

ABSTRACT

This research concerns the teaching and learning of Mathematics in the presence of computational tools, particularly in relation to the construction of fractals. The main aim of the work is to contribute to the analysis of the *computational transposition* process (Balacheff, 1994) of Fractal Geometry in four computational learning environments. The study entails an analysis of the nature of the meanings that each of the educational software environments affords for particular mathematical notions, that is, its “*epistemological domain of validity*”. Additionally the idea of *microworld* is used in relation to “*Dynamic Geometry*” environments and those based of “*Turtle Geometry*”. The theoretical approach adopted warrants and guides the methodological choices, which involve the qualitative analysis of the set of chosen computational environments. The analyses show that the construction of fractals in two of these environments made use of the concept of the *Chaos Game*. In these two cases, the formal structures of both softwares present tools based on geometrical transformations of the plane, although with distinct kinds of control, especially in terms of the extent to which relationships between tools and the concepts involved are made explicit. The results also enable the proposal of teaching situations concerning the construction of fractals in these environments, aimed at providing a context for the exploration of notions related to geometrical transformations of the plane and their geometrical and algebraic representations.

Keywords: fractal geometry; computational learning environments; informatical transposition; geometrical transformations.

SUMÁRIO

APRESENTAÇÃO	11
CAPÍTULO 1: A GEOMETRIA FRACTAL	15
1.1 Afinal, o que são Fractais ?	15
1.1.1 Principais Características da Geometria Fractal.....	17
1.1.2 Dimensões Fractais.....	20
1.1.3 Tipos de Fractais.....	22
1.2 Fractais “Iniciador-Gerador”	24
1.2.1 Conjunto de Cantor.....	25
1.2.2 Fractais de Von Koch.....	27
1.2.3 Fractais de Sierpinski e Esponja de Menger.....	29
1.2.4 Fractais de Peano.....	31
1.2.5 Curva de Hilbert.....	34
1.3 Fractais Gerados por Processos Algébricos	36
1.3.1 Enfoque Matemático e Sistemas Dinâmicos Não-Lineares.....	36
1.4 O Conjunto de Mandelbrot (Mandelbrot Set)	39
1.5 Jogo do Caos	43
1.5.1 Sistemas de Funções Iteradas e Transformações Afins.....	44
1.5.2 Algoritmos Determinísticos.....	47
1.5.2.1 Samambaia determinística (<i>deterministic fern</i>).....	47
1.5.2.2 Outros exemplos de fractais.....	49
1.6 Considerações Finais	52

CAPÍTULO 2: CONCEPÇÕES TEÓRICAS E	
FERRAMENTAS INFORMÁTICAS.....	54
2.1 Considerações Iniciais.....	54
2.2 Objetivos e Problemas de Pesquisa.....	55
2.3 Pressupostos Conceituais da Fundamentação Teórica.....	57
2.4 Transposição Informática.....	60
2.4.1 Domínio de Validade Epistemológica.....	62
2.4.2 Micro-mundos.....	65
2.4.3 Geometria da Tartaruga e Geometria Dinâmica.....	67
2.5 MicroWorlds LOGO.....	71
2.6 Cabri-Gèomètre II.....	78
2.7 Geometer’s SketchPad.....	84
2.8 GeomeTricks.....	89
CAPÍTULO 3: ANÁLISE DA CONSTRUÇÃO DE FRACTAIS	
EM DIFERENTES AMBIENTES INFORMÁTICOS.....	90
3.1 Considerações Iniciais.....	90
3.2 Fractais “Iniciador-Gerador”.....	91
3.2.1 Linguagem LOGO a partir do MicroWorlds.....	91
3.2.2 Cabri Gèomètre II.....	93
3.2.3 Geometer’s SketchPad.....	97
3.3 Fractais Gerados por Seqüências Complexas:	
o Conjunto de Mandelbrot.....	100
3.4 Fractais Gerados pelo “Jogo do Caos”.....	105
3.4.1 Linguagem LOGO a partir do MicroWorlds.....	105
3.4.1.1 Triângulo de Sierpinski.....	106
3.4.1.2 Samambaia determinística.....	110
3.4.2 Geometria Dinâmica a partir do GeomeTricks.....	114
3.4.2.1 Construção de fractais.....	116
3.4.2.2 Triângulo de Sierpinski.....	118
3.4.2.3 Samambaia determinística.....	120

CAPÍTULO 4: TRANSFORMAÇÕES GEOMÉTRICAS A PARTIR DO JOGO DO CAOS.....	122
4.1 Considerações Iniciais.....	122
4.1.1 Pressupostos Iniciais e Escolhas de Âmbito Global.....	123
4.1.2 Concepções Teórico-Metodológicas das Situações de Ensino.....	125
4.2 Análise e Desenvolvimento da Seqüência de Atividades.....	127
4.2.1 Parte I: Fractais no GeomeTricks.....	127
4.2.1.1 Exemplo: construindo o “triângulo de Sierpinski”	129
4.2.1.2 Atividade I: construindo a “cauda do dragão”	131
4.2.1.3 Atividade II: construindo o “ piso fractal”	134
4.2.2 Parte 2: Fractais no MicroWorlds LOGO.....	137
4.2.2.1 Atividade III: construindo o “ piso fractal”	139
4.2.2.2 Atividade IV: construindo a “cauda do dragão”	142
CONCLUSÃO.....	145
REFERÊNCIAS BIBLIOGRÁFICAS.....	150
ANEXO I: TRANSFORMAÇÕES GEOMÉTRICAS NO PLANO – APRESENTAÇÃO MATEMÁTICA	i
ANEXO II: MODELO DAS FICHAS DE ATIVIDADE PARA A PROPOSTA DE SITUAÇÕES DE ENSINO.....	ix

APRESENTAÇÃO

Nas últimas décadas, o grande interesse despertado pela Educação Matemática repousa fundamentalmente na constatação de que a Matemática a ser ensinada possui especificidades que tornam seu processo de ensino e de aprendizagem extremamente complexo e digno de estudos sistematizados e aprofundados. A partir desta constatação, juntamo-nos ao grupo de pesquisa “*Tecnologias e Meios de Expressão em Matemática*” (TecMEM), presente no Programa de Estudos Pós-Graduados em Educação Matemática da PUC-SP, dentro da linha de pesquisa “Tecnologias da Informação e Educação Matemática”, para desenvolvermos nosso estudo.

Nessa perspectiva, o presente trabalho engloba três grandes questões abordadas pelo grupo de pesquisa:

- a análise de características específicas às representações informatizadas de objetos matemáticos, considerando as relações entre estes objetos e os ambientes informatizados nos quais eles podem ser potencialmente estudados;
- a incorporação e transposição de “novos” objetos matemáticos, ainda não explorados enquanto objetos de ensino, assim como sua utilização didática em conjunto com ambientes informáticos;
- a idéia da contextualização de conceitos matemáticos a partir da integração de recursos tecnológicos e da criação de cenários de aprendizagem voltados à resolução de problemas.

Dessa forma, relativa à primeira questão, pretendemos inserir o trabalho na problemática da utilização das novas tecnologias no processo didático, visto que atualmente já não se discute “se” os computadores devem ser incorporados ao processo de ensino e de aprendizagem, mas “como” essa utilização pode ser potencializada. Por sua vez, na segunda questão, elegemos a Geometria Fractal, visto que esta nova “geometria da natureza”, apesar de conhecida, ainda é pouco

explorada enquanto objeto de ensino. Finalmente, a terceira questão visa relacionar as anteriores ao assumir, como pressuposto, que o desenvolvimento de cenários de aprendizagem integrando recursos tecnológicos, e voltados à resolução de problemas, pode fornecer ferramentas úteis para a contextualização de noções matemáticas abstratas. Nesse sentido, pretendemos discutir o desenvolvimento de situações de ensino, visando explorar conceitos relacionados à construção de fractais em ambientes computacionais.

No sentido de contribuir, ao menos inicialmente, para um aprofundamento destas questões, abordaremos os seguintes problemas de pesquisa:

(i) Que transformações resultam do esforço de representação decorrente da passagem dos modelos matemáticos, utilizados na construção de objetos fractais, para modelos computáveis, utilizados em ambientes informáticos de aprendizagem? Quais as conseqüências dessas transformações numa perspectiva didática?

(ii) Que contribuições podem advir da utilização dos processos de construção de fractais, assim como dos conceitos matemáticos a eles relacionados, no sentido da contextualização do ensino e aprendizagem de determinadas noções matemáticas?

Assim, o objetivo deste trabalho concentra-se especificamente em abordar aspectos da *Transposição Informática* (Balacheff, 1994) de objetos fractais, em particular, realizando uma análise da dimensão epistemológica de um conjunto de ambientes informáticos de aprendizagem humana. Em outros termos, esse estudo visa analisar as possibilidades de construção de objetos fractais em softwares educativos, no que se refere às ferramentas, representações, tipos de controles e feedback por eles fornecidos.

Em termos teóricos e metodológicos, o trabalho fundamenta-se essencialmente na noção de Transposição Informática, desenvolvida por Nicolas Balacheff (1994), assim como em outros trabalhos de sua co-autoria que ampliam esta noção (Balacheff & Sutherland, 1994; Balacheff & Kaput, 1996). Além disso, também nos interessamos pela noção de *micro-mundo* e em concepções relativas à engenharia de softwares educativos (Hoyles, 1993).

Desta forma, dentro das questões relativas à Transposição Informática, destaca-se a análise das possibilidades disponibilizadas por um dado ambiente, a partir da delimitação de seu “*Domínio de Validade Epistemológica*” (Balacheff & Sutherland, 1994). Esta questão engloba essencialmente o sentido que um dado software educativo permite construir para uma determinada noção matemática, ou seja, analisa-se a natureza dos significados possibilitados pelas propriedades dos sistemas de representação, que são disponibilizados pelos softwares. Em particular, na análise de ambientes voltados ao ensino da Geometria, foco deste trabalho, tal concepção conduz a dois tipos de ambientes informáticos: os micro-mundos voltados à “Geometria Dinâmica” e os baseados na “Geometria da Tartaruga” (*Turtle Geometry*), cujas concepções integram-se a nosso estudo.

Nessa perspectiva, a noção de *domínio de validade epistemológica* de um micro-mundo visa fornecer tanto um referencial teórico, como criar um conjunto de ferramentas metodológicas para a análise qualitativa desses ambientes. Assim, ainda segundo Balacheff & Sutherland (1994, p. 148), a noção de *domínio de validade epistemológica* de um micro-mundo levanta, pelo menos, quatro dimensões teórico-metodológicas a serem consideradas na análise de um dado domínio de conhecimento matemático, e isto faz com que tais dimensões se tornem a base de nossas análises.

Estruturamos este trabalho de maneira a ser possível explorar separadamente e de forma abrangente os diversos componentes nos quais ele se baseia, com as devidas fundamentações teóricas e metodológicas utilizadas. Assim, adotamos a seguinte seqüência para o corpo do trabalho.

O capítulo 1 apresenta três “famílias” de objetos fractais, enfatizando suas características matemáticas, trajetórias histórico-epistemológicas e formas originais de construção. Como o objetivo deste capítulo é situar o leitor dentro da Geometria Fractal, adotamos uma abordagem mais abrangente do que a necessária para este trabalho, porém tal postura se justifica dentro da proposta do capítulo de responder a questões ligadas às origens matemáticas dos fractais e sobre suas formas originais de construção, situando-a enquanto objeto matemático.

O capítulo 2 expõe detalhadamente as fundamentações teóricas e metodológicas adotadas no trabalho e propõe uma análise inicial dos ambientes

informáticos escolhidos. Como dissemos, nos inspiramos na noção de Transposição Informática e, mais especificamente, no “*domínio de validade epistemológica*” (Balacheff & Sutherland, 1994) de micro-mundos educativos. Assim, este capítulo pretende analisar quatro ambientes informáticos, no que se refere à natureza das ferramentas e objetos fornecidos em suas estruturas formais, assim como certos aspectos relativos à fenomenologia dessas estruturas.

O capítulo 3 propõe uma análise mais aprofundada de certos aspectos do “*domínio de validade epistemológica*” desses ambientes informáticos, estudando as possibilidades específicas de construção de diferentes tipos de fractais em cada um dos ambientes informáticos analisados. Com isso, pretendemos retomar e aprofundar as análises desses ambientes, focando os tipos de controle e “feedback” específicos existentes nessas estruturas.

Finalmente, o capítulo 4 dedica-se à discussão da última dimensão de análise, proposta pela noção de “*domínio de validade epistemológica*”, ou seja, identificar conjuntos de problemas que podem ser propostos em um determinado micro-mundo. Além disso, este capítulo concentra-se no desenvolvimento e análise de uma proposta voltada à criação de situações de ensino, baseadas em dois dos micro-mundos estudados. Com tal proposta, pretendemos explorar as potencialidades dessas ferramentas computacionais, quanto à relação entre a construção de fractais e o conceito de Transformações Geométricas no Plano, no sentido da contextualização desse conceito e da utilização de suas representações geométricas e algébricas.

CAPÍTULO 1

A GEOMETRIA FRACTAL

Por que a geometria é freqüentemente descrita como “fria” e “seca”? Uma razão repousa em sua inabilidade de descrever a forma de uma nuvem, uma montanha, uma linha costeira ou uma árvore. Nuvens não são esferas, montanhas não são cones, linhas costeiras não são círculos, e uma casca de árvore não é lisa, tampouco um feixe de luz viaja em linha reta. (...) Eu afirmo que muitos dos padrões da Natureza são tão irregulares e fragmentados, que, se comparados com a geometria tradicional, exibem não somente um grau mais alto, mas um nível de complexidade completamente diferente¹.

(Mandelbrot, 1977, p. 1)

1.1 Afinal, o que são Fractais ?

O corpo de conhecimentos que hoje denotamos por Geometria Fractal engloba uma enorme classe de figuras que dificilmente podem ser agrupadas em poucas categorias devido a grande variedade de formas, características e tipos de construção que possuem. De fato, sequer existe uma relação ou classificação para os diversos tipos de fractais que são atualmente explorados pela Matemática e demais ciências, uma vez que esta é uma área ainda em efervescente desenvolvimento, mais voltada ao crescimento do que ao aprofundamento e na qual novas descobertas e criações são anunciadas com novos enfoques e novas aplicações.

Segundo Mandelbrot (1991, pp. 207-208), a Geometria Fractal, “apesar de largamente aceita (...) ainda não se tornou ‘acadêmica’, mantendo uma

¹ Traduzido por nós a partir do original em inglês.

diversidade que é intrínseca, rara, divertida e importante. Não só levanta ainda questões fundamentais, como continua a desencadear polêmicas”. Apesar de relativamente antiga, percebemos que esta afirmação ainda se mantém atual, motivo pelo qual optamos por um estudo abrangente de diversos tipos de objetos fractais, enfocando suas principais características históricas e epistemológicas, assim como os processos matemáticos originais que possibilitaram suas construções.

A maior prova da diversidade presente na Geometria Fractal resume-se no fato de que sequer há uma definição formal que seja consensual e definitiva do que venha a ser um fractal, o que pode ser constatado em algumas das definições que foram dadas em diferentes obras. Em um de seus primeiros trabalhos sobre o tema², Mandelbrot afirmou que:

Uma vez definido qualquer conceito fractal de dimensão, chegando ao valor D , pode-se **tentar** [grifo nosso] definir um fractal como sendo, ou um conjunto para o qual D é um real não-inteiro, ou um conjunto para qual D é um inteiro, mas o todo é “irregular”. (...)
Do ponto de vista concreto, tais exemplos seriam insuportáveis. É para os evitar que prescindo de definir o conceito de conjunto fractal.

(Mandelbrot, 1991, p. 176)

Já em seu segundo livro, Mandelbrot nos fornece uma definição mais formal de um conjunto fractal, afirmando que “*um fractal é por definição um conjunto para o qual a dimensão de Hausdorff-Besicovitch estritamente excede a dimensão topológica*”. (Mandelbrot, 1977, p. 15). Fica claro que, para uma mínima compreensão desta nova definição, seria preciso explorar diversos conceitos da Topologia, especialmente relacionados aos Espaços Métricos, conceitos utilizados para a definição da dimensão de *Hausdorff-Besicovitch*, porém, tentaremos explorar melhor o conceito de dimensão fractal em momento oportuno.

Iremos apresentar também o trabalho de Michael Barnsley, que tratou a Geometria Fractal com grande rigor matemático, enunciando diversas definições e teoremas com o objetivo de melhor caracterizar o tema. Em princípio, Barnsley conceitua a geometria fractal como preocupada com “a estrutura de subconjuntos de vários espaços ‘geométricos’ muito simples (...) ao passo que o espaço é

² Deve-se ressaltar que seguimos a cronologia dos trabalhos de Mandelbrot e que a disparidade de datas se deve ao fato de termos utilizado a tradução portuguesa da obra “*Les Objects Fractals*”, editada originalmente em 1975.

simples, o subconjunto fractal pode ser geometricamente complicado” (Barnsley, 1993, p. 6), concluindo após diversas proposições que “o espaço dos fractais é um espaço métrico completo³” (*ibid.*, pp. 29-35).

Enfim, do que foi dito até agora, pode-se chegar a duas conclusões principais. A primeira nos leva a afirmar que uma análise rigorosa e formal da geometria fractal do ponto de vista da Matemática pura nos conduzirá principalmente à Topologia e Espaços Métricos e a segunda que as definições formais e a análise rigorosa da geometria fractal, dado seu alto grau de abstração e complexidade, contribuem pouco no sentido de uma compreensão concreta e minimamente objetiva do que realmente são os fractais.

Com essa afirmação não pretendemos de forma alguma desprezar o enfoque rigoroso e formal dado pela Matemática ao tema e sim justificar o motivo pelo qual não iremos nele nos aprofundar. Ao contrário, nossa postura para a análise da geometria fractal pode ser resumida nas palavras de Barnsley (1993, p. 35) ao afirmar que “os fractais não são definidos por uma pequena definição formal, mas pelas diversas figuras e contextos que se referem a eles”.

1.1.1 Principais Características da Geometria Fractal

Visando uma melhor compreensão da geometria fractal em termos mais objetivos e concretos, iremos explorar agora alguns dos principais aspectos que a diferenciam da geometria euclidiana tradicional. Apesar de enorme variedade de formas e figuras que hoje são “catalogadas” como fractais, é possível enumerar três características essenciais que estão presentes em todos os fractais e que podem servir, pelo menos inicialmente, para caracterizar esse novo conjunto de formas geométricas. A primeira dessas características está relacionada à aparência dos fractais e, apesar de ser essencialmente intuitiva e visual, é

³ Segundo Lima (1977, p.1), um *espaço métrico* é um par (M, d) , onde M é um conjunto e d é uma métrica em M . Por sua vez, uma métrica num conjunto M é uma função $d: M \times M \rightarrow \mathbb{R}$, que associa a cada par ordenado de elementos $x, y \in M$ um número real $d(x, y)$, chamado distância de x a y , de modo a satisfazer as seguintes condições para quaisquer $x, y, z \in M$:

- d1) $d(x, x) = 0$;
- d2) Se $x \neq y$ então $d(x, y) > 0$;
- d3) $d(x, y) = d(y, x)$;
- d4) $d(x, z) \leq d(x, y) + d(y, z)$.

importante para um reconhecimento inicial dessas figuras. A segunda se relaciona à maneira como os fractais são construídos, uma vez que, mesmo utilizando objetos matemáticos completamente distintos, teremos sempre um mesmo enfoque básico na construção de todos os tipos de fractais. A terceira e última característica se relaciona a um aspecto teórico dos fractais, importantíssimo do ponto de vista matemático e principal responsável pela grande ruptura que a geometria fractal causou na Matemática tradicional.

Visualmente, os fractais possuem como característica básica a “auto-semelhança” (*self-similarity*), o que, em termos simples, significa dizer que pequenas partes da curva repetem a forma da curva como um todo, ou seja, se fizermos uma ampliação de uma região específica de um fractal (mesmo quando ampliada milhares de vezes), iremos encontrar uma réplica do fractal como um todo.

Essa característica, como dissemos, é principalmente visual e intuitiva, visto que nem todos os fractais apresentam réplicas exatas da curva como um todo em todas as escalas de ampliação, porém encontramos uma certa semelhança intrigante na forma e na aparência das ampliações de determinadas regiões dos fractais, quando comparadas com a figura como um todo, chamando, nesses casos, a auto-semelhança de “estatística”. Entretanto, nos fractais tipo “iniciador-gerador”, onde o processo de construção geométrica é repetido, em tese, infinitas vezes, são produzidas efetivamente réplicas exatas da curva como um todo em qualquer escala de ampliação, nesses casos, originando o que se definiu como auto-semelhança “estrita”.

A segunda característica dos fractais se refere a sua construção, que sempre utiliza, de uma forma ou outra, algum tipo de “Processo Iterativo”, significando que, na construção de qualquer fractal, iremos **repetir** um determinado procedimento infinitamente, seja este procedimento um conjunto de cálculos algébricos ou uma determinada construção geométrica.

Está característica básica da construção dos fractais é, em grande parte, a responsável pelo grande fascínio que estas figuras provocam pois, na maioria dos casos, os fractais são construídos a partir de “elementos” extremamente simples mas que, apesar disso, dão origem a figuras com extraordinária complexidade e

riqueza de detalhes justamente graças às infinitas iterações⁴ presentes em sua construção. Infelizmente, devido a grande diversidade de formas de construção dos fractais, não nos é possível ainda deixar esta característica menos abstrata, bastando dizer que o termo “elementos” que utilizamos a pouco engloba funções complexas, sistemas de funções reais, construções geométricas, entre outros. De qualquer forma, deve-se sempre ter em mente algum tipo de processo iterativo como o enfoque principal presente na construção de quaisquer fractais.

A terceira e última característica dos fractais que iremos explorar é mais teórica e se refere à sua dimensão. Até então, a dimensão de uma figura era tradicionalmente representada por números inteiros, assim, na geometria euclidiana, pontos têm dimensão zero, retas têm dimensão um, planos têm dimensão dois e o espaço, dimensão três e todas as formas geométricas se encaixam perfeitamente em alguma dessas categorias. Mesmo a Topologia utilizava sempre valores inteiros para definir uma dimensão. Porém, nos fractais criou-se a necessidade de uma nova abordagem, uma vez que as formas geradas não se encaixavam nas categorias euclidianas ou topológicas. Dessa forma, as figuras criadas pela geometria fractal acabaram por sugerir um outro tipo de análise, a “dimensão fractal”, o que acabou fazendo com que o valor que irá expressar a dimensão de um objeto fractal seja, em geral, um valor não-inteiro.

Na verdade, foi essa característica das curvas fractais que deu origem a seu nome. Segundo o pai da geometria fractal, o matemático Benoit Mandelbrot (1977, p. 4), palavra fractal foi “criada” quando ele folheava o dicionário de latim de seu filho, onde encontrou o adjetivo *fractus*, do verbo *frangere*, significando quebrar, fraturar, criar fragmentos irregulares. A associação com os cognatos ingleses *fracture* e *fraction* parecia adequada e assim Mandelbrot criou a palavra *fractal* (Gleick, 1990, p. 93). De fato, a “dimensão fractal” é uma característica tão fundamental para o tema que iremos dedicar uma seção específica para uma abordagem mais completa do assunto.

⁴ Deve ficar claro que as “infinitas” iterações que citamos partem de um enfoque teórico da construção dos fractais, visto que, na prática, o processo será repetido até o limite da

1.1.2 Dimensões Fractais

A idéia de uma dimensão matemática que resultasse num valor não-inteiro é, na verdade, anterior ao estudo sistemático da geometria fractal e tem suas primeiras bases na grande “crise” sofrida pela Matemática entre os anos de 1875 e 1925 que, especialmente nos campos da Topologia e Espaços Métricos, acabou por criar diversas definições de dimensão fractal. Dentre elas, a primeira a ser desenvolvida foi a de Hausdorff-Besicovich, em 1919 (Mandelbrot, 1991, p.176).

Em sua maioria, essas novas formas de se medir a dimensão se referem a figuras muito gerais e partem de conceitos matemáticos extremamente abstratos, normalmente iniciando-se a partir de um dado espaço métrico onde as definições formais de dimensão fractal são construídas. Tais definições estão presentes em diversos trabalhos já publicados sobre o tema e, portanto, não será nosso objetivo reproduzi-las, entretanto, pretendemos apresentar uma versão simplificada de dimensão fractal, que resulta no mesmo resultado das definições mais abstratas quando aplicada a uma determinada família de fractais – a “*dimensão de similaridade*”. Esta definição é particularmente útil para caracterizar a construção de fractais regulares, em particular do tipo “iniciador-gerador”, obtendo nessas figuras o mesmo valor dado pelas definições mais abstratas de dimensão, como a de Hausdorff-Besicovich.

O conceito de dimensão de similaridade, que denotaremos por D_S , está intimamente ligado com a idéia de homotetia interna e a simplicidade de sua estrutura em relação às demais definições de dimensão fractal se justifica devido ao fato de sua validade se aplicar apenas a um conjunto restrito de fractais regulares. Dessa forma, o conceito de dimensão de similaridade que demonstraremos agora será válido apenas para fractais que possuem auto-similaridade estrita, ou seja, em fractais onde são formadas réplicas exatas da figura como um todo em qualquer escala de ampliação, garantindo, com isso, que tais figuras se encaixem no domínio de validade da dimensão de similaridade, por possuírem homotetia interna.

Nas diversas obras sobre o tema, são encontradas definições para a dimensão de similaridade com a mesma estrutura mas com notações e

visualização gráfica possibilitada pelas ferramentas que geram a imagem, via de regra, ferramentas informáticas.

demonstrações ligeiramente diferentes, dessa forma, optamos aqui pela demonstração utilizada na obra “Fractal and Chaos” (Addison, 1997, pp. 14-16) por ser bastante completa e acessível aos nossos propósitos⁵.

Sejam um segmento de reta, uma superfície e um sólido quaisquer, divididos respectivamente por “sub-comprimentos”, “sub-áreas” e “sub-volumes” homotéticos com lados de comprimento “ ε ” de forma que possamos considerar “ ε ” como um padrão de escala. Dessa forma, iremos considerar a idéia de auto-semelhança estrita como sinônimo de homotetia interna, ressaltando porém que para a geometria fractal é dado um significado mais amplo para este conceito. Para simplificar o raciocínio, assumiremos que o comprimento, L , a área, A e o volume, V , são unitários.

Consideremos primeiramente o segmento de reta. Se o segmento for dividido em N segmentos auto-semelhantes de comprimento ε , ou seja, segmentos de mesmo comprimento, então poderemos considerar ε como um padrão de escala e assumir que $\varepsilon/L = \varepsilon$, uma vez que $L = 1$. Portanto:

$$L = N \varepsilon = 1 \quad (a)$$

Este raciocínio faz com que o segmento unitário seja composto de N partes auto-semelhantes, cada uma medindo $\varepsilon = 1/N$.

Consideremos agora a superfície unitária. Se dividirmos novamente a superfície em N partes, cada uma com área ε^2 , teremos:

$$A = N \varepsilon^2 = 1 \quad (b)$$

O que faz com que a superfície unitária seja composta de N partes auto-semelhantes, cada uma medindo $\varepsilon = 1/N^{1/2}$.

Com o mesmo raciocínio, podemos obter, para o volume unitário, a expressão:

$$V = N \varepsilon^3 = 1 \quad (c)$$

⁵ Um tratamento mais formal e abrangente do tema, focado à Topologia e Análise, é dado em Mandelbrot (1991, pp. 175-185) e Bloch, I (2000, pp. 66-76).

Novamente fazendo com que o volume unitário seja composto de N partes auto- semelhantes que medem $\varepsilon = 1/N^{1/3}$.

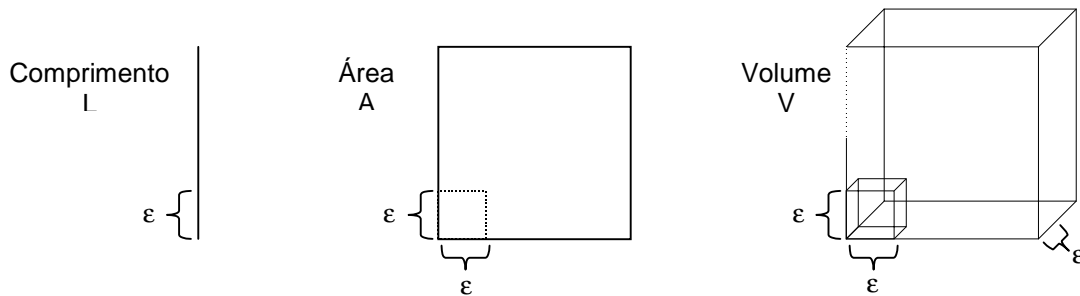


Figura 1.1: Determinação de sub-comprimentos, sub-áreas e sub-volumes.

Examinando as expressões a, b e c, pode-se notar que o expoente de ε em cada caso reflete a medida da dimensão de similaridade de cada objeto, o que, generalizando, nos fornece:

$$N \varepsilon^{D_s} = 1$$

Finalmente, após o devido tratamento matemático, chegamos a expressão:

$$D_s = \frac{\log(N)}{\log(1/\varepsilon)}$$

Expressão esta que nos fornece uma forma para se determinar a dimensão de similaridade para fractais que possuem homotetia interna.

1.1.3 Tipos de Fractais

Como dissemos, seria impossível explorar todos os tipos de fractais existentes dada a abrangência deste trabalho, portanto, iremos aqui nos ater a três “famílias” de fractais, ressaltando que essa separação foi por nós desenvolvida somente com fins didáticos para uma introdução mais confortável aos principais aspectos da Geometria Fractal. O critério utilizado para essa separação levou em conta os métodos de construção dos fractais, portanto, agruparemos em cada “família”, figuras que são construídas da mesma maneira.

A primeira dessas famílias de fractais será gerada por um “iniciador-gerador”, que utiliza processos de construção essencialmente geométricos,

semelhantes às construções tradicionais com régua e compasso. A segunda família irá utilizar o resultado gráfico da iteração de seqüências de números complexos plotados no plano cartesiano, englobando essencialmente o “conjunto de Mandelbrot” e os “conjuntos de Julia”. Finalmente, a terceira família de fractais que iremos explorar também utiliza o resultado gráfico da iteração de seqüências de números plotados no plano cartesiano, porém utilizando sistemas de funções reais para esse fim, no processo tradicionalmente conhecido como “Jogo do Caos”.

1.2 Fractais “Iniciador-Gerador”

Uma das formas de se gerar um fractal, na verdade a primeira delas, baseia-se em um processo iterativo simples de iniciador-gerador que foi utilizado por grandes matemáticos no final do séc. XIX e início do séc. XX para construir curvas bizarras e anômalas que desafiavam a análise, álgebra e geometria. É importante ressaltar que a maioria dessas figuras foi construída décadas antes da geometria fractal se tornar um objeto matemático e antes mesmo de existir a palavra “fractal” que hoje as denomina, portanto, o enfoque e o interesse matemático que estas “curvas monstro” ou “patológicas” despertaram não é exatamente o mesmo que temos hoje ao estudar um fractal.

Esses fractais partem de uma figura inicial chamada de “*iniciador*”, que, em tese, pode ser qualquer figura geométrica regular, porém, em geral são utilizados segmentos de reta, triângulos ou quadrados. Em seguida é definido um “*gerador*”, que por sua vez, consiste um conjunto de segmentos de reta consecutivos que irá substituir cada um dos segmentos de reta do iniciador. Na maioria dos casos, o gerador consiste em N segmentos de reta, cada um medindo ε , onde ε é uma fração do comprimento dos segmentos de reta que serão substituídos no iniciador.

Como regra, a disposição dos N segmentos de reta deve ser tal que a distância do início ao fim do gerador seja igual ao comprimento dos segmentos de reta a serem substituídos no iniciador. O processo iterativo então se inicia ao se substituir cada segmento do iniciador pelo gerador, o que irá definir o *nível* de iterações. Cada novo nível do fractal é construído repetindo-se o processo original, onde os novos segmentos da figura são novamente substituídos por uma réplica do gerador em escala reduzida, avançando-se um nível na iteração. Para se determinar o fractal, o número de níveis de interação será, em tese, infinito. Finalmente, uma vez que o fractal deve ser uma curva contínua, algumas regras devem ser observadas para a construção do gerador, que não pode ter pontos de intersecção nem se sobrepor (self-overlapping), mesmo após as infinitas iterações.

Os fractais construídos pelo processo “iniciador-gerador” possuem auto-semelhança estrita, ou seja, devido ao próprio processo de construção, serão formadas réplicas exatas da curva como um todo em qualquer escala de ampliação, fazendo com que esta família de fractais forme figuras invariantes em escala. Para uma idéia mais clara e palpável desse tipo de fractal, iremos expor com detalhes alguns dos mais famosos fractais do tipo “iniciador-gerador” a seguir.

1.2.1 Conjunto de Cantor

Uma das primeiras figuras citadas em praticamente todas as bibliografias relacionadas à geometria fractal é o “Conjunto de Cantor” ou “Poeira de Cantor”, desenvolvida por Georg Cantor (1845-1918). Este fractal primitivo, que está ligado ao trabalho mais relevante de Cantor – a Teoria dos Conjuntos, apesar de não ser tão atraente como a maioria dos fractais, possui características matemáticas bastante incomuns. A figura é construída partindo-se de um segmento de reta com comprimento unitário que é subdividido em três partes iguais. Feito isso, o terço médio do segmento é retirado, repetindo-se o mesmo processo nos dois segmentos restantes, e assim sucessivamente. O conjunto de Cantor é a “poeira” de pontos que fica após as infinitas iterações onde, mesmo restando infinitos pontos, possui extensão total zero.

A dimensão de similaridade do conjunto é não-inteira e, como em cada etapa, são obtidos dois novos segmentos com um terço do segmento anterior, temos $N = 2$ e $\varepsilon = 1/3$, portanto:

$$D_s = \frac{\log (N)}{\log (1/\varepsilon)} = \frac{\log 2}{\log 3} \cong 0,6309$$

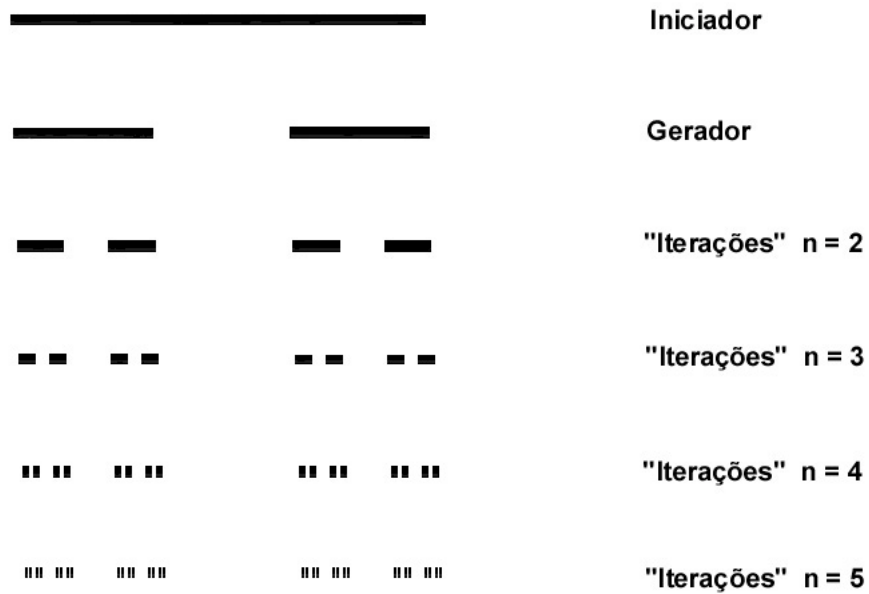


Figura 1.2: Processo iniciador-gerador para os primeiros níveis de iteração do Conjunto de Cantor.

Analisando o Conjunto de Cantor matematicamente, pode-se considerar o segmento iniciador como o conjunto de pontos contidos no intervalo fechado $I_0 = [0,1]$. Na primeira iteração, ao se retirar o terço médio, forma-se o intervalo $I_1 = [0, 1/3] \cup [2/3, 1]$. Na segunda iteração, forma-se o intervalo $I_2 = [0, 1/9] \cup [2/9, 3/9] \cup [6/9, 7/9] \cup [8/9, 1]$. Como o Conjunto de Cantor é obtido repetindo-se o processo indefinidamente, este será formado por todos os pontos que não foram retirados. É interessante notar que os pontos extremos de cada “sub-segmento” gerado nas sucessivas iterações farão parte do conjunto, uma vez que, em cada nível de iteração, são retirados apenas pontos interiores de cada intervalo (Carvalho et al., s.d., pp. 24-26).

1.2.2 Fractais de Von Koch

O matemático Helge von Koch (1870-1924) foi o idealizador de alguns dos mais famosos fractais regulares existentes – a “Curva de Koch” e o “Floco de Neve de Koch”. Ambas as figuras foram exibidas pela primeira vez em 1904 (Stevens, 1990, p. 93) e ganharam tamanha notoriedade devido às características patológicas e incomuns que ambas possuem. Entre outras características, a Curva de Koch, apesar de ser uma curva limitada, possui comprimento infinito e o floco de neve é uma figura geométrica regular composta por uma área finita limitada por um perímetro infinito. Além disso, ambas mantêm a característica topológica de uma curva de dimensão unitária, uma vez que elas não se interceptam em nenhum ponto.

A curva de Koch possui um segmento de reta como iniciador e o gerador divide este segmento em três partes iguais onde o terço médio é retirado e substituído por dois lados de um triângulo eqüilátero, como mostrado abaixo.

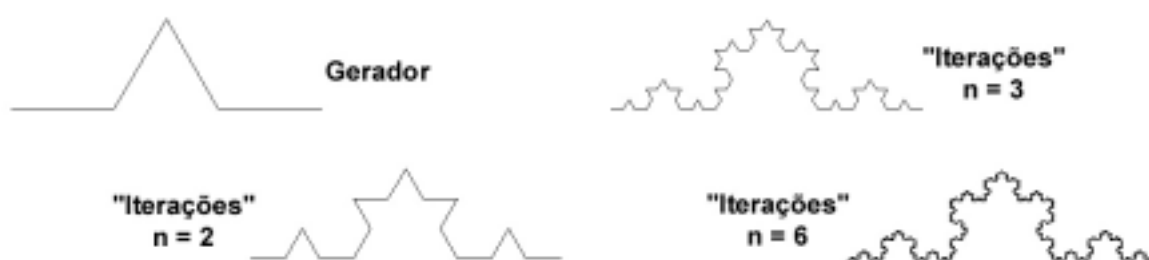


Figura 1.3: Processo "iniciador-gerador" para a construção da Curva de Koch.

Já o floco de neve de Koch possui um triângulo eqüilátero como iniciador e o mesmo gerador da curva de Koch. No floco de neve pode-se perceber mais facilmente as características incomuns desses fractais, pois temos uma figura regular fechada de lados e perímetro infinitos cercado uma área finita, que, em termos simples, nunca será maior que a área da circunferência circunscrita ao triângulo eqüilátero iniciador. As expressões que fornecem a área e o perímetro exatos para o floco de neve podem ser facilmente determinadas através da soma dos infinitos termos de uma Progressão Geométrica que, partindo-se de um triângulo eqüilátero de lado unitário, nos dará um perímetro de $3 \cdot \left(\frac{4}{3}\right)^n$, onde n

corresponde ao número de iterações, e área de $\frac{2\sqrt{3}}{5}$ (uma demonstração detalhada e ilustrada dessas afirmações pode ser da obra de Carvalho, Maria Cecília C. S. et al., s.d., pp. 159-163).

A dimensão de similaridade, tanto do floco de neve como da curva de Koch são iguais, já que ambas possuem o mesmo gerador e, uma vez que, tanto o segmento de reta iniciador da curva de Koch como cada lado do triângulo iniciador do floco de neve são substituídos por quatro segmentos iguais com $1/3$ do comprimento do segmento anterior, temos $N = 4$ e $\epsilon = 1/3$, portanto :

$$D_s = \frac{\log(N)}{\log(1/\epsilon)} = \frac{\log 4}{\log 3} \cong 1,2617$$

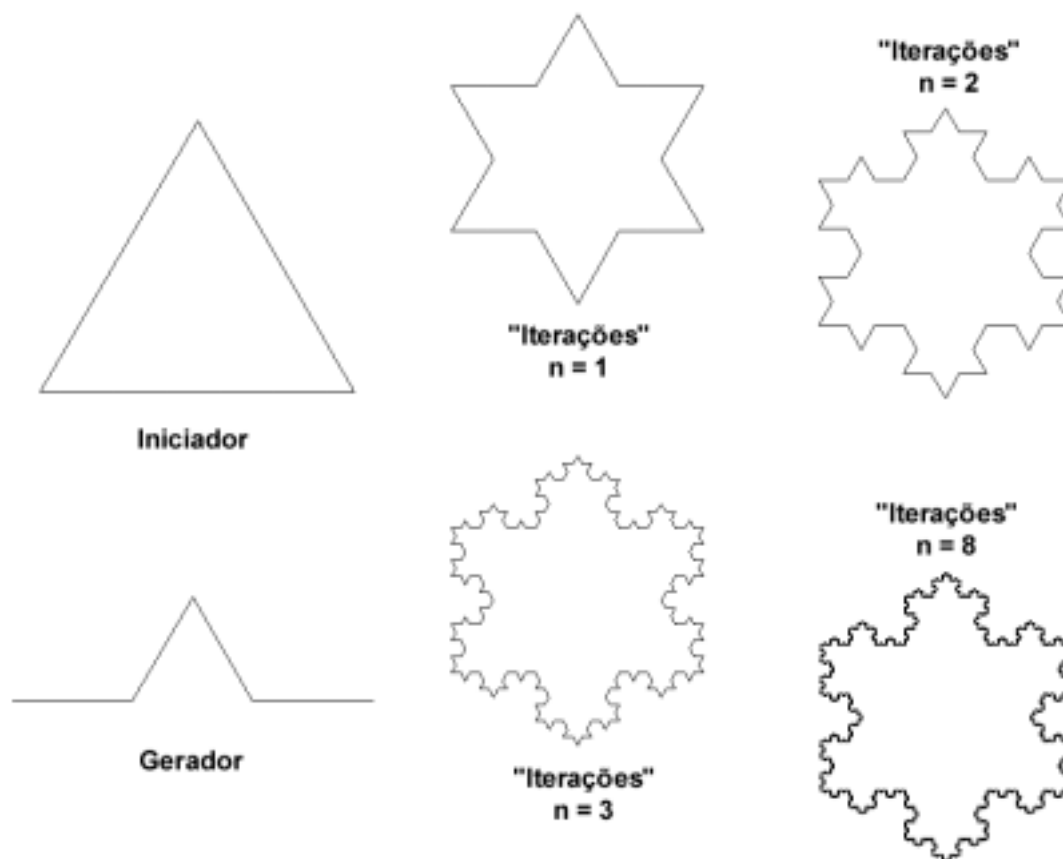


Figura 1.4: Processo "iniciador-gerador" para a construção do Floco de Neve de Koch.

Além das características bizarras citadas acima, o que tornou os fractais de Koch interessantes para grandes matemáticos de sua época, como Hilbert, Peano e Poincaré, se deve a aspectos ainda mais abstratos dessas construções que

“não só não tem tangente em nenhum ponto mas tem a notável propriedade de que, dados dois pontos quaisquer sobre a curva, o comprimento do arco entre os dois pontos é infinito.” (Boyer, 1974, p. 448). Finalmente, é interessante notar que ao se observar o segmento que constitui a “base” da curva de Koch, iremos encontrar uma réplica exata do “Conjunto de Cantor”.

1.2.3 Fractais de Sierpinski e Esponja de Menger

Outro grande matemático dessa época que deixou seu nome marcado na geometria fractal foi o polonês Waclaw Sierpinski (1882-1969) com a criação, em 1916, de um fractal triangular que recebeu seu nome. O triângulo de Sierpinski tem as mesmas características dos fractais geométricos analisados até aqui mas acabou se tornando, junto com as curvas de Koch, um dos fractais mais famosos e citados pelas bibliografias especializadas. Um detalhe interessante e particular sobre o triângulo de Sierpinski que o torna diferente dos demais fractais que analisaremos é a variedade de formas como podemos construí-lo, contando com alguns processos “iniciador-gerador” completamente distintos, além de outro que utiliza o Jogo do Caos e que será analisado mais adiante.

Além disso, o triângulo de Sierpinski parece possuir uma certa universalidade bizarra que aparece em algumas obras matemáticas ao ser associado com domínios totalmente diversos das suas origens. Um interessante exemplo disso aparece em Peitgen, Jürgens e Saupe (1991a, pp. 96-102) onde são apresentados os números que constituem o triângulo de Pascal, dispostos graficamente de forma a se ressaltar os números pares e ímpares com cores diferentes, obtendo como resultado uma figura assustadoramente parecida com o triângulo de Sierpinski.

Objetivamente, o triângulo de Sierpinski original parte de um triângulo equilátero como iniciador e a partir deste é criado um novo triângulo equilátero menor cujos vértices são os pontos médios do triângulo original, o processo tem continuidade ao se retirar este novo triângulo do original e se repetir a mesma construção nos três triângulos gerados e assim sucessivamente, como mostrado abaixo.

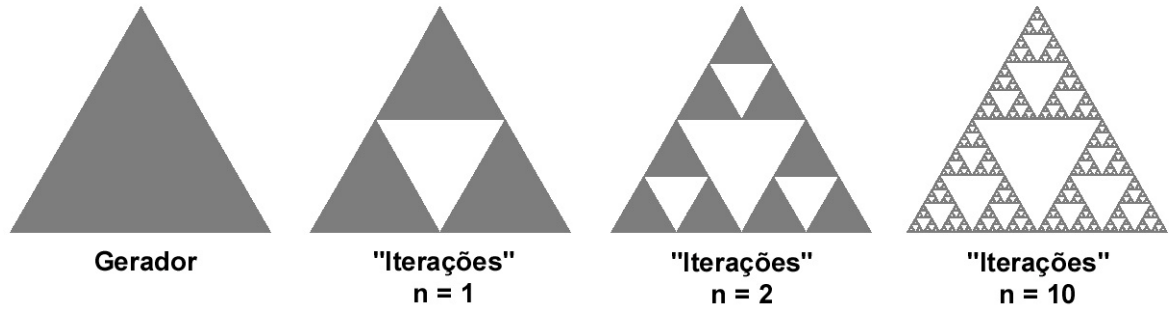


Figura 1.5: Processo original de construção do Triângulo de Sierpinski.

Outra forma de se construir o Triângulo de Sierpinski parte de um segmento de reta como iniciador e um gerador formado por um trapézio sem sua base maior. Este trapézio é obtido a partir de um triângulo equilátero formado pelo segmento iniciador cuja base menor passa pelos pontos médios de dois dos lados desse triângulo.

A dimensão de similaridade do triângulo de Sierpinski é obtida ao se analisar o gerador dessa figura, que possui três segmentos de reta iguais ($N = 3$), com metade do comprimento do segmento iniciador ($\epsilon = 1/2$), logo:

$$D_s = \frac{\log(N)}{\log(1/\epsilon)} = \frac{\log 3}{\log 2} \cong 1,58496$$

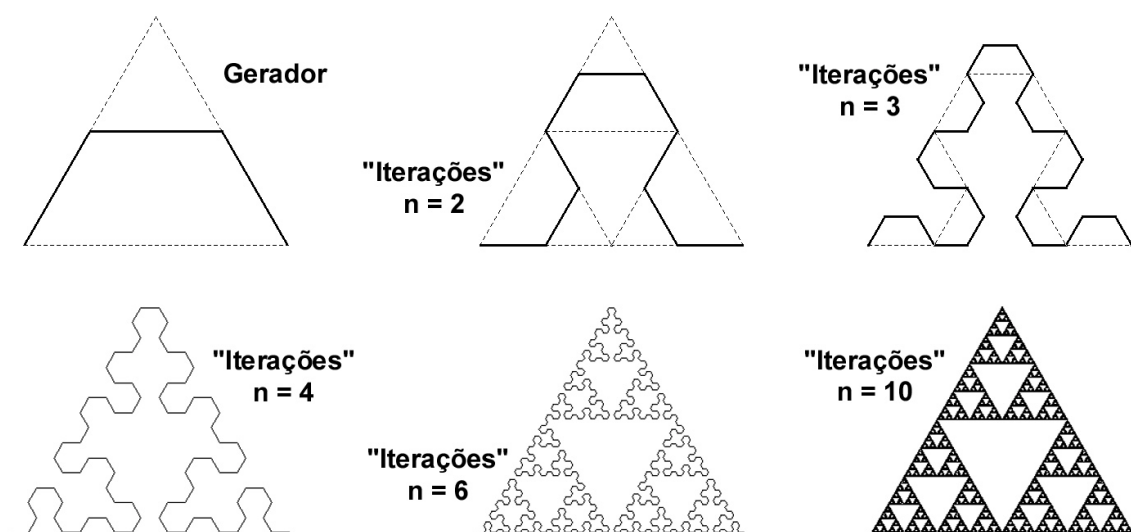


Figura 1.6: Processo "iniciador-gerador" para a construção do Triângulo de Sierpinski.

Outras duas figuras interessantes que se assemelham ao triângulo de Sierpinski e mostram quão bizarros podem ser os fractais, ficaram conhecidas como “Tapete de Sierpinski” e “Esponja de Menger”. O tapete de Sierpinski é, de certa forma, uma versão quadrada do triângulo, sendo obtido com um quadrado iniciador do qual é retirado o nono central, repetindo-se o processo nos oito quadrados restantes e assim sucessivamente. Já a esponja de Menger é um objeto tridimensional análogo ao tapete de Sierpinski, “um rendado de aparência sólida que tem uma área de superfície infinita, e não obstante, um volume nulo.” (Gleick, 1990, p. 96).

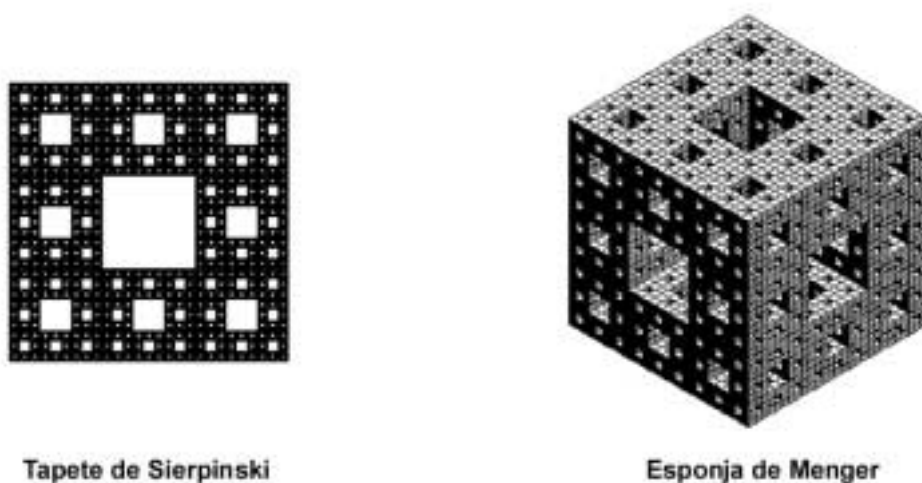


Figura 1.7: Representação dos fractais “Tapete de Sierpinski” e “Esponja de Menger”.

1.2.4 Fractais de Peano

As “Curvas de Peano” foram assim batizadas em homenagem ao matemático Giuseppe Peano (1858-1932) que descreveu a primeira dessas curvas em 1890. De fato, a maioria delas não foi descoberta por Peano, portanto a denominação é utilizada para descrever uma classe de curvas que possuem características comuns ao invés de se referirem ao seu descobridor. Até agora foram analisadas curvas que se caracterizam pela dimensão de similaridade não-inteira, na maioria curvas que possuem dimensões entre 1 e 2. Este fato implica que, independente do número de iterações realizadas, estas curvas nunca preencheriam todo o plano. Em contraposição a isso, todas as curvas de Peano possuem dimensão fractal igual a 2.

De fato, o interesse matemático de Peano ao construir as primeiras curvas dessa família era exatamente esse – curvas contínuas capazes de preencher completamente o plano, ou em outras palavras:

O séc. XIX se iniciou com a descoberta de que curvas e funções não precisam ser do tipo bem comportado que até então dominara o campo, e Peano a partir de 1890 mostrou até que ponto a Matemática podia insultar o senso comum quando construiu curvas contínuas que enchem o espaço – isto é, curvas dadas por equações paramétricas $x = f(t)$, $y = g(t)$, onde f e g são funções reais contínuas no intervalo $0 \leq t \leq 1$, cujos pontos preenchem totalmente o quadrado unitário $0 \leq x \leq 1$, $0 \leq y \leq 1$. Este paradoxo, é claro, combina perfeitamente com a descoberta de Cantor de que não há mais pontos no quadrado unitário que no segmento de reta unitário...

(Boyer, 1974, p. 437)

A curva de Peano original contraria uma das regras para a construção de fractais, fazendo com que sua curva se intercepte em uma infinidade de pontos (Stevens, 1990, p. 149). A curva de Peano parte de um segmento de reta como iniciador⁶ e, devido à quantidade de intersecções é quase impossível descrever como o gerador da curva é desenhado, porém, pode-se notar que este é formado por nove segmentos iguais dispostos em dois quadrados simétricos possuindo dois outros segmentos partindo dos vértices comuns aos dois quadrados.

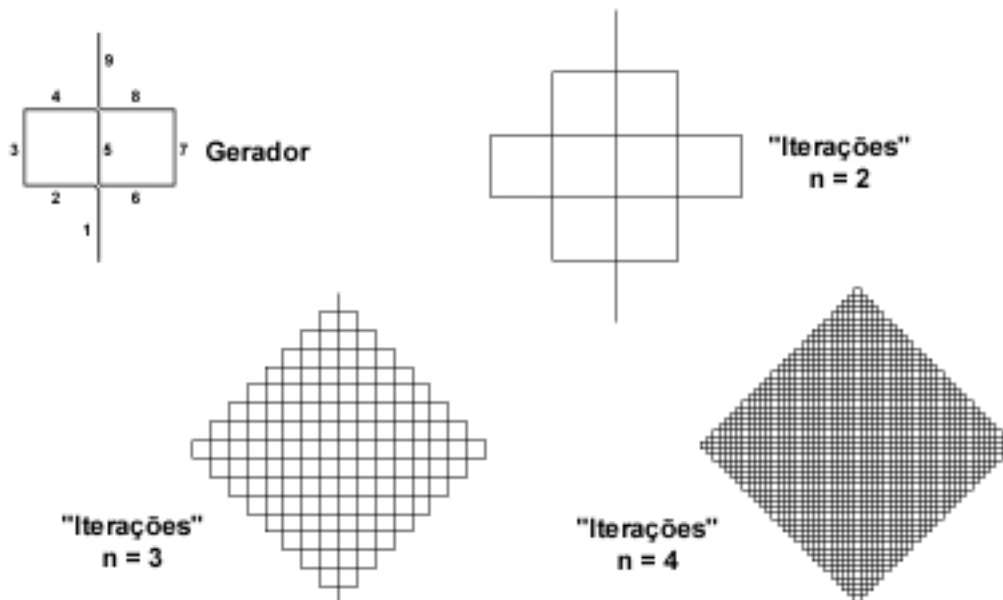


Figura 1.8: Processo “iniciador-gerador” para a construção da Curva de Peano original.

⁶ Ao considerarmos um segmento de reta como iniciador, tanto nesta como nas demais “Curvas de Peano” adotamos um enfoque essencialmente geométrico e mais simples de análise, porém vale ressaltar que originalmente, a construção dessas curvas possuía um enfoque muito mais algébrico e abstrato, considerando na maioria das vezes o iniciador como o intervalo $[0, 1]$ da reta real.

A dimensão de similaridade da Curva de Peano se deve ao gerador, que é formado por nove segmentos de reta ($N = 9$), tendo um terço do comprimento do segmento de reta iniciador ($\varepsilon = 1/3$), logo :

$$D_s = \frac{\log 9}{\log 3} = 2$$

Outra curva desta família é a “Curva de Peano-Gosper”, construída por G. Peano e W. Gosper com o objetivo de sanar a deficiência da curva de Peano original criando uma curva contínua sem nenhum ponto de intersecção. Novamente, o iniciador dessa curva é um segmento de reta, porém seu gerador é um pouco mais complexo que os estudados até aqui, e está associado a uma malha de triângulos equiláteros.

A dimensão de similaridade dessa curva também é baseada no seu gerador, onde temos sete segmentos de reta ($N = 7$), cada um com um comprimento $\varepsilon = \frac{1}{\sqrt{7}}$ em relação ao segmento de reta iniciador, logo:

$$D_s = \frac{\log 7}{\log \sqrt{7}} = 2$$

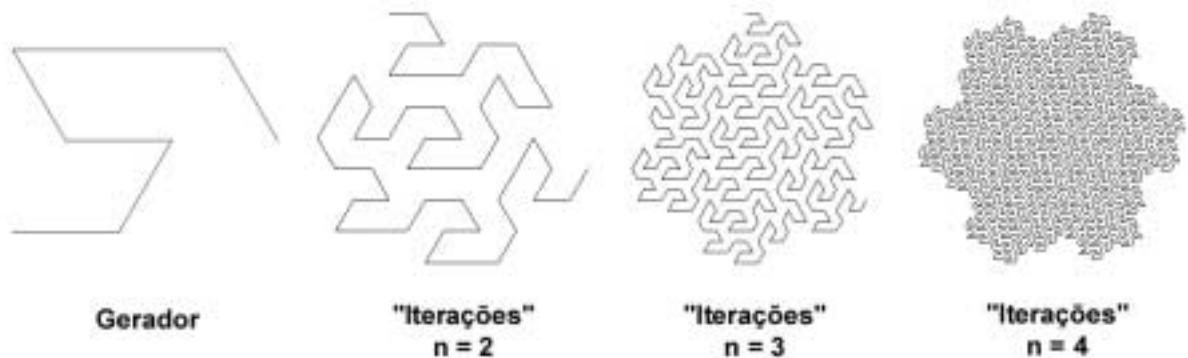


Figura 1.9: Processo “iniciador-gerador” para a construção da Curva de Peano-Gosper.

A última das curvas de Peano que será analisada aqui foi construída por B. Mandelbrot e ficou conhecida como “Floco de Neve com Sete Segmentos de Peano”. Esta curva, assim como as demais curvas de Peano, possui dimensão 2, porém, neste caso o gerador é mais complicado por não ser formado por segmentos com comprimentos iguais o que impede a determinação da dimensão

através da similaridade, sendo necessários outros processos para o cálculo da dimensão fractal da figura.

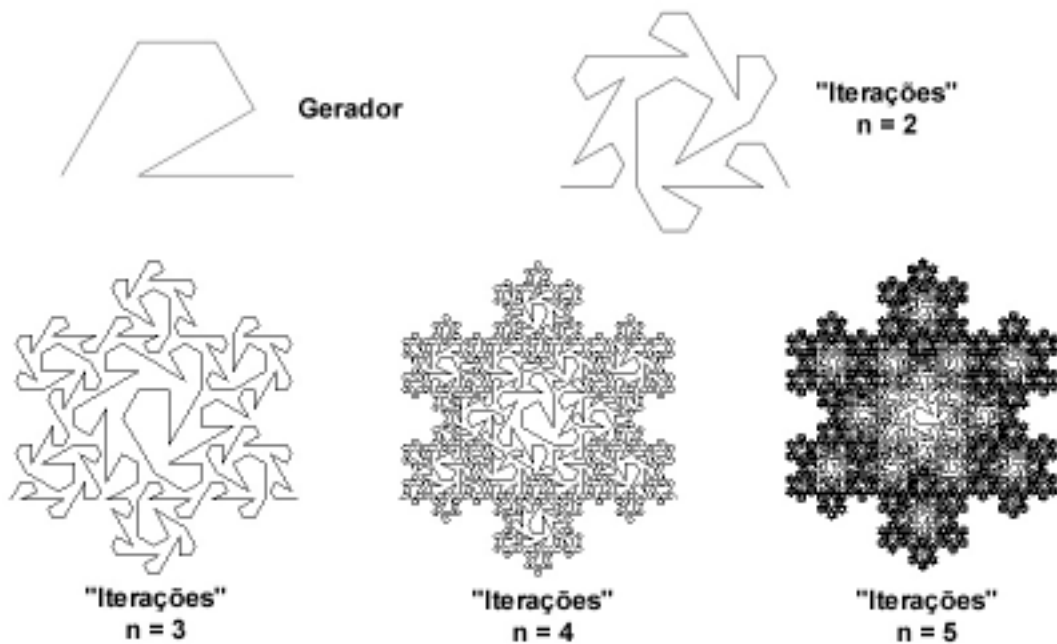


Figura 1.10: Processo “iniciador-gerador” para o Floco de Neve com Sete Segmentos de Peano.

Devido ao fato do gerador desta figura ser mais complicado que os anteriores, existirão quatro possíveis posições para o seu gerador, e estas posições devem ser cuidadosamente escolhidas em cada nível de iteração e em cada segmento a ser substituído para garantir que não ocorram intersecções nem sobreposições na curva. (Stevens, 1990, p.185)

1.2.5 Curva de Hilbert

A curva de Hilbert é mais um exemplo da família das curvas de Peano, porém, ela possui algumas características peculiares que a tornam única. Quando o matemático alemão David Hilbert (1862-1943) a apresentou pela primeira vez, possuía o mesmo objetivo de Peano com a vantagem de ser uma curva de construção mais simples que a de seu colega.

A curva de Hilbert é formada pelo mesmo processo recursivo que foi utilizado até aqui, porém com algumas particularidades. Seu iniciador é um quadrado unitário e o gerador consiste em dividi-lo em quatro quadrados iguais, porém, a curva é formada não pelos quadrados e sim pelos segmentos de reta que se originam da ligação dos pontos centrais de cada um dos quadrados formados nas sucessivas iterações.

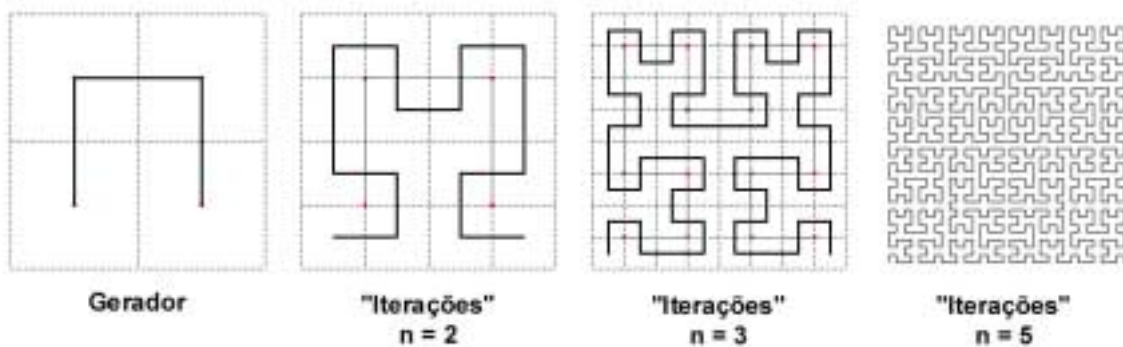


Figura 1.11: Processo “iniciador-gerador” para a construção da Curva de Hilbert.

A dimensão de similaridade da curva de Hilbert fica clara ao considerar que o iniciador é dividido em quatro quadrados iguais ($N = 4$), cada um com metade do lado original ($\varepsilon = 1/2$), então:

$$D_s = \frac{\log 4}{\log 2} = 2$$

A curva de Hilbert, como as demais, tem comprimento infinito limitado a uma área finita, neste caso, a área do quadrado unitário e, apesar de ser topologicamente equivalente a uma linha, é tão dobrada e contorcida que ocupa todo o plano, comportando-se como um objeto bi-dimensional. A curva passa através dos pontos de uma malha quadrada progressivamente mais refinada a cada iteração, porém, sem nunca passa por uma área onde ela já exista, ou seja, a curva não se intercepta em nenhum ponto. “É claro que a curva limite desse processo passará por todos os pontos do quadrado; incidentalmente esse é outro exemplo de curva contínua que não é diferenciável em nenhum ponto” (Boyer, 1974, p. 447).

1.3 Fractais Gerados por Processos Algébricos

Existem duas famílias de fractais geradas por processos algébricos, a primeira delas é gerada a partir de conjuntos de números complexos que são plotados no sistema cartesiano convencional. De forma simples, este processo utiliza a representação gráfica tradicional de números complexos, ou seja, decompondo-os em sua parte real e imaginária, sendo a parte real plotada no eixo das abscissas e a parte imaginária no eixo das ordenadas. Os fractais gerados através desse processo baseiam-se nas técnicas de construção do “Conjunto de Mandelbrot” e dos “Conjuntos de Julia”, idealizados por Benoit Mandelbrot em meados de 1979, porém, o processo não é tão simples como se poderia imaginar e para entendermos como tais figuras são geradas teremos que ir um pouco mais fundo em algumas novas concepções matemáticas acerca da construção de seqüências de números complexos.

A segunda família de fractais que iremos estudar baseia-se em seqüências de números reais determinados por sistemas de funções no \mathbb{R}^2 , que geram pares ordenados também plotados no sistema cartesiano convencional. A técnica que produz esta família de fractais ficou popularmente conhecida por “Jogo do Caos” e foi idealizada por Michael Barnsley.

1.3.1 Enfoque Matemático e Sistemas Dinâmicos Não-Lineares

Antes de uma análise dos fractais algébricos, temos que tecer algumas considerações importantes sobre o enfoque matemático que é dado para a construção desses fractais. De fato, a idéia de “enfoque matemático” é bastante adequada neste caso pois, de fato, a geometria fractal não cria uma nova teoria ou sistema de axiomas matemáticos, ela simplesmente se utiliza de diversas ferramentas clássicas da álgebra nas suas construções, de uma forma tão simples que praticamente poderia ser acessível a um aluno do ensino médio. Porém, o que torna a geometria fractal tão complexa e inovadora é exatamente a

maneira como essas ferramentas simples são utilizadas e é neste novo enfoque que estamos interessados.

Na construção dos fractais algébricos, utilizam-se essencialmente seqüências numéricas, tanto reais como complexas, sistemas de funções e representação gráfica de pontos no sistema cartesiano. Porém, esses elementos são utilizados de forma bem diferente, baseando as construções em novas ferramentas que permitem a utilização de cálculos matemáticos e representações gráficas em ambientes informáticos, e importando da Teoria do Caos concepções acerca do comportamento de sistemas dinâmicos não-lineares.

Quando nos referimos a sistemas dinâmicos não-lineares, somos remetidos imediatamente ao estudo do comportamento de funções matemáticas e seqüências numéricas porém, com uma diferença essencial. Quando lidamos com funções matemáticas tradicionais, nos acostumamos e somos induzidos a esperar um certo grau de previsibilidade nos resultados, ou seja, para quaisquer “leis de formação” de uma função, esperamos sempre um de três resultados possíveis :

- a função tende a um valor fixo;
- a função tende ao infinito ($+\infty$ ou $-\infty$);
- a função assume um ciclo periódico de valores repetitivos.

Ou seja, em todos os casos, esperamos que a função atinja, após certo número de iterações, um estado de equilíbrio estável e perfeitamente previsível. Com isso, enfoque cartesiano tradicional, no qual transformamos funções em curvas, supõe a “resolução” de funções a partir de um conjunto de valores que a satisfaçam, fazendo com que as soluções de uma dada função resultem em uma forma geométrica, descrita estaticamente no plano cartesiano a partir dos pontos que a satisfazem. Assim, uma função de comportamento constante tende sempre a um valor fixo, funções lineares ou polinomiais, via de regra, rumam ao infinito e várias funções trigonométricas e circulares assumem ciclos periódicos de valores repetitivos.

Entende-se por comportamento estático essa “postura” de “resolução” de funções, ou seja, uma função “descreve” um comportamento previsível, com o qual nos acostumamos a lidar, mesmo quando estudamos funções bem mais complicadas. Dessa forma, quando passamos a estudar seqüências numéricas, também nos condicionamos a esperar o mesmo comportamento, ou seja,

seqüências que “convergem” ou “não convergem”, porém são previsíveis nesse comportamento e, para uma mesma “lei de formação”, teremos, em geral, somente um desses dois resultados possíveis.

Porém, quando lidamos com sistemas dinâmicos não-lineares, alguns novos e intrigantes resultados irão vir à tona e abrirão possibilidades para um imenso campo novo de estudos – temos o Caos. De forma muito simples, o estudo de sistemas dinâmicos partiu de uma nova postura na análise de funções matemáticas, as funções não são mais “resolvidas”, mas “repetidas” num *loop* de retroalimentação. Assim, a função “se torna um processo em lugar de uma descrição, dinâmica em lugar de estática” (Gleick, 1990, p. 219), onde cada número analisado gera um novo número que será novamente analisado, gerando um novo número, e assim sucessivamente. As formas geométricas resultantes de uma função são definidas plotando-se pontos que produzem um determinado comportamento e não mais quando “satisfazem” a função. Assim, um comportamento pode ser um regime constante, ou uma convergência para uma repetição periódica de valores, ou uma tendência para o infinito (*ibid.*)

O que se observou com essa nova postura de análise criada com os sistemas dinâmicos é que, além dos comportamentos que estamos habituados a lidar, surgem também comportamentos caóticos, ou seja, as seqüências fornecem resultados novos e imprevisíveis, sem se repetir e sem nunca atingir um estado de equilíbrio, vagando caoticamente em uma órbita de valores que nunca se repetem nem tendem ao infinito. Este novo regime de comportamento foi a principal mola propulsora para a construção da Teoria do Caos e, segundo seus idealizadores, esse comportamento dinâmico e imprevisível é muito mais “regra” do que exceção na Matemática, aparecendo em funções definidas por leis extremamente simples. Além disso, como veremos a seguir, uma mesma função pode conter regimes de comportamento completamente diferentes quando “alimentada” com números distintos.

1.4 O Conjunto de Mandelbrot (Mandelbrot Set)

Basicamente, o conjunto de Mandelbrot é uma coleção de números complexos definidos a partir de uma função dinâmica, analisada com o enfoque descrito acima. Esta função parte de uma expressão quadrática bastante simples que define uma “lei de formação” para uma seqüência de números complexos. Esta lei, bastante conhecida para aqueles que se interessam por fractais, é assim definida:

- $z_{n+1} = z_n^2 + c$; onde z_0 e c são números complexos;
- ou na forma $f: C \rightarrow C$ definida por $f(z) = z^2 + c$.

Visto que a construção do Conjunto de Mandelbrot baseia-se na utilização de seqüências de números complexos, consideramos importante destacar algumas características deste conjunto numérico. Como sabemos, os números complexos são, por definição, representados na forma $z = a + bi$, onde a e b são números reais e $i = \sqrt{-1}$; decorre desse fato que convencionou-se decompor os números complexos em uma parte real e uma parte imaginária. Com essa decomposição, convencionou-se também que a representação gráfica de um número complexo seria feita como um ponto num plano cartesiano – o Plano de Argand – cujas coordenadas teriam a parte real como abscissa e a parte imaginária como ordenada. Apesar de plenamente conhecidas, essas características devem ser tidas em mente para compreendermos a estrutura de construção do conjunto de Mandelbrot.

Uma vez que o Conjunto de Mandelbrot baseia-se numa função dinâmica não linear, a expressão que a define representa um teste para identificarmos se um determinado número complexo pertence ou não ao conjunto, ou seja, se o número complexo testado produzir um determinado regime de comportamento, ele pertence ao conjunto, caso contrário, não pertence.

Dessa forma, o número complexo a ser testado assume o valor “ c ” na função dada acima e a primeira iteração da seqüência é feita com $z_1 = 0 + c$, em seguida, toma-se este resultado (z_1), eleva-se ao quadrado e acrescenta-se o

número original “c”, portanto: $z_2 = c^2 + c$. Feito isso, toma-se novamente o resultado anterior (z_2), eleva-se ao quadrado e acrescenta-se o número original “c”, portanto: $z_3 = (c^2 + c)^2 + c$ – e assim sucessivamente. O regime de comportamento desejado é bastante simples e consiste em: se a seqüência assumir valores finitos no decorrer das sucessivas iterações (quando o número se fixar num loop repetitivo, manter-se constante ou vagar caoticamente), teremos um número que pertence ao conjunto. Caso contrário, se o número escolhido projetar-se para o infinito no decorrer das iterações, ele não pertence ao conjunto.

Porém, o comportamento dos números complexos submetidos a este sistema dinâmico não é tão simples, uma vez que não há forma de se prever quais números irão gerar uma seqüência que tende ao infinito e quais geram seqüências que se mantêm em repetições finitas. Dessa forma, em termos práticos, pode-se estabelecer que todos os números complexos que pertencem ao conjunto de Mandelbrot irão gerar uma seqüência de valores que se mantêm próxima à origem do plano de Argand. Assim, o critério usual para a determinação desses números estabelece que todos os valores da seqüência gerada por um determinado número “c” permaneçam delimitados por um círculo fixo de centro na origem do plano e raio 2, ou seja, se um número complexo qualquer gerar uma seqüência que possua algum termo que exceda essa delimitação, pode-se afirmar que esta seqüência certamente rumará para o infinito. É claro que este critério é principalmente utilizado para determinar quais valores *não* pertencem ao conjunto, uma vez que nem todos os números complexos situados na região interna ao círculo pertencem ao conjunto de Mandelbrot. Nesse caso, restam somente a expressão e o processo de tentativa e erro para a efetiva determinação do conjunto, realizada a partir da limitação do número de iterações a ser executada. Esse processo evidentemente nos remete para uso de computadores no que essas máquinas fazem melhor, cálculos simples e repetitivos.

Por outro lado, o programa para gerar o conjunto de Mandelbrot é, do ponto de vista técnico, bastante simples e se resume em testar uma grade de valores com uma expressão quadrática que se repete continuamente para cada um de milhares de pontos. Se os resultados escaparem para o infinito, plota-se o ponto testado de uma cor, branco por exemplo, e se os resultados se mantiverem finitos, plota-se o ponto de outra cor, por exemplo, preto. Assim, em termos informáticos, a grade de números complexos determinada para a representação

gráfica do Conjunto de Mandelbrot é convertida num conjunto de “*pixels*”⁷ cujas coordenadas são testadas pela expressão e os pontos testados são plotados de uma determinada cor.

Porém, é obvio que o número de iterações que o programa faz para tomar essa decisão não pode ser infinito, então, cria uma nova variável para o estudo do Conjunto de Mandelbrot – a escala de ampliação desejada para a apresentação de partes do conjunto – pois, dada a enorme complexidade das formas geradas na região limítrofe entre os pontos que pertencem e os que não pertencem ao conjunto, uma “viagem” pelas diferentes escalas de ampliação desta região é ainda mais fascinante que a construção do Conjunto de Mandelbrot em si. Assim, para escalas de ampliação entre 1 e 20 vezes, algo entre 100 e 500 iterações por ponto é, com frequência, bastante, já para escalas de ampliação maiores, que podem ir além de 10^{10} vezes, chega-se a utilizar até 100 000 iterações por ponto.

Além disso, a forma como a figura é colorida no processo depende do gosto do programador e os softwares mais atuais são elaborados de forma a substituir os pontos brancos (fora do conjunto) por gradações de cores para uma imagem mais atraente. Tais gradações são definidas em função do número de iterações necessárias para cada ponto escapar para o infinito. Com isso teremos uma visão dos contornos da região imediatamente externa ao conjunto em si, representando a “velocidade” com que os números complexos testados escapam para o infinito.

⁷ *Pixel* é a abreviatura de “picture element” e corresponde a cada uma das minúsculas subdivisões que compõe a “tela” de um computador. E assim, visto que os *pixels* são definidos a partir de um sistema de coordenadas cartesianas do “plano” da tela, um programa que visa a tradução da construção do Conjunto de Mandelbrot para a interface gráfica de um computador é obtida de forma relativamente simples.

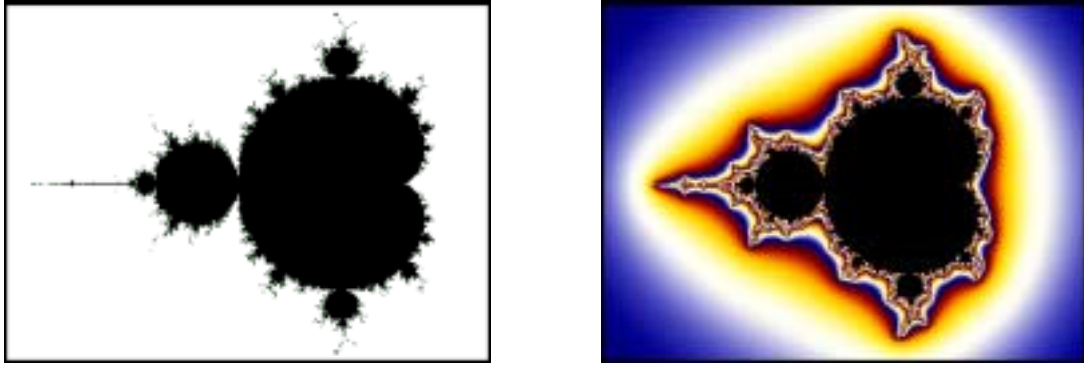


Figura 1.12: Representações gráficas do Conjunto de Mandelbrot.

Em resumo, o Conjunto de Mandelbrot parte de alguns novos e interessantes enfoques matemáticos para sua construção. O objetivo aqui não é o de resolver uma equação a partir de um valor dado e sim testar o comportamento desse valor em repetições sucessivas. A “lei de formação” do conjunto não irá gerar o conjunto em si, mas serve como teste de valores que farão ou não parte do conjunto. Outro fato intrigante surge ao se notar que as seqüências geradas por cada ponto são interessantes em si mesmas e nos dão uma idéia da complexidade do conjunto como um todo uma vez que cada novo ponto testado nos dá uma nova resposta, com uma nova configuração que nos leva a um novo ponto que pode ter uma configuração semelhante a anterior ou outra completamente nova, pode escapar para o infinito, se fixar em um loop repetitivo ou, finalmente, vagar caoticamente. O mais interessante é que, mesmo depois de adquirir certa experiência na manipulação desses valores, nunca se pode prever com certeza qual será a posição do próximo ponto.

1.5 Jogo do Caos

A última forma de construção figuras fractais aqui analisada é o processo conhecido como “Jogo do Caos”, criado por Michael Barnsley no Georgia Institute of Technology em meados da década de 1980. Formalmente, o jogo do caos é descrito como “Construção Global de Fractais por meio de Sistemas de Funções Iteradas⁸”. Essa técnica leva o nome de “Jogo do Caos” devido ao componente aleatório do processo, pois, como num jogo de dados, cada função do sistema é aplicada aleatoriamente pelo programa.

Em termos simples, a técnica idealizada por Barnsley utiliza-se da aleatoriedade para criar modelos de formas naturais, como nuvens, florestas, galáxias, samambaias, flores, etc. Ela se resume na definição de um sistema de funções matemáticas que, essencialmente, nos dizem como “levar” um ponto em outro em um dado sistema de eixos ortogonais. A partir dessas regras, escolhe-se um ponto de partida qualquer nesse sistema e um gerador de números aleatórios (o “cara e coroa” de uma moeda seria suficiente, por exemplo). Feito isso, começamos a iteração do sistema, ou seja, a “jogar a moeda para o alto” e a marcar os pontos resultantes das funções sorteadas. O fascinante nesse processo é que, “se abandonarmos os primeiros pontos, veremos que o jogo do caos produz não um campo aleatório de pontos, mas uma forma, que se torna cada vez mais nítida à medida que o jogo prossegue” (Gleick, 1990, pp. 228-229).

O fascínio despertado pelo processo idealizado por Barnsley se resume no fato desconcertante de um programa simples, munido de três ou quatro funções reais, cada uma delas realizando algumas transformações geométricas, ser capaz de gerar figuras com enorme riqueza de detalhes a partir das coordenadas de um único ponto inicial. Além disso, quanto maior for a ordem fractal oculta no objeto que se deseja criar, mais simples serão as funções para decodificá-lo e, neste particular, Mandelbrot fez a observação fundamental de que grande parte da natureza tem essa ordem oculta.

Cabe ressaltar que a aleatoriedade serve apenas como uma ferramenta na técnica de Barnsley. Os resultados são deterministas e previsíveis, ou seja, figuras idênticas são construídas ao se iterar um mesmo sistema de funções. À

⁸ É freqüente nas bibliografias o uso da sigla IFS, da notação inglesa para “Iterated Function Systems”.

medida que os pontos surgem na tela do computador, ninguém consegue saber onde o próximo ponto aparecerá: isso depende do resultado da “moeda” interna do programa, o que torna o resultado “local” da construção aleatório, ou seja, ao se analisar partes específicas da seqüência de pontos gerados por um mesmo sistema, (seqüências com algumas dezenas de pontos, por exemplo) serão encontrados resultados completamente distintos. Não obstante, o resultado “global” da iteração, ou seja, conjuntos formados com algo da ordem de milhares de pontos, resultará invariavelmente na mesma imagem. “Sob esse aspecto, o papel do acaso é uma ilusão. ‘A aleatoriedade é uma pista falsa’ disse Barnsley. Com a probabilidade de um, conseguiremos sempre a mesma imagem” (Gleick, 1990, p. 231).

1.5.1 Sistemas de Funções Iteradas e Transformações Afins

A partir dessa introdução ao processo de construção de fractais com o jogo do caos, cabe-nos a exploração de como os sistemas de funções iteradas são definidos matematicamente. Nesse sentido, a obra “Fractals Everywhere” (Barnsley, 1993) foi sem dúvida a que mais rigorosa e detalhadamente abordou o tema em termos matemáticos. Os fractais descritos por ele são definidos como “subconjuntos de Espaços Métricos”, situando grande parte de seus estudos na Topologia. Segundo Barnsley (1993, pp. 29-35) os fractais podem ser encontrados num espaço métrico, cujos elementos são subconjuntos compactos não-vazios desse espaço que, sob certas condições, geram um espaço métrico completo no qual as seqüências de Cauchy convergem⁹.

Dessa forma, os fractais são definidos, representados e analisados com o auxílio de transformações elementares de um espaço nele mesmo, em particular, utilizando transformações afins ou transformações de Möbius. Assim, para se trabalhar com estes fractais, é indispensável certa familiaridade com algumas

⁹ Segundo Lima (1977, pp. 161-169), um espaço métrico M é *completo* quando toda seqüência de Cauchy em M é convergente.

Por sua vez, uma seqüência (x_n) num espaço métrico M chama-se uma seqüência de Cauchy quando, para todo $\varepsilon > 0$ dado, existe $n_0 \in \mathbb{N}$ tal que $m, n > n_0 \Rightarrow d(x_m, x_n) < \varepsilon$.

Além disso, um tratamento mais formal sobre a Geometria Fractal, enfocando Espaços Métricos e outros tópicos relacionados à Topologia, pode ser encontrado Nagata, J-I (1968, pp. 115-131).

famílias básicas de transformações em \mathbb{R} (reta real), \mathbb{R}^2 (plano euclidiano), \mathbb{C} (plano complexo) ou $\hat{\mathbb{C}}$ (esfera de Riemann).

Também é necessário conhecer bem a relação entre as “fórmulas” de uma dada transformação e seu significado geométrico, isto é, para o jogo do caos será mais importante entender o que as transformações fazem com uma dada figura do que como elas agem em pontos específicos, ou seja, é mais útil entender como uma transformação no \mathbb{R}^2 age sobre uma reta, circunferência ou triângulo, do que saber como ela transforma um ponto específico, por exemplo. Isso se justifica pelo fato de um conjunto fractal conter uma infinidade de pontos cuja organização é tão complicada que não seria possível descrever o conjunto a partir da posição específica de cada ponto. Ao contrário, o conjunto que determina o fractal pode ser melhor definido pelas relações entre as partes que o constituem.

Por motivos óbvios, não iremos abordar todas as possibilidades de construção dos fractais do Jogo do Caos, porém, iremos nos aprofundar nos processos básicos dessa construção limitando-a a um espaço específico e mais acessível a nossos propósitos. Assim sendo, estaremos interessados em definir um conjunto de elementos de transformações geométricas – as transformações afins no plano euclidiano – que fornecerão um repertório básico de ferramentas para a construção de um conjunto específico de fractais. Além disso, iremos focar nossas definições em transformações realizadas no \mathbb{R}^2 , apesar destas definições também serem válidas para outros espaços.

Visto que o objetivo desta seção é o de fornecer um conjunto específico de ferramentas para a construção de fractais, não estaremos interessados em definir as transformações geométricas completamente. Da mesma forma, não iremos nos interessar na reprodução das demonstrações formais dos conceitos que iremos introduzir pela simples razão de que tais demonstrações podem ser facilmente obtidas nas obras que nos serviram de referência, bastando-nos, portanto, apenas enunciá-los.

Segundo Barnsley (1993, pp. 50-55), a expressão geral para cada transformação afim de um sistema de funções iteradas será a transformação $w: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, na forma:

$$w(x_n, y_n) = (ax + by + e, cx + dy + f),$$

onde os parâmetros a, \dots, f são números reais. Serão utilizadas também as seguintes notações equivalentes:

$$w(x) = w \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = Ax + t$$

Onde “A” é uma matriz real de ordem 2 e “t” é um vetor coluna que não será distinguido do par ordenado $(e, f) \in \mathbb{R}^2$. A transformação afim geral $w(x) = Ax + t$ no \mathbb{R}^2 consiste numa composta de transformações lineares, A, a qual deforma o espaço relativamente à origem, seguida de uma translação definida pelo vetor t. Dessa forma, cada função do sistema executa, a partir dos parâmetros reais a, \dots, f , uma transformação afim “w” composta de homotetias, rotações, reflexões e translações nas coordenadas (x, y) de um vetor “x”, gerando um novo vetor em uma posição representada por (x_n, y_n) .

Para o sistema funcionar corretamente, a homotetia deve resultar na redução das distâncias entre os pontos, caso contrário, as iterações irão resultar em pontos extrapolando ao infinito. Caso a transformação w seja aplicada a uma figura geométrica, esta poderá ser, “encolhida”, rotacionada, refletida e transladada para uma nova posição.

Além disso, a matriz A também poderá ser sempre escrita na forma:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} r_1 \cos \hat{e}_1 & -r_2 \sin \hat{e}_2 \\ r_1 \sin \hat{e}_1 & r_2 \cos \hat{e}_2 \end{pmatrix}$$

onde (r_1, θ_1) são as coordenadas polares do ponto (a, c) e $(r_2, (\theta_2 + \pi/2))$ são as coordenadas polares do ponto (b, d), onde já é possível notarmos o papel da composta de diferentes transformações geométricas na obtenção dos parâmetros

que irão definir os sistemas de funções iteradas, necessários para a construção de fractais a partir do jogo do caos.

1.5.2 Algoritmos Determinísticos

Dentro das técnicas de construção de fractais desenvolvidas por Barnsley em seus trabalhos, existe uma versão determinística do jogo do caos que produz algumas figuras em particular quando, como em um jogo com dados “viciados”, são introduzidas diferentes probabilidades para cada função (Stevens ,1990, pp. 421-423). Como dissemos, estas figuras são formadas essencialmente por um conjunto de pontos, portanto, para uma construção bem detalhada, serão necessários entre 10.000 e 100.000 pontos. Para uma idéia mais clara dessa “família” de fractais, iremos apresentar alguns deles a seguir.

1.5.2.1 Samambaia determinística (*deterministic fern*)

Este fractal, desenvolvido por Barnsley, tornou-se praticamente o maior e mais famoso representante do Jogo do Caos, pois a samambaia determinística reflete maravilhosamente um dos objetivos de Barnsley ao apresentar sua técnica, ou seja, criar modelos matemáticos que representassem objetos naturais. A samambaia determinística, que é definida por um sistema simples com quatro funções iteradas, corresponde perfeitamente à imagem dessa espécie de planta. O próprio Barnsley afirmou: “Era uma imagem espantosa, correta sob todos os aspectos. Nenhum biólogo teria problemas em identificá-la” (Gleick, 1990, p. 231).

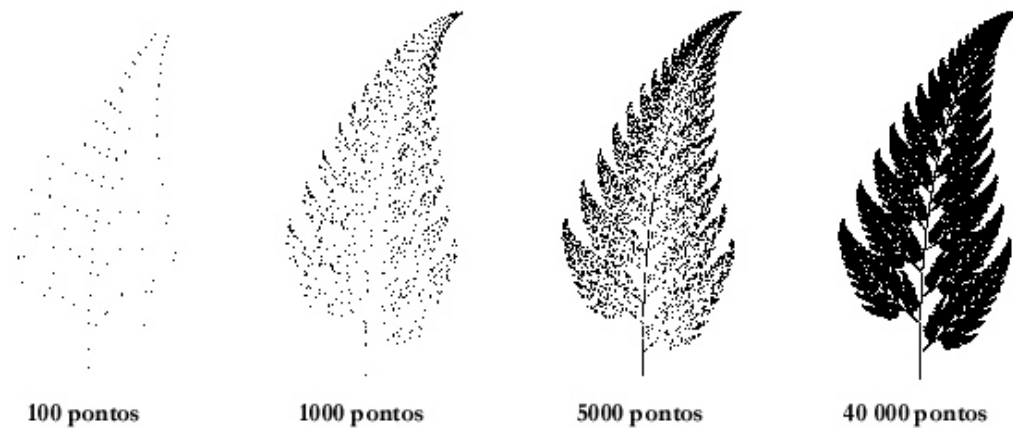


Figura 1.13: Processo de construção para a samambaia determinística.

A figura 1.13 apresenta o processo de construção da samambaia determinística com maiores detalhes. As imagens foram todas concebidas a partir dos mesmos parâmetros, onde somente variamos o número de iterações e, conseqüentemente, o número de pontos produzido pelo programa. Nela, pode-se perceber mais claramente como o fractal construído vai lentamente tomando forma, na medida em que novos pontos são gerados pelo programa.

Os parâmetros para a obtenção das quatro funções iteradas do sistema e as probabilidades para a construção da samambaia são dados a seguir (Barnsley, 1993, p. 87).

Parâmetros para a “Samambaia Determinística”.

w	a	b	c	d	e	f	Prob.
1	0	0	0	0,16	0	0	0,01
2	0,2	-0,26	0,23	0,22	0	0,16	0,07
3	-0,15	0,28	0,26	0,24	0	0,16	0,07
4	0,85	0,04	-0,04	0,85	0	0,44	0,85

A partir desses parâmetros, é possível determinar a forma matricial das quatro transformações afins que definem o sistema de funções iteradas para a construção, como segue:

$$w_1 \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0,16 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$w_2 \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} 0,2 & -0,26 \\ 0,23 & 0,22 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0,16 \end{pmatrix}$$

$$w_3 \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} -0,15 & 0,28 \\ 0,26 & 0,24 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0,16 \end{pmatrix}$$

$$w_4 \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} 0,85 & 0,04 \\ -0,04 & 0,85 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0,44 \end{pmatrix}$$

Finalmente, para determinarmos o sistema de funções iteradas para este fractal, basta traduzir a representação matricial das transformações afins para a representação algébrica tradicional. Com isso, determina-se um sistema com quatro pares de funções, como apresentado abaixo:

$$\begin{array}{ll} \text{(i)} & x_n = 0 \quad ; \quad y_n = 0,16y \\ \text{(ii)} & x_n = 0,2x - 0,26y \quad ; \quad y_n = 0,23x + 0,22y + 0,16 \\ \text{(iii)} & x_n = -0,15x + 0,28y \quad ; \quad y_n = 0,26x + 0,24y + 0,16 \\ \text{(iv)} & x_n = 0,85x + 0,04y \quad ; \quad y_n = -0,04x + 0,85y + 0,44 \end{array}$$

Com base nesse sistema, é possível gerar a figura de um galho de samambaia se adicionarmos o sistema de funções acima e a probabilidade com que cada função deve ser aplicada a um programa de computador com interface gráfica. Dessa forma, a função (i) deve ter 1% de probabilidade de ser aplicada; as funções (ii) e (iii), probabilidade de 7% cada uma e a função (iv) com 85% de probabilidade.

1.5.2.2 Outros exemplos de fractais

Descreveremos a seguir os parâmetros para a construção de outros quatro fractais que serão importantes na seqüência deste trabalho. Estes fractais, ao contrário da samambaia determinística, não representam formas da natureza, mas são traduções, para o jogo do caos, de fractais “iniciador-gerador” clássicos. Tais traduções fornecerão uma representação algébrica para os fractais completamente diferente da apresentada no início deste trabalho. Além disso,

estes fractais são equiprováveis, ou seja, cada função do sistema possui a mesma probabilidade de ocorrer.

O primeiro fractal que apresentamos, o “Triângulo de Sierpinski”, teve os parâmetros definidos por Barnsley (1993, p. 85), a “Cauda do Dragão” foi apresentada em Darst, Palagallo e Price (1998, p. 13) e os demais, o “Piso Fractal” e a “Curva de Koch”, tiveram seus parâmetros determinados por nós no decorrer desta pesquisa.

Parâmetros para o “Triângulo de Sierpinski”.

w	a	b	c	d	e	f
1	0,5	0	0	0,5	0	0
2	0,5	0	0	0,5	0,5	0,5
3	0,5	0	0	0,5	1	0

Parâmetros para a “Cauda do Dragão”.

w	a	b	c	d	e	f
1	0,5	-0,5	0,5	0,5	1	0
2	0,5	-0,5	0,5	0,5	-1	0

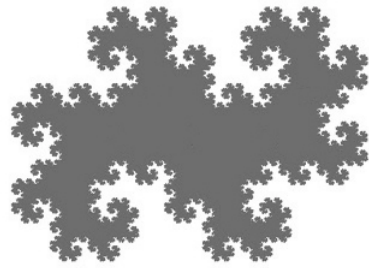
Parâmetros para o “Piso Fractal”.

w	a	b	c	d	e	f
1	0,5	-0,5	0,5	0,5	-1	0
2	-0,5	0,5	0,5	0,5	1	0

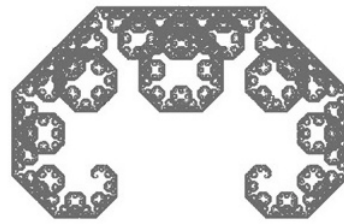
Parâmetros para a “Curva de Koch”.

w	a	b	c	d	e	f
1	0,333	0	0,333	0	0	0
2	0,167	-0,289	0,289	0,167	0,5	0
3	-0,167	0,289	0,289	0,167	1	0
4	0,333	0	0,333	0	1	0

Visto que já descrevemos detalhadamente o processo de determinação dos sistemas de funções iteradas a partir da definição dos parâmetros para a samambaia, nos restringimos a fornecer somente as tabelas contendo os parâmetros de construção dos fractais desejados, deixando para capítulos posteriores um estudo mais aprofundado de cada uma de suas construções.



Cauda do Dragão



Piso Fractal

Figura 1.14: Representação de fractais gerados pelo Jogo do Caos.

A figura 1.14 apresenta somente dois dos fractais aqui definidos uma vez que o triângulo de Sierpinski e a curva de Koch terão a mesma aparência que suas respectivas “versões” iniciador-gerador, já apresentadas anteriormente nesse trabalho.

1.6 Considerações Finais

Ao encerrarmos este primeiro capítulo, voltado à apresentação da Geometria Fractal enquanto objeto matemático, julgamos relevante tecer algumas considerações preliminares acerca do enfoque nele adotado. Em primeiro lugar, a ordem escolhida para a descrição das “famílias” de fractais seguiu aproximadamente a cronologia histórica do desenvolvimento da geometria fractal, ou seja, iniciamos o trabalho com os fractais “iniciador-gerador”, que em sua maioria foram criados entre 1890 e 1920, seguido pelo Conjunto de Mandelbrot, desenvolvido a partir da década de 1970 (apesar de ter sido principalmente “inspirado” nos trabalhos de Fatou e Julia, de meados de 1920), para, finalmente, concluirmos nossa trajetória com o Jogo do Caos, desenvolvido no início da década de 1980.

Além disso, uma característica importantíssima para a compreensão da Geometria Fractal é que, de fato, os diferentes tipos de fractais aqui analisados, assim como outras famílias que não foram aqui incluídas, têm muito pouco em comum além do fato de serem fractais. Como dissemos no início deste capítulo, os critérios utilizados para considerar uma determinada classe de figuras como objetos fractais, ou se referem à características muito subjetivas – como o critério da “auto-semelhança” – ou extremamente abstratas, como a determinação de uma “dimensão fractal” a qual, em geral, se refere a conceitos bastante complexos da Topologia.

Em outras palavras, não se identifica imediatamente uma conexão ou relação pré-estabelecida capaz de englobar todas as diferentes famílias de fractais, ou seja, mesmo que duas construções resultem em figuras com a mesma aparência, não se pode afirmar que ambas se referem a uma mesma família de fractais. Para melhor explicitar esta importante característica, pode-se traçar uma analogia com a construção de cônicas a partir de secções planas de cones, caracterizadas por um enfoque essencialmente geométrico; e sua determinação enquanto curvas expressas pelo gráfico de funções analíticas, dadas pela geometria algébrica. Em ambos os casos, os resultados gráficos são idênticos, porém os domínios matemáticos que ambas as construções suscitam é completamente distinto, ou seja, apesar de tratarem-se somente de diferentes

representações de um mesmo objeto, seu tratamento em termos matemáticos, e principalmente didáticos, acaba por tratar estas duas construções como objetos praticamente distintos.

Nessa perspectiva, uma questão fundamental pode ser levantada: como, afinal, podemos definir um fractal? A chave para a compreensão dessa questão está em um conceito bem mais objetivo que a idéia de auto-semelhança e muito menos abstrato que as definições topológicas de dimensão – sua construção.

O fato é que, ao analisarmos os processos de construção dos fractais, podemos perceber que todos, sem exceção, se baseiam em algum tipo de processo iterativo. Assim, apesar da construção de fractais utilizar-se de objetos matemáticos completamente diferentes, como pudemos constatar pelo apresentado até aqui, tais construções invariavelmente se baseiam em alguma forma de iteração.

Com isso, pode-se afirmar que este enfoque presente na construção de fractais é, provavelmente, o principal responsável pela grande relação de afinidade entre a Geometria Fractal e as tecnologias informáticas. Esta relação é tão forte que nos arriscamos a dizer que o próprio desenvolvimento da Geometria Fractal só foi possível graças ao advento dos computadores. Claro que a construção de fractais sem esse recurso é até possível, não nos atreveríamos a dizer o contrário, porém, sob inúmeros aspectos ela seria quase impraticável.

Finalmente, cabe-nos uma última conexão entre a construção de objetos fractais e o contexto da Educação Matemática, no qual se insere este trabalho. Assim sendo, dada a relação entre a construção de fractais e as novas tecnologias, nos interessaremos especificamente por ambientes computacionais voltados à Educação Matemática que potencializem a integração didática dos objetos fractais, em particular, aqueles que permitam a representação destes objetos via diferentes construções.

Dessa forma, visando um estudo aprofundado das potencialidades das ferramentas de alguns desses ambientes informáticos, tanto no que se refere a uma eventual transposição de objetos fractais, quanto à sua efetiva utilização educacional, é que apresentamos, no próximo capítulo, um conjunto de referenciais teóricos e de ambientes educacionais que julgamos pertinentes e adaptáveis a estes propósitos.

CAPÍTULO 2

CONCEPÇÕES TEÓRICAS E FERRAMENTAS INFORMÁTICAS

2.1 Considerações Iniciais

Este capítulo dedica-se à definição e explicitação dos objetivos e problemas de pesquisa desse trabalho, de sua fundamentação teórica e das ferramentas informáticas nele utilizadas. Assim sendo, após a apresentação dos objetivos, introduzimos as principais idéias teóricas que embasam nosso estudo e o enfoque que pretendemos desenvolver, seguido de um conjunto de análises acerca dos ambientes informáticos de aprendizagem que utilizaremos no decorrer deste trabalho.

Cabe ressaltar que, dentro dos quadros teóricos existentes na Educação Matemática, fundamentamos nossas análises essencialmente em ferramentas conceituais da Didática Francesa, especificamente na noção de Transposição Informática, introduzida por Nicolas Balacheff (1994, 1998), com ênfase, em particular, à delimitação do “*domínio de validade epistemológica*” de ambientes informáticos. Além disso, utilizamos outras idéias que ampliam esta noção ou são voltadas para a problemática do ensino e aprendizagem de Matemática com ambientes informáticos, assim, visto que esse contexto cria a necessidade de uma maior abrangência em outros trabalhos relativos à Informática na Educação, nos atemos também à noção de “*micro-mundo*” e em concepções relativas à elaboração e funcionamento dos softwares educativos que serão utilizados neste trabalho.

2.2 Objetivos e Problemas de Pesquisa

Este trabalho apóia-se em três temáticas principais, como base de sustentação conceitual: a apresentação da Geometria Fractal, enquanto objeto do saber matemático; a análise de ambientes informáticos de aprendizagem humana, enquanto meios de representação desses objetos matemáticos; e a utilização didática desses ambientes, visando a identificação de situações de ensino que explorem a construção de fractais.

No sentido de aprofundar e relacionar tais temáticas, formulamos as seguintes questões de pesquisa:

(i) Que transformações resultam do esforço de representação decorrente da passagem dos modelos matemáticos, utilizados na construção de objetos fractais, para modelos computáveis, utilizados em ambientes informáticos de aprendizagem? Quais as conseqüências dessas transformações numa perspectiva didática?

(ii) Que contribuições podem advir da utilização dos processos de construção de fractais, assim como dos conceitos matemáticos a eles relacionados, no sentido da contextualização do ensino e aprendizagem de determinadas noções matemáticas?

Assim, em termos objetivos, o trabalho aborda, num primeiro momento, a análise da dimensão epistemológica de um conjunto ambientes informáticos de aprendizagem humana, no que se refere à construção de objetos fractais. Para tanto, enfocamos noções voltadas à transposição informática da Geometria Fractal. E, num segundo momento, concentramo-nos no desenvolvimento e análise de uma proposta voltada à criação de situações de ensino, buscando eventuais relações entre a construção de fractais em diferentes ambientes informáticos e os conceitos matemáticos envolvidos. Em síntese, o objetivo principal deste trabalho consiste em contribuir para a delimitação do “*domínio de validade epistemológica*” de quatro ambientes informáticos, no que se refere à construção de diferentes “famílias” de objetos fractais.

A escolha dos ambientes informáticos utilizados no trabalho foi condicionada, principalmente, pelo fato de serem ambientes educativos, nos quais tínhamos uma prévia experiência de utilização e que permitem a construção de algum tipo de fractal. Além disso, optamos por estudar ambientes que utilizam a plataforma Windows e que se encaixam na noção de “*micro-mundos*” de Geometria, em particular, no que se refere às concepções de “Geometria da Tartaruga” (*Turtle Geometry*), baseada na linguagem de programação LOGO, e “Geometria Dinâmica”. Com base nessas considerações, elegemos os seguintes ambientes para serem analisados:

- *MicroWorlds LOGO 2.05* (Papert & Silverman, 2000);
- *Cabri-Gèomètre II* (Laborde & Bellemain, 1994);
- *Geometer’s Sketchpad 3.0* (Jackiw & Klotz, 1997);
- *GeomeTricks 2.37* (Sadolin, 1998a).

A partir da apresentação das questões de pesquisa, dos objetivos do estudo e da escolha dos ambientes a serem analisados, dedicamos a próxima seção ao detalhamento das ferramentas conceituais que embasam nossas análises.

2.3 Pressupostos Conceituais da Fundamentação Teórica

Os pressupostos conceituais utilizados por Balacheff (1998)¹ na análise de um dado “*ambiente informático de aprendizagem humana*”², e que tomaremos como nossos, são baseados em três idéias fundamentais, apresentadas a seguir.

A primeira idéia considera a aprendizagem como um processo de adaptação, a qual implica um “*sujeito*” e um “*meio*”. Dessa forma, Balacheff (1998) se interessa pelas condições de evolução das interações do sujeito com o meio, ou seja, nas características presentes na interação entre o aluno, sujeito humano, e a máquina, meio artificial, onde são analisadas as ações do aluno sobre a máquina e as retroações da máquina sobre o aluno. Nessa perspectiva, o conhecimento é tido como um estado de “*equilíbrio dinâmico*” na interação entre o sujeito e o meio, ou seja, considera-se que existe conhecimento se o sistema for capaz de reencontrar um novo estado de equilíbrio quando as condições da interação sujeito/meio são desequilibradas.

Desta primeira idéia derivam dois “*postulados*”. O primeiro afirma que as propriedades dos conhecimentos construídos pelo aluno na interação sujeito/meio dependem de seus conhecimentos anteriores e o segundo afirma que estas interações dependem das características do meio com o qual o aluno interage. Nota-se, nesta primeira idéia, alguns dos referenciais teóricos da Didática Francesa utilizados por Balacheff (1998), em particular com alusões à Teoria das Situações e “*Milieu*” (Brousseau, 1986; 1988), referenciais estes que também estão implicitamente presentes em nossas análises.

A segunda idéia visa delimitar a abrangência da interação sujeito/meio aos ambientes informáticos de aprendizagem humana. Com isso, Balacheff (1998) caracteriza a problemática de se determinar os procedimentos, os comportamentos e os controles sobre o funcionamento do sistema, no que diz respeito à interação entre esses dois agentes. Assim sendo, esta idéia refere-se à

¹ Estas considerações são baseadas na transcrição de uma conferência realizada por Balacheff em 1998, disponibilizada on-line em < <http://www-didactique.imag.fr/Balacheff/PubliBalacheffPartiel.html>> (acesso em 29/05/2003). Ressaltamos que, por tratar-se de um documento on-line, suas citações carecem da numeração de páginas.

² Traduzido do termo em francês para “*Environnement Informatique d’Apprentissage Humain*”, denotado pela abreviação EIAH.

interação sujeito/meio quando o meio é um dispositivo informático, ou seja, na interação de dois sistemas cognitivos: a máquina que “conhece” e o sujeito humano que conhece, situando o problema na modelização dessa interação e na análise de sua complexidade sob o ponto de vista cognitivo. Além disso, o autor interessa-se particularmente pela possibilidade de estudar essa interação sem a obrigatoriedade de uma análise acerca do funcionamento psicológico do sujeito, o que justifica sua abordagem.

A terceira idéia relaciona-se ao clássico paradoxo semiótico: “*toda representação transforma o representado*” e esta representação algumas vezes “permite ver” e outras vezes “impede de ver” algumas propriedades do objeto que é representado (Balacheff, 1998). Em outras palavras, quando se cria um meio para representar um determinado objeto, este meio irá tanto ressaltar como ocultar certas propriedades do objeto representado.

Este paradoxo é essencial na concepção do processo de Transposição Informática, visto que as transformações que dele derivam não só são inevitáveis na elaboração do projeto de um ambiente informático de aprendizagem, como levam a uma enorme quantidade de escolhas específicas que surgem por parte da engenharia de softwares do ambiente.

A partir dessas idéias preliminares, o autor afirma que um dispositivo informático divide o “*mundo*” em três regiões que devem ser devidamente delimitadas (Balacheff, 1994, pp 365-368):

- O “*universo interno*”, constituído pelos diversos componentes eletrônicos e pelo modelo formal de programação, cujas articulações permitem o funcionamento do dispositivo informático. Este modelo formal, que será representado numa linguagem de programação, é implantado graças a um sistema de representação determinado pelos estados físicos da máquina, que vão implicar propriedades de componentes materiais ou softwares de diversos níveis (processadores, compiladores, etc.). Tais propriedades materiais geram, portanto, uma complexidade interna que acrescenta uma dimensão específica à transformação que está sendo operada, ou seja, a representação de um modelo matemático formal numa linguagem informática de programação.

- A “*interface*”, compreendendo a “tela” do computador, ou qualquer outro periférico de comunicação entre o usuário humano e o dispositivo informático. Em particular, a tela de um computador é um espaço finito e composto por um conjunto também finito de pontos, os “pixels”, que estão num certo estado de iluminação. É sobre este conjunto de limitações que os conhecimentos disponibilizados pela interação serão explicitados, onde se passam as coisas e onde é possível observar o que se passa. É sobre essa pequena tela que vão se observar os fenômenos, estudá-los e utilizar o nosso conhecimento para dar sentido ao que é “realizado” pelo aluno e “mostrado” pela máquina.

- O “*universo externo*”, onde se encontra o usuário humano ou notadamente, os conhecimentos mobilizados e as representações construídas pelo sujeito, a partir das interações com os dispositivos informáticos.

Além disso, o autor afirma que programar é criar um sistema formal, ou seja, criar um programa que “faz aritmética”, que “faz álgebra”, ou que “desenha objetos geométricos”, por exemplo, é criar um sistema formal e este modelo, ao ser implantado em um dispositivo informático, gera uma fonte de complexidade que deve ser analisada rigorosamente. Esta problemática justifica então a criação de um conjunto de concepções no sentido de melhor situar ao menos parte da problemática apresentada acima. Estas concepções serão o foco de análise da Transposição informática, discutida a seguir.

2.4 Transposição Informática

A noção de Transposição Informática parte do pressuposto fundamental de que a passagem de uma determinada representação à outra, implica uma transformação. Dessa forma, o problema em questão é compreender o que resulta desse esforço de representação/transformação quando, a partir de um modelo matemático de referência, seja ele algébrico, geométrico ou numérico, tentamos criar um modelo computável, um modelo que será manipulado por um dispositivo artificial. O que se passa durante esta transformação? Como passar das representações que estamos habituados a lidar nos livros, sobre o papel, ou no decorrer de interações verbais, para representações simbólicas que sejam manipuláveis por um dispositivo informático? Quais serão as conseqüências destas transformações?

Em outras palavras, “a questão fundamental neste contexto é determinar a distância entre o universo da tela com a qual se interage, e a partir da qual se decide a legitimidade, a pertinência e a validade das operações realizadas; e o modelo matemático de referência³” (Balacheff, 1998). Assim, define-se a Transposição Informática a partir desta problemática, ou seja, do processo de transformação, que compreende a passagem de um sistema de representação externo (o qual se compartilha classicamente na Matemática) a um sistema de representação interno, bem como o processo a ele subjacente (Balacheff, 1994, p. 369).

A Transposição Informática levanta três questões principais no que diz respeito às possibilidades disponibilizadas por um dado software numa perspectiva de aprendizagem. A primeira está relacionada ao que Balacheff (1994, p. 370) chamou de “*Domínio de Validade Epistemológica*” de um ambiente informático, a qual engloba essencialmente o sentido que um dado software educativo permite construir para uma determinada noção matemática, ou seja, analisa-se a natureza dos significados possibilitados pelas propriedades dos sistemas de representação, que são disponibilizados pelos softwares. Em síntese, esta questão ressalta a complexidade criada pelo fato de um mesmo objeto

³ Traduzido por nós do original em francês.

matemático não possuir a mesma natureza em dois ambientes distintos, isto é, ambientes diferentes apresentam, via de regra, tipos de controle e tipos de construção diferentes para uma mesma noção matemática.

A segunda questão considera o problema da aprendizagem como um problema de modelização, que implica, “por um lado, a passagem de um domínio de fenomenologia a um sistema de conhecimento – e o trabalho nesse sistema de conhecimento – e, por outro lado, o retorno desse sistema sob forma de interpretação. Este duplo movimento constitui o ciclo normal da modelização, isto é, o ciclo da interação entre aquele que modeliza e o objeto da modelização”⁴ (Balacheff, 1998).

A última questão explora o “*Domínio de Validade Didática*” que, em termos simples, explora a margem de controle que o professor pode ter sobre o ambiente informático, posto à disposição de seus alunos. Admite-se como pressuposto que há diferenças entre o que ocorre e o que se espera que ocorra na interação didática com um dado ambiente informático. Este fato torna delicada a gestão das situações de aprendizagem, em particular, fazendo com que o professor perca uma parte da capacidade de controle que é indispensável à condução da classe. Dessa forma, a questão levantada com o *domínio de validade didática* centra-se na antecipação das complexidades didáticas inerentes à escolha de um ambiente informático de aprendizagem e em como descrever os limites de sua utilização eficaz e fiel no âmbito da prática em sala de aula (Balacheff, 1998).

Nesta pesquisa, pretendemos abordar principalmente os aspectos relativos ao *domínio de validade epistemológica* de um conjunto de ambientes informáticos de aprendizagem e alguns aspectos da modelização de situações de ensino, assim como o domínio de fenomenologia destas situações. Devido a este fato, a questão do *domínio de validade didática* será apenas citada, sem maiores aprofundamentos. Além disso, devemos ressaltar que alguns termos por nós utilizados terão somente a abrangência dada por Balacheff em seus trabalhos, em particular, quanto às idéias de “*modelização*” e domínios de “*fenomenologia*”, que são utilizadas em outras áreas da Educação Matemática, com significados e abrangência por vezes distintos dos quais estaremos interessados.

⁴ Traduzido por nós do original em francês.

2.4.1 Domínio de Validade Epistemológica

Enquanto o impacto da tecnologia na prática diária ainda não correspondeu às expectativas de duas ou três décadas atrás, seu impacto epistemológico é mais profundo do que o esperado. Este impacto é baseado numa reificação de objetos matemáticos e relações que os alunos podem, como nunca antes, utilizar para atuar diretamente sobre esses objetos e relações. Esta nova realidade matemática, quando associada com o fato do computador ter se tornado um novo parceiro no contrato didático, força-nos a estender a transposição didática das matemáticas para uma transposição informática (Balacheff & Kaput, 1996, p. 469).

A noção de *domínio de validade epistemológica* de um ambiente informático é apontada no sentido de prover uma contrapartida teórica para o impacto epistemológico, criado pela introdução da tecnologia no processo didático. Assim, esta noção refere-se essencialmente à natureza do conhecimento e das relações por ele disponibilizadas, a partir de um ambiente informático específico. Portanto, o domínio de validade epistemológica visa criar um conjunto de ferramentas conceituais que permitam analisar as características de um ambiente informático em relação a um determinado domínio de conhecimento matemático, assim como desenvolver mecanismos que possibilitem a distinção entre diferentes ambientes, no que diz respeito aos potenciais específicos nos quais cada um deles pode contribuir nas diferentes etapas do processo de ensino e aprendizagem.

Nessa perspectiva, a noção de *domínio de validade epistemológica* de um ambiente informático de aprendizagem visa fornecer tanto um referencial teórico, voltado a delimitação da natureza das representações e significados disponibilizados por um determinado ambiente computacional, quanto criar um conjunto de ferramentas metodológicas para a análise desses ambientes. Dessa forma, segundo Balacheff & Sutherland (1994, p. 148), a noção de *domínio de validade epistemológica* de um micro-mundo levanta pelo menos quatro dimensões a serem consideradas na análise de um dado domínio de conhecimento matemático:

- a natureza das ferramentas e objetos fornecidos pela estrutura formal do micro-mundo;
- a natureza da fenomenologia sobre esta estrutura formal;
- o tipo de controle que o micro-mundo disponibiliza aos usuários e o “feedback” fornecido;
- o conjunto de problemas que o micro-mundo permite propor.

Essas quatro dimensões de análise serão centrais, tanto para delimitar a abordagem teórica deste trabalho, como para justificar seu desenvolvimento metodológico. Em outras palavras, partimos de uma abordagem metodológica voltada à análise qualitativa de um conjunto de ambientes informáticos, no que diz respeito a um determinado domínio de conhecimentos matemáticos. Esta abordagem se baseia diretamente na concepção do *domínio de validade epistemológica*, resumida nas dimensões citadas acima, portanto, nas próximas seções deste trabalho, pretendemos desenvolver e analisar cada uma dessas dimensões.

Dessa forma, as seções finais deste capítulo são dedicadas ao estudo das duas primeiras dimensões de análise, propostas pela noção de *domínio de validade epistemológica*. Nessas seções, pretendemos analisar, num primeiro momento, a natureza das ferramentas e objetos fornecidos pela estrutura formal de cada um dos micro-mundos estudados. Assim, uma vez que elegemos ambientes voltados ao ensino e aprendizagem de Geometria, pretendemos analisar o tratamento dado aos conceitos e objetos geométricos que são apresentados em cada ambiente, as ferramentas e operações por eles disponibilizadas e as regras que associam tais objetos, ferramentas e operações. Em suma, pretendemos apresentar as características particulares de cada um dos ambientes analisados de forma geral, assim como propor algumas comparações entre as estruturas formais presentes em cada um dos micro-mundos.

Num segundo momento, iremos nos concentrar na natureza da fenomenologia existente sobre a estrutura formal de cada um dos micro-mundos analisados. Em síntese, esta dimensão visa relacionar os objetos, ferramentas e operações matemáticas, disponibilizadas em cada ambiente, aos fenômenos que “aparecem” na tela do computador em função das ações realizadas pelo usuário, ou seja, analisamos a interface de cada ambiente e como esta interface traduz um modelo de referência matemático, nesse caso a Geometria, para um conjunto de informações apresentadas num periférico informático de comunicação – a tela do computador.

É importante salientar que não temos, absolutamente, a pretensão de analisar todas as ferramentas, possibilidades de construção e operações disponibilizadas nesses ambientes, objetivo este que, por si só, superaria em

muito a abrangência deste trabalho, mas sim de apresentar um conjunto mínimo de ferramentas e objetos indispensáveis ao desenvolvimento das demais dimensões aqui analisadas.

Feito isso, conduziremos o foco de nossas análises para os diferentes tipos de controle que cada micro-mundo disponibiliza aos usuários e o “feedback” por eles fornecido e, nesse ponto, concentraremos nossas análises em um domínio matemático mais específico, ou seja, na Geometria Fractal. Para tanto, dedicamos todo o terceiro capítulo para explorar as possibilidades de construção das diferentes famílias de fractais – apresentadas matematicamente no início do trabalho – em cada um dos ambientes informáticos aqui analisados. Com isso, pretendemos estudar como cada família de fractais pode ser construída nos diferentes micro-mundos analisados e que tipo de controle e de “feedback” cada micro-mundo disponibiliza a seus usuários para tais construções.

Finalmente, a última dimensão visa analisar o conjunto de problemas que cada micro-mundo permite propor e, dada a óbvia abrangência desta dimensão, adotamos uma abordagem ainda mais específica para essa dimensão, relacionando-a ao tratamento didático que daremos à construção de fractais. Assim sendo, dedicamos o quarto, e último, capítulo do trabalho a este tópico, onde relacionamos as três questões principais presentes neste trabalho, ou seja, a Geometria Fractal, os ambientes informáticos de aprendizagem e as Transformações Geométricas no Plano.

Além disso, no último capítulo pretendemos apresentar um conjunto de situações de ensino, focadas nas Transformações Geométricas no Plano, analisando-as a partir de um conjunto de aplicações envolvendo a construção de fractais em diferentes ambientes informáticos de aprendizagem. Para tanto, elegemos uma família específica de fractais – o Jogo do Caos – e dois ambientes informáticos distintos, que permitam a construção desses fractais, a partir de conceitos relativos às Transformações Geométricas no Plano.

O fato do domínio de validade epistemológica fornecer um conjunto de concepções teóricas para a análise de ambientes informáticos estabelece um vínculo imediato com diversas pesquisas relacionadas à utilização desses ambientes. Em particular, para a análise de ambientes voltados ao ensino da

Geometria, foco desta pesquisa, esta noção relaciona-se particularmente à problemática dos “micro-mundos” (*microworlds*), que será melhor explicitada na próxima seção deste trabalho.

2.4.2 Micro-mundos

A análise da dimensão epistemológica dos saberes e representações disponibilizadas por um dado ambiente informático de aprendizagem cria um vínculo natural com o desenvolvimento dos softwares educativos e, em particular para os domínios de conhecimento da Matemática, este vínculo leva-nos ao conceito de “micro-mundos”. Dessa forma, faremos uma síntese desta noção para melhor situar nossas intenções, seguido de uma análise dos dois enfoques utilizados na engenharia dos softwares estudados numa perspectiva epistemológica, para então analisar as características particulares de cada um dos quatro softwares escolhidos.

Segundo Hoyles (1993, pp. 1-2), o termo “micro-mundo” (*microworld*) foi usado pela primeira vez pelos cientistas da inteligência artificial para descrever “um pequeno e coerente domínio de objetos e atividades implementadas na forma de um programa de computador que correspondesse a uma parte do mundo real”. Uma vez que os objetos correspondentes do mundo real eram em geral muito complexos, os primeiros micro-mundos representavam versões simplificadas da realidade, agindo como experimentos para testar teorias comportamentais. Em meados da década de 1970, Seymour Papert propôs uma modificação importante nesta concepção, na qual o “domínio simples e coerente se tornou parte de um domínio de conhecimento com significado epistemológico” (*ibid.*, p. 1). Conseqüentemente os micro-mundos passaram a ser vistos como “a incorporação concreta dos domínios da Matemática”, ou seja, no centro da idéia de micro-mundo está um domínio de conhecimento a ser investigado pela interação com o software.

Nesse sentido, o conhecimento é reconhecido como complexo, inter-relacionado e em evolução, e estas características devem ser refletidas no ambiente. Em particular, supõe-se fundamental que os elementos primitivos do

ambiente possam ser combinados, formando novos elementos que estendam seu “vocabulário” facilmente, ou seja, não limitando o ambiente a um conjunto predeterminado de ações permitidas. Assim, o objetivo fundamental de um micro-mundo é o de fornecer ferramentas informáticas nas quais o conhecimento possa se desenvolver interativamente na busca de objetivos epistemologicamente ricos (Hoyles, 1993, pp. 2-4).

Em Balacheff & Kaput (1996, p. 471) ressalta-se que o objetivo conceitual de um “micro-mundo” é o de “prover uma base semântica para um sistema formal, permitindo ao aluno explorar simultaneamente a estrutura de objetos acessíveis, suas relações e a representação que os tornem acessíveis”.

Dessa forma, pode-se dizer que um micro-mundo consiste das seguintes características essenciais (Balacheff & Sutherland 1994, p. 142):

- i) Um conjunto de objetos primitivos, operações elementares com esses objetos e regras que expressam a maneira como essas operações podem ser realizadas e associadas – a qual é uma estrutura usual de um *sistema formal*, no sentido matemático.
- ii) Um domínio de fenomenologia que relacione os objetos e ações matemáticas subjacentes aos fenômenos que “aparecem na tela”. Este domínio de fenomenologia irá determinar o tipo de feedback que o micro-mundo produz como consequência das ações e decisões do usuário.
- iii) A possibilidade de transformar operações complexas ou objetos em novas operações ou objetos disponíveis para uso posterior, ou seja, o ambiente permite a criação e incorporação de “novas” ferramentas não fornecidas *a priori* pelo ambiente. Nesse sentido, se costuma dizer que “um micro-mundo se desenvolve enquanto o usuário aprende”.

No sentido de melhor explorar os micro-mundos estudados nesse trabalho, é necessário fazer uma distinção introdutória entre dois enfoques utilizados na engenharia dos quatro softwares que serão analisados. Por um lado, teremos um ambiente baseado em uma “*linguagem de programação*”, neste caso a linguagem LOGO, adotada pelo software MicroWords e, por outro lado, teremos ambientes

baseados na “*geometria dinâmica*”, utilizada no Cabri-Géomètre, no Geometer’s SkechPad e no Geometricricks.

Dada a importância desses enfoques, tanto na noção de micro-mundos de Geometria, quanto na delimitação do *domínio de validade epistemológica* desses ambientes, apresentamos, na próxima seção, um estudo mais específico das concepções de geometria da tartaruga e geometria dinâmica.

2.4.3 Geometria da Tartaruga e Geometria Dinâmica

Dentro das potencialidades oferecidas pela linguagem LOGO, estaremos particularmente interessados na “*Geometria da Tartaruga*” (Turtle Graphics), utilizada no MicroWorlds, cuja estrutura se baseia essencialmente na Geometria Diferencial. Dessa forma, pode-se dizer que LOGO é uma linguagem de programação completamente definida por um conjunto de ações elementares e objetos, como números e listas, e uma sintaxe cujo objetivo é combinar ações e manipular objetos.

A estrutura central de controle da linguagem LOGO é a recursividade. Assim, a partir do ponto de vista da “*Geometria da Tartaruga*” (turtle graphics), torna-se possível a utilização de estruturas recursivas para modelizar fenômenos geométricos, como os fractais, por exemplo. A esta estrutura formal subjacente está associada um domínio fenomenológico que, neste caso, é um conjunto de traços deixados por uma pequena “tartaruga” durante sua “jornada” pela tela, a qual é definida por uma seqüência organizada de ações que irá constituir um certo programa (Balacheff & Sutherland, 1994, p. 143). Além disso, o Microworlds permite a incorporação de “novas” ferramentas a partir da possibilidade de se transformar um sistema organizado de ações elementares em ações complexas através de “procedimentos” de programação.

Por outro lado, temos os softwares baseados na “*Geometria Dinâmica*”, que pode ser completamente definida por um conjunto de objetos elementares, como pontos, retas, segmentos e circunferências, e ações elementares (desenhar

uma reta perpendicular dados um ponto e uma reta, retas paralelas, etc.), essencialmente baseados na Geometria Euclidiana⁵.

Segundo Balacheff & Sutherland (1994, p. 143), o domínio de fenomenologia que está associado à Geometria Dinâmica permite que os desenhos que aparecem na tela possam ser diretamente manipulados ao se “pegar” e “arrastar” quaisquer objetos que possuem um certo grau de liberdade (alguns objetos não podem ser movidos, como pontos de intersecção de duas retas ou uma reta paralela, por exemplo). Os ambientes baseados na geometria dinâmica permitem a transformação de um conjunto de ações elementares em ações complexas através do uso, por exemplo, de “*macro-construções*”, no Cabri, ou através de “*scripts*”, no Sketchpad. Deve-se ressaltar, entretanto, que o GeomeTricks, apesar de ser baseado nas concepções da geometria dinâmica, não dispõe dessa última característica essencial, ou seja, uma ferramenta que possibilite a transformação de ações elementares em ações complexas.

Segundo Healy & Hoyles (2001, pp. 236-237), “a diferença crítica entre os ambientes de programação e as interfaces com manipulação direta giram em torno de sua ênfase na interação com um controle simbólico, no primeiro caso, em oposição a um controle visual, no segundo”⁶.

Entre os pesquisadores que privilegiam a Geometria da Tartaruga, argumenta-se que a presença necessária da representação formal de uma linguagem de programação é um bônus, visto que uma linguagem de programação provê tanto um texto no qual conjecturar, quanto um rico meio lingüístico com o qual os aprendizes podem desenvolver seus próprios “vocabulários” e comunicar suas estratégias de resolução de problemas. Os defensores dos ambientes baseados em linguagens de programação alegam que é por meio desse vocabulário que os alunos são capazes de expressar suas idéias matemáticas (Healy & Hoyles, 2001, pp. 236-237).

Em contrapartida, outros pesquisadores defendem a idéia de que as expressões formais presentes nessas linguagens limitam a acessibilidade. Em Laborde (1993, p. 59), por exemplo, argumenta-se que o esforço necessário para

⁵ Deve-se ressaltar que o modelo euclidiano adotado pelos ambientes baseados na geometria dinâmica, parte de um enfoque fenomenológico, ou seja, tendo-se em vista o que é apresentado ao usuário do ambiente. Entretanto, o modelo computável, utilizado pelo sistema interno de representação desses ambientes, baseia-se na fortemente na Geometria Analítica.

acompanhar a sintaxe de um ambiente de programação pode ser tão grande que, para o aprendiz, os problemas geométricos seriam reduzidos a problemas de linguagem, ou seja, é através da manipulação direta, disponibilizada pela Geometria Dinâmica, que o usuário poderia lidar e modificar não somente objetos como as relações entre eles.

Para os que privilegiam a Geometria Dinâmica, ressalta-se que a proposta principal dos ambientes nela baseados, ou seja, a introdução de estratégias clássicas de resolução de problemas geométricos, via régua e compasso, em um ambiente informatizado, é cognitivamente potencializada pela possibilidade da manipulação direta das construções realizadas. Dessa forma, a principal característica desses ambientes repousa no fato de uma construção poder ser continuamente modificada enquanto preserva suas propriedades geométricas quando um de seus elementos independentes é arrastado. Tal enfoque garante um maior controle e um rico feedback das propriedades geométricas estudadas em uma determinada construção, uma vez que essas propriedades podem ser visualizadas como invariantes sob o contínuo movimento da figura. Segundo Laborde (1993, p. 56), “a novidade aqui é que a variabilidade inerente à figura é expressa com um significado gráfico de representação e não somente na linguagem. Uma nova dimensão é adicionada ao espaço gráfico enquanto meio para a geometria: o movimento”⁷.

Em contrapartida, há pesquisadores que ressaltam o fato de que, apesar de ser relativamente simples construir e explorar figuras geométricas complexas a partir dos menus de rolagem dos ambientes baseados na Geometria Dinâmica, ainda estão em aberto problemas relacionados com as diferenças entre as estratégias de resolução de problemas dos alunos em relação às realizadas por matemáticos experientes (Healy & Hoyles, 2001, pp. 236-237). Em outras palavras, enfatiza-se a importância de se reconhecer que as interações com micro-mundos computacionais por aqueles que já conhecem as matemáticas não é necessariamente a mesma daqueles que ainda estão em fase de elaboração de conceitos matemáticos e processos relevantes.

⁶ Traduzido por nós do original em inglês.

⁷ Traduzido por nós do original em inglês.

Independentemente das preferências pela Geometria da Tartaruga ou pela Geometria Dinâmica, o fato é que o feedback fornecido por ambos os tipos de micro-mundo não é somente de natureza perceptiva, mas também de natureza conceitual. Nos ambientes baseados na Geometria da Tartaruga, o feedback ocorre na confrontação entre a figura esperada e a figura efetivamente produzida após a definição dos procedimentos de programação. Por sua vez, nos ambientes baseados na Geometria Dinâmica, o aprendiz é confrontado com a idéia de certos invariantes da figura construída, a partir do movimento das mesmas, ou seja, o feedback ocorre quando a utilização imprecisa de conceitos geométricos implica na imprecisão do resultado obtido após a manipulação da figura. Em ambos os casos, é fornecido um feedback visual que requer antecipação e conseqüentemente algum conhecimento do aprendiz (Laborde, 1993, p. 66).

A partir dessas características introdutórias, passaremos a analisar mais detalhadamente cada um dos ambientes estudados nesse trabalho.

2.5 MicroWorlds LOGO

A linguagem de programação LOGO e a “tartaruga” desenvolvida nos ambientes que a utilizam foram inicialmente introduzidas por Seymour Papert, na década de 1960, com o objetivo de apresentar a Matemática e os computadores às crianças na escola primária. Porém, com o desenvolvimento da informática, da inteligência artificial e das pesquisas referentes aos ambientes informáticos de aprendizagem, esta idéia inicial se desenvolveu tremendamente e, atualmente, existem diversos softwares que utilizam-se da linguagem LOGO e da Geometria da Tartaruga, direcionados para diferentes propósitos e níveis de ensino. Assim sendo, apesar de seus objetivos iniciais, atualmente seria incorreto considerar o LOGO como uma linguagem direcionada a crianças do ensino primário.

Segundo Harold & DiSessa (1992), a tradição de chamar as criaturas que aparecem nos ambientes baseados no LOGO de “*tartarugas*” começou com Grey Walter, um neurofisiologista que, no começo da década de 1960, realizou experimentos na Inglaterra com pequeninas criaturas robóticas que ele chamou de “*tortoises*”. Isto inspirou as primeiras “tartarugas” (*turtles*) projetadas no MIT – pequenos robôs controlados por computador que se moviam no chão em resposta aos comandos FORWARD (para frente) e RIGHT (para a direita). O trabalho no presente contexto matemático e computacional herdou e seguiu diretamente as terminologias dessas primeiras tartarugas.

Conforme já dito, LOGO é uma linguagem de programação e como tal, se baseia em um conjunto de comandos e uma sintaxe que irá determinar as ações realizadas na tela do computador, ou seja, seu domínio fenomenológico se baseia em figuras que são desenhadas por uma tartaruga que se move na tela de acordo com um conjunto de comandos especificados na sintaxe de programação. Cabe ressaltar que os exemplos e comandos apresentados a seguir foram baseados no software MicroWorlds, portanto outros softwares baseados na linguagem LOGO podem ter a sintaxe de comandos ligeiramente diferente da que será aqui apresentada.

Os movimentos da tartaruga são controlados por um conjunto de comandos simples (chamados “primitivos”) que determinam ações específicas e previamente “conhecidas” pela tartaruga. Os principais primitivos permitem que a tartaruga “ande” para frente (fd ou forward) e para trás (bk ou back), assim como

“vire” para direita (rt ou right turn) e esquerda (lt ou left turn). A tartaruga possui uma caneta amarrada em sua cauda que, ao se mover, pode deixar uma linha por onde passa, se a caneta estiver “abaixada” (pd, ou pen down) ou se mover sem deixar a linha (pu, ou pen up).

Os comandos primitivos relativos aos movimentos da tartaruga devem ser seguidos por um valor numérico ou parâmetro que especifique o número de “passos” dado pela tartaruga, quando ela é mandada para frente ou para trás, e quantos “graus” ela deve virar à direita ou esquerda. Dessa forma, o MicroWorlds disponibiliza um padrão de medida baseada na quantidade de “passos” dados pela tartaruga e um sistema de orientação baseado na direção apontada pela “cabeça” da tartaruga (heading). Com esses primitivos, é possível fundamentar as ações da tartaruga tanto a partir de seus movimentos relativos quanto a partir de um sistema de coordenadas cartesianas ou polares.

Para o sistema de coordenadas cartesianas, convencionou-se o centro da tela como origem e o número de passos da tartaruga como escala de medida, utilizando, portanto, um sistema de coordenadas baseado nas dimensões da tela em “passos”. Para o MicroWorlds, a dimensão da tela é definida pelos intervalos $[-160, 160]$ para abscissas e $[-100, 100]$ para ordenadas. Os comandos setx e sety enviam a tartaruga, respectivamente, para uma determinada abscissa ou ordenada.

No sistema polar, utiliza-se igualmente o número de passos como medida de distâncias e o ângulo, em graus, da orientação da tartaruga como direção. Convencionou-se o sentido horário como padrão de orientação e a direção “vertical, para cima” como ângulo de 0° , assim, o comando “seth 90” (que aponta a tartaruga para o ângulo de 90°), por exemplo, orienta a tartaruga na horizontal para a direita.

No sentido de melhor delimitar o domínio fenomenológico do MicroWorlds, podemos, por exemplo, combinar esses comandos primitivos, para desenhar um triângulo equilátero de lado 100 “passos” com a seguinte seqüência de comandos: fd 100 rt 120 fd 100 rt 120 fd 100 rt 120. Visto que neste exemplo estamos utilizando somente os movimentos relativos da tartaruga, o triângulo terá o primeiro lado construído na mesma direção que a apontada pela tartaruga antes da introdução dos comandos (fig. 2.1). O ângulo de 120° representa o ângulo

externo do triângulo, uma vez que a tartaruga é orientada em função de sua direção anterior.

Neste exemplo, pode-se notar que dois comandos são repetidos (fd 100 rt 120). Tal fato é muito comum nas construções com o LOGO e, devido a isto, existe um primitivo específico para facilitar a iteração de comandos na programação, o comando “repeat”. Este primitivo pode ser utilizado para repetir os comandos “fd 100” e “rt 120”, resumindo a sintaxe de construção de nosso triângulo eqüilátero com a instrução: repeat 3 [fd 100 rt 120].

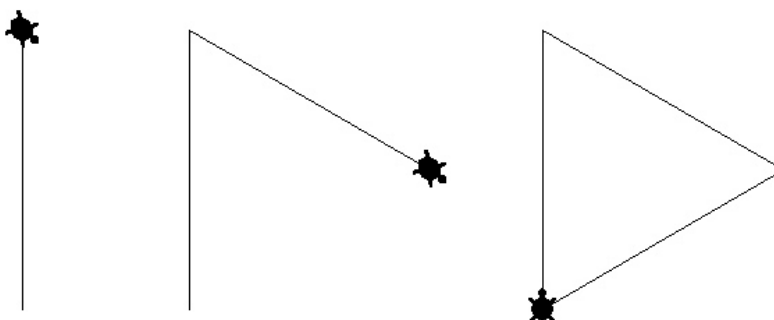


Figura 2.1: Construção de um triângulo eqüilátero no MicroWorlds LOGO.

Outra característica fundamental da linguagem LOGO é a possibilidade de se “criar” novas ferramentas a partir dos objetos inicialmente fornecidos pelo ambiente. No LOGO, essa característica é disponibilizada pelos “procedimentos” de programação e pelo uso de “parâmetros”. Assim, no exemplo do triângulo, temos duas considerações importantes que explicitam o uso de procedimentos e parâmetros: em primeiro lugar, a tartaruga “esquece” a construção do triângulo imediatamente após sua construção, ou seja, para a construção de um novo triângulo eqüilátero, é necessário digitar novamente todos os comandos; em segundo lugar, com os comandos desenvolvidos no exemplo, somente é possível construir um triângulo eqüilátero de lado 100.

O uso de “procedimentos” de programação no LOGO cria então uma importante solução para a primeira consideração feita acima, ou seja, a partir desse recurso é possível estender o vocabulário do ambiente com a criação de novos comandos que utilizam combinações de comandos primitivos. Assim, é possível criar o comando “triangulo” (ou qualquer outro nome) que constrói

automaticamente o triângulo equilátero do exemplo, a partir do seguinte procedimento:

```
to triangulo  
repeat 3 [fd 100 rt 120]  
end
```

Dessa forma, um procedimento irá “ensinar” a tartaruga a executar tarefas mais complexas a partir da combinação de comandos primitivos que ela já “conhece”, ou seja, ao se introduzir esse procedimento em um programa, basta digitar o comando “triangulo” que a tartaruga irá automaticamente desenhar um triângulo equilátero de lado 100.

Para a segunda consideração, existe a possibilidade de se inserir um conjunto de “parâmetros” num procedimento do LOGO no sentido de atribuir um maior grau de generalidade ao procedimento e às construções realizadas. No exemplo, o novo comando “triangulo” é limitado a somente construir triângulos equiláteros de lado 100, porém é possível criar um procedimento que desenhe triângulos equiláteros com lados de qualquer medida, permitindo a representação de classes de triângulos equiláteros. Este resultado é obtido com a introdução do parâmetro “lado”, como segue:

```
to triangulo :lado  
repeat 3 [fd :lado rt 120]  
end
```

A introdução do parâmetro “lado” impõe ao procedimento um valor para a medida do lado dos novos triângulos construídos, portanto, ao se utilizar esse comando, deverá ser introduzido um valor numérico para o novo parâmetro, ou seja, para se construir um triângulo equilátero de lado 50, por exemplo, deveremos utilizar o comando “triangulo 50”. Deve-se ressaltar que um procedimento pode ter vários parâmetros em função das necessidades da construção projetada.

Como dissemos anteriormente, a estrutura central de controle da linguagem LOGO é a recursividade e, em termos muito simples, tal fato implica na criação de procedimentos que executam a si próprios, ou seja, entre os comandos existentes dentro do procedimento, introduz-se a execução do próprio

procedimento impondo a este um “loop” de repetições. É imediato que a introdução deste artifício em um programa conduziria a um ciclo infinito de repetições do procedimento programado, portanto, para a utilização da recursividade é imperativo que se utilize um comando que defina “quando” a tartaruga deve “parar” a execução de um determinado procedimento que é sucessivamente repetido.

Para tornar esta característica mais concreta, iremos propor como exemplo a construção de uma “espiral” recursiva formada por triângulos eqüiláteros de lados sucessivamente menores até um determinado limite, no qual a tartaruga será instruída a “parar” a construção. Neste programa, faremos uso de dois procedimentos: o primeiro constrói os triângulos eqüiláteros, já explicitados anteriormente, com a introdução do comando “fd: lado” que irá posicionar a tartaruga sobre o segundo vértice do triângulo construído. O segundo procedimento do programa define uma espiral formada por triângulos eqüiláteros com lados cada vez menores. Os procedimentos e o resultado da construção são dados abaixo:

```
to espiral :lado  
if :lado < 1 [stop]  
triangulo :lado  
rt 30  
espiral :lado - 4  
end
```

```
to triangulo :lado  
repeat 3 [fd :lado rt 120]  
fd :lado  
end
```

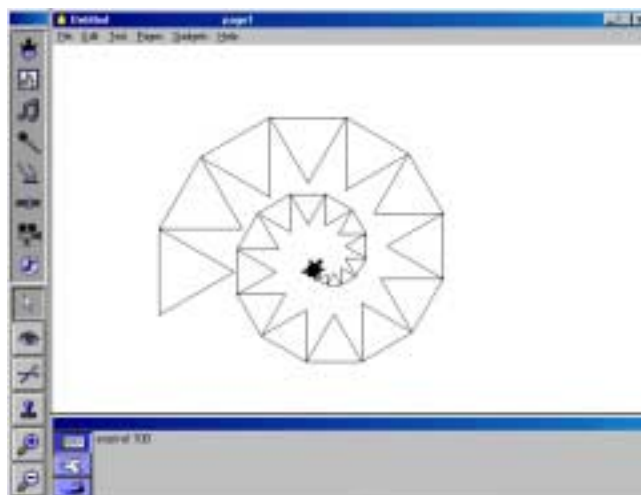


Figura 2.2: Espiral recursiva no MicroWorlds LOGO.

Analisando o programa, temos a introdução do procedimento “espiral” que, após alguns comandos iniciais, executa a si próprio com o novo parâmetro de entrada “lado – 4”, definido em função do “lado” original. Além disso, é introduzido o comando “if :lado < 1 [stop]”, que instrui a tartaruga a “parar” a construção “se” o lado do triângulo for menor que “1 passo”.

Dessa forma, ao se digitar o comando “espiral 100”, por exemplo, o programa realiza a seguinte seqüência de ações:

- interrompe a construção “se” o parâmetro “lado” for menor que 1;
- caso contrário, executa o comando “triângulo :lado”, que realiza o procedimento “to triângulo” para construir um triângulo equilátero;
- “gira” a tartaruga 30° para a direita (rt 30);
- impõe a estrutura recursiva ao executar novamente o comando “espiral”, porém com o parâmetro “lado” medindo quatro passos a menos que o comando original (neste exemplo, o segundo triângulo será construído com lado igual a 96 passos).

Nesse programa pode-se notar como uma estrutura recursiva é inserida na linguagem LOGO, visto que o procedimento “espiral” irá gerar um “loop” que repete a si mesmo até o limite de “1 passo”, construindo réplicas sucessivamente menores do triângulo equilátero original. Com isso, é imediata a analogia com a construção de fractais pois, como vimos no primeiro capítulo deste trabalho, tais construções são essencialmente baseadas em estruturas recursivas.

Para as construções de fractais com o MicroWorlds, que apresentaremos nas próximas seções, será necessária a discussão de um último recurso disponível na linguagem LOGO: a possibilidade da introdução de variáveis e fórmulas algébricas. Em diversas construções, principalmente naquelas onde é necessária a introdução de expressões algébricas de algum tipo, é comum a necessidades de se fixar valores numéricos específicos para a utilização de fórmulas. No LOGO, tal recurso é possibilitado através do uso de “variáveis” e do comando “make”, que visa tanto criar uma variável, que receberá um nome, quanto armazenar e atribuir um valor numérico para essa variável. Assim, caso se queira, por exemplo, criar uma variável “x” cujo valor é “10”, utiliza-se o comando “make "x 10”. Nesse comando, são usadas aspas para atribuir o nome a uma

variável e esta, após definida, poderá ser utilizada em outros comandos ou procedimentos com a mesma sintaxe utilizada em qualquer parâmetro (como o comando “fd :x”, por exemplo).

A criação e utilização de variáveis pelo LOGO é fundamental para a introdução de fórmulas ou expressões algébricas em um procedimento de programação. Assim, para se inserir uma “fórmula” de cálculo baseada em uma função algébrica, por exemplo, uma função iterada na forma $x_{n+1} = -5x_n + 2$, cujo valor inicial de “x” é “0” e o próximo valor de “x” é dado em função do anterior, utiliza-se o procedimento que segue:

```
to formula  
make "xn -5 * :x + 2  
make "x :xn  
end
```

Este procedimento supõe a criação prévia da variável “x” com o valor “0” em outra parte do programa. Isto se deve à utilização de uma estrutura recursiva para este procedimento, que segue a seguinte ordem:

- executa-se (em outro procedimento) o comando “make "x 0”, criando a variável “x” e se define um procedimento recursivo que utiliza o procedimento “formula”.
- no procedimento “formula” é criada a variável “xn”, representando o próximo valor “ x_n ” da função, cujo valor é calculado pela expressão “ $x_n = -5x + 2$ ”.
- o ciclo recursivo se encerra ao se atribuir a “x” o valor de “xn” (calculado na linha anterior), dando prosseguimento ao programa.

Cabe ressaltar que os exemplos aqui desenvolvidos apresentam somente alguns procedimentos muito simples, visando apenas uma introdução inicial aos recursos disponibilizados pela Geometria da Tartaruga. Nas construções de fractais que apresentaremos no próximo capítulo serão utilizados comandos mais avançados que exigem conhecimentos mais aprofundados da linguagem LOGO e que serão descritos nessa ocasião.

2.6 Cabri-Gèomètre

O Cabri-Gèomètre II, idealizado por Baulac, Bellemain & Laborde (1994), pode ser considerado um micro-mundo de geometria dinâmica, que, em termos simples, se propõe a realizar construções clássicas da geometria tradicional com régua e compasso na tela do computador, porém com o diferencial de possibilitar a interação com as figuras geradas via manipulação direta destas. A palavra CABRI foi cunhada a partir da abreviação da expressão francesa para **Cahier de Brouillon Interactif**, ou Caderno de Rascunho Interativo, nome que define bem seus objetivos fundamentais.

A possibilidade de manipulação direta das construções, apesar de ser um conceito aparentemente simples, que a olhos descuidados poderia parecer sem grande importância, visa elevar o status das produções dos alunos de meros desenhos para figuras geométricas próximas ao sentido ideal dessas⁸. Isto se deve ao fato desse ambiente ter sido projetado de forma a fazer com que a manipulação das figuras carregue consigo as propriedades geométricas de sua construção, ou seja, somente uma construção “bem feita” no sentido geométrico irá manter suas propriedades geométricas iniciais, ou *invariantes*, após a manipulação de quaisquer de seus elementos (Laborde, 1993).

Conforme já ressaltamos, a natureza das ferramentas fornecidas pela estrutura formal dos ambientes baseados na Geometria Dinâmica está essencialmente relacionada com a Geometria Euclidiana, portanto, os objetos e ações disponibilizados no Cabri estão diretamente ligados às construções geométricas tradicionais com régua e compasso assim como às propriedades clássicas presentes nesse modelo matemático.

Dessa forma, o domínio de fenomenologia associado à Geometria Dinâmica visa transformar a tela do computador em uma grande folha de papel em branco, na qual os objetos geométricos são desenhados e manipulados a partir de um conjunto de menus de rolagem e do mouse. Nesses menus, é fornecido um conjunto de ferramentas que representam os elementos clássicos da geometria euclidiana (como ponto, reta, segmento, circunferência...), assim

⁸ Segundo Laborde (1993), um desenho corresponde à representação de um objeto geométrico em um suporte material (folha de papel, tela do computador, lousa, ...), enquanto que a figura

como diversas ferramentas de construção geométrica (reta paralela ou perpendicular, ponto médio, simetrias...). Uma característica que deve ser destacada quanto ao domínio de fenomenologia é o fato de que as ferramentas de construção presentes no Cabri assumem um duplo significado no sentido de representarem tanto uma ferramenta de construção quanto as propriedades geométricas intrínsecas a essa construção. Assim, a ferramenta “reta perpendicular”, por exemplo, impõe tanto a construção de uma reta perpendicular a outra quanto garante e carrega consigo as propriedades geométricas da perpendicularidade (Balacheff & Sutherland, 1994, p. 141).

Outra característica importante a ser destacada refere-se à manipulação dos objetos e desenhos construídos. Como dissemos, um importante enfoque associado à manipulação direta das construções no Cabri repousa no fato desta “carregar” consigo as propriedades geométricas das construções, ou seja, as propriedades geométricas de uma dada construção podem ser visualizadas como “invariantes” sob o movimento dos desenhos. Conseqüentemente, este enfoque presente na manipulação direta impõe aos objetos utilizados em uma construção diferentes “graus de liberdade” quanto à sua manipulação. Como retoma Jahn (1998) em sua tese, são estabelecidos três “graus de liberdade” para os objetos geométricos presentes no ambiente – graus de liberdade 0, 1 e 2.

Objetos com grau de liberdade 2 são livremente manipulados na construção, podendo ser movidos em qualquer direção, ou seja, nas duas dimensões do plano. Em geral, objetos com este grau de liberdade são aqueles utilizados no início das construções, como pontos livres, retas e segmentos de reta, por exemplo.

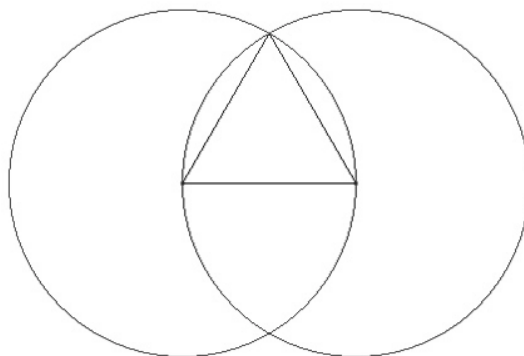
Objetos com grau de liberdade 1 também podem ser livremente manipulados na construção, porém têm seu movimento restringido por algum outro objeto ao qual esteja relacionado por alguma propriedade geométrica. Em geral, os objetos com grau de liberdade 1 se caracterizam por “estarem sobre” outros objetos. Assim, por exemplo, um ponto “sobre” uma reta somente poderá ser movido na direção definida pela reta e uma semi-reta cuja origem foi definida sobre um ponto de intersecção poderá assumir qualquer direção porém sua origem não poderá ser manipulada diretamente.

refere-se a possíveis significados atribuídos a tais representações, com base num modelo de referência.

Finalmente, os objetos com grau de liberdade 0 se caracterizam por não poderem ser diretamente manipulados, ou seja, são objetos que não podem ser “arrastados”. Estes objetos são resultado de ferramentas dotadas de propriedades geométricas ou de construções que impõe um caráter de dependência geométrica a tais objetos. Assim, um ponto de intersecção ou uma reta paralela, por exemplo, não podem ser “arrastadas” diretamente pelo mouse. Isto não significa que esses objetos sejam “estáticos” visto que, se os objetos que foram utilizados em uma construção com grau de liberdade 0 forem movidos, a construção como um todo irá se movimentar, ou seja, uma reta paralela não pode ser diretamente arrastada pelo mouse, porém ela é definida a partir de um ponto e uma outra reta e estes objetos podem, em geral, serem movidos. Dessa forma, quando a reta ou o ponto que definem uma reta paralela são movidos, esta também o é, caracterizando o domínio de fenomenologia associado à idéia dos invariantes geométricos de uma determinada construção.

Como exemplo, seguindo a mesma linha da seção anterior, propomos a construção de um triângulo equilátero a partir do Cabri com o objetivo de melhor expor e delimitar as principais características do domínio fenomenológico deste ambiente. Como dissemos, o Cabri é um ambiente essencialmente interativo que, ao contrário do MicroWorlds, baseia-se essencialmente na manipulação direta dos objetos construídos na tela do computador, não dispondo portanto de um texto ou rotina de programação que defina as ações realizadas. Assim sendo, representaremos as construções do exemplo a partir da reprodução das telas do próprio Cabri, seguidas de comentários a respeito das mesmas.

Das diversas formas possíveis para a construção de triângulos disponibilizadas pela geometria euclidiana, e passíveis de reprodução a partir do Cabri, iremos optar por uma das mais tradicionais, utilizando duas circunferências, como segue:



Nesta construção poder-se-ia ressaltar diferenças fundamentais no que se refere aos domínios de validade epistemológica entre o MicroWorlds e o Cabri, que pode-se estender, em geral, para ambientes de programação LOGO e ambientes de geometria dinâmica. Como apontado por Balacheff & Sutherland (1994, pp. 144-145) existem diferenças significativas entre os modelos matemáticos de referência nos dois tipos de ambiente e estas irão gerar diferentes níveis de complexidade das construções, gerando diferenças significativas também quanto ao nível cognitivo. Isto faz com que o resultado das interações com estes ambientes levem a construções em “milieux” completamente diferentes, apesar das construções se referirem a uma mesma classe de figuras.

No MicroWorlds, como vimos, a natureza dos conhecimentos geométricos utilizados (explicitamente ou nas ações) para a construção de um triângulo equilátero são relativos, pelo menos, ao fato de:

- os lados de um triângulo equilátero serem iguais;
- os ângulos internos de um triângulo equilátero serem iguais;
- as relações entre os ângulos internos e externos de um triângulo.

Por sua vez, para a construção de um triângulo equilátero são necessários, pelo menos, os seguintes conhecimentos:

- como criar pontos, segmentos e circunferências;
- características de um ponto de intersecção entre dois objetos;
- igualdade de os lados de um triângulo equilátero;
- igualdade de raios de duas circunferências.

Além disso, também cabe ressaltar o fato de que nos ambientes de programação LOGO há a utilização explícita e imprescindível de valores numéricos para lados e ângulos nas construções utilizadas, ao contrário dos ambientes de geometria dinâmica que, apesar de possuírem ferramentas que manipulem valores numéricos, privilegiam em geral propriedades geométricas das figuras não relacionadas necessariamente à utilização de representações numéricas.

Outra característica fundamental do Cabri é a possibilidade de se “criar” novas ferramentas a partir dos objetos inicialmente fornecidos pelo ambiente. Essa possibilidade é disponibilizada a partir das “macro-construções”, são elas que permitem ao Cabri a construção e incorporação de novas ferramentas, e esta é uma característica essencial na concepção de um micro-mundo. Dessa forma, é possível criar no Cabri uma nova ferramenta que possibilite a construção de, por exemplo, um triângulo equilátero qualquer.

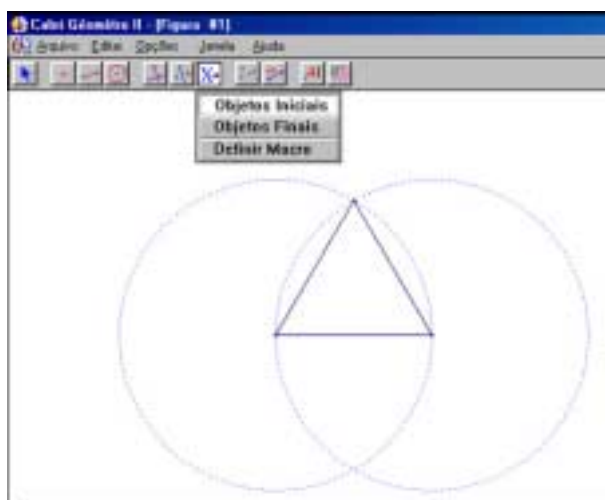


Figura 2.4: Tela do Cabri com o menu de macro-construções e construção de um triângulo equilátero .

A figura 2.4, que representa o menu das macro-construções, mostra as ferramentas necessárias para se definir uma macro-construção. Partindo-se de uma construção geométrica já concluída (nesse caso o triângulo equilátero), definimos os objetos iniciais, que servem de base para a construção, e os objetos finais que serão efetivamente construídos. Dessa forma, optamos por selecionar dois pontos quaisquer (que serão dois dos vértices do futuro triângulo) como

objetos iniciais⁹ e o triângulo definido por esses vértices e o ponto de intersecção entre as duas circunferências, como objeto final.

Com esses procedimentos, que basicamente consistem em se selecionar as opções desejadas no menu das macro-construções e clicar nos objetos da construção, concluí-se o procedimento com a opção “Definir macro”.

Um ponto que merece destaque no Cabri é que, a partir das macro-construções é possível se reformular totalmente o ambiente, criando-se novas ferramentas ou, inclusive, substituindo-se ferramentas previamente fornecidas nos menus de construção por outras, em função das necessidades didáticas que se deseja incorporar a uma dada atividade neste ambiente. Um bom exemplo disto são os recentes suplementos desenvolvidos pelos criadores do Cabri para o estudo de geometrias não-euclidianas, como o modelo hiperbólico de Poincaré, por exemplo, os quais modificam completamente as ferramentas originais disponíveis na versão original do ambiente.

Por outro lado, pode-se dizer que o Cabri é um software essencialmente geométrico, portanto é natural afirmar que ele não é voltado à representações gráficas que envolvam outros aspectos matemáticos, em especial os algébricos. Outro ponto relevante se refere à inexistência de alguma ferramenta específica que permita a utilização da recursividade na construção das figuras, ou seja, construções que se baseiam neste tipo de estrutura são, em geral dificultadas, mesmo com a utilização das macro-construções. Este fato trará, como veremos no próximo capítulo, certas dificuldades na construção de fractais.

⁹ Ressaltamos o fato de termos optado por estes objetos iniciais pois existem outras possibilidades para esta escolha como, por exemplo, um segmento qualquer que poderia servir igualmente como objeto inicial, neste caso, como um dos lados do triângulo.

2.7 Geometer's Sketchpad

O Geometer's Sketchpad 3.10, idealizado por Nicholas Jackiw, constitui outro ambiente fundamentado na geometria dinâmica, portanto o modelo matemático de referência presente em sua estrutura é o mesmo do Cabri, ou seja, os elementos fundamentais da geometria euclidiana. Além disso, o Sketchpad também pode ser considerado um micro-mundo uma vez que o este permite a criação e a incorporação de novas ferramentas através do uso de “scripts”.

Entretanto o Sketchpad possui algumas características diferentes em relação ao Cabri uma vez que este software, segundo seu idealizador, visa “explorar o processo de expressão das relações geométricas visualmente e através de demonstrações. Este processo denomina-se ‘*sketching*’ (rascunhando ou esboçando) e foi desenvolvido através da noção de ‘*spatial programming*’.”¹⁰ (Jackiw & Finzer, 2002). Em outras palavras, o Sketchpad se propõe a introduzir uma linguagem de programação baseada nas estruturas de dependência entre os objetos geométricos de uma determinada construção. Esta linguagem possui uma semântica baseada nos próprios elementos da geometria euclidiana que são dispostos da mesma forma que uma rotina de programação, fornecendo um roteiro próximo a uma “demonstração” de cada passo das construções geométricas realizadas. Por outro lado, apesar das afirmações do autor em relação às potencialidades dos “scripts” gerados pelo Sketchpad enquanto linguagem de programação, cabe-nos a ressalva de que, até onde pudemos

¹⁰ Traduzido por nós do original em inglês.

observar neste trabalho, tais “scripts” não são diretamente editáveis, ou seja, não nos foi possível alterar as relações neles criadas.

Visto que o Sketchpad segue basicamente o mesmo modelo de referência adotado pelo Cabri, podemos afirmar que suas estruturas básicas de funcionamento são praticamente idênticas, ou seja, afora as diferenças óbvias na aparência dos dois ambientes, a interface de ambos possui basicamente as mesmas ferramentas. Assim sendo, grande parte das afirmações feitas na seção anterior em relação ao Cabri são igualmente válidas para o Sketchpad, portanto iremos concentrar nossas análises a partir do ponto onde os ambientes divergem.

A primeira diferença importante entre os dois ambientes ocorre na forma como estes estendem suas ferramentas. De fato, encontramos diferenças importantes entre a definição das macro-construções, no Cabri, e dos scripts, no Sketchpad e, devido a isto, iremos propor um exemplo baseado na utilização desse recurso.

O primeiro ponto que destacamos é a utilização de um script para a construção, com os mesmos elementos utilizados no Cabri, de um triângulo equilátero. Analisando a fig. 2.5, podemos observar como o Sketchpad descreve a definição de uma construção geométrica em seus scripts e como o ambiente apresenta esta ferramenta. Como podemos observar, a interface do Sketchpad conta com duas janelas distintas, a janela “sketch” onde as construções geométricas são realizadas e manipuladas, representando a “folha de papel em branco” onde o aprendiz realiza seus “esboços”, e a janela “script” onde estes são definidos e gravados, e onde a descrição geométrica das ações realizadas é apresentada em forma de texto.

Nesse ponto, é possível deixar mais claras as intenções do idealizador do Sketchpad ao propor a introdução de uma linguagem de programação em seu ambiente, ou seja, basicamente um script possui uma dupla função no ambiente: ele tanto se presta à criação de novas ferramentas de construção, aumentando a acessibilidade do ambiente, quanto descreve cada passo da construção geométrica executada para a definição da ferramenta programada.

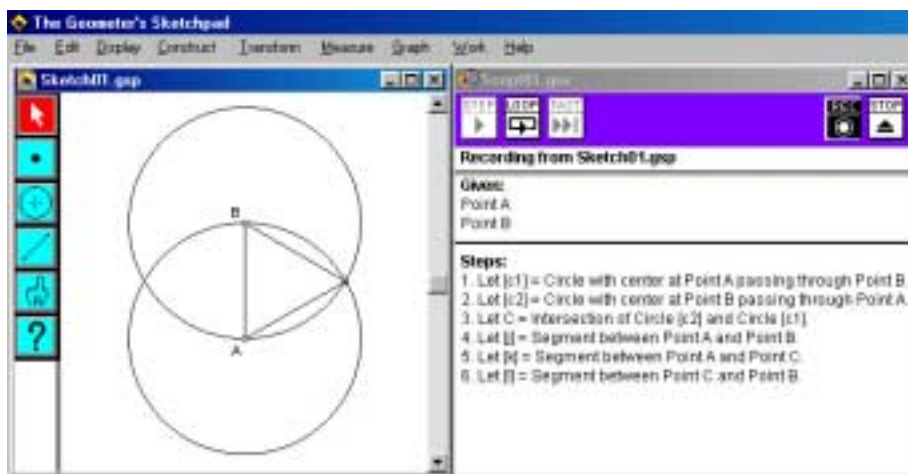


Figura 2.5: Tela do Sketchpad com o “sketch” de construção de um triângulo equilátero (à esq.) e seu respectivo “script” (à dir.).

Dessa forma, ao se gravar um script (iniciada com o botão “REC”), o ambiente automaticamente gera um texto descrevendo cada passo da construção realizada, utilizando-se de uma semântica praticamente idêntica daquela utilizada em “roteiros” didáticos de construção geométrica e com isso estabelecendo explicitamente as estruturas de dependência geométrica entre os objetos utilizados.

Ainda segundo Jackiw & Finzer (2002), o processo se dá a partir da determinação dos “elementos básicos” da construção, chamados “dados” (*given*), que possuem grau de liberdade 2, podendo ser manipulados livremente, e dos “passos” (*steps*) dados em cada etapa da construção. Esta estrutura, que é apresentada no texto fornecido pelo script, fornece formalmente a dependência entre os objetos utilizados na construção e cria uma fonte de generalização e abstração a partir de um desenho para uma classe de figuras relacionadas. Além disso, como já dissemos, o script também se presta à criação de novas ferramentas para o ambiente e, após sua gravação, este script também pode ser utilizado para a construção de classes de triângulos equiláteros a partir da seleção de dois pontos quaisquer na tela “sketch” do ambiente.

O texto mostrado no script é construído paralelamente às construções geométricas realizadas, ou seja, cada novo objeto adicionado ao “sketch” é simultaneamente traduzido para o script, mostrando as relações e dependências geométricas presentes na construção. Esta característica é outro diferencial deste ambiente em relação ao Cabri, pois aqui a definição de um script tem de ser

realizada concomitantemente às construções geométricas que o definem, ao contrário das macro-construções do Cabri, que podem ser definidas a partir de uma construção já concluída (apenas selecionando-se os objetos iniciais e finais). Porém, como o ambiente não permite a edição das construções já realizadas, caso ocorra algum erro no processo de definição do script, este erro irá obrigar o usuário a apagar toda a construção e recomeçar a gravação do script desde seu início, fato este que implica numa certa rigidez na definição de um script.

O segundo ponto importante a ser destacado relaciona-se ao fato do Sketchpad contar com uma ferramenta específica voltada à repetição sucessiva de instruções programadas anteriormente, ou seja, o ambiente permite a utilização da recursividade nas construções programadas em um script. Para tornar a utilização deste recurso mais clara, iremos propor um exemplo de construção de uma espiral recursiva formada por triângulos equiláteros, semelhante à apresentada na seção dedicada ao MicroWorlds.

Na figura 2.6, apresentamos o resultado gráfico da construção realizada e a reprodução do script utilizado, no qual é possível observar como a espiral recursiva é definida a partir de dois pontos dados e de nove passos de construção.

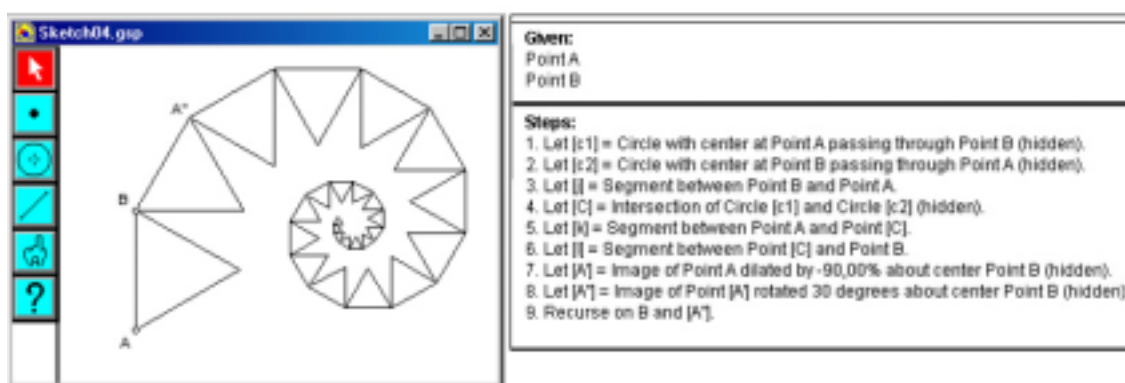


Figura 2.6: Tela do Sketchpad com o “sketch” de construção da espiral recursiva (à esq.) e seu respectivo “script” (à dir.).

Os seis primeiros passos do script realizam a construção de um triângulo equilátero, como já analisado no exemplo anterior, os dois passos seguintes (sétimo e oitavo) utilizam ferramentas de transformações geométricas para o posicionamento do ponto A”, que representa um dos vértices do triângulo

construído na próxima recursão executada na construção, e o último passo introduz a ferramenta recursiva na construção.

Um ponto que merece destaque nessa construção refere-se a utilização das ferramentas de transformação geométrica. Comparando esta espiral com a construída a partir do MicroWorlds, pode-se perceber algumas diferenças na aparência final do desenho que são devidas à utilização de objetos matemáticos distintos. Na construção realizada no MicroWorlds, a recursão foi definida com o próximo triângulo com lado medindo 4 passos a menos e uma rotação de 30° em relação ao triângulo anterior. Já na construção com o Sketchpad, utilizamos duas transformações geométricas para reproduzir estas operações, uma “homotetia” (dilata) de centro B e fator 0,9, responsável pela diferença na aparência da construção, e uma rotação de 30° em relação ao triângulo anterior.

Finalmente destacamos a introdução de estruturas recursivas, possibilitada pelo Sketchpad, esta ferramenta é disponibilizada na opção “loop” do script e deve ser definida a partir da seleção dos mesmos objetos geométricos “dados” (given) no início da definição do script, ou seja, numa construção baseada em dois pontos dados, como nesse exemplo, a recursão deve ser definida com a seleção de dois pontos quaisquer, instruindo o programa a repetir todos os passos anteriores do script a partir destes novos pontos. Como destacado no MicroWorlds, uma estrutura recursiva gera um ciclo infinito de repetições de um dado comando, portanto é imperativo que se instrua “quando” a recursão deve ser interrompida. No Sketchpad, esta instrução é inserida em uma janela, que surge quando se executa o script, solicitando o número de recursões desejado (neste exemplo utilizamos 25 recursões).

2.8 GeomeTricks

O GeomeTricks 2.37, desenvolvido por Vigo Sadollin, foi recentemente traduzido para o português pelos Dr. Marcelo C. Borba e Dra. Miriam Godoy Penteadó do GPIMEM, UNESP - Rio Claro. Este ambiente também segue a linha da geometria dinâmica, possuindo basicamente a mesma interface e a maioria das características e ferramentas básicas dos ambientes concebidos a partir desta concepção e apresentados anteriormente.

Porém deve-se ressaltar que o GeomeTricks possui uma diferença importante em relação aos demais ambientes analisados – ele não possui uma opção que permita a incorporação de novas ferramentas ou a criação de novos ambientes. Esta característica contraria a definição utilizada neste trabalho para ser considerado um micro-mundo mas, apesar disso, o GeomeTricks foi incluído neste trabalho por possuir uma ferramenta específica para a construção de fractais que, como veremos nos próximos capítulos, será fundamental para a determinação de parte do problema de pesquisa abordado neste trabalho. Esta ferramenta se baseia na combinação de ternas de pontos e será melhor explorada na análise específica, dedicada à construção de fractais.

Dessa forma, visto que a interface do GeomeTricks é muito semelhante à de outros ambientes, por se basear na geometria dinâmica, e devido ao fato de interessarmos-nos principalmente pela ferramenta de construção de fractais nele presente, optamos por realizar uma análise aprofundada desse ambiente no próximo capítulo do trabalho. Esta escolha se justifica pelo fato de, no terceiro

capítulo do trabalho, propormos uma análise específica dos ambientes informáticos aqui apresentados, focada na construção de fractais.

CAPÍTULO 3

ANÁLISE DA CONSTRUÇÃO DE FRACTAIS EM DIFERENTES AMBIENTES INFORMÁTICOS

3.1 Considerações Iniciais

Neste capítulo, iremos nos dedicar à construção das diferentes famílias de fractais a partir de quatro ambientes informáticos de aprendizagem humana. Dessa forma, conforme introduzido no capítulo anterior, pretendemos estudar potenciais estratégias de construção de alguns representantes de cada uma das famílias de fractais, nos diferentes micro-mundos analisados. Com isso, focaremos o estudo nos diferentes tipos de controle e “feedbacks”, disponibilizados aos usuários em cada um deles.

Nessa perspectiva, optamos por listar as três famílias de figuras fractais, elegendo um ou dois representantes de cada uma delas para estudar a possibilidade de sua construção, em cada um dos ambientes escolhidos. Portanto, adotaremos, como abordagem, a construção dos fractais de cada família em todos os ambientes que viabilizem tais construções, para então analisá-las.

Com isso, este capítulo visa relacionar os tópicos estudados nos dois primeiros capítulos deste trabalho, ou seja, analisar como cada tipo de fractal pode ser construído, em cada um dos ambientes informáticos estudados. Para isso, adotamos uma metodologia baseada na análise qualitativa destes ambientes, fundamentada no *domínio de validade epistemológica* de micro-mundos.

3.2 Fractais “Iniciador-Gerador”

As figuras desta família foram os primeiros fractais construídos com processos iterativos e muitas delas remontam ao início do séc. XX, quando sequer a palavra “fractal” havia sido criada. Tais figuras foram idealizadas em uma época de grandes rupturas com a Matemática tradicional e têm, como característica, uma ênfase essencialmente geométrica em sua construção.

Devido a este enfoque geométrico, sua representação é viabilizada através de três dos softwares aqui analisados, o *MicroWorlds LOGO*, o *Cabri Gèomètre II* e o *Geometer’s SketchPad*. Por sua vez, a construção de fractais “iniciador-gerador” com o *Geometricricks* é, em termos práticos, inviabilizada devido a inexistência de um dispositivo que permita a incorporação de novas ferramentas de construção geométrica.

Optamos por analisar dois fractais desta família, a *Curva de Koch* e o *Floco de Neve de Koch*, com o objetivo de tornar as comparações, tanto entre os ambientes, como entre os processos de construção utilizados em cada ambiente, mais claras e acessíveis.

3.2.1 Linguagem LOGO a partir do MicroWorlds

A tradução do processo de construção do *Floco de Neve de Koch* para a linguagem LOGO, pode ser feita a partir de um programa composto por dois procedimentos relativamente simples, tornando a construção desta figura bastante rápida e prática. Além disso, dadas as características dos micro-mundos de geometria da tartaruga, o mesmo programa nos permite construir este fractal em diversos tamanhos e, mais importante, com diferentes níveis de iteração.

A figura abaixo mostra o resultado gráfico do programa para a construção do *Floco de Neve de Koch* a partir do *MicroWorlds*, assim como o programa utilizado para construí-lo. Nesta figura, são fornecidas quatro representações do fractal, construídas com diferentes níveis de iteração, cabendo-nos ressaltar que estas representações foram obtidas a partir dos mesmos procedimentos de programação, apenas modificando a posição inicial da “tartaruga” e variando o parâmetro inicial, relativo ao nível de iteração.

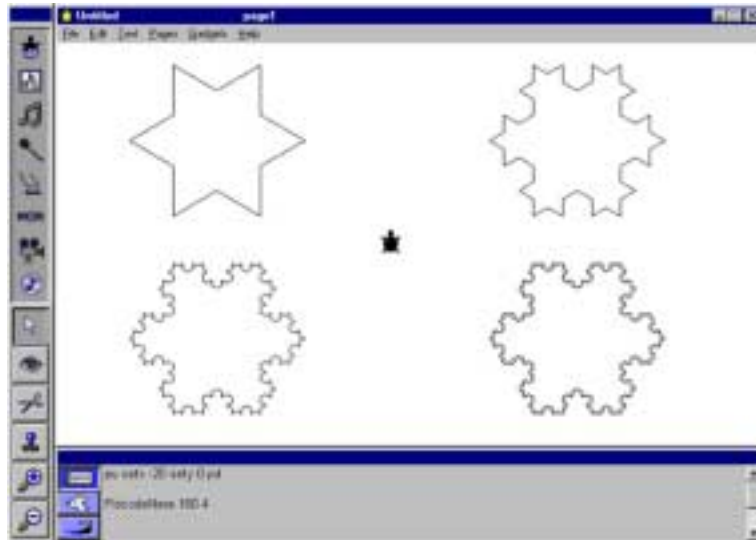


Figura 3.1: Construção do *Floco de Neve de Koch* a partir do MicroWorlds LOGO.

Para a construção dessas figuras, foram utilizados os seguintes procedimentos de programação:

```
to FlocoDeNeve :lado :nivel
repeat 3 [CurvaDeKoch :lado :nivel rt 120]
end
```

```
to CurvaDeKoch :lado :nivel
if :nivel = 0 [fd :lado stop]
CurvaDeKoch :lado / 3 :nivel - 1
lt 60 CurvaDeKoch :lado / 3 :nivel - 1
rt 120 CurvaDeKoch :lado / 3 :nivel - 1
lt 60 CurvaDeKoch :lado / 3 :nivel - 1
end
```

A partir do programa desenvolvido acima, passaremos a analisar as operações executadas em cada um de seus procedimentos.

Dessa forma, o procedimento “**to FlocoDeNeve**” cria o comando de construção da figura e introduz dois parâmetros numéricos de entrada, o parâmetro “lado”, que determina o comprimento em “passos” do segmento de reta iniciador da figura e portanto, seu tamanho final; e o parâmetro “nivel”, que define o nível de iterações utilizado na construção. São estes dois parâmetros que determinam o controle sobre as dimensões e nível de iteração da construção do *Floco de Neve de Koch*.

A partir dessas informações, e lembrando que o iniciador do *Floco de Neve de Koch* é um triângulo equilátero, o programa é instruído a repetir 3 vezes

o procedimento “**CurvadeKoch**” com os mesmos parâmetros de comprimento e nível de iteração, seguido do comando “rt 120”, que irá girar a tartaruga 120° para a direita, representando o ângulo externo de um triângulo equilátero. Em síntese, este procedimento possui a mesma estrutura daquele utilizado para a criação de um triângulo equilátero, apresentado no capítulo anterior.

Por sua vez, o procedimento “**to CurvaDeKoch**” visa a construção da *Curva de Koch* tradicional a partir dos mesmos parâmetros de tamanho e nível de iteração fornecidos no procedimento anterior. Assim, este programa apresenta também um procedimento que, utilizado individualmente, possibilita a construção da *Curva de Koch*. Este procedimento cria uma estrutura recursiva complexa para a construção da figura, que é definida em função do nível de iteração escolhido e executa as seguintes operações:

- move a tartaruga “lado” passos para frente e interrompe o programa “se” o parâmetro “nivel” for zero, ou seja, quando não houver mais iterações a serem feitas;
- caso contrário, executa o procedimento “CurvadeKoch” recalculando os parâmetros “lado” e “nivel” com um terço do comprimento do segmento iniciador e menos um nível de iteração;
- “gira” a tartaruga 60° para a esquerda e executa novamente o procedimento “CurvadeKoch” com os parâmetros “lado / 3” e “nivel – 1”;
- “gira” a tartaruga 120° para a direita e executa o procedimento “CurvadeKoch” com os mesmos parâmetros da operação anterior;
- “gira” a tartaruga 60° para a esquerda e executa o novamente o procedimento “CurvadeKoch” com os parâmetros da operação anterior.

3.2.2 Cabri Gèomètre II

A construção de fractais “iniciador-gerador” no Cabri é, em termos práticos, viabilizada por meio do uso de macro-construções, uma vez que tais construções dependem essencialmente de processos iterativos, com a repetição sucessiva de construções geométricas complexas. Porém, mesmo com a utilização das macro-construções, o Cabri não possui uma ferramenta específica destinada à utilização

da recursividade e tal característica faz com que a construção de fractais nesse ambiente seja relativamente longa e trabalhosa.

Por outro lado, esse componente “braçal” na construção de fractais possui uma característica interessante, que se torna inevitável devido aos tipos de controle disponibilizados no Cabri. Uma vez que o ambiente impossibilita a automatização do processo de construção, o usuário acaba sendo obrigado a dedicar maior atenção ao significado e importância dos processos iterativos, presentes na construção desses fractais. Nesse sentido, o Cabri acaba apresentando a construção dos fractais “iniciador-gerador” de forma mais explícita e transparente para o usuário, se comparado com as construções pré-programadas dos outros ambientes analisados.

Devido a esta característica, optamos por analisar apenas a *Curva de Koch* nesta seção, ressaltando que a única diferença existente entre esta e o *Floco de Neve* é o iniciador. Visto que o iniciador da *Curva de Koch* é um segmento de reta, enquanto o iniciador do *Floco de Neve* é um triângulo equilátero, a construção do último acabaria apenas “triplicando” o trabalho.

Assim sendo, o primeiro passo é a construção do gerador da *Curva de Koch*, ressaltando aqui os aspectos geométricos da divisão de segmentos em partes iguais e a construção de um triângulo equilátero, conforme apresentado na figura 3.2.

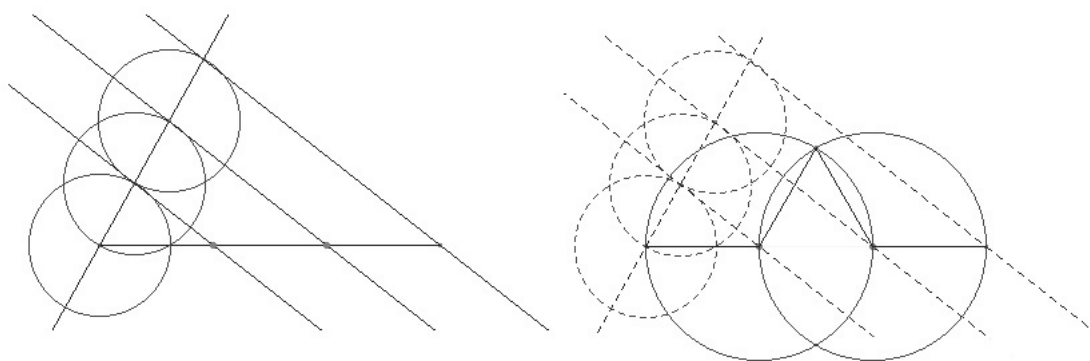


Figura 3.2: Processo geométrico para definição da macro-construção para a Curva de Koch a partir do software Cabri Géomètre II.

Devemos ressaltar que a determinação do gerador deste fractal decorre de certas escolhas específicas quanto ao enfoque geométrico adotado e, portanto, não representa a única forma de construí-lo. Além disso, foram realizadas diversas escolhas no sentido de quais objetos seriam “escondidos” e quais teriam sua cor modificada, escolhas estas condicionadas em função do correto funcionamento da macro-construção.

Em seguida, executamos a definição da macro-construção propriamente dita, na qual partimos de um único objeto inicial, o próprio segmento de reta iniciador da curva e, como objetos finais, selecionamos os quatro segmentos de reta que compõe o gerador do fractal. Esta etapa é, sem dúvida, a principal responsável pela maior parte das dificuldades encontradas na construção desse fractal, pois nela surgem diversos problemas decorrentes de características informáticas do software. Em particular, é necessária atenção especial para a definição dos segmentos e sua ordem, quais objetos devem ser “escondidos” e “mostrados” e sobre como colorir adequadamente alguns dos segmentos para conseguir uma apresentação adequada.

Dessa forma, tivemos que dar atenção a posição dos pontos extremos dos segmentos de reta utilizados pois, em tentativas anteriores, constatou-se que a distinção, realizada pelo ambiente, entre os “pontos iniciais” e os “pontos finais” de um segmento, afeta o funcionamento correto da macro-construção. Ainda que estes pontos não constassem dos objetos iniciais da macro-construção, sua ordenação interna pelo ambiente pode provocar, por exemplo, o posicionamento “invertido” do gerador do fractal em um determinado segmento.

Além disso, na construção geométrica do gerador da *Curva de Koch*, colorimos o segmento de reta iniciador de amarelo, no sentido de evitar a “poluição visual” do desenho, após sucessivas aplicações da macro-construção. Esta escolha se justifica se considerarmos que, ao deixá-lo “escondido”, o ambiente não irá permitir a seleção deste segmento como um novo objeto inicial da macro-construção, inviabilizando as sucessivas aplicações, necessárias à construção dos próximos níveis de iteração. Optamos também por “esconder” os pontos extremos dos segmentos que compõe o fractal por motivos estéticos, uma vez que sua presença dificulta a visualização da figura nos níveis de iteração mais altos. Entretanto, não “escondemos” os pontos extremos do segmento iniciador da *Curva de Koch*, com o objetivo de preservar o grau de liberdade da construção.

Outro inconveniente da macro-construção, que também está relacionado a características informáticas do software, advém da sobreposição de segmentos, uma vez que, a cada aplicação da macro-construção, um novo segmento de reta é gerado sobre os anteriores, acumulando diversos segmentos sobrepostos. Entretanto, este inconveniente não chega a dificultar a construção do fractal, se o usuário tiver em mente que será sempre o último segmento da lista que deve ser selecionado.

Apresentamos, na figura 3.3, a reprodução da tela do Cabri, na qual as figuras representam a construção dos quatro primeiros níveis de iteração da *Curva de Koch*. Ressaltamos que estas construções foram feitas a partir da mesma macro-construção, aplicada em diferentes segmentos.

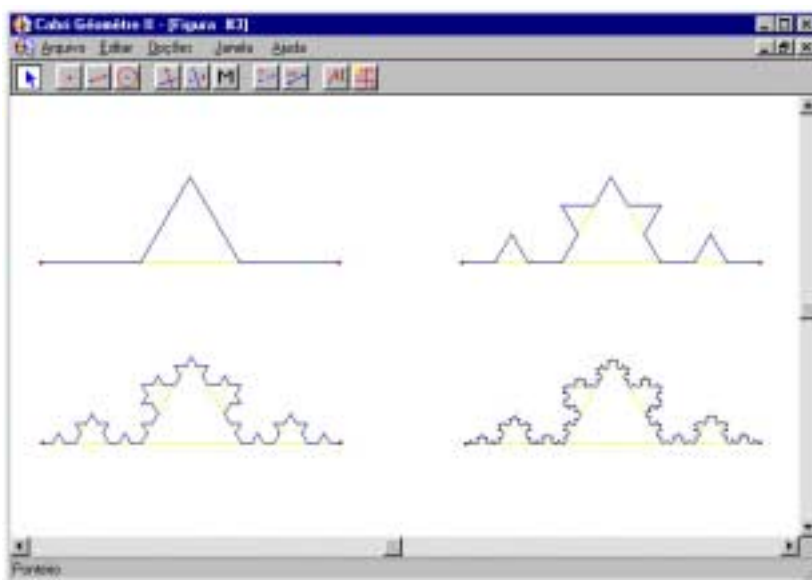


Figura 3.3: Construção da Curva de Koch a partir do software Cabri Gèomètre II.

Como o componente “braçal” da construção torna-se inevitável, devido a inexistência de uma ferramenta recursiva específica no Cabri, descrevemos o processo de construção destas quatro figuras a seguir, apenas para ressaltar o trabalho realizado. Assim, a primeira figura (canto superior esquerdo) representa o gerador do fractal, obtido na primeira iteração, com a aplicação da macro-construção uma vez. A segunda figura (canto superior direito), que apresenta o segundo nível de iteração, pode ser obtida a partir da primeira, com a utilização da macro-construção mais quatro vezes (uma vez em cada segmento do gerador). Para a terceira figura (canto inferior esquerdo), aplicou-se a macro-

construção outras dezesseis vezes, em relação à construção anterior. Por fim, no quarto desenho, a macro-construção foi aplicada mais 64 vezes.

Como o Cabri é um ambiente de geometria dinâmica, é natural destacar a característica dinâmica disponibilizada pela manipulação direta das construções realizadas. Neste caso, tal controle é obtido a partir do deslocamento das extremidades do segmento iniciador da *Curva de Koch*, cujos pontos não foram escondidos como os demais, gerados pela execução da macro-construção. Assim, a possibilidade de manipulação desses pontos pode representar uma forma de controle da “robustez” da construção, do funcionamento da macro-construção nos diferentes níveis de iteração, das dimensões e posição da figura, etc.

Porém, devemos ressaltar que a manipulação da Curva de Koch é cada vez mais dificultada, conforme avançamos nos níveis de iteração, devido à complexidade das construções definida por esta macro-construção, ou seja, por utilizar muitos objetos geométricos, a macro-construção acaba por sobrecarregar a memória de processamento de imagens do ambiente, tornando a manipulação direta da figura progressivamente mais lenta.

3.2.3 Geometer’s SketchPad

Como introduzido no capítulo anterior deste trabalho, o SketchPad possui uma ferramenta específica para processos iterativos e esta ferramenta será particularmente útil na criação de fractais “iniciador-gerador”. Além disso, contaremos com os “scripts”, que traduzem as construções realizadas no ambiente em forma de um texto, cuja sintaxe se aproxima dos roteiros de construção da geometria euclidiana tradicional. Este fato cria um novo tipo de controle das construções realizadas e pode, ao menos potencialmente, servir como meio de ligação entre o processo informático de construção de uma figura e a forma como esta figura seria tradicionalmente definida pela Geometria.

A figura 3.4 mostra três representações do *Floco de Neve de Koch* em diferentes níveis de iteração, assim como o “script” que a define. Este “script” é

definido a partir de dois pontos iniciais (A e B) e é a partir da posição desses pontos que a dimensão final da figura é determinada.

Além disso, estas representações foram obtidas com o mesmo “script”, visto que o SketchPad permite a escolha do número de iterações usada na construção. Porém, cabe aqui a mesma ressalva feita em relação ao Cabri, ou seja, que a manipulação direta do fractal está condicionada às limitações de memória de processamento de imagens, fato este que limita o nível de iterações máximo possibilitado pelo ambiente. Ressaltamos também que o “script” grava automática e simultaneamente cada ação realizada no processo de construção geométrica da figura, ou seja, após a escolha dos objetos iniciais para a construção e o início da gravação do “script”, cada passo da construção geométrica executada na tela é automaticamente traduzida e apresentada dentro do “script”.

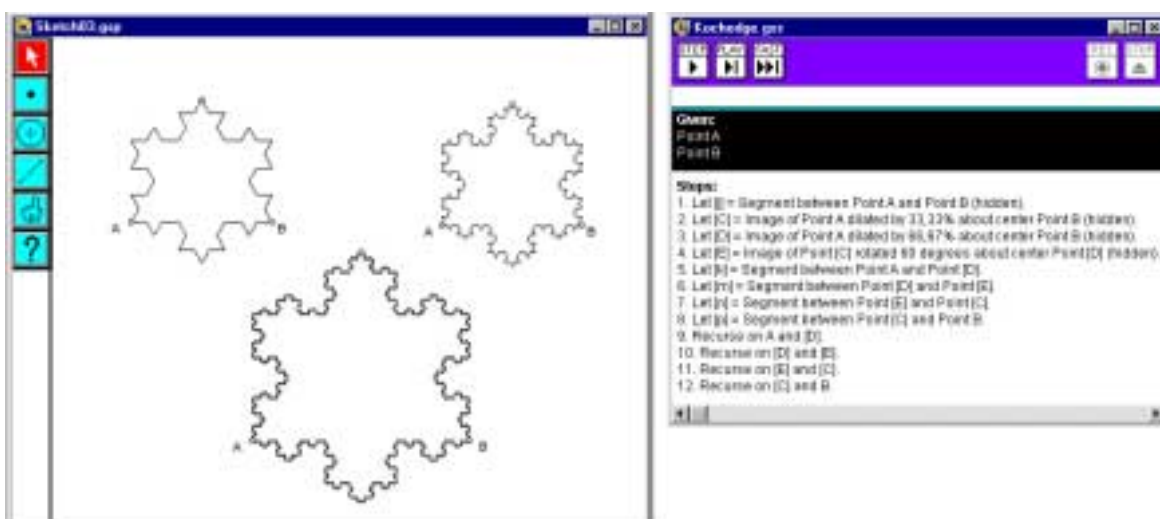


Figura 3.4: Construção do Floco de Neve de Koch com o “script” a partir do Geometer’s SketchPad.

Neste “script” é possível notar a seqüência de informações do procedimento de programação do ambiente, mostrada com a mesma terminologia das construções geométricas tradicionais. Além disso, é interessante ressaltar o nível de generalidade que este processo possibilita, no sentido de não serem utilizadas medidas para a representação das figuras e sim uma estrutura de dependência geométrica completamente baseada na posição e construção de pontos. Além disso, cabe ressaltar que, analogamente às construções realizadas no MicroWorlds, o presente “script” visa construir a *Curva de Koch* a partir de dois

pontos dados, portanto, na construção das figuras apresentadas, utilizou-se o artifício de se determinar um terceiro ponto, representando o vértice de um triângulo equilátero, para então combiná-los em sucessivas aplicações do “script”.

Outro detalhe importante se refere à utilização de transformações geométricas nas construções, neste caso homotetias (*dilate*) e rotações (*rotate*), que substituem o processo geométrico clássico de divisão de segmentos, utilizado no Cabri. Embora seja possível utilizar tais ferramentas no Cabri, o SketchPad apresenta uma diferença nos “argumentos” de transformação, ou seja, em seus elementos característicos. No primeiro, a utilização destas ferramentas em uma macro-construção impõe a definição de dois valores numéricos como objetos iniciais (a razão de homotetia e o ângulo de rotação), impondo sua seleção manual todas as vezes que a macro-construção é utilizada; enquanto que no segundo, como vimos, é possível a introdução destes “argumentos” diretamente no “script” de construção, automatizando o processo.

3.3 Fractais Gerados por Seqüências Complexas: o Conjunto de Mandelbrot

O *Conjunto de Mandelbrot*, como visto no primeiro capítulo, é um fractal gerado no plano complexo, cujas coordenadas são testadas a partir de seqüências de números complexos, pela expressão $z_{n+1} = z_n^2 + c$, onde “c” representa cada ponto a ser testado na grade de valores complexos que representa o conjunto. Apesar de ser um processo aparentemente simples, a construção deste fractal demanda uma enorme quantidade de cálculos para ser gerado. Além disso, o processo de construção do *Conjunto de Mandelbrot* possui características essencialmente algébricas e, devido a estas características, apenas o micro-mundo de Geometria da Tartaruga, sustentado pela linguagem LOGO, viabiliza sua construção.

A tradução do processo de construção do *Conjunto de Mandelbrot* para a linguagem LOGO parte de um conjunto de procedimentos extremamente complexo e, como conseqüência, este fato implica numa construção bastante lenta, demandando vários minutos para a confecção de cada figura. Além disso, o programa de construção do *Conjunto de Mandelbrot* só poderá ser completamente compreendido por usuários que possuam profundos conhecimentos, tanto do processo de construção desse fractal como, principalmente, da sintaxe de programação da linguagem LOGO.

A seguir, apresentamos o resultado gráfico do programa para a construção deste fractal, a partir do MicroWorlds, e o programa utilizado para construí-lo. Cabe-nos ressaltar que este programa foi adaptado por nós a partir do original obtido em “The Fractal Umbrella” (Scott & Beaumont, 2002, s.p.)¹.

¹ Disponível on-line em <<http://www.maths.adelaide.edu.au/pure/pscott/fractals/index.html>> (último acesso em 16/06/03).

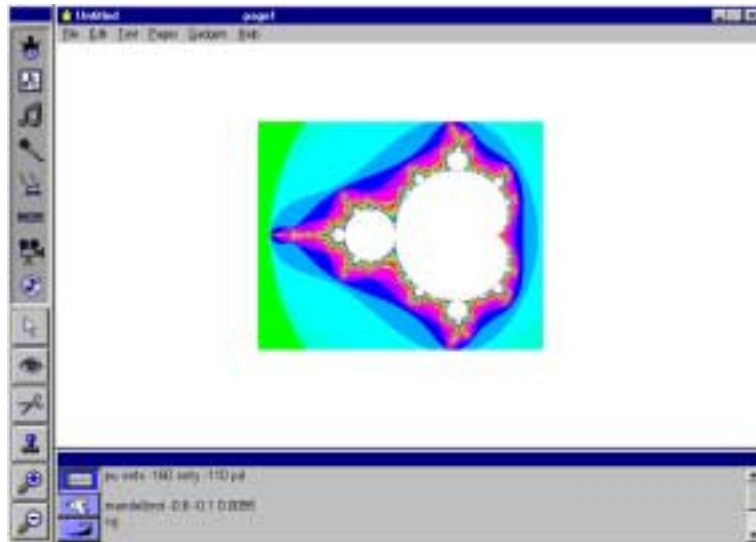


Figura 3.5: Construção do *Conjunto de Mandelbrot* a partir do software MicroWorlds LOGO.

Nesta construção, foram utilizados os seguintes procedimentos, nos quais a seta “→” indica que a linha de comando continua na próxima linha do texto impresso:

to mandelbrot :xc :yc :scale

```
seth 90 ht
make "y -110 repeat 241 [make "x -160 repeat 301 [setc 54 + 10 * mandelcolour :xc + →
:x * :scale:yc + :y * :scale dot se :x :y make "x :x + 1] make "y :y + 1]
end
```

to mandelcolour :x :y

```
make "a 0 make "b 0 make "n 0
while [and :a * :a + :b * :b < 4 :n < 160] [make "newb 2 * :a * :b + :y make "a :a * :a - →
:b * :b + :x make "b :newb make "n :n + 1] if :n > 159 [make "n -5.4]
output :n
end
```

to while :cond :do

```
if not run :cond [stop]
run :do
while :cond :do
end
```

to dot :pos

```
pu setpos :pos pd
fd 0.01 bk 0.01
end
```

A partir do programa desenvolvido acima, passaremos a analisar as principais características dos procedimentos apresentados. Entretanto, dada a complexidade da sintaxe adotada neste programa, antecipamos que esta análise abordará somente alguns aspectos relativos aos principais tipos de controle

disponibilizados pelo programa. Dessa forma, deixamos a trabalhos futuros uma análise mais detalhada acerca do funcionamento das diversas variáveis e das operações realizadas pelos procedimentos.

Nessa perspectiva, os principais controles disponibilizados pelo procedimento “**to mandelbrot**” relacionam-se à determinação das dimensões e posição da figura construída na tela do computador. Assim, neste programa optou-se pela construção de uma figura com dimensões de 341 x 241 “passos”, iniciada a partir da posição (-160 , -110) da tela do MicroWorlds (posição da extremidade esquerda inferior da figura). Além disso, este procedimento impõe a estrutura iterativa do programa, ao repetir e executar um algoritmo que define as variáveis introduzidas no procedimento “to mandelcolour”.

O procedimento “**to mandelcolour**”, por sua vez, traduz para a linguagem LOGO a expressão que testa cada número complexo da grade de valores que representará o *Conjunto de Mandelbrot*, definindo o número de iterações para a construção e as cores utilizadas para a apresentação gráfica de cada ponto da grade. Para compreender o funcionamento dos algoritmos utilizados neste procedimento, é essencial entender como a expressão que testa a grade de valores do *Conjunto de Mandelbrot* é aplicada na prática, funcionamento este que passamos a resumir².

Já vimos que a expressão que testa os valores da construção do *Conjunto de Mandelbrot*, “ $z_{n+1} = z_n^2 + c$ ”, gera uma seqüência de números complexos e que a determinação dos valores que pertencem ao conjunto é feita quando esta expressão resulta numa seqüência que se mantém finita, após um determinado número de iterações (visto que, em termos práticos, o número de iterações não poderá ser infinito). Pois bem, cabe-nos agora determinar como esse conjunto de considerações teóricas é utilizada na prática, ou seja, num conjunto de procedimentos computáveis através de um dispositivo informático.

O primeiro ponto a ser destacado decorre do fato dessa expressão utilizar números complexos em sua representação o que, como sabemos, impõe algumas adaptações para seu tratamento, a partir do modelo adotado pelo ambiente – o plano cartesiano com dois eixos *reais*. Portanto, devemos traduzi-la

² Para uma melhor compreensão deste processo, sugerimos que o leitor retome as considerações realizadas no primeiro capítulo deste trabalho acerca da construção matemática deste fractal.

para um conjunto de expressões reais equivalentes, decompondo os números complexos da expressão $z_{n+1} = z_n^2 + c$ em suas respectivas partes reais e imaginárias. Para tanto, adotaremos as seguintes notações (onde $i = \sqrt{-1}$):

- $z_{n+1} = a_{n+1} + b_{n+1} \cdot i$;
- $z_n = a_n + b_n \cdot i$;
- $c = x + yi$.

Estas notações justificam-se quando as aplicamos na expressão que testa os valores da grade, uma vez que o número complexo “c” assume um valor constante em todas as iterações da seqüência gerada (daí as diferentes notações para z_n e c). Com essa notação, a expressão original assumirá a seguinte forma:

$$- a_{n+1} + b_{n+1} \cdot i = (a_n + b_n \cdot i)^2 + (x + yi) = (a_n^2 - b_n^2 + x) + (2a_n b_n + y)i.$$

Com essa última expressão, resta-nos apenas um último problema para sua efetiva transformação em algoritmos computáveis – a unidade imaginária “i”. Porém, este problema é facilmente resolvido ao se desmembrar a expressão acima em dois algoritmos distintos, um para a *parte real* do número complexo e outro para sua *parte imaginária*, obtendo os algoritmos utilizados no programa, ou seja:

- “ $b_{n+1} \cdot i = (2a_n b_n + y) i$ ” é transformado em “**newb = 2.a.b + y**” e
- “ $a_n = a_n^2 - b_n^2 + x$ ” é transformado em “**a = a² - b² + x**”.

A partir dessas considerações, podemos concluir que o procedimento “**to mandelcolour**” realiza, em termos muito simples, as seguintes operações para testar cada um dos pontos da grade de valores do *Conjunto de Mandelbrot*:

- cria a variável “n”, que limita em 160 iterações a determinação do regime de comportamento de cada ponto testado;
- define um conjunto de algoritmos para testar esse regime em cada um dos pontos da grade de valores, executando-os como mostrado a seguir;
 - “enquanto” [$a^2 + b^2 < 4$ e $n < 160$] “executar”, nessa ordem, os algoritmos [**newb = 2.a.b + y ; a = a² - b² + x ; b = newb ; n = n + 1**];
 - “se” [$n > 159$] “executar” [$n = - 5.4$].

Estas operações têm por objetivo a determinação do regime de comportamento dos pontos testados e, conseqüentemente, a cor com a qual este ponto será representado graficamente. Assim, os primeiros algoritmos desse programa ($a^2 + b^2 < 4$ e $n < 160$) determinam quais pontos se mantêm em iterações finitas e quais rumam ao infinito.

Em outras palavras, quando a seqüência gerada por um determinado ponto permanecer delimitada por um círculo fixo de centro na origem do plano e raio 2 (daí a expressão " $a^2 + b^2 < 4$ ") no decorrer das 160 iterações, o ponto pertence ao conjunto e será representado pela cor branca (definida no programa quando " $n = -5.4$ "). Caso contrário, se $a^2 + b^2 \geq 4$, o procedimento se encerra, gerando outro valor para " n " e, portanto, determinando outra cor para a representação gráfica do ponto testado. Finalmente, enquanto nenhuma destas condições é atingida, o procedimento continua executando os algoritmos " $newb = 2.a.b + y$; $a = a^2 - b^2 + x$; $b = newb$ e $n = n + 1$ ", prosseguindo iterativamente o teste de cada um dos pontos da representação gráfica do *Conjunto de Mandelbrot*.

3.4 Fractais Gerados pelo “Jogo do Caos”

Conforme apresentado no primeiro capítulo deste trabalho, a construção de fractais pelo Jogo do Caos baseia-se matematicamente na aplicação iterada de um sistema de funções, onde tais funções realizam uma composição de transformações geométricas a um conjunto de pontos do plano. Este fato implica um processo de construção com características essencialmente algébricas, que, implícita ou explicitamente, devem fazer parte do software utilizado na construção destas figuras. Devido a isto, a construção desta família de fractais é viabilizada através dos ambientes MicroWorlds e Geometricks, visto que os demais ambientes analisados neste trabalho possuem um enfoque essencialmente geométrico em suas construções.

Dessa forma, utilizamos o MicroWorlds, que, por se basear na linguagem LOGO e na Geometria da Tartaruga, nos permite a tradução de funções algébricas em procedimentos de programação, assim como sua representação gráfica posterior, e o GeomeTricks, que possui uma ferramenta destinada especificamente à construção de fractais dessa família.

Estudamos detalhadamente a construção de dois fractais desta família, a versão para o jogo do caos do “*Triângulo de Sierpinski*” e a “*Samambaia Determinística*” (*Deterministic Fern*), ambos apresentados na obra que primeiro apresentou o Jogo do Caos (Barnsley, 1993, pp. 86-88).

3.4.1 Linguagem LOGO a partir do MicroWorlds

A forma mais usual de se apresentar o conjunto de transformações geométricas, necessária à construção dos fractais através do Jogo do Caos, se dá através de sua representação algébrica a partir de um Sistema de Funções Iteradas. Portanto, para tais construções, cabe-nos a tradução dessas funções para a linguagem LOGO e a incorporação, ao programa de construção, de procedimentos que possibilitem a representação gráfica dos pontos calculados pelas sucessivas iterações, realizadas pelas funções que compõe o sistema. Uma vez que a base matemática para este processo já foi apresentada, passamos agora para os detalhes de construção de fractais dessa família, especificamente através do MicroWorlds LOGO.

O desenvolvimento do programa de construção para estes fractais parte da determinação do sistema de funções que o define, seguido pela tradução desse sistema para a sintaxe da linguagem LOGO. A isso segue a incorporação dos procedimentos que executam o cálculo das coordenadas de cada um dos pontos da figura a ser construída, assim como a representação gráfica desses pontos na tela do computador. Partimos aqui do sistema de funções iteradas propriamente ditas, uma vez que o desenvolvimento matemático que dá origem a esse sistema já foi anteriormente descrito nesse trabalho.

3.4.1.1 Triângulo de Sierpinski

Nesta seção, pretendemos apresentar a versão para o jogo do caos do *Triângulo de Sierpinski* e construí-lo a partir do MicroWorlds. Para isso, partimos dos coeficientes já apresentados no início deste trabalho, retomando sua representação matricial e tradução para a forma de sistemas de funções iteradas, para então desenvolver o programa de construção deste fractal no ambiente.

Cabe-nos ressaltar que este fractal é, de fato, uma versão adaptada do fractal originalmente desenvolvido pelo processo “iniciador-gerador”, portando, apesar da aparência final das figuras, os processos de construção e os objetos matemáticos em jogo são completamente diferentes. Assim sendo, a representação matricial das funções iteradas para a construção do *Triângulo de Sierpinski* é a seguinte:

$$\begin{aligned} \mathbf{w}_1 \begin{pmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{pmatrix} &= \begin{pmatrix} 0,5 & 0 \\ 0 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \mathbf{w}_2 \begin{pmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{pmatrix} &= \begin{pmatrix} 0,5 & 0 \\ 0 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} + \begin{pmatrix} 0,5 \\ 0,5 \end{pmatrix} \\ \mathbf{w}_3 \begin{pmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{pmatrix} &= \begin{pmatrix} 0,5 & 0 \\ 0 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{aligned}$$

O sistema de funções iteradas para este fractal é obtido traduzindo-se a representação matricial acima para a representação algébrica tradicional. Com isso, determina-se um sistema com três pares de funções, como segue:

$$\begin{array}{lll}
 \text{(i)} & x_n = 0,5x & ; \quad y_n = 0,5y \\
 \text{(ii)} & x_n = 0,5x + 0,5 & ; \quad y_n = 0,5y + 0,5 \\
 \text{(iii)} & x_n = 0,5x + 1 & ; \quad y_n = 0,5y
 \end{array}$$

A partir desse sistema, desenvolvemos o programa³ que irá gerar o *Triângulo de Sierpinski* a partir dos procedimentos apresentados a seguir. Entretanto, cabe ressaltar que estes procedimentos foram concebidos para atender certas necessidades didáticas deste trabalho e, portanto, não representam nem a única, e provavelmente nem a mais simples, maneira de se construir este fractal.

```

to sierp :n
make "x 0 make "y 0
repeat :n [run pick [regra1 regra2 regra3] dot]
end

to dot
setx :x * 250 - 200 sety :y * 250 - 100
pd fd 0.1 bk 0.1 pu
end

to regra1
setc "red
make "x 0.5 * :x
make "y 0.5 * :y
end

to regra2
setc "green
make "x 0.5 * :x + 0.5
make "y 0.5 * :y + 0.5
end

to regra3
setc "blue
make "x 0.5 * :x + 1
make "y 0.5 * :y
end

```

A partir do programa desenvolvido acima, passamos a analisar as operações executadas em cada um de seus procedimentos.

O procedimento “**to sierp :n**” cria o comando de construção do *Triângulo de Sierpinski*, introduzindo o parâmetro “n” que representa o número de pontos a serem calculados e plotados pelo programa. Na segunda linha de comandos, o procedimento cria as variáveis x e y, atribuindo a ambas o valor zero. E, na terceira linha, repete “n” vezes os comandos “[run pick [regra1 regra2 regra3] dot]”.

Este último conjunto de comandos (linha 3) introduz o algoritmo determinístico essencial do Jogo do Caos, ao escolher e executar aleatoriamente,

³ Adaptado por nós a partir do original apresentado em “The Fractal Umbrella” (Scott & Beaumont, 2002, s.p.).

com os comandos “**run**” e “**pick**”, cada uma das três funções iteradas do sistema (traduzidas pelas três “regras” de cálculo). Além disso, este conjunto de comandos se encerra com o procedimento “**dot**”, que possui a função de plotar um ponto no plano a partir das coordenadas dadas.

Neste ponto, julgamos importante tecer algumas considerações acerca dos tipos de controle, disponibilizados pelos comandos desse primeiro procedimento de construção. Primeiramente, ressaltamos que a utilização dos comandos “**run**” e “**pick**”, que introduzem o componente aleatório à construção do fractal, são válidos somente para a construção de fractais com distribuição equiprovável de pontos, ou seja, quando cada função do sistema tiver a mesma probabilidade de ser aplicada. Em termos objetivos, o comando “**pick**” tem a função de escolher aleatoriamente o que for colocado entre colchetes a sua frente, ou seja, ou a “**regra1**” ou a “**regra2**” ou a “**regra3**”, e o comando “**run**” executa esta escolha na forma de procedimento. Feito isso, um novo par de coordenadas é calculado e armazenado na memória interna do programa, o qual encerra uma iteração ao plotar o ponto resultante das coordenadas calculadas, com o procedimento “**dot**”.

O procedimento “**to dot**” tem, como dissemos, a função de plotar cada um dos pontos calculados pelo algoritmo do programa. Porém, isso é realizado com o auxílio de certos artifícios, que adicionam a este procedimento uma parte do controle fornecido pelo ambiente para a construção de fractais, pois é a partir deste procedimento que se definem a posição e as dimensões da figura na tela do computador.

Este controle é fundamental para o resultado gráfico apresentado na tela do computador, no sentido de ajustar o valor numérico das coordenadas calculadas pelas funções do sistema às dimensões do modelo cartesiano adotado pela interface gráfica do MicroWorlds. Assim, são utilizados os comandos “**setx :x * 250 – 200**” e “**sety :y * 250 – 100**” que definem, a partir das coordenadas calculadas pelas funções do sistema, a posição final dos pontos efetivamente plotados na tela, num processo que pode ser assim resumido:

- a partir das operações realizadas pelo programa, define-se um ponto de coordenadas (x , y), cujos valores são armazenados na memória do programa;
- os comandos “**setx**” e “**sety**” enviam a tartaruga respectivamente para as ordenadas x e y do sistema de coordenadas do MicroWorlds;

- estas ordenadas são determinadas pelos os valores $250x - 200$, para abscissas, e $250y - 100$, para ordenadas, definidos em função das coordenadas originais calculadas pelo programa;

- o coeficiente que multiplica as coordenadas (neste caso, 250) determina as dimensões do fractal desenhado na tela

- os coeficientes somados (neste caso, -200 para x e -100 para y) definem a origem relativa a partir da qual o fractal será construído, ou seja, o ponto $(0, 0)$ para o sistema de funções iteradas. Neste caso, a origem relativa para a construção do fractal será de 200 “passos” à esquerda e 100 “passos” abaixo da origem adotada na tela do MicroWorlds;

- com a tartaruga posicionada nas coordenadas, o procedimento desenha um pequeno ponto na tela com os comandos “`pd fd 0.1 bk 0.1 pu`”.

Os procedimentos “**to regra1**” até “**to regra3**” representam as funções iteradas do sistema, devidamente traduzidas para a sintaxe da linguagem LOGO, e têm por objetivo calcular as coordenadas de um novo ponto a partir dos valores das coordenadas do ponto anterior e armazená-los na memória para a próxima recursão do programa.

Nestes procedimentos, adicionamos um outro dispositivo de controle que julgamos interessante sob o ponto de vista cognitivo, ou seja, a partir do comando “**setc**”, impomos diferentes cores aos pontos calculados em cada uma das regras. Este recurso pode criar um “feedback” interessante, ao mostrar graficamente quais pontos foram efetivamente produzidos por cada uma das funções do sistema, tornando tanto o processo de construção desse fractal, quanto as relações entre as funções do sistema e os pontos produzidos, mais visuais e acessíveis, se comparadas com uma construção realizada com somente uma cor.

O resultado gráfico deste programa é apresentado na figura 3.6, na qual foram utilizados 50.000 pontos, para uma visualização apropriada. Nesta figura, os pontos produzidos pela regra 1 foram plotados de vermelho, os pontos da regra 2 de verde e os pontos da regra 3 de azul.

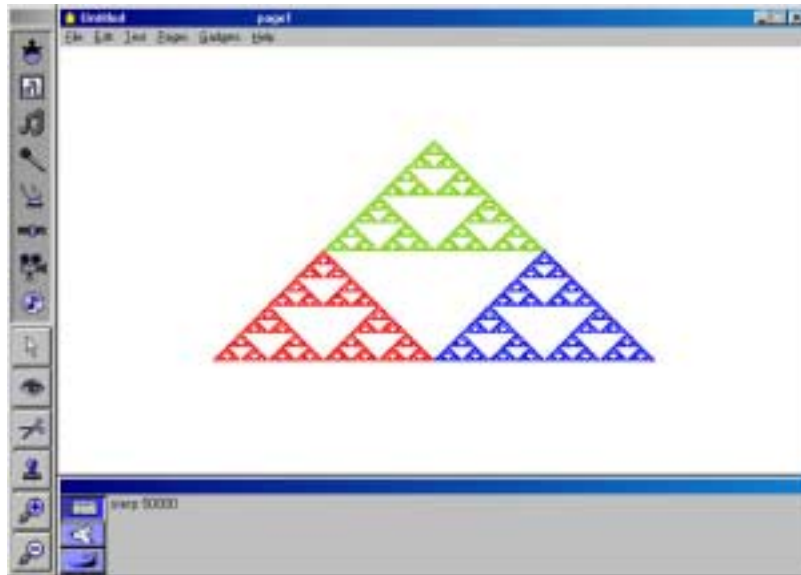


Figura 3.6: Construção do Triângulo de Sierpinski a partir do software MicroWorlds LOGO.

Finalmente, apresentamos o resumo das operações realizadas pelo programa de construção do *Triângulo de Sierpinski*, assim como a estrutura iterativa de construção desse fractal, sintetizadas nas seguintes operações:

- repetir “n” vezes os comandos “[run pick [regra1 regra2 regra3] dot]”;
- em cada repetição, escolher aleatoriamente e executar um dos procedimentos “[regra1 regra2 regra3]”;
- com a “regra” escolhida, calcular as coordenadas (x , y) de um novo ponto a partir das coordenadas armazenadas na iteração anterior, assim, na primeira iteração do programa, atribuem-se para x e y as coordenadas (0 , 0), calcula-se um novo par de coordenadas e armazenam-se estes novos valores para a próxima iteração do programa;
- o ponto calculado é plotado na construção com o procedimento “dot” e o processo é novamente iniciado.

3.4.1.2 Samambaia determinística

Como vimos no primeiro capítulo deste trabalho, a construção da *Samambaia Determinística* parte do sistema com quatro funções iteradas apresentado abaixo. Ressaltamos que a escolha destas funções pelo algoritmo do Jogo do Caos não resultará em uma distribuição equiprovável de pontos,

portanto, devemos ter em mente que a função (i) desse sistema deve ser aplicada com probabilidade de 1%; a função (ii) com probabilidade de 85% e as funções (iii) e (iv) com probabilidade de 7% cada uma.

$$\begin{array}{ll}
 \text{(i)} & x_n = 0 & ; & y_n = 0,16y \\
 \text{(ii)} & x_n = 0,85x + 0,04y & ; & y_n = -0,04x + 0,85y + 0,16 \\
 \text{(iii)} & x_n = 0,2x - 0,26y & ; & y_n = 0,23x + 0,22y + 0,16 \\
 \text{(iv)} & x_n = -0,15x + 0,28y & ; & y_n = 0,26x + 0,24y + 0,08
 \end{array}$$

Seguindo a mesma mesma estrutura básica de procedimentos que nos permitiu a construção do *Triângulo de Sierpinski* e a partir do sistema apresentado acima, desenvolvemos a seguir o programa de construção da *Samambaia Determinística* para o Jogo do Caos.

```

to fern :n
  pu make "x 0 make "y 0
  repeat :n [EscolherRegra dot]
  end

to EscolherRegra
  make "r random 100
  if :r = 0 [regra1 stop]
  if :r < 86 [regra2 stop]
  if :r < 94 [regra3 stop]
  regra4
  end

to dot
  setx :x * 300 sety :y * 300 - 150
  pd fd 0.1 bk 0.1 pu
  end

to regra1
  setc "black
  make "x 0 make "y 0.16 * :y
  end

to regra2
  setc "green
  make "xn 0.85 * :x + 0.04 * :y
  make "y -0.04 * :x + 0.85 * :y + 0.16
  make "x :xn
  end

to regra3
  setc "blue
  make "xn 0.2 * :x - 0.26 * :y
  make "y 0.23 * :x + 0.22 * :y + 0.16
  make "x :xn
  end

to regra4
  setc "red
  make "xn -0.15 * :x + 0.28 * :y
  make "y 0.26 * :x + 0.24 * :y + 0.08
  make "x :xn
  end

```

Como dissemos, a estrutura deste programa é basicamente a mesma do anterior, portanto iremos focar nossas análises nas principais diferenças entre os dois programas, visto que diversos comandos aqui utilizados já foram analisados na seção anterior. Assim, a principal diferença entre os dois fractais repousa no fato do algoritmo determinístico para a *Samambaia Determinística* resultar em uma distribuição não-equiprovável de pontos, o que impõe a criação de um procedimento diferente do anterior para a aplicação das funções do sistema.

As operações executadas pelos procedimentos desse programa podem ser resumidas com o procedimento “**to fern :n**”, que cria o comando de construção da *Samambaia Determinística* e segue a mesma estrutura do procedimento de criação do *Triângulo de Sierpinski*. Neste procedimento é introduzido o comando “EscolherRegra”, que escolhe aleatoriamente uma das “regras” existentes no programa e a executa, calculando um novo par de coordenadas (x , y) para o procedimento “dot”, porém impondo ao programa as diferentes probabilidades de aplicação de cada uma das “regras”.

Dessa forma, o procedimento “**to EscolherRegra**” cria o algoritmo determinístico do Jogo do Caos, a partir do qual são escolhidas e executadas as funções iteradas do sistema, segundo as probabilidades impostas para cada uma delas. Para incorporar as diferentes probabilidades ao programa, este procedimento se inicia com a criação do parâmetro “r” (linha 2) e atribui a “r” um valor aleatório (definido pelo programa) entre 0 e 100. Feito isso, o procedimento “escolhe” qual das quatro funções (regras) será aplicada às coordenadas da figura. Em síntese, o procedimento executa as seguintes operações:

- se $r = 0$ (o que ocorre com 1% de chance), aplica-se a regra 1;
- se $0 < r < 86$ (o que ocorre com 85% de chance), aplica-se a regra 2;
- se $86 \leq r < 94$ (o que ocorre com 7% de chance), aplica-se a regra 3;
- se $94 \leq r \leq 100$ (o que ocorre com 7% de chance), aplica-se a regra 4.

Os procedimentos “**to regra1**” até “**to regra4**” representam as funções iteradas do sistema, devidamente traduzidas para a sintaxe da linguagem LOGO, e têm por objetivo calcular as coordenadas de um novo ponto a partir dos valores das coordenadas do ponto anterior e registrá-los na memória para a próxima iteração do programa. Em alguns destes procedimentos (regras 2, 3 e 4) foi criada a variável “xn” para auxiliar nos cálculos dos pontos e sua função decorre de características informáticas do ambiente.

De fato, a criação da variável “xn” é um artifício informático de cálculo necessário quando as funções iteradas do sistema utilizam simultaneamente os valores de x e y em seus cálculos e justifica-se pelo fato das funções do sistema utilizarem os valores de coordenadas calculadas anteriormente e armazenadas na memória do programa. Assim, em cada iteração do programa, os procedimentos relativos às funções acessam as variáveis x e y armazenadas na memória,

calculam novos valores para estas variáveis e registram estes novos valores no lugar dos anteriores, para então dar prosseguimento ao programa.

Com isso, sem o artifício “xn”, cria-se uma incoerência lógica na construção quando o comando de cálculo da variável x (aplicado antes da variável y) calcular e registrar um novo valor de x, sendo este valor novamente acessado nos cálculos da variável y. Então, com a variável “xn”, temos a criação, cálculo e armazenamento de um novo valor na memória, não alterando o valor da variável x, que será utilizada corretamente nos cálculos da variável y. Finalmente, o procedimento se encerra com o comando “make x :xn” que atribui e registra o novo valor de xn na variável x.

O resultado gráfico deste programa é apresentado na figura 3.7, na qual foram utilizados 50.000 pontos para uma visualização apropriada.



Figura 3.7: Construção da Samambaia Determinística a partir do software MicroWorlds.

Para esta figura, também utilizamos cores diferentes para os pontos calculados por cada uma das regras, dessa forma, os pontos coloridos de preto representam a regra 1 (“caule” da samambaia), os pontos verdes representam a regra 2, os pontos azuis representam a regra 3 e os pontos vermelhos representam a regra 4.

3.4.2 Geometria Dinâmica a partir do GeomeTricks

Já vimos que os fractais baseados no Jogo do Caos foram construídos originalmente a partir de um Sistema de Funções Iteradas, onde cada função do sistema define uma composição de transformações afins no plano. Até aqui, esse sistema de funções foi apresentado em sua forma algébrica, conforme ilustrado com o MicroWorlds, porém esta não é a única forma de representar transformações afins.

Uma vez que o objetivo do sistema de funções iteradas é, basicamente, o de promover um conjunto de transformações afins nos pontos do \mathbb{R}^2 , podemos defini-las por meio da aplicação de certas funções, em diferentes sistemas de coordenadas do plano. Dessa forma, é possível criar um outro processo de construção para os fractais do Jogo do Caos, sem a utilização de representações algébricas, substituindo-as por um processo mais visual e intuitivo, baseado na representação geométrica das transformações, e é exatamente isso que o GeomeTricks se propõe a fazer.

É importante ressaltar que os processos de construção de fractais com o GeomeTricks e com o MicroWorlds são similares do ponto de vista teórico, visto que ambos se baseiam na definição de um conjunto de transformações afins no plano. Porém, as representações matemáticas em jogo, a forma como as informações de entrada (inputs) são usadas e a interface de ambos os softwares são bem diferentes. Segundo o criador do GeomeTricks, Prof. Viggo Sadolin (1998b, p. 1), o objetivo dessa nova forma de representação do Jogo do Caos visa uma abordagem menos abstrata das transformações afins, que pudesse ser explorada por alunos mais jovens e não somente por estudantes universitários, a partir de uma visão mais geométrica, experimental e lúdica do processo.

Neste processo de construção, define-se um sistema de coordenadas a partir de uma terna de pontos não colineares. Dessa forma, definido um sistema de coordenadas ABC, por exemplo, qualquer ponto no plano fica determinado por um, e somente um, par ordenado de coordenadas nesse sistema. Visto que essa terna de pontos define um sistema de coordenadas, a posição e a ordem

escolhida para cada ponto da terna é fundamental no processo de determinação do sistema.

Segundo Sadolin (1998b, pp. 1-2), a partir desse sistema de coordenadas inicial, define-se um outro sistema de coordenadas XYZ, utilizando uma segunda terna de pontos não colineares. A relação entre esses dois sistemas de coordenadas determina uma função biunívoca T do plano euclidiano nele próprio ($\mathbb{R}^2 \rightarrow \mathbb{R}^2$), ou seja, para qualquer ponto P do plano, representado no sistema de coordenadas ABC, teremos um único ponto $T(P)$, representado no sistema XYZ. Utilizaremos a notação $T: (ABC) \rightarrow (XYZ)$ para representar uma função assim definida.

A partir da função $T: (ABC) \rightarrow (XYZ)$, pode-se visualizar uma transformação afim ao considerarmos os sistemas de coordenadas utilizados como triângulos, ou seja, ao se definir o sistema de coordenadas ABC, defini-se também um triângulo de vértices A, B e C. Dessa forma, considerando o conjunto ΔABC , dos pontos internos do ΔABC , uma dada transformação afim poderá ser visualizada a partir da imagem do conjunto ΔXYZ obtida pela função T , isto é, transformando os pontos internos do ΔABC , no conjunto dos pontos internos do ΔXYZ , onde $X = T(A)$; $Y = T(B)$; $Z = T(C)$ e $T(\cdot)$ são os pontos internos do ΔXYZ .

Com isso, a ferramenta de construção de fractais disponibilizada no GeomeTricks fornece também um mecanismo para definir e visualizar transformações afins no plano, a partir da construção das imagens da função T , definida por duas ternas de pontos. Esta ferramenta é obtida na opção “Níveis” que, dessa forma, assume uma dupla função no ambiente, pois ela é utilizada tanto para construir os diferentes níveis de iteração de um fractal, como para representar visualmente a imagem de uma transformação afim em particular. Esta transformação, dependendo das ternas escolhidas, pode representar tanto uma única transformação como uma composta de transformações quaisquer.

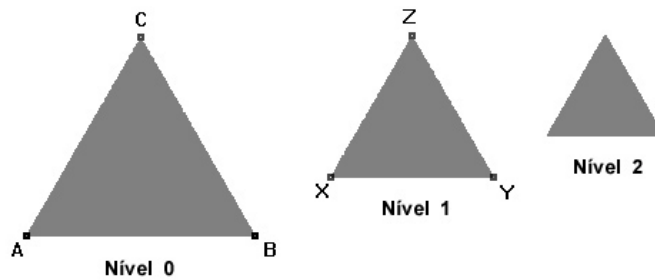


Figura 3.8: Níveis de iteração do triângulo ABC, transformado a partir do GeomeTricks.

Nesta ferramenta, as transformações afins são aplicadas iterativamente pelo software, assim, dada uma transformação $T: (ABC) \rightarrow (XYZ)$, por exemplo, no “*nível 0*” de iteração, o software gera o conjunto dos pontos internos do ΔABC (portanto, sem aplicar a transformação). No “*nível 1*” de iteração, o software aplica a transformação T ao conjunto, cuja imagem é o ΔXYZ . Dando continuidade ao processo, no “*nível 2*” é aplicada a transformação $T(T(\quad))$, gerando a imagem do ΔXYZ por T , construindo um novo triângulo transformado. Conseqüentemente, em cada nível de iteração, a transformação T será aplicada aos pontos internos do triângulo gerado pelo nível imediatamente anterior.

3.4.2.1 Construção de fractais

Uma vez que as transformações afins utilizadas para a construção de fractais no GeomeTricks são baseadas em sistemas de coordenadas, a posição dos pontos que as definem e a ordem como esses pontos são selecionados têm a mesma importância para o sucesso da construção que os parâmetros utilizados no sistema de funções algébricas, utilizadas no MicroWorlds.

Devemos lembrar que o GeomeTricks é um software baseado na geometria dinâmica, portanto a criação e o posicionamento inicial dos pontos que geram o fractal é feito via construções geométricas e manipulação direta. Dessa forma, caso se queira efetivamente controlar os resultados obtidos, a criação e o posicionamento de pontos não deve ser feito de forma aleatória. Tal controle engloba a construção e o posicionamento desses pontos, assim como a determinação da ordem das ternas que definem o fractal desejado.

Em síntese, a interface do GeomeTricks para a determinação de fractais impõe o seguinte processo de construção:

- define-se um conjunto de pontos no plano, a partir das ferramentas de construção geométrica do software. Esses pontos serão utilizados para a determinação das ternas que definem os sistemas de coordenadas e, conseqüentemente, as transformações afins do sistema de funções iteradas do fractal desejado.

- a primeira terna de pontos selecionada gera o que chamamos de “triângulo de controle” e cada uma das ternas seguintes define uma transformação afim composta, que será aplicada ao triângulo de controle em função da posição e ordem em que cada ponto dessas ternas é selecionado.

- o GeomeTricks conta com a opção “Níveis”, que não foi disponibilizada nos procedimentos desenvolvidos no MicroWorlds, permitindo ao usuário verificar visualmente o resultado gráfico de todas as transformações executadas pelo programa, através da construção de diferentes níveis de iteração. Como dissemos, essas transformações são aplicadas iterativamente pelo programa, ou seja, no “*nível 1*” de iteração, as transformações são aplicadas no triângulo de controle; no “*nível 2*”, em todos os triângulos gerados pelo “*nível 1*” e em cada nível subsequente, nos triângulos gerados pelo nível imediatamente anterior.

- como o fractal é obtido por um nível de iteração que tende ao infinito, o GeomeTricks possui a opção “Desenhar fractal”, que constrói o fractal propriamente dito.

Apresentamos a seguir dois exemplos de fractais construídos com o GeomeTricks, assim como a explicitação do processo para defini-los. Seguindo a mesma ordem utilizada nos exemplos apresentados no MicroWorlds, construiremos o *Triângulo de Sierpinski* e a *Samambaia Determinística*.

3.4.2.2 Triângulo de Sierpinski

Este fractal pode ser construído a partir de seis pontos (A até F), onde os três primeiros representam os vértices de um triângulo ABC e os demais representam os pontos médios dos lados desse triângulo. Neste exemplo, construímos um triângulo equilátero, porém cabe ressaltar que poderíamos partir de um triângulo qualquer, considerando, em seguida, os pontos médios de seus lados.

A partir do posicionamento destes pontos, prosseguimos com a determinação das transformações afins que definem o fractal e, uma vez que o *Triângulo de Sierpinski* é definido a partir de um sistema composto por três funções, faz-se necessário traduzir tais funções para um conjunto de transformações, determinadas a partir de um de sistemas de coordenadas no plano. Dessa forma, definimos as seguintes transformações, que podem ser visualizadas no “nível 1” de iterações dado na figura 3.9:

$T_1: (ABC) \rightarrow (ADF)$, que leva o triângulo ABC no triângulo ADF;

$T_2: (ABC) \rightarrow (FEC)$, que leva o triângulo ABC no triângulo FEC;

$T_3: (ABC) \rightarrow (DBE)$, que leva o triângulo ABC no triângulo DBE;

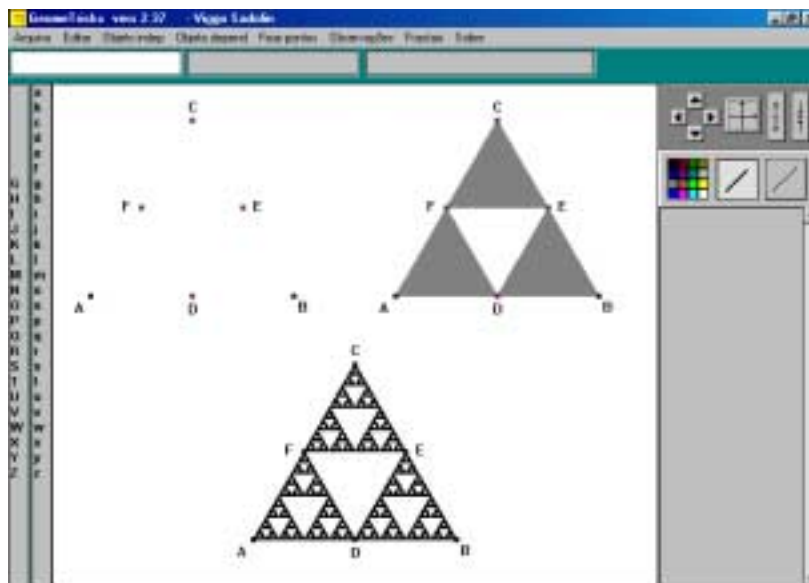


Figura 3.9: Tela do GeomeTricks com diferentes níveis de iteração para o triângulo de Sierpinski.

Para a determinação das ternas que irão definir os sistemas de coordenadas deve-se ter em mente que para “n” ternas, serão definidas sempre “n – 1” transformações. Assim, o *Triângulo de Sierpinski*, deve ser definido a partir de 4 ternas, selecionadas na seguinte ordem:

- ABC; ADF, FEC e DBE.

A figura 3.9 reproduz a tela do GeomeTricks (adaptada para comportar três construções diferentes) e nela são apresentados os pontos iniciais da construção; o “nível 1” de iterações, que representa visualmente a imagem do sistema de transformações realizadas; e o resultado final da construção do *Triângulo de Sierpinski*. Devemos ressaltar que todas essas construções foram geradas no GeomeTricks, porém em momentos distintos que posteriormente foram “editados” na confecção da figura final apresentada.

Como dissemos, todas as transformações são inicialmente aplicadas no “triângulo de controle” ABC, que representa o sistema de coordenadas inicial onde são aplicadas todas as demais transformações. No decorrer do processo iterativo, as transformações são aplicadas em todos os “novos” triângulos gerados pelo nível de iteração imediatamente anterior. Assim, na primeira iteração (nível 1) são construídos os triângulos ADF, FEC e DBE; na segunda iteração, as três transformações são aplicadas em cada um desses triângulos, gerando nove triângulos transformados, que se transformam em 27 novos triângulos na terceira iteração.

3.4.2.3 Samambaia determinística

Este fractal será construído a partir de doze pontos (A até L), conforme ilustra a fig. 3.10 (a). Devido a extrema dependência da posição inicial dos pontos que irão defini-lo, os pontos foram posicionados e ajustados manualmente a partir de procedimentos de tentativa e erro pois, ao contrário do *Triângulo de Sierpinski*, a *Samambaia Determinística* parte de um sistema bem mais complexo de transformações (que inclui, entre outras, compressões verticais e horizontais).

Este componente dificulta a associação do posicionamento dos pontos com um processo simples de construção geométrica, assim, o resultado da construção da *Samambaia Determinística* não será exatamente igual à realizada com o MicroWorlds. No entanto, cabe ressaltar que este fato não é regra, sendo específico a alguns fractais, visto que no *Triângulo de Sierpinski*, assim como em várias outras construções, obtivemos resultados idênticos com ambos os processos de construção, visto que o posicionamento inicial dos pontos foi obtido a partir de suas efetivas construções geométricas.

As três funções necessárias à construção do fractal são apresentadas abaixo e podem ser visualizadas na fig. 3.10 (b), onde ressaltamos os triângulos determinados pelas funções:

$T_1: (ABC) \rightarrow (DEF)$, que leva o triângulo ABC no triângulo DEF;

$T_2: (ABC) \rightarrow (GHI)$, que leva o triângulo ABC no triângulo GHI;

$T_3: (ABC) \rightarrow (JKL)$, que leva o triângulo ABC no triângulo JKL;

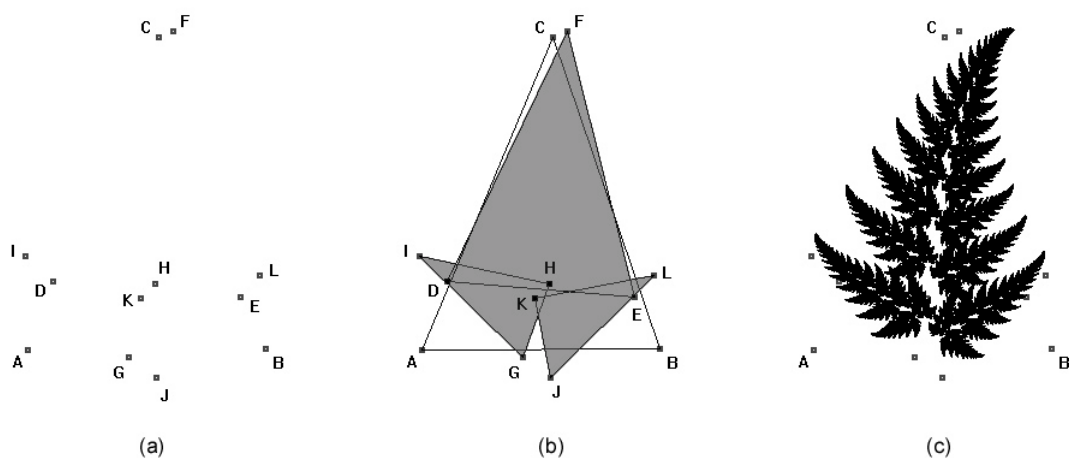


Figura 3.10: Processo de construção da Samambaia Determinística a partir do geomeTricks.

Utilizamos apenas três funções para definir a *Samambaia Determinística*, ao contrário das quatro utilizadas para defini-la com o MicroWorlds. Isto deve-se ao fato de não termos incluído, neste fractal, a função que representa o “caule” da samambaia. Portanto, serão utilizadas apenas três funções para a construção desse fractal, introduzidas no software a partir das quatro ternas dadas abaixo:

- ABC; DEF, GHI e JKL.

O resultado final da construção da *Samambaia Determinística* com o GeomeTricks é dado na fig. 3.10 (c). Nessa figura retiramos parte das letras que denominam os pontos iniciais do fractal no sentido de possibilitar uma melhor visualização do resultado final da construção, mantendo somente as letras relativas ao triângulo de controle, ABC.

CAPÍTULO 4

TRANSFORMAÇÕES GEOMÉTRICAS A PARTIR DO JOGO DO CAOS

4.1 Considerações Iniciais

O objetivo principal deste capítulo resume-se na apresentação de uma proposta focada na modelização de situações didáticas, visando o estudo das Transformações Geométricas no Plano Euclidiano. Esta proposta parte da construção de objetos fractais gerados pelo Jogo do Caos, construídos com o auxílio de dois ambientes informáticos distintos, o GeomeTricks e o MicroWorlds LOGO. Nesta perspectiva, pretendemos explicitar o que consideramos como um dos resultados obtidos nesta pesquisa em termos de aplicações didáticas, a partir da análise de um conjunto de atividades de construção de fractais em ambientes informáticos.

Nesta proposta, a Geometria Fractal possui um papel central – ela aglutina e inter-relaciona os diferentes objetos e conceitos que estamos explorando. Este fato se justifica de uma forma simples, pois, em primeiro lugar, seria quase impossível abordar o Jogo do Caos sem, de alguma forma, explorar as transformações geométricas, visto que são essencialmente nelas que a criação do processo de construção de fractais pelo Jogo do Caos se baseia. Em segundo lugar, a construção de fractais, dada a sua própria natureza, nos remete quase que obrigatoriamente à utilização de computadores.

Com isso, retomamos neste capítulo a última dimensão de análise do *domínio de validade epistemológica* de um micro-mundo, ou seja, o conjunto de problemas que cada micro-mundo permite propor. Entretanto, dada a abrangência deste tópico – visto que definir o conjunto de situações ou problemas que dão sentido a um conceito não é tarefa simples – focaremos nossas análises em uma classe específica de problemas, envolvendo o conceito de Transformações Geométricas no Plano, contextualizado a partir de aplicações do Jogo do Caos.

4.1.1 Pressupostos Iniciais e Escolhas de Âmbito Global

Visto que nosso enfoque será o estudo das Transformações Geométricas no Plano, nos apoiaremos em um conjunto de pressupostos, fundamentados em diversas pesquisas, para desenvolver as situações didáticas. Tais pressupostos decorrem tanto de componentes matemáticos, relativos aos objetos do saber com os quais lidamos, quanto informáticos, determinados pelas características específicas dos ambientes utilizados. Assim sendo, tentaremos explicitar as principais escolhas realizadas no desenvolvimento das situações didáticas, assim como os motivos que as justificam.

O primeiro pressuposto refere-se ao enfoque dado à resolução de problemas no ensino e aprendizagem da Matemática. Nesse sentido, entendemos que uma estratégia efetiva de modelização demanda a construção de situações problema, que, por um lado, utilizem efetivamente os conceitos que se deseja tratar e, por outro lado, convidem e desafiem o aprendiz a tomar tais problemas como seus. Para isso, adotamos diversas concepções presentes na Teoria das Situações Didáticas (Brousseau, 1986) que, em termos simples, supõe a modelização de situações didáticas que podem ser interpretadas como “perturbadoras”, ao exigir dos sujeitos novas formas de interpretação e descrição dos conceitos em jogo. Assim, adotamos esta concepção principalmente no sentido de atribuir às atividades alguns componentes “a-didáticos”. E ainda, adotamos idéias relativas à noção de “milieu” (Brousseau, 1988), que se atém essencialmente à interação do sujeito com o meio que, neste caso, enfoca principalmente as interações dos sujeitos com ambientes informáticos.

O segundo pressuposto refere-se a um conjunto de recomendações, no sentido de uma maior contextualização do processo de ensino e aprendizagem das transformações como um todo. Muitos dos trabalhos nos quais nos inspiramos referem-se a pesquisas no campo da Álgebra Linear (Dorier, 2000), porém, dados os objetivos estabelecidos nessas pesquisas, consideramos que a aplicação dessas recomendações em nossa abordagem se justifica.

Nessa linha, inspiramo-nos em três idéias principais: a primeira enfoca a suposição de que diversos conceitos presentes no ensino e aprendizagem da Álgebra Linear – inclusive as transformações – são, em geral, estanques e não correlacionados (Dorier, 1997; 1998). A segunda aborda a grande dificuldade apresentada pelos estudantes quanto ao formalismo e abstração exigidos por esses conceitos, onde podemos citar trabalhos que chegam a considerá-los como um “*obstáculo ao formalismo*”¹. A terceira refere-se às deficiências quanto às inter-relações das diferentes representações semióticas, presentes no ensino das Transformações Lineares e Afins², fazendo com que as possíveis relações e articulações entre as diferentes representações de um mesmo objeto matemático não sejam, em geral, concretamente identificadas e tratadas pelos estudantes.

Nesse contexto, a proposta de criação de situações didáticas que desenvolvemos, parte da resolução de problemas envolvendo o conceito de Transformações Afins no Plano, contextualizadas com a construção de objetos fractais. Com isso, projetamos as atividades de modo que a necessidade do uso de múltiplas representações destes conceitos surja na resolução de problemas, a partir da utilização de diferentes ferramentas informáticas.

Além disso, partimos do pressuposto informático fundamental de que os significados susceptíveis de serem construídos, a partir de diferentes ambientes, na construção de um mesmo objeto são efetivamente distintos. Assim, a utilização de ambientes distintos, para a construção de um mesmo fractal, implica aspectos

¹ Ressaltamos as pesquisas de J.-L. Dorier, A. Robert, J. Robinet e M. Rogalski (Dorier, 1998), além de nos inspirarmos na proposta de Ghershon Harel sobre os princípios de concretização, necessidade e generalização, presentes na reforma do ensino de Álgebra Linear proposta por ele nos EUA (Harel, 2000).

² Ressaltamos os trabalhos de Hillel e Sierpinski (1995) sobre a necessidade de conversão entre representações de transformações lineares e de Marlene Alves Dias (1998) que propõe a exploração da articulação de diferentes quadros e representações para espaços de pequena dimensão.

e representações diferentes para esse fractal. Esse pressuposto é o ponto de partida para a presente proposta, a qual baseia-se na utilização de dois softwares distintos, o *GeomeTricks* e o *MicroWorlds LOGO*.

A escolha desses ambientes informáticos se deu devido ao fato de ambos possibilitarem a construção dos mesmos fractais a partir de processos de construção diferentes, ou seja, o *GeomeTricks* explora principalmente a representação geométrica das transformações, onde são aplicadas composições de transformações geométricas definidas por conjuntos de ternas de pontos e, por sua vez, o *MicroWorlds LOGO* utiliza estas mesmas transformações a partir de sistemas de funções iteradas, adotando a representação algébrica das transformações. Pois bem, ao invés de simplesmente caracterizar esta diferença, pretendemos utilizá-la didaticamente no desenvolvimento de situações de ensino.

4.1.2 Concepções Teórico-Methodológicas das Situações de Ensino

Apesar de utilizarmos, no corpo deste trabalho, uma fundamentação que privilegie principalmente concepções teóricas relacionadas à noção de Transposição Informática (Balacheff, 1994; 1998), esta proposta de criação de situações didáticas foi concebida e fortemente influenciada por diversos aspectos da Teoria das Situações e da noção de “Milieu” (Brousseau, 1986; 1988). Além disso, a análise da seqüência de atividades proposta segue certas concepções desta teoria, assim como algumas abordagens inspiradas na Engenharia Didática (Artigue, 1988), principalmente no que diz respeito à “*análise a priori*” de situações de ensino e em considerações acerca das “*variáveis didáticas*”, utilizadas para embasar e justificar diversas escolhas, realizadas na análise desta seqüência de atividades.

Dessa forma, o foco de nossas análises didáticas concentra-se em um conjunto de atividades que visa explorar as relações existentes entre as representações geométricas e algébricas das transformações lineares e afins, sendo destinadas a estudantes universitários, dos cursos de graduação em Matemática, Ciências da Computação ou afins, nos quais o estudo das transformações geométricas no plano e da Álgebra Linear seja objeto de seus currículos.

A proposta de situações didáticas parte do desenvolvimento de uma seqüência de atividades dividida em duas partes: a primeira dedicada à construção de fractais a partir do ambiente *GeomeTricks* e a segunda propondo a utilização do ambiente *MicroWorlds LOGO*.

As atividades que utilizam o *GeomeTricks* iniciam-se com um exemplo propondo a construção do “Triângulo de Sierpinski”, seguido de outras duas atividades que propõem, respectivamente, a construção da “Cauda do Dragão” e do “Piso Fractal”. Por sua vez, a parte que utiliza o *MicroWorlds LOGO* é composta por duas atividades, propondo a construção do “Piso Fractal” e da “Cauda do Dragão”, respectivamente.

Para uma melhor compreensão dos conceitos matemáticos envolvidos nestas atividades, desenvolvemos uma síntese das principais definições e características a respeito das Transformações Geométricas no Plano no Anexo I deste trabalho. Além disso, as “fichas de atividade” para esta proposta de situações didáticas podem ser encontradas no Anexo II.

4.2 Análise e Desenvolvimento da Seqüência de Atividades

Nesta seqüência de atividades, desenvolvemos a construção de três fractais: o “Triângulo de Sierpinski”, a “Cauda do Dragão” e o “Piso Fractal”. Esses fractais, assim como a ordem na qual os apresentamos, foram escolhidos devido ao fato de possibilitarem representações idênticas, tanto no *GeomeTricks* quanto no *MicroWorlds*, e por apresentarem diferentes “níveis de dificuldade” de construção, explicitando os principais objetivos da seqüência com relação às propostas didáticas desenvolvidas.

Além disso, destacamos que estas escolhas também foram condicionadas pelo fato do “Triângulo de Sierpinski” se tratar de uma das figuras mais conhecidas e clássicas da geometria fractal. Por sua vez, os demais fractais – a “Cauda do Dragão” e o “Piso Fractal” – foram escolhidos devido a uma característica importante, do ponto de vista da geometria fractal: são dois fractais totalmente diferentes na aparência, mas praticamente idênticos quanto à forma de construção, e esta característica é extremamente representativa para uma introdução, ao menos intuitiva, tanto à Geometria Fractal como à Teoria do Caos.

4.2.1 Parte I: Fractais no GeomeTricks

Nesta primeira parte da seqüência de atividades, pretendemos abordar a construção de fractais a partir de um conjunto de representações geométricas das transformações lineares e afins, visando desenvolver uma primeira abordagem visual e intuitiva desses conceitos. Com essa proposta, pretende-se exercitar a visualização das transformações utilizadas, focando o desenvolvimento de uma noção intuitiva e geométrica do comportamento das transformações lineares e afins, baseada na observação do “movimento” das figuras transformadas.

Nessa linha, julgamos relevante explicitar algumas escolhas específicas a respeito da forma como propomos estas atividades. A primeira delas relaciona-se à construção geométrica dos pontos necessários à execução das atividades, pois seria possível propor aos sujeitos que construíssem os conjuntos de ternas necessárias para a construção dos fractais. Entretanto, uma vez que essas atividades visam uma introdução à construção e visualização de fractais a partir

de transformações geométricas, e não à exploração de construções geométricas em ambientes informáticos, optamos por fornecer um arquivo contendo os pontos necessários para as atividades, sem maiores comentários sobre seu posicionamento e construção geométrica.

Apesar disso, ressaltamos enfaticamente a importância do correto posicionamento dos pontos para a definição das composições de transformações geométricas. Essa escolha também garante que não surgirão problemas quanto ao posicionamento correto dos pontos e quanto à manipulação das ferramentas geométricas do software.

Entre outras escolhas, optamos por mantermos a atividade “aberta”, não impondo uma única estratégia possível para a sua execução. Com isso, é possível prever que alguns sujeitos poderão adotar estratégias de “tentativa e erro”, principalmente para a atividade I, podendo-se antecipar, neste caso, duas possibilidades na adoção dessa estratégia: na pior delas, os sujeitos descartam as instruções no sentido de observar as transformações e realizam as atividades por combinação de ternas, até chegar ao resultado desejado; e na melhor delas, os sujeitos observam a existência e a importância das transformações geométricas mas não se empenham na descrição formal destas.

Nesta atividade, as estratégias por tentativa podem ter resultados satisfatórios para os sujeitos, ou seja, mesmo adotando tais estratégias, é possível aos mesmos alcançarem os objetivos propostos. Entretanto, como esse comportamento não é o desejável para estas atividades, concebemos a atividade II de forma a dificultar o sucesso com a adoção deste tipo de estratégia, a partir da utilização das composições de transformações geométricas como informação inicial, na qual acreditamos que somente uma estratégia que privilegie a observação atenta das transformações geométricas envolvidas poderá chegar a um resultado satisfatório.

4.2.1.1 Exemplo: construindo o “triângulo de Sierpinski”

Um dos principais problemas existentes na concepção desta seqüência se refere à utilização de diferentes ambientes informáticos, uma vez que não é possível garantir que os sujeitos, aos quais a seqüência se destina, possuam efetivamente o grau de familiaridade com os ambientes, necessário para a execução das atividades.

➤ **Exemplo : Construindo o “Triângulo de Sierpinski”**

1) Abra o arquivo “**sierp.tri**” a partir do *GeomeTricks*.
Neste arquivo, os pontos A, B e C representam o ΔABC (chamado de “*triângulo de controle*”). Os pontos D, E e F são os pontos médios de cada um dos lados do triângulo de controle **ABC** (Fig 1a).




Fig. 1: Conjunto dos pontos para a construção do triângulo de Sierpinski e o nível I de iterações da construção a partir do GeomeTricks.

2) Na barra de ferramentas, selecione a opção “**Fractais**” > “**Definir Fractal**”. O programa mostrará uma janela pedindo o número de ternas. Para esta figura utilizaremos **quatro ternas** (digite “4” e clique em “OK”).
A construção de fractais no *GeomeTricks* se baseia na combinação de **ternas ordenadas de pontos**, ou seja, o **número de ternas** e a **ordem dos pontos** de cada terna definem o fractal.

3) O programa espera que as **4 ternas** sejam selecionadas.
Como a **ordem** dos pontos escolhidos em cada terna é essencial para a construção do fractal, clique os pontos na seguinte ordem:

- **ABC; AFD, FBE e DEC**

Quadro 1: Exemplo de construção de fractal no GeomeTricks.

Devido a isso, optamos por apresentar um exemplo “passo a passo” do processo de construção do “Triângulo de Sierpinski” no início desta parte da seqüência (Quadro 1), antes de propor as atividades propriamente ditas. Esta escolha se justifica no sentido de familiarizar os usuários com as ferramentas e procedimentos básicos de construção de fractais no ambiente informático utilizado (Quadro 2).

4) Tente construir os primeiros níveis de iteração para esta figura.

Para isso selecione “**Fractais**” > “**Níveis**”. O programa mostrará uma janela pedindo o número de **níveis** desejado. Escolha alguns valores inteiros para ver o resultado da construção.

Notas:

- **Após concluir uma figura, pare a construção clicando no botão**



- Você pode mudar a cor das figuras geradas, clicando as cores desejadas na paleta de cores no canto superior direito da tela.

Quadro 2: Procedimentos de construção do fractal no GeomeTricks.

Na seqüência, aproveitamos este exemplo para introduzir as principais características matemáticas da construção de fractais do Jogo do Caos nesse ambiente: a explicitação de uma construção baseada em Transformações Geométricas.

➤ *Análise matemática da construção*

A construção de fractais a partir do *GeomeTricks* se baseia em Transformações Geométricas, que são definidas a partir da **posição inicial** dos pontos que irão gerá-lo (fornecidos na Fig. 1a) e da **ordem** na qual esses pontos são selecionados em cada terna.

Assim, a primeira terna selecionada define um “**triângulo de controle**” e cada uma das demais ternas selecionadas define uma **composição de transformações geométricas** que será aplicada ao “triângulo de controle”.

Dessa forma, a terna ABC define o “triângulo de controle”, no qual serão feitas as composições de transformações definidas pelas demais ternas e a **posição** e **ordem** dos pontos das demais ternas irão definir quais composições de transformações geométricas serão aplicadas ao “triângulo de controle”, como pode ser visto na Fig. 1b, ou seja:

- A transformação (ABC) → (AFD) leva o **triângulo de controle** no **triângulo I**;
- A transformação (ABC) → (FBE) leva o **triângulo de controle** no **triângulo II**;
- A transformação (ABC) → (DEC) leva o **triângulo de controle** no **triângulo III**.

Observe que estas transformações são aplicadas iterativamente pelo software, fazendo com que, em cada nível de iteração, as composições de transformações geométricas sejam aplicadas em todos os triângulos gerados pelo nível anterior.

Você poderá constatar estas informações se construir, por exemplo, o **nível 2** de iterações, onde todas as transformações são aplicadas a cada um dos três novos triângulos gerados no nível anterior.

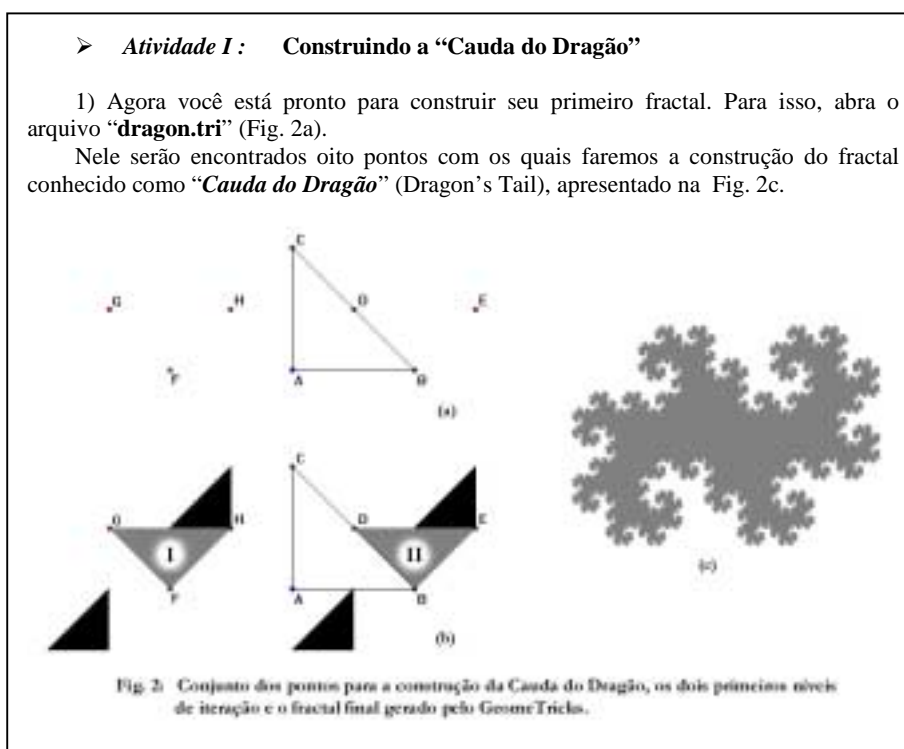
O **fractal final** será gerado após “*infinitas*” iterações.

Quadro 3: Características matemáticas da construção de fractais no GeomeTricks.

4.2.1.2 Atividade I: construindo a “cauda do dragão”

Com a apresentação, no exemplo anterior, das principais ferramentas de construção de fractais no ambiente, propomos, nesta primeira atividade, a construção da “Cauda do Dragão” a partir do conjunto de pontos necessários à sua construção com o GeomeTricks. Esta atividade, para a qual também será fornecido um arquivo contendo os pontos necessários à construção do fractal, se inicia com um conjunto de orientações baseadas numa figura, reproduzida no Quadro 4.

Nessa figura, são apresentados os pontos necessários à construção do fractal (reproduzindo os pontos obtidos no arquivo), os dois primeiros níveis de iteração desenvolvidos no ambiente e uma reprodução do fractal final, obtida após a correta determinação das ternas que o definem.



Quadro 4: Introdução da atividade I no GeomeTricks.

Partindo-se destas informações, o objetivo da atividade é a determinação das ternas de pontos necessárias à construção do fractal, assim como sua ordem, o que deverá ser feito a partir da identificação de duas composições de transformações geométricas, utilizadas na determinação do fractal.

Para tanto, a atividade exige a identificação dos elementos característicos de cada transformação, que deverá ser feita a partir da observação da figura que apresenta o nível 1 de iteração para o fractal (em cinza) e o nível 2 (em preto). Nessa perspectiva, o primeiro passo para a construção do fractal será a determinação das transformações geométricas necessárias para “transformar” o **triângulo retângulo isósceles ABC no triângulo I**.

Para realizar este “desafio”, supomos que os sujeitos sejam capazes de desenvolver, na atividade, a seqüência de transformações geométricas apresentadas na figura 4.1 e discutidas matematicamente, em seguida. Entretanto, adiantamos que, apesar deste ser o objetivo “ideal” desta atividade, não acreditamos que os sujeitos aos quais ela se destina empenhem-se efetivamente em sua execução, suposição que será discutida no final dessa seção.

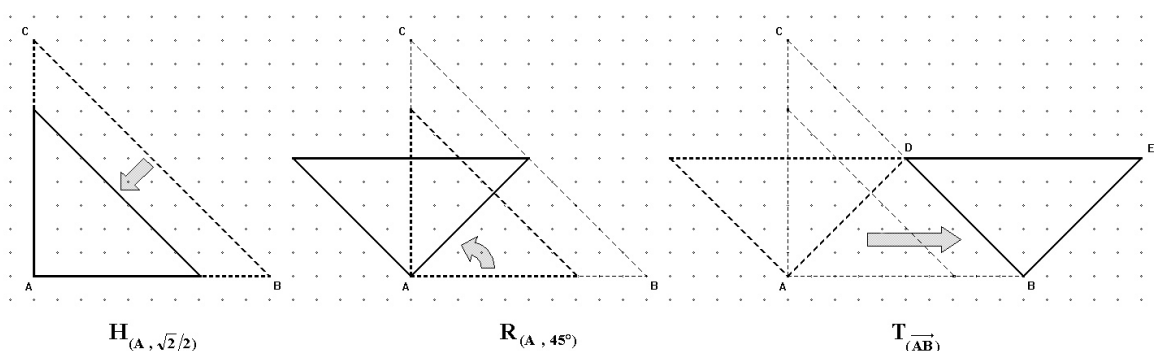


Figura 4.1: Seqüência de transformações geométricas da “Cauda do Dragão”.

Esta seqüência de ações estabelece uma função composta por três transformações geométricas que, adotando o ponto A como referencial, são assim definidas:

- uma *homotetia* que “encolhe” o ΔABC de modo a gerar outro triângulo cujo lado equivale à metade da hipotenusa do ΔABC , resultando, após alguns cálculos, na transformação $\mathbf{H}_{(A, \sqrt{2}/2)}$;
- uma *rotação* que leva o triângulo transformado pela homotetia a uma posição tal que sua hipotenusa torna-se perpendicular ao lado AC do ΔABC , resultando na transformação $\mathbf{R}_{(A, 45^\circ)}$;

- uma translação que leva o segundo triângulo transformado do ponto A (onde se localiza um de seus vértices) para o ponto B, definindo o vetor que resulta na transformação $T_{(AB)}$.

Definidas as transformações geométricas utilizadas nesta primeira composição, pode-se determinar a função composta que representa a primeira transformação afim aplicada ao ΔABC como:

$W_I = T_{(AB)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$; que, traduzida para a notação baseada em ternas de pontos do GeomeTricks, fornece a transformação:

$$W_I = T(ABC) \rightarrow T(BED).$$

Seguindo o mesmo raciocínio, pode-se determinar a segunda composição de transformações geométricas necessárias à “transformação” do **triângulo retângulo isósceles ABC** no **triângulo II**, também adotando o ponto A como referencial. Tendo-se em vista que, nesta transformação, são aplicadas a mesma homotetia e a mesma rotação da função anterior, variando-se apenas a translação que, neste caso, leva o triângulo transformado pela rotação do ponto A para o ponto F, define-se outro vetor que resulta na transformação $T_{(AF)}$.

Com isso, a função composta que representa a segunda transformação afim aplicada ao ΔABC é assim definida:

$W_{II} = T_{(AF)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$; que, traduzida para a notação baseada em ternas de pontos do GeomeTricks, fornece a transformação:

$$W_{II} = T(ABC) \rightarrow T(FHG).$$

Terminado este processo, segue-se apenas a efetiva construção da “Cauda do Dragão” no *GeomeTricks*, realizada a partir dos pontos dados e definida com as três ternas dadas pelas funções compostas acima, ou seja, as ternas:

- ABC; BED e FHG.

Do ponto de vista didático, pode-se dizer que a atividade I representa essencialmente uma *situação de ação* (Freitas, 1999, p. 78) pois, apesar de propormos tarefas de reconhecimento formal das transformações geométricas envolvidas, acreditamos que o enfoque adotado pelos sujeitos será o da

exploração experimental das características de construção de fractais, dadas pelo ambiente, e de possíveis relações entre a ordem das ternas e o fractal final gerado.

Dessa forma, dada a quantidade de novas informações fornecidas, nos parece previsível, e até natural, supor que a abordagem mais provável será um misto de estratégias de tentativa e erro, associadas a um reconhecimento implícito, operacional e imediatista de alguma possível relação entre as transformações geométricas com o fractal a ser construído.

Assim, apesar desta tarefa ter sido proposta, nossa expectativa com relação à atividade I não será a explicitação de argumentos de natureza teórica, mas sim uma situação na qual o sujeito tenha condições de agir e buscar uma solução satisfatória para o problema proposto. Acreditamos que os sujeitos utilizarão aspectos predominantemente experimentais de conhecimentos teóricos que ele já possui, além de reconhecer, mesmo que de forma intuitiva e informal, a existência e importância da relação entre um conjunto de composições de transformações geométricas e a construção de um fractal pelo Jogo do Caos, mesmo que não se empenhe efetivamente à sua explicitação.

4.2.1.3 Atividade II: construindo o “piso fractal”

A construção do “Piso Fractal” representa, de certa forma, um complemento da construção da “Cauda do Dragão”, na qual pretendemos introduzir uma característica extremamente relevante do ponto de vista da geometria fractal. De fato, estes dois fractais apresentam processos de construção muito semelhantes, ou seja, ambos são definidos praticamente a partir das mesmas transformações, gerando porém, fractais que possuem aparências completamente diferentes.

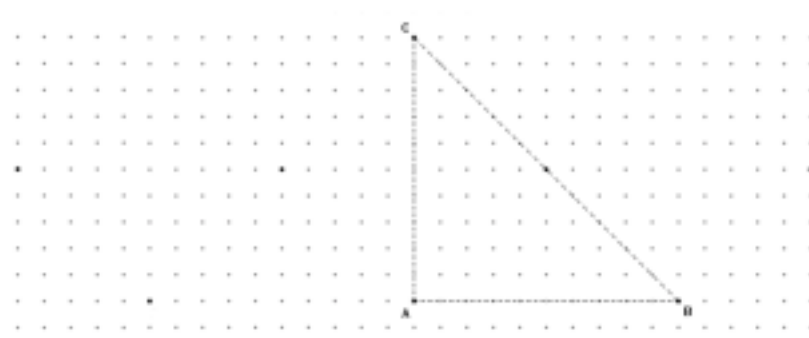
Em termos objetivos, a atividade de construção do “Piso Fractal” parte da apresentação das duas composições de transformações geométricas, necessárias à determinação das ternas de pontos para a construção do fractal a partir do *GeomeTricks*. Tais transformações deverão ser aplicadas aos mesmos pontos já utilizados para a construção da “Cauda do Dragão”. Neste ponto,

introduzimos um conjunto de notações formais, relativas a representação geométrica das transformações, fornecendo aos sujeitos a nomenclatura adotada para as transformações geométricas utilizadas e os “elementos característicos” de cada transformação.

Com isso, a proposta para a determinação da ordem correta dos pontos de cada terna utiliza um modelo baseado na tradução gráfica (através de desenhos) das composições de transformações, em uma figura contendo os pontos necessários à construção, apresentados numa grade de pontos (no Quadro 5).

➤ **Desafios:**

(i) Sua tarefa é desenhar os dois primeiros níveis de iterações do fractal e assim determinar a ordem correta das ternas com base nas transformações geométricas dadas acima.



Ordem das ternas :

Quadro 5: Grade de pontos para a determinação das ternas do “Piso Fractal”.

Assim sendo, a atividade de construção do “Piso Fractal” inicia-se com as composições de transformações geométricas:

$$W_I = T_{(AB)} \circ S_{(AC)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$$

$$W_{II} = T_{(AF)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$$

Analisando estas duas funções, percebemos que a única diferença, em relação às funções definidas com a “Cauda do Dragão”, situa-se na introdução da reflexão $S_{(AC)}$ na função W_I .

Dessa forma, cabe aos sujeitos a interpretação do efeito causado por esta nova transformação à ordem das ternas de pontos que definirão o fractal no *GeomeTricks* e esta interpretação poderá ser desenvolvida através de um esboço

dos dois primeiros níveis de iteração das funções, quando aplicadas ao triângulo de controle ABC.

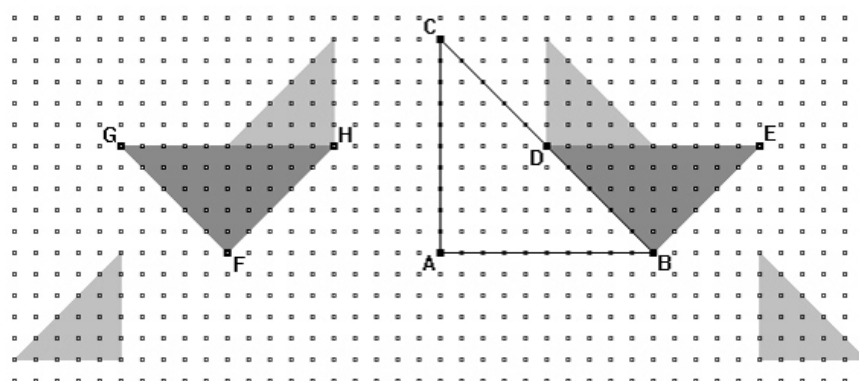


Figura 4.2: Esboço dos dois primeiros níveis de iteração para o “Piso Fractal”.

A partir desse esboço, espera-se que os sujeitos sejam capazes de determinar a ordem correta das ternas que definem o “Piso Fractal”, traduzindo as duas composições de transformações geométricas fornecidas para a notação baseada em ternas de pontos, utilizada no *GeomeTricks*, ou seja, traduzindo as funções dadas para a notação:

$$W_I = T(ABC) \rightarrow T(BDE)$$

$$W_{II} = T(ABC) \rightarrow T(FHG)$$

Esta proposta foi planejada de forma a evitar uma mera repetição das tarefas realizadas com a “Cauda do Dragão”, esperando com isso estender o impacto didático das representações das transformações geométricas. Além disso, a atividade também visa realçar o enfoque iterativo, presente na construção de fractais, visto que os sujeitos devem construir ao menos os dois primeiros níveis de iteração, para perceber qualquer diferença em relação ao fractal da atividade I, uma vez que, na primeira iteração, ambas as construções são idênticas.

Na atividade II temos ainda certas características de *situações de ação*, porém, nossa intenção é a transição para uma *situação de formulação* (Freitas, 1999, p. 79-80). Este enfoque pode ser notado na apresentação de alguns modelos teóricos explícitos aos sujeitos, que, nesta atividade, são representados

pelas notações formais dadas às transformações geométricas utilizadas. A partir da apresentação desses modelos, as composições de transformações, necessárias à construção do fractal, são descritas formalmente, segundo a notação apresentada e, com isso, partiremos da suposição de que os sujeitos adotem uma postura mais evidente de trabalho com um conjunto de informações teóricas, na resolução dos desafios apresentados.

Esperamos ainda que haja, por parte dos sujeitos, uma primeira tentativa implícita no sentido da validação de suas ações, o que se dará pela construção do “Piso Fractal” e a comparação deste com os primeiros níveis de iteração, por eles desenhados em um dos “*desafios*” propostos pela atividade.

Ainda nesta linha, propomos uma comparação entre as construções realizadas nas duas atividades, onde se espera dos sujeitos uma análise teórica das características de construção da “Cauda do Dragão” e do “Piso Fractal” e a relação entre as transformações geométricas com os fractais finais construídos.

Entretanto, é possível prever a possibilidade de não haver uma boa “*devolução*” (Freitas, 1999) desta situação, caracterizada pela insistência nas estratégias de tentativa e erro, na qual os sujeitos ignoram os dados e instruções dadas pela atividade e insistem em buscar a solução pela combinação aleatória dos pontos fornecidos pela atividade. Dadas as características da proposta, essa postura será uma possibilidade que, até a atividade II, não poderá ser controlada, entretanto, iremos retomar este fato mais adiante, nas atividades com o *MicroWorlds LOGO*.

4.2.2 Parte 2: Fractais no MicroWorlds LOGO

Nesse ponto, iniciamos a análise da segunda parte da seqüência de atividades, mudando o enfoque geométrico, abordado no *GeomeTricks*, para um conjunto de representações algébricas e formais das transformações geométricas no plano. Para isso, apresentamos aos sujeitos um novo ambiente, o *MicroWorlds*, que como vimos, se baseia na linguagem de programação *LOGO*.

Nessa fase, pretendemos aprofundar o caráter formal das representações algébricas das transformações. Assim, na primeira atividade, apresentamos uma

espécie de “exemplo disfarçado” de construção de fractais, que é seguido de uma análise matemática detalhada das construções, onde introduzimos uma revisão de alguns conceitos formais sobre a representação de transformações lineares e afins. Além disso, o fractal escolhido para esta atividade já terá sido construído pelos sujeitos na etapa anterior, fato que, supomos, irá facilitar a mudança de representação que pretendemos explorar.

Seguindo o mesmo raciocínio adotado na primeira parte das atividades, julgamos relevante explicitar algumas escolhas específicas a respeito da forma como concebemos esta segunda parte da seqüência de atividades. A primeira delas relaciona-se à proposta de mudança de representações e, quanto a isso, optamos pela construção, no *MicroWorlds*, dos mesmos fractais construídos com o *Geometricks*. Assim, as duas atividades propostas na segunda parte da seqüência propõem a construção de fractais já conhecidos pelos sujeitos. Com isso, esperamos que seja possível aos sujeitos uma maior compreensão do processo matemático que permite a construção dos fractais pelo Jogo do Caos, além de potencializar a criação de uma ponte entre as representações geométricas e algébricas das transformações no plano.

Além disso, no sentido de exercermos um maior controle sobre as atividades e um enfoque mais preciso para nossas análises didáticas, optamos por minimizar os impactos da mudança de ambiente informático para, com isso, nos concentrarmos nas interações matemáticas realizadas pelos sujeitos, a partir das novas formas de representação algébrica utilizadas. Essa preocupação foi especialmente relevante neste caso, uma vez que utilizamos um ambiente baseado numa linguagem de programação. Devido a essa característica, torna-se indispensável que os sujeitos interajam com a sintaxe de programação da linguagem *LOGO* para uma abordagem matemática mais aprofundada.

Dessa forma, optamos pela apresentação dos programas para a construção dos fractais parcialmente prontos nas atividades desta fase, esperando, assim, que uma potencial falta de familiaridade dos sujeitos com a linguagem de programação não os impossibilite de realizar as tarefas com sucesso. Essa abordagem também justifica a proposta da primeira atividade, no sentido de concentrarmos a ênfase de sua execução nas interações matemáticas com as representações algébricas introduzidas, minimizando os aspectos informáticos advindos da mudança de ambiente e da linguagem de programação.

4.2.2.1 Atividade III: construindo o “ piso fractal”

Concluída a determinação do fractal a partir do *GeomeTricks*, partiremos para sua construção a partir do *MicroWorlds*. O objetivo desta segunda fase da atividade concentra-se no desenvolvimento de noções mais formais e abstratas acerca do conceito de transformações geométricas, a partir da tradução das representações geométricas, utilizadas no *GeomeTricks*, para suas respectivas representações matriciais e algébricas, necessárias à determinação do sistema de funções iteradas utilizado no *MicroWorlds*.

Uma vez que, nesta parte da seqüência de atividades, também não é possível se garantir que os sujeitos possuam a devida familiaridade com o ambiente, optamos por introduzir um conjunto de informações, relativas às representações algébricas do Jogo do Caos, nesta primeira atividade. Com isso, descartamos a necessidade de um exemplo “passo a passo” de construção, o que pode fornecer um maior dinamismo e objetividade ao processo.

Entretanto, para lidar tanto com as mudanças de representação introduzidas, quanto com o novo ambiente utilizado, adicionamos a esta atividade uma análise matemática detalhada do processo de construção realizado, onde revisamos alguns conceitos formais acerca das transformações geométricas³, assim como um conjunto de instruções acerca das ferramentas informáticas e da utilização do ambiente (em destaque no Quadro 5). Esta escolha se deve ao duplo objetivo de fornecer alguns elementos indispensáveis aos sujeitos, tanto dos objetos matemáticos essenciais para uma compreensão adequada dos problemas propostos pela atividade, quanto dos aspectos informáticos do software e da linguagem de programação usada.

³ A “ficha” completa para esta atividade encontra-se no Anexo II.

Os procedimentos para a criação do “*Piso Fractal*” já se encontram prontos no arquivo “*pisotxt*”. É possível utilizá-los com as ferramentas “copiar” e “colar” do próprio Windows, seguindo as instruções abaixo:

1) No “*Bloco de Notas*” do Windows, abra o arquivo “*pisotxt*” e clique em “**Editar**” > “**Selecionar tudo**”.

Feito isso, copie o conteúdo selecionado com o comando “**Copiar**” ou com o atalho “**Ctrl+C**”.

2) No software “*MicroWorlds*”, selecione a página de procedimentos clicando em “**Pages**” > “**Procedures**”.

Feito isso, “cole” o conteúdo do arquivo com o comando “**Editar**” > “**Colar**” ou com o atalho “**Ctrl+V**”.

3) Para executar o programa, volte para a **página de desenhos** do software clicando em “**Pages**” > “**Page1**”.

4) Construa o “*Piso Fractal*” digitando o comando “*pisotxt*” na barra de comandos (quadro cinza) situada na parte inferior da tela do “*MicroWorlds*”.

Quadro 6: Instruções informáticas fornecidas na atividade III.

No sentido de minimizar os impactos advindos da mudança de ambiente informático e da necessidade de utilização da linguagem de programação LOGO, contamos com a utilização de um arquivo de apoio nesta parte da atividade⁴. Este arquivo, onde fornecemos a sintaxe do programa utilizado na construção de fractais no *MicroWorlds*, encontra-se incompleto, cabendo aos sujeitos a introdução das regras de cálculo necessárias à construção.

Além disso, para enfatizar as interações matemáticas realizadas pelos sujeitos, a partir das novas formas de representação algébrica utilizadas, propomos, como “*desafio*”, a determinação dos coeficientes resultantes das composições de transformações geométricas, necessários à construção do fractal, a partir da utilização das informações dadas.

Para isso, partimos das funções compostas definidas no *GeomeTricks*, propondo aos sujeitos a tradução da notação geométrica, utilizada nas atividades anteriores, para a notação matricial tradicional, adotada matematicamente em cada transformação geométrica. Com isso, determinar-se-á uma equação matricial, composta pelas matrizes que representam cada transformação utilizada e estas, quando operadas, determinam a expressão matricial final para a função composta desejada.

⁴ A necessidade deste arquivo se deve ao fato de termos utilizado a “*versão demo*” do ambiente, que não permite a gravação de “*projetos*” de construção.

Realizada esta tarefa, o último passo do processo simplesmente traduz a expressão matricial obtida, para a notação algébrica tradicional do sistema de funções iteradas, que será utilizado no *MicroWorlds*. Em termos objetivos, este processo segue os passos apresentados a seguir.

Para a primeira transformação, traduz-se a notação geométrica para a notação matricial equivalente, ou seja, partindo-se da função

$W_I = T_{(\overline{AB})} \circ S_{(\overline{AC})} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$; chega-se a expressão:

$$W_I \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \frac{\sqrt{2}}{2} \cdot \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos 45^\circ & -\text{sen} 45^\circ \\ \text{sen} 45^\circ & \cos 45^\circ \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$W_I \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} -0,5 & 0,5 \\ 0,5 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Traduzindo-se esta expressão matricial para a representação algébrica tradicional, teremos o primeiro par de funções iteradas do sistema, assim definidas:

$$\begin{cases} x_n = -0,5x + 0,5y + 1 \\ y_n = 0,5x + 0,5y \end{cases}$$

Analogamente, obtemos a segunda transformação a partir da tradução da função geométrica para a representação matricial equivalente, assim, a partir da função

$W_{II} = T_{(\overline{AF})} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$ obtém-se a expressão:

$$W_{II} \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \frac{\sqrt{2}}{2} \cdot \begin{pmatrix} \cos 45^\circ & -\text{sen} 45^\circ \\ \text{sen} 45^\circ & \cos 45^\circ \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0,5 & -0,5 \\ 0,5 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

Que, traduzida para a representação algébrica tradicional, determina o segundo par de funções iteradas do sistema, assim definidas:

$$\begin{cases} x_n = 0,5x - 0,5y - 1 \\ y_n = 0,5x + 0,5y \end{cases}$$

Para construir o “Piso Fractal”, basta-nos traduzir estes dois pares de funções iteradas para a linguagem LOGO e inseri-los no programa de construção de fractais, apresentado no arquivo de apoio. Cabe novamente ressaltar que

pretendemos enfatizar o processo matemático desta construção, portanto fornecemos na atividade a tradução das funções iteradas para a sintaxe de programação LOGO. Além disso, como se trata da primeira atividade com este modelo de representação, fornecemos aos sujeitos a resposta do “desafio” proposto, ou seja, a expressão algébrica final do sistema de funções iteradas.

Sob o ponto de vista didático, apesar das construções serem fornecidas parcialmente desenvolvidas, nosso objetivo com esta atividade será a *institucionalização* (Freitas, 1999, p. 82-83) de alguns conceitos e notações fundamentais à execução desta segunda fase das atividades, onde esperamos fornecer um caráter de objetividade e universalidade para os conceitos apresentados, no sentido de superar potenciais limitações impostas pelas particularidades dos problemas propostos. Além disso, uma vez que o MicroWorlds possui uma interface mais transparente, quando comparado com o GeomeTricks, nos é possível propor atividades que explorem mais diretamente os aspectos matemáticos relevantes a estas situações didáticas, fazendo com que as potenciais limitações dos sujeitos, relativas ao desconhecimento da linguagem LOGO, sejam superadas sem maiores problemas.

4.2.2.2 Atividade IV: construindo a “cauda do dragão”

Como introduzido no *GeomeTricks*, esperamos abordar duas características importantes do ponto de vista didático com as construções do “Piso Fractal” e da “Cauda do Dragão”: por um lado, esperamos um maior aprofundamento das noções acerca das diferentes representações das Transformações Geométricas; e, por outro lado, uma apresentação, ao menos intuitiva, da extrema dependência existente entre os coeficientes iniciais do sistema de funções iteradas e a aparência final do fractal produzido, visto que esta última característica é muito representativa para uma introdução tanto à Geometria Fractal como à Teoria do Caos.

A construção da “Cauda do Dragão” com o *MicroWorlds* propõe novamente a determinação dos coeficientes necessários ao sistema de funções iteradas, que definem a representação algébrica do fractal. Dessa forma, esta

atividade será basicamente idêntica à realizada na seção anterior, portanto seguiremos a mesma seqüência de ações já apresentada na construção do “Piso Fractal”, ou seja, traduzir as transformações geométricas determinadas no *GeomeTricks*, para as representações matriciais e algébricas.

Assim, na primeira transformação, traduz-se a notação geométrica para a notação matricial equivalente, ou seja, parte-se da função

$$W_I = T_{(AB)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)} ; \text{ que, traduzida para a notação matricial,}$$

nos fornece a expressão:

$$W_I \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \frac{\sqrt{2}}{2} \cdot \begin{pmatrix} \cos 45^\circ & -\text{sen} 45^\circ \\ \text{sen} 45^\circ & \cos 45^\circ \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0,5 & -0,5 \\ 0,5 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Traduzindo-se esta expressão matricial para a representação algébrica tradicional, teremos o primeiro par de funções iteradas do sistema, assim definidas:

$$\begin{cases} x_n = 0,5x - 0,5y + 1 \\ y_n = 0,5x + 0,5y \end{cases}$$

Da mesma forma, obtemos a segunda transformação a partir da tradução da função geométrica definida no *GeomeTricks*, para sua notação matricial equivalente, portanto, temos a função

$$W_{II} = T_{(AF)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)} ; \text{ que, também traduzida para a notação}$$

matricial, nos fornece a expressão:

$$W_{II} \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \frac{\sqrt{2}}{2} \cdot \begin{pmatrix} \cos 45^\circ & -\text{sen} 45^\circ \\ \text{sen} 45^\circ & \cos 45^\circ \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0,5 & -0,5 \\ 0,5 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

Traduzindo-se novamente a expressão matricial para a representação algébrica tradicional, determina-se o segundo par de funções iteradas do sistema, assim definidas:

$$\begin{cases} x_n = 0,5x - 0,5y - 1 \\ y_n = 0,5x + 0,5y \end{cases}$$

Como esta atividade propõe vários “desafios” distintos, para os quais serão dadas poucas informações, escolhemos a “Cauda do Dragão” cuja construção é muito semelhante ao “Piso Fractal”, utilizado na atividade anterior, visto que essa semelhança já terá sido explorada na primeira parte das atividades. Além disso, esta atividade visa retomar a construção com o *GeomeTricks*, no sentido de exigir dos sujeitos a utilização das transformações geométricas exploradas na atividade I. Dessa forma, esta atividade também fornece um controle da produção realizada no *GeomeTricks*, uma vez que os sujeitos serão obrigados a rever suas construções, no caso de terem optado por estratégias de tentativa e erro na atividade I. Esta postura também justifica o motivo de termos invertido a ordem, na qual as atividades foram propostas, na primeira e na segunda parte da seqüência de atividades.

Em termos didáticos, a atividade IV propõe uma *situação de formulação* com vistas à *validação* dos resultados (Freitas, 1999, p. 79-82), visto que será proposto aos sujeitos a “tradução” das composições de transformações geométricas do fractal “Cauda do Dragão”, já construído com o *GeomeTricks*, para um sistema de funções iteradas, num contexto focado nas representações matriciais e algébricas das transformações. Feito isso, segue-se outra “tradução” do sistema de funções obtido para a sintaxe da linguagem *LOGO*, procedendo-se a validação dos resultados, a partir da construção do fractal, quando o programa é executado.

A afirmação de que a atividade conta com aspectos de *formulação* e *validação* dos resultados, deve-se ao fato desta solicitar que os sujeitos interpretem as informações dadas no problema, manipulando-as a partir de um conjunto de notações e representações simbólicas e procedendo à validação dos resultados ao substituir, testar e visualizar graficamente os resultados obtidos.

CONCLUSÃO

Iniciamos este trabalho com o objetivo de estudar potenciais aplicações da Geometria Fractal, associada à utilização de ambientes computacionais, visando a criação de situações didáticas. Assim, partimos da hipótese – até então uma mera suspeita – de que a utilização coordenada da Geometria Fractal e de ambientes informáticos de aprendizagem poderia criar um terreno potencialmente fértil para a exploração didática de objetos matemáticos a serem ensinados.

Uma vez que a utilização de ambientes informáticos de aprendizagem é fundamental nesse trabalho, situamos grande parte das análises na noção de Transposição Informática de ambientes computacionais (Balacheff, 1994), dando especial atenção à utilização de *micro-mundos* de Geometria e ao *domínio de validade epistemológica* desses ambientes. Nesse contexto, atribuímos aos computadores o papel de ferramentas, o que acarreta a constatação de que a efetiva utilização didática de novas tecnologias demanda uma análise criteriosa, tanto das ferramentas disponibilizadas pelo ambiente informático utilizado, como dos objetos matemáticos específicos que nos propomos a ensinar com tais ambientes.

Dessa forma, aproveitamos esta seção para retomar e inter-relacionar os objetivos e considerações realizadas nos capítulos anteriores, apresentando uma síntese dos principais tópicos estudados.

No primeiro capítulo, concentramo-nos em estabelecer um conjunto suficientemente amplo de conceitos matemáticos acerca da Geometria Fractal, incluindo processos de construção que se utilizam, dentre outros conceitos, das Transformações Geométricas no Plano. Dessa forma, situamos a Geometria Fractal enquanto objeto matemático.

Feito isso, o segundo capítulo dedica-se à delimitação dos principais referenciais teóricos e metodológicos do estudo: a delimitação de um conjunto de

noções de natureza epistemológica, acerca das transformações ocorridas em objetos de um modelo matemático de referência, quando transposto para um sistema de representações simbólicas, manipuláveis por diferentes dispositivos artificiais, assim como o trabalho e análise desses dispositivos. Além disso, realizamos uma análise a respeito das ferramentas específicas, fornecidas em cada um dos ambientes informáticos estudados.

Nessa perspectiva, nos propomos a situar o trabalho no contexto de pesquisas recentes em Educação Matemática, qual seja, a problemática da utilização de micro-mundos de Geometria e da transposição informática de ambientes informáticos de aprendizagem, a partir de uma metodologia baseada na análise qualitativa do *domínio de validade epistemológica* desses ambientes.

Por sua vez, o terceiro capítulo apresenta um estudo específico voltado à construção de fractais nestes ambientes informáticos, onde exploramos detalhadamente a viabilidade de construção de cada tipo de fractal em diferentes micro-mundos. Dessa forma, a partir de um enfoque baseado no “como construir” fractais em diferentes micro-mundos, abordamos mais detalhadamente duas das dimensões de análise, propostas pela delimitação do *domínio de validade epistemológica* dos ambientes informáticos estudados.

Finalmente, a última parte do trabalho propõe a análise de situações de ensino, envolvendo uma classe específica de problemas, contextualizada a partir da construção de fractais do Jogo do Caos e da utilização de dois ambientes informáticos distintos. Tal proposta baseia-se na modelização de situações didáticas e visa explorar a relação entre a construção de fractais e as representações geométricas e algébricas das Transformações Geométricas no Plano.

Nossas análises apontam um conjunto de resultados, em relação aos problemas de pesquisa inicialmente propostos pelo trabalho. Em particular, podemos observar objetivamente os efeitos advindos das transformações decorrentes do esforço de representação de um modelo matemático de referência, para um modelo computável, implementado em ambientes informáticos de aprendizagem. Tal fato pode ser constatado nas diversas construções de fractais realizadas, onde, para cada ambiente informático utilizado, obtêm-se tipos de representação distintos para um mesmo objeto fractal.

Destacamos abaixo as principais características do *domínio de validade epistemológica* de dois dos ambientes analisados – o *GeomeTricks* e o *Microworlds LOGO*.

Os fractais que podem ser gerados nesses ambientes são baseados na concepção do Jogo do Caos e, com isso, decorre a relação entre sua construção e o conceito de Transformações Geométricas no Plano. Do ponto de vista matemático, essa relação foi evidenciada no capítulo 1, quando da definição do Jogo do Caos.

Na implementação das ferramentas computacionais de construção de fractais do Jogo do Caos nesses ambientes, utilizou-se de representações algébricas no *MicroWorlds*, e geométricas no *GeomeTricks*.

Os feedback presentes em ambos são basicamente de natureza “visual” (preceptiva), relacionados às representações figurais obtidas na construção de fractais. Entretanto, os tipos de controle neles disponibilizados diferem substancialmente, em termos das ferramentas disponibilizadas e na explicitação das relações entre tais ferramentas e os conceitos matemáticos em jogo, em particular, as transformações geométricas.

Dessa forma, pode-se dizer que o *MicroWorlds* apresenta ferramentas mais explícitas, visto que, para a construção de um fractal, é necessária a elaboração de um procedimento (em linguagem de programação), que, por sua vez, exige a determinação dos coeficientes dos sistemas de funções iteradas que o definem. Com relação a proposta do nosso estudo, ressaltamos ainda as considerações de Laborde (1993, p. 59), no que se refere à possíveis dificuldades originadas pela acessibilidade à sintaxe da linguagem LOGO.

No caso do *GeomeTricks*, destacamos que não há uma relação imediata entre a determinação, o posicionamento e a ordem das ternas de pontos que definem o fractal com a concepção do Jogo do Caos, sequer com a utilização de conjuntos de composições de transformações geométricas no plano. Assim, pode-se afirmar que a estrutura formal, presente no *GeomeTricks*, possui características essencialmente implícitas. Desta forma, referindo-se a estrutura formal do *GeomeTricks*, podemos supor que, para se obter um efetivo controle da construção de fractais nesse ambiente, faz-se necessário associar sua utilização a atividades, formuladas de modo a tornar mais evidentes as relações entre a ferramenta de construção de fractais e as noções e propriedades matemáticas

relativas ao Jogo do Caos. E mais, as concepções ou caracterizações de objetos fractais da família “iniciador-gerador” – habitualmente apresentadas no ensino de Matemática – não são direta e eficazmente transpostas para construções com o *GeomeTricks*, podendo inclusive dificultar a compreensão de tais processos.

Assim, pretendemos ressaltar que o enfoque baseado na utilização da Geometria Fractal e desses ambientes, como contextualização das Transformações Geométricas no Plano, pode fornecer um repertório de aplicações envolvendo diferentes apreensões e representações desse objeto.

Dada a complexidade e amplitude das questões envolvendo a Geometria Fractal e o uso de ferramentas computacionais, algumas questões abordadas nesse trabalho ficaram “em aberto” ou não foram devidamente analisadas. A principal delas refere-se à investigação da aplicabilidade das situações didáticas que desenvolvemos no capítulo 4, por meio de experimentos de ensino ou engenharias didáticas, por exemplo, realizadas com o perfil de estudantes aos quais as situações se destinam. Com isso, seria possível complementar o conjunto de análises proposto pela Transposição Informática, com o estudo do *domínio de validade didática* dessas situações.

Apesar disso, esperamos contribuir para o campo de pesquisas em Educação Matemática, no sentido de explicitar um conjunto de concepções acerca de potenciais utilizações da Geometria Fractal, assim como pela indicação de caminhos para a análise e utilização de ambientes informáticos de aprendizagem neste processo, passíveis de estudos mais completos e rigorosos.

Em termos de perspectivas de continuidade do trabalho, podemos citar outros enfoques para a utilização do Jogo do Caos, que poderiam ser facilmente adaptados para contextualizar aplicações relacionadas à Álgebra Linear, ao estudo da Topologia e Espaços Métricos (supondo contextos em espaços de pequena dimensão). Além dessas, podemos sugerir, inclusive, aplicações voltadas ao Ensino Médio, como a contextualização e operações de matrizes, estudo de sistemas de funções e seqüências numéricas, seguidas de suas respectivas representações gráficas, etc.

Além destas, vislumbramos outras possibilidades de utilização da Geometria Fractal, que sequer puderam ser consideradas no formato final do

trabalho. Assim, podemos propor pesquisas que utilizem, por exemplo, o Conjunto de Mandelbrot para a contextualização do conceito de números complexos, abordando a idéia de seqüências complexas e representações no plano de Argand. Na mesma linha, pode-se explorar o conceito de intervalo infinito e o estudo de números irracionais, além de uma introdução à própria Teoria do Caos.

Enfim, além da enorme gama de enfoques que a Geometria Fractal suscita, finalizamos essa etapa da investigação com a convicção, ainda que subjetiva, de que os processos de construção de fractais em ambientes informatizados podem ser utilizados no ensino e aprendizagem de objetos matemáticos de forma concreta e contextualizada. Tais características podem se refletir na possibilidade de estimular a imaginação e curiosidade dos aprendizes, não somente apresentando, na tela do computador, figuras extremamente belas e intrigantes, mas também permitindo o tratamento de diversos conceitos e processos matemáticos numa perspectiva exploratória, experimental e, porque não, divertida.

REFERÊNCIAS BIBLIOGRÁFICAS

- ADDISON, P. S. – *Fractals and Chaos: an illustrated course*, Londres: Institute of Physics Publishing (IoP), 1997.
- ALVES-DIAS, M. – *Les problèmes d’articulation entre points de vue “cartésien” et “paramétrique” dans l’enseignement de l’algèbre linéaire*, Tese de Doutorado, Universidade de Paris 7, França, 1998.
- ARTIGUE, M. – “Ingénierie didactique”, *Recherches en Didactiques des Mathématiques*, Grenoble: La pensée Sauvage, vol. 9, n. 3, 1988, pp. 281-308.
- BALACHEFF, N. – “Eclairage didactique sur les EIAH”, In : *Actes du "Colloque annuel de la Société de Didactique des mathématiques du Québec"*, 1998, Québec. Disponível em: <<http://www-didactique.imag.fr/Balacheff/PubliBalacheffPartiel.html>> Acesso em: 29 mai. 2003.
- BALACHEFF, N. – “La transposition informatique. Note sur un nouveau problème pour la didactique”, In: ARTIGUE, M. et al. (eds.), *Vingt ans de didactique des mathématiques en France*, Grenoble: La pensée Sauvage, 1994, pp. 364-370.
- BALACHEFF, N.; KAPUT, J. – “Computer-Based Learning Environments in Mathematics”, In: BISHOP, A. et al. (ed.), *International Handbook in Mathematics Education*, Dordrecht: Kluwer Academic Publishers, 1996, pp. 469-501.

- BALACHEFF, N.; SUTHERLAND, R. – “Epistemological domain of validity of microwords: the case of LOGO and Cabri-Géomètre”, In: LEWIS, R.; MENDELSON, P. (ed.), *Lessons from Learning*, Amsterdam: North-Holland/Elsevier Science B. V., IFIP Transactions A46, 1994, pp. 137-150.
- BARBOSA, R. M. – *Descobrimos a Geometria Fractal para a sala de aula*, Belo Horizonte: Autêntica, 2002.
- BARNESLEY, M. F. – *Fractals Everywhere*, New York: Academic Press, 1993.
- BAULAC, Y.; BELLEMAIN, F.; LABORDE, J.-M. – *Cabri-géomètre, un logiciel d'aide à l'enseignement de la géométrie, logiciel et manuel d'utilisation, version 1.0, Macintosh de Apple*, Paris: Nathan-Logiciels, 1988.
- BAULAC, Y.; BELLEMAIN, F.; LABORDE, J.-M. – *Cabri-Géomètre II v. 1.0*, Texas Instruments, Dallas, IMAG-CNRS, Universidade Joseph Fourier, França, 1994, MS Windows, software educativo (geometria dinâmica).
- BLOCH, I. – *L'enseignement de l'analyse à la charnière lycée / université Savoirs, connaissances et conditions relatives à la validation*, Tese de Doutorado, Universidade Bordeaux I, França, 2000.
- BOYER, C. – *História da Matemática*, Tradução de Elza Gomide, São Paulo: Edgard Blücher, 1974.
- BOLDRINI, J. L. et al. – *Álgebra Linear*, São Paulo: Harper & Row, 1978.
- BROUSSEAU, G. – “Fondements et Méthodes de la Didactique des Mathématiques”, *Recherches en Didactiques des Mathématiques*, Grenoble: La pensée Sauvage, vol. 7, n. 2, 1986, pp. 33-116.
- _____, – “Le Contrat Didactique: le milieu”, *Recherches en Didactiques des Mathématiques*, Grenoble: La pensée Sauvage, vol. 9, n. 3, 1988, pp. 309-336.

- CALLIOLI, C. A.; DOMINGUES, H. H.; COSTA, R. F. – *Álgebra Linear e Aplicações*, 2ª ed., São Paulo: Atual, 1978.
- CARVALHO, M. C. e S., et alli. – *Fractais: uma breve introdução*, s.c., s.l.p., s.d.
- CHEVALLARD, Y. – “La Transposition Didactique: du savoir savant au savoir enseigné”, *Recherches en Didactiques des Mathématiques*, Grenoble: La Pensée Sauvage, vol. 12, n. 23, 1991.
- DARST, R.; PALAGALLO, J.; PRICE, T. – “Fractal Tilings in the Plane”, *Mathematics Magazine*, vol. 71, nº 1, 1998, pp. 12-23.
- DORIER, J.-L. (ed.), – *On the Teaching of Linear Algebra*, Dordrecht: Kluwer Academic Publishers, 2000.
- _____, – “État de l’Art de la Recherche en Didactique à Propos de L’Enseignement de l’Algèbre Linéaire”, *Recherches en Didactiques des Mathématiques*, Grenoble: La Pensée Sauvage, vol. 18, n. 2, 1998, pp. 191-230.
- _____, – *L’Enseignement de l’Algèbre Linéaire en Question*, Grenoble: La Pensée Sauvage, 1997.
- FREITAS, J. L. M. – “Situações Didáticas”, In: MACHADO, S. D. A. (org.), *Educação Matemática: uma Introdução*, São Paulo: EDUC, pp. 65-87, 1999.
- GLEICK, J. – *Caos: A criação de uma nova ciência*, Tradução de Waltensir Dutra, Rio de Janeiro: Campus, 1990.
- GOLDENBERG, E. P.; CUOCO, A. – “What is Dynamic Geometry?”, In: LEHRER, R.; CHAZAN, D. (ed.), *New Directions in the Teaching and Learning of Geometry*, Erlbaum: Hillsdale, pre-print, 1996.
- HANNA, G.; JAHNKE, N. – “Proof and Proving”, In: BISHOP, A. et al. (ed.), *International Handbook in Mathematics Education*, Dordrecht: Kluwer Academic Publishers, 1996, pp. 877-908.

- HAREL, G. – “Three Principles of Learning and Teaching Mathematics: Particular Reference to Linear Algebra – Old and New Observations”, In: DORIER, J.-L. (ed.), *On the Teaching of Linear Algebra*, Dordrecht: Klüwer Academic Publishers, 2000, pp. 177-189.
- HAROLD, A; DISESSA, A. – *Turtle Geometry*, 8^a ed., Massachusetts: MIT Press, 1992.
- HEALY, L.; HOYLES, C. – “Software Tools for Geometrical Problem Solving: Potentials and Pitfalls”, *International Journal of Computers for Mathematical Learning*, Dordrecht: Klüwer Academic Publishers, vol. 6, 2001, pp. 235-256.
- HILLEL J., SIERPINSKA, A. – “One Persistent Mistake in Linear Algebra”, *Proceeding of PME 18 (4vol.)*, vol. 4, Université de Lisbonne, 1995, pp. 65-72.
- HOYLES, C. – “Microworlds/Schoolworlds: The Transformation of an Innovation”, *Learning from Computers: Mathematics Education and Technology*, NATO ASI Series, Berlin: Springer-Verlag, vol. 121, 1993, pp. 1-17.
- HOYLES, C.; NOSS, R. – “Synthesizing Mathematical Conceptions and their Formalization through the Construction of a Logo-based School Mathematics Curriculum”, *International Journal of Mathematical Education in Science and Technology*, Loughborough: Taylor & Francis, vol. 18, n. 4, 1987, pp. 581-595.
- JACKIW, N.; FINZER, W. – *The Geometer’s SketchPad: Programming by Geometry*, Disponível em: <<http://www.acypher.com/wwid/Chapters/13Sketchpad.html>> Acesso em: 26 mar. 2002.
- JACKIW, N.; KLOTZ, G. – *The Geometer’s SketchPad v. 3.10*, Key Curriculum Press. Berkeley, CA, 1997, MS Windows, software educativo (geometria dinâmica).
- JAHN, A. P. – *Des transformations des figures aux applications ponctuelles: étude d’une séquence d’enseignement avec Cabri-Géomètre. Relations entre aspects géométriques et fonctionnels en classe de Seconde*, Tese de Doutorado, Universidade Joseph Fourier (Grenoble I), França, 1998.

- LABORDE, C. – “The computer as part of the learning environment: the case of Geometry”, In: KEITHEL, C.; RUTHVEN, K. (eds.), *Learning from Computers: Mathematics Education and Technology*, NATO ASI Series, Berlim: Springer-Verlag, v. 121, 1993, pp. 48-67.
- LIMA, E. L. – *Coordenadas no Plano*, 4^a ed., Rio de Janeiro: Sociedade Brasileira de Matemática, 2002.
- _____, – *Espaços Métricos*, Rio de Janeiro: Instituto de Matemática Pura e Aplicada, CNPq, 1997.
- MANDELBROT, B. B.; JAFFARD, S. – “Peano-Pólya Motions, When Time is Intrinsic or Binomial”, *The Mathematical Intelligencer*, New York: Springer-Verlag, vol. 19, n. 4, 1997, pp. 21-26.
- MANDELBROT, B. B. – *Objectos Fractais: Forma, Acaso e Dimensão*, Tradução a partir da 3^a Edição francesa de Carlos Fiolhais, Lisboa: Gradiva, 1991.
- _____, – *The Fractal Geometry of Nature*, Nova York: Freeman, 1977.
- NAGATA, J.-I. – *Modern General Topology*, Amsterdan: North Holland Publishing, 1968.
- NOSS, R.; HOYLES, C. – *Windows on Mathematical Meanings: learning cultures and computers*, Dordrecht: Kluwer Academic Publishers, 1996.
- PAPERT, S.; SILVERMAN, B. – *MicroWorlds*, v. 2.05, Logo Computer Systems Inc., Cambridge, 2000, MS Windows, software educativo (geometria da tartaruga).
- PEITGEN, H.-O.; JÜRGENS, H.; SAUPE, D. – *Fractals for the Classroom – Part One: Introduction to Fractals and Chaos*, New York: Springer-Verlag, 1991a.

_____, *Fractals in the Classroom: Strategic Activities*, New York: Springer-Verlag, vol. 1, 1991b.

PENROSE, R. – *A Mente Nova do Rei: computadores, mentes e leis da física*, Tradução de Waltensir Dutra, Rio de Janeiro: Campus, 1991.

SADOLIN, V. – *GeomeTricks v. 2.37*, Copenhagen, Versão em Português e Espanhol de Dr. Marcelo C. Borba e Dra. Miriam Penteado, São Paulo: GPIMEM, UNESP-Rio Claro, 1998(a), MS Windows, software educativo (geometria dinâmica).

SADOLIN, V. – *Exploring fractals defined by triples of points*, 14 abr. 1998(b). Arquivo (399 bytes). Word 7.0.

SCOTT, P.; BEAUMONT, B. – *The Fractal Umbrella*, última modificação: 20 fev. 2002, Disponível em: <<http://www.maths.adelaide.edu.au/pure/pscott/fractals/index.html>> Acesso em: 16 jun. 2003.

SIERPINSKA, A. et al. – “Evaluation of a Teaching Design in Linear Algebra”, *Recherches en Didactiques des Mathématiques*, Grenoble: La Pensée Sauvage, vol. 19, n. 1, 1999, pp. 7-40.

STEVENS, R. T. – *Fractal: Programing in Turbo Pascal*, Redwood City: M&T Books, 1990.

SUTHERLAND, R.; BALACHEFF, N. – “Didactical Complexity of Computational Environments for the Learning of Mathematics”, *International Journal of Computers for Mathematical Learning*, Holanda, Vol. 4, No. 1, 1999, pp. 1-26.

ANEXO I

TRANSFORMAÇÕES GEOMÉTRICAS NO PLANO – APRESENTAÇÃO MATEMÁTICA

Transformações Geométricas no Plano – Apresentação Matemática

Neste anexo, estamos interessados em definir um conjunto de transformações geométricas no plano euclidiano que servirão como repertório básico de ferramentas para a construção de fractais a partir do Jogo do Caos. Focaremos tais definições nas transformações no \mathbb{R}^2 , embora estas pudessem ser estendidas para outros espaços.

Visto que o objetivo desta seção é o de fornecer um conjunto específico de ferramentas para a construção de fractais, não estamos interessados em definir as Transformações Geométricas completamente. Não nos interessaremos na reprodução de demonstrações formais dos conceitos introduzidos, pela simples razão de que tais demonstrações podem ser facilmente obtidas nas obras que nos serviram de referência¹, bastando-nos, portanto, apenas enunciá-los.

Como já ressaltamos, a principal ferramenta utilizada para a construção de fractais no Jogo do Caos são as transformações afins no \mathbb{R}^2 . Dessa forma, iremos explorar somente um conjunto de conceitos indispensáveis à compreensão desse objeto, no sentido de sua utilização na construção de fractais.

Apresentamos a seguir três definições fundamentais, que especificam a construção formal de transformações afins no plano euclidiano, adaptadas de Lima (2002, p. 137).

(i) Define-se uma transformação afim no plano euclidiano como uma função $T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, isto é, uma correspondência que associa a cada ponto P do plano outro ponto $P_1 = T(P)$ do plano, chamado sua *imagem* por T .

(ii) Dadas duas transformações $S, T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, a composta $S \circ T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ é a transformação que associa a cada ponto P do plano \mathbb{R}^2 o ponto $S(T(P))$ do

¹ Para maiores detalhes sobre as demonstrações formais das transformações geométricas no plano, sugerimos as obras de Elon Lages Lima (2002), Boldrini (1978) e Callioli & Domingues (1978), nas quais se fundamentaram nossas pesquisas.

plano. Portanto, por definição, $(S \circ T)(P) = S(T(P))$, ou seja, $S \circ T$ consiste em aplicar primeiro T e depois S .

(iii) Uma vez que um sistema de coordenadas no \mathbb{R}^2 seja estabelecido, uma transformação T pode ser escrita por suas equações, isto é, pelas expressões das coordenadas (x_1, y_1) do ponto $P_1 = T(P)$, obtido pela aplicação de T ao ponto $P = (x, y)$.

Definimos a seguir as principais Transformações Geométricas utilizadas neste trabalho, pois é a partir de sua composição que os sistemas de funções iteradas do Jogo do Caos são determinados. Em particular, quando se deseja definir uma função composta de transformações, que deformam o espaço relativamente à origem, ou seja, quando são utilizadas “homotetias”, “rotações” ou “reflexões”, estas são obtidas pela multiplicação das matrizes que as definem. Por sua vez, se desejarmos introduzir uma translação ao sistema, deve-se somar a matriz que a define diretamente ao sistema.

Homotetias

Definimos a homotetia como uma transformação $H: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, associada a um centro “ O ” e uma razão, ou fator, “ f ” real que leva cada ponto $P \in \mathbb{R}^2$ no ponto $P_1 = H(P)$ tal que $\overrightarrow{OP_1} = f \cdot \overrightarrow{OP}$.

Em particular, a homotetia transforma retas paralelas em retas paralelas e retas perpendiculares em retas perpendiculares, preservando os ângulos. Dessa forma, ao aplicarmos uma homotetia a uma dada figura, como um triângulo, por exemplo, sua imagem será um triângulo semelhante ao original.

Dado um sistema OXY de eixos ortogonais, com origem no centro O da homotetia, as coordenadas do ponto $P_1 = (x_1, y_1)$, imagem do ponto $P = (x, y)$ pela homotetia são:

$$\begin{cases} x_1 = f \cdot x + 0 \cdot y \\ y_1 = 0 \cdot x + f \cdot y \end{cases} \quad \text{ou} \quad \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = f \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f & 0 \\ 0 & f \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

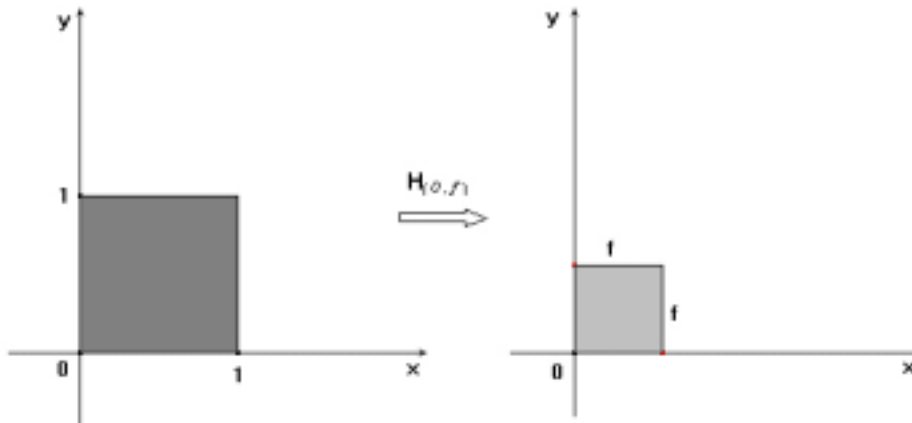


Figura I.1: Homotetia de razão f , aplicada em um quadrado de lado unitário.

A composta de uma homotetia em um dado sistema de eixos ortogonais com a origem em seu centro O com outra transformação qualquer será obtida diretamente tanto pela multiplicação do fator f , como da matriz $\begin{pmatrix} f & 0 \\ 0 & f \end{pmatrix}$, pelas matrizes que definem as demais transformações.

Neste trabalho, adotamos a notação $H_{(O, f)}$ para definir uma homotetia de centro O e razão f . Cabe também ressaltar que para a construção de fractais os fatores de homotetia utilizados devem sempre resultar em *redução* de escala, portanto, $f \in]0, 1[$.

Rotações

Definimos a rotação como uma transformação $R: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ associada a um centro O e um ângulo θ , medido no sentido anti-horário em dado sistema de eixos ortogonais no \mathbb{R}^2 .

Dado um sistema OXY de eixos ortogonais, com origem no centro O da rotação, esta leva o ponto $P = (x, y)$ no ponto $P_1 = (x_1, y_1)$ segundo as seguintes equações:

$$\begin{cases} x_1 = x \cdot \cos \theta - y \cdot \text{sen} \theta \\ y_1 = x \cdot \text{sen} \theta + y \cdot \cos \theta \end{cases} \quad \text{ou} \quad \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} \cos \hat{e} & -\text{sen} \hat{e} \\ \text{sen} \hat{e} & \cos \hat{e} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Em particular, a rotação preserva distâncias e ângulos. Dessa forma, ao aplicarmos uma rotação a uma dada figura, sua imagem será congruente à figura original. Adotaremos a notação $R_{(O, \theta)}$ para definir uma rotação de centro O e ângulo θ .

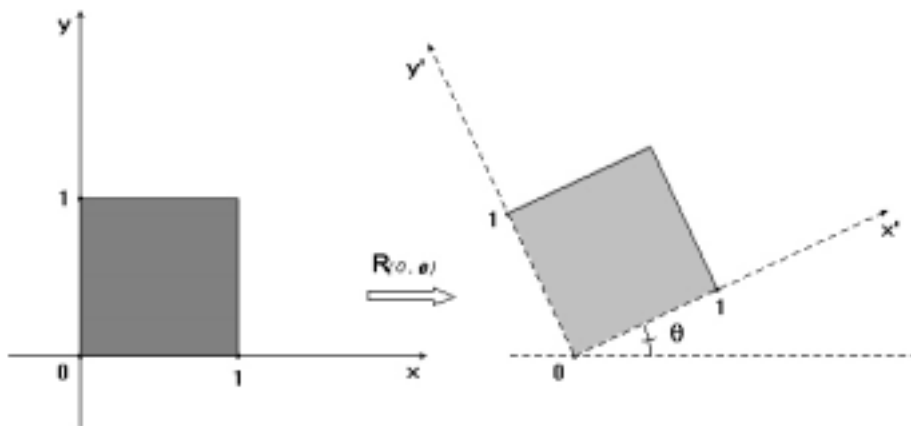


Figura I.2: Rotação de centro O e ângulo θ , aplicada em um quadrado de lado unitário.

A composta de uma rotação em um dado sistema de eixos ortogonais, com origem em seu centro O , com outras transformações quaisquer será obtida diretamente pela multiplicação da matriz $\begin{pmatrix} \cos \theta & -\text{sen} \theta \\ \text{sen} \theta & \cos \theta \end{pmatrix}$ pelas demais matrizes que definem a transformação composta.

Reflexões

A reflexão em torno de um dado eixo r é definida como uma transformação S que faz corresponder a cada ponto A do plano o ponto $A' = S(A)$, simétrico de A em relação à r . Definimos o ponto A' como “*simétrico*” do ponto A em relação ao eixo r , quando r é a mediatriz do segmento AA' . O eixo r pode ser definido tanto a partir de uma reta qualquer como de um segmento no qual se apóie.

Seja um OXY um sistema de eixos ortogonais no plano. A reflexão S , em torno do eixo r , que passa pela origem e faz um ângulo θ com o eixo OX , transforma o eixo OX no eixo OX' , obtido de OX por rotação de ângulo 2θ , e transforma OY no eixo OY' com rotação de ângulo $(180^\circ + \theta)$. Assim, a reflexão S transforma o ponto $A(x, y)$ no ponto $A' = (x_1, y_1)$, tal que:

$$\begin{cases} x_1 = x \cdot \cos 2\theta + y \cdot \sin 2\theta \\ y_1 = x \cdot \sin 2\theta - y \cdot \cos 2\theta \end{cases} \quad \text{ou} \quad \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Esta matriz será particularmente útil na definição de compostas com outras transformações.

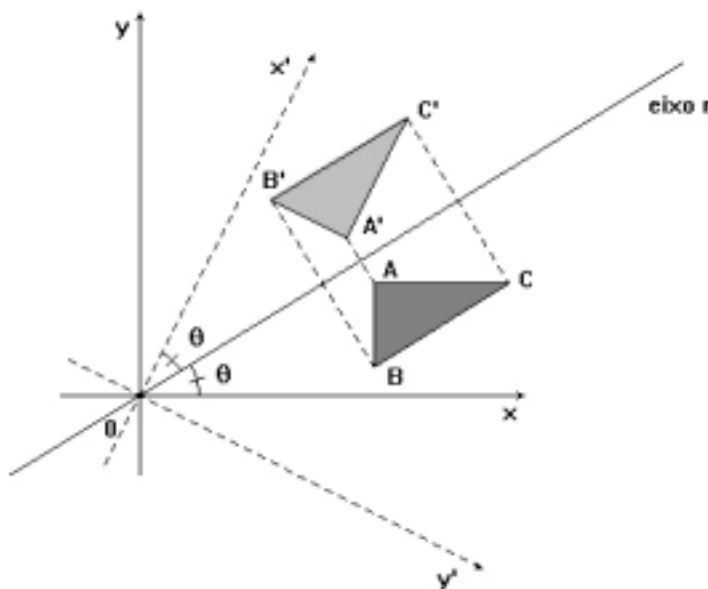


Figura I.3: Reflexão de um ΔABC em torno do eixo r .

Em particular, quando se deseja uma reflexão em torno do próprio eixo OX, teremos $\theta = 0^\circ$, então:

$$\begin{pmatrix} \cos 0^\circ & \text{sen } 0^\circ \\ \text{sen } 0^\circ & -\cos 0^\circ \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Analogamente, quando se deseja uma reflexão em torno do eixo OY, teremos $\theta = 90^\circ$, então:

$$\begin{pmatrix} \cos 180^\circ & \text{sen } 180^\circ \\ \text{sen } 180^\circ & -\cos 180^\circ \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

Utilizamos a notação $S_{(\overline{AB})}$ para representar uma reflexão (ou “simetria axial”) em torno de um eixo definido por um segmento $\overline{\hat{A}\hat{A}}$ dado, supondo que o segmento $\overline{\hat{A}\hat{A}}$ se apóie sobre o eixo r.

Translações

A translação $T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, determinada por um vetor “ v ”, é a transformação que leva cada ponto P do \mathbb{R}^2 no ponto $T(P) = P + v$. Adotaremos a notação $T_{(\overline{AB})}$ supondo a possibilidade de definir-se o vetor v em função de um segmento AB dado, ou seja, iremos considerar $v = \overrightarrow{AB}$.

Se, num dado sistema de eixos ortogonais, as coordenadas de v são (α, β) então, para cada ponto $P = (x, y)$ tem-se $T(P) = (x + \alpha, y + \beta)$. Dessa forma, a translação T transforma toda figura F numa figura $T(F)$ cujos pontos $P + v$ são obtidos transladando-se os pontos P de F pelo vetor v .

Um sistema de eixos ortogonais OXY é transformado por T num sistema $O'X'Y'$, cujos eixos são paralelos e têm o mesmo sentido de OX e OY .

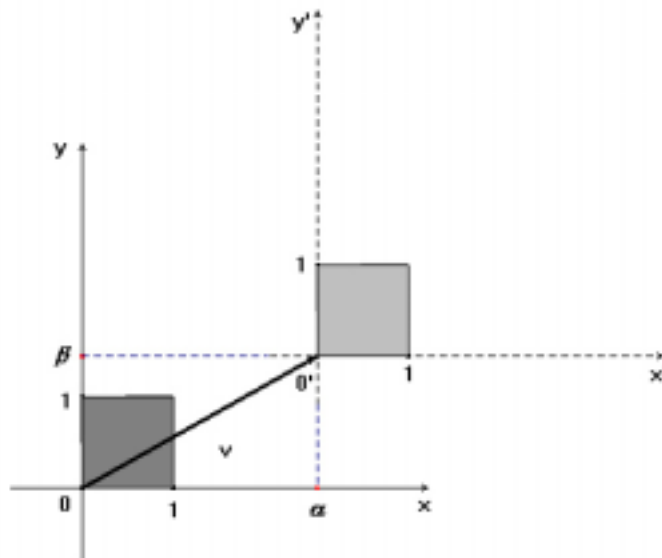


Figura I.4: Translação de um quadrado de lado unitário, a partir de um vetor $v = (\alpha, \beta)$.

Sejam as coordenadas $T(P) = (x_1, y_1)$ da imagem do ponto $P = (x, y)$ no sistema de eixos OXY , transladado pelo vetor $v = (\alpha, \beta)$. As equações que definem a translação T são dadas por:

$$\begin{cases} x_1 = x + \alpha \\ y_1 = y + \beta \end{cases} \quad \text{ou} \quad \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Estas equações podem ser representadas pelo vetor coluna $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ que poderá ser utilizado na determinação de transformações compostas.

ANEXO II

MODELO DAS FICHAS DE ATIVIDADE PARA A PROPOSTA DE SITUAÇÕES DE ENSINO

Parte 1 : Fractais no *GeomeTricks*

➤ *Exemplo* : Construindo o “Triângulo de Sierpinski”

1) Abra o arquivo “**sierp.tri**” a partir do *GeomeTricks*.

Neste arquivo, os pontos A, B e C representam o ΔABC (chamado de “*triângulo de controle*”). Os pontos C, D e E são os pontos médios de cada um dos lados do triângulo de controle ABC (Fig 1a).



Fig. 1: Conjunto dos pontos para a construção do triângulo de Sierpinski e o nível I de iterações da construção a partir do *GeomeTricks*.

2) Na barra de ferramentas, selecione a opção “**Fractais**” > “**Definir Fractal**”. O programa mostrará uma janela pedindo o número de ternas. Para esta figura utilizaremos **quatro ternas** (digite “4” e clique em “OK”).

A construção de fractais no *GeomeTricks* se baseia na combinação de **ternas ordenadas de pontos**, ou seja, o **número de ternas** e a **ordem dos pontos** de cada terna definem o fractal.

3) O programa espera que as **4 ternas** sejam selecionadas.


Como a **ordem** dos pontos escolhidos em cada terna é essencial para a construção do fractal, clique os pontos na seguinte ordem:

- **ABC; AFD, FBE e DEC**

4) Tente construir os primeiros níveis de iteração para esta figura.

Para isso selecione “**Fractais**” > “**Níveis**”. O programa mostrará uma janela pedindo o número de **níveis** desejado. Escolha alguns valores inteiros para ver o resultado da construção.

Notas:

- Após concluir uma figura, pare a construção clicando no botão 
- Você pode mudar a cor das figuras geradas, clicando as cores desejadas na paleta de cores no canto superior direito da tela.

5) Depois de visualizar alguns níveis, você está pronto para construir o fractal propriamente dito.

Para isso apague os níveis desenhados na opção “**Fractais**” > “**Apagar desenho do fractal**”. Feito isso, selecione “**Fractais**” > “**Desenhar Fractal**” para ver o resultado final da construção.

➤ **Análise matemática da construção**

A construção de fractais a partir do *GeomeTricks* se baseia em Transformações Geométricas que são definidas a partir da **posição inicial** dos pontos que irão gerá-lo (fornecidos na Fig. 1a) e da **ordem** na qual esses pontos são selecionados em cada terna.

Assim, a primeira terna selecionada define um “**triângulo de controle**” e cada uma das demais ternas selecionadas define uma **composição de transformações geométricas** que será aplicada ao “triângulo de controle”.

Dessa forma, a terna ABC define o “triângulo de controle”, no qual serão feitas as composições de transformações definidas pelas demais ternas e a **posição** e **ordem** dos pontos das demais ternas irão definir quais composições de transformações geométricas serão aplicadas ao “triângulo de controle”, como pode ser visto na Fig. 1b, ou seja:

- A transformação (ABC) → (AFD) leva o **triângulo de controle** no **triângulo I**;
- A transformação (ABC) → (FBE) leva o **triângulo de controle** no **triângulo II**;
- A transformação (ABC) → (DEC) leva o **triângulo de controle** no **triângulo III**;

Observe que estas transformações são aplicadas iterativamente pelo software, fazendo com que, em cada nível de iteração, as composições de transformações geométricas sejam aplicadas em todos os triângulos gerados pelo nível anterior.

Você poderá constatar estas informações se construir, por exemplo, o **nível 2** de iterações, onde todas as transformações são aplicadas a cada um dos três novos triângulos gerados no nível anterior.

O **fractal final** será gerado após “*infinitas*” iterações.

➤ **Atividade I : Construindo a “Cauda do Dragão”**

1) Agora você está pronto para construir seu primeiro fractal. Para isso, abra o arquivo “**dragon.tri**” (Fig. 2a).

Nele serão encontrados oito pontos com os quais faremos a construção do fractal conhecido como “**Cauda do Dragão**” (Dragon’s Tail), apresentado na Fig. 2c.

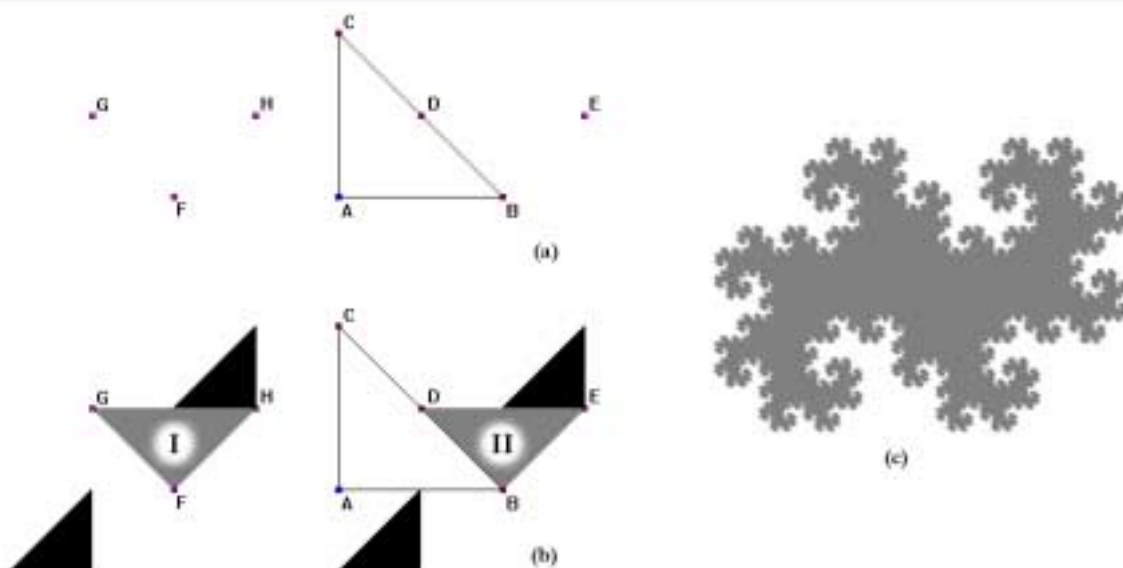


Fig. 2: Conjunto dos pontos para a construção da Cauda do Dragão, os dois primeiros níveis de iteração e o fractal final gerado pelo GeomeTricks.

2) A partir do triângulo de controle **ABC**, iremos construir o fractal utilizando **duas composições de transformações** realizadas em duas outras ternas. Portanto, iremos construir um fractal com **3 ternas**, sendo que a primeira gera o triângulo de controle **ABC** (Fig. 2a).

3) Observe os níveis de iteração 1 (**cinza**) e 2 (**preto**) na Fig. 2b. É a partir desta figura que você irá definir as composições de transformações geométricas que definirão a ordem correta das ternas para gerar o fractal.

➤ **Desafios :**

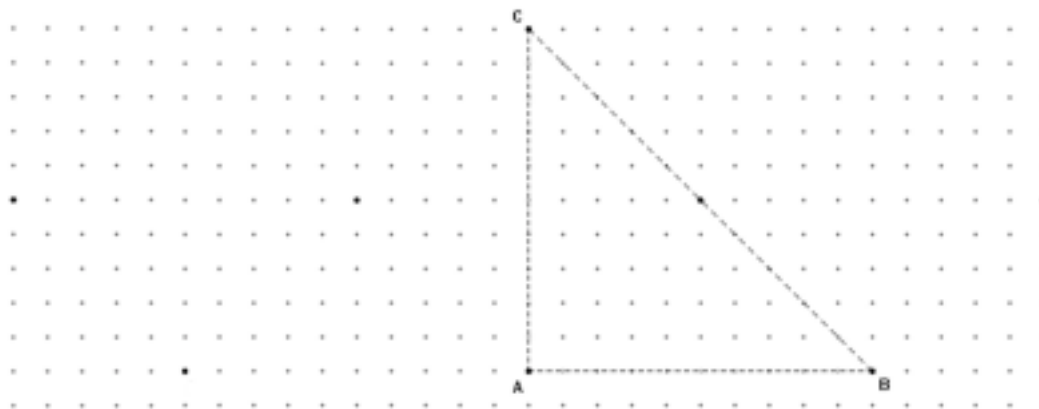
Para se determinar a ordem correta das ternas, é necessário identificar as **duas composições de transformações geométricas** realizadas na construção do fractal, assim sendo, você deve:

(i) Definir a primeira composição de transformações geométricas sabendo que serão necessárias **três** delas para transformar o **triângulo retângulo isósceles ABC** no **triângulo retângulo isósceles I**. Você poderá contar com o esboço em escala dos pontos, dado abaixo.

Em cada transformação, você deve identificar seus elementos característicos, ou seja:

- o **centro** e o **fator** (ou razão) para *Homotetias*;
- o **centro** e o **ângulo** para *Rotações*;
- o **eixo** para *Simetrias Axiais* (ou *Reflexões*);
- o **vetor** para *Translações*.

Descrição das transformações geométricas :



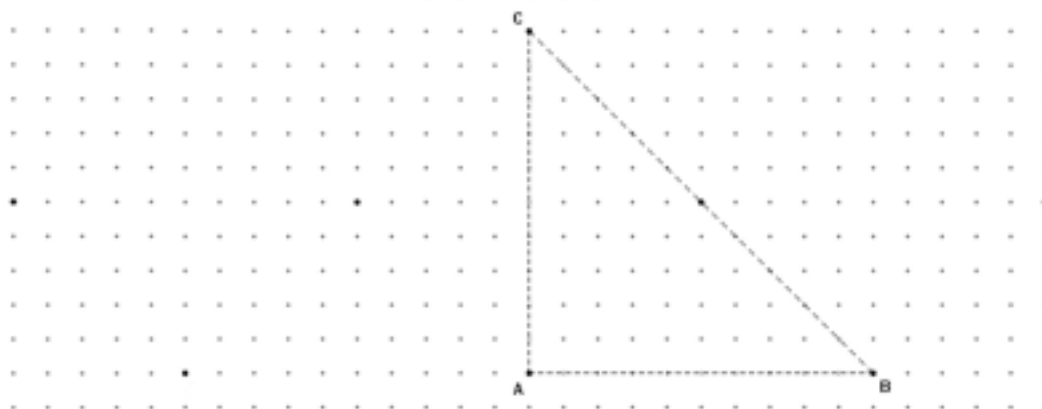
- 1ª Transformação:

- 2ª Transformação:

Primeira Composição:

(ii) Com o mesmo raciocínio, defina a segunda composição de transformações geométricas, sabendo que também serão necessárias três transformações para transformar o **triângulo retângulo isósceles ABC** no **triângulo retângulo isósceles II**, definindo assim a ordem correta das ternas que compõe o fractal.

Descrição das transformações geométricas :



- 1ª Transformação:

- 2ª Transformação:

- 3ª Transformação:

Segunda Composição:

Ordem das ternas :

(iii) Agora, a partir das transformações que você descreveu, defina as ternas e a ordem correta de seus pontos. Teste seu resultado construindo os **dois** primeiros níveis (comparando-os com a Fig. 2b) e construa a **Cauda do Dragão** (Fig. 2c).

➤ **Desafio Extra :** Qual será a combinação correta das ternas se o triângulo de controle for definido com a terna **CAB**?

Ordem das ternas :

➤ **Atividade II : Construindo o “Piso Fractal”**

1) Nesta atividade, vamos construir o fractal conhecido como “*Piso Fractal*” (Fractal Floor). Para isso, utilizaremos **o mesmo triângulo de controle** e os **mesmos pontos** da atividade anterior, mas com novos conjuntos de transformações.

Iremos adotar as seguintes notações para a descrição das composições de transformações :

- $H_{(A, f)}$: que define uma **homotetia** ou “mudança de escala” de centro “A” e fator “f”.
- $S_{(\overline{AB})}$: que define uma **simetria axial** sobre um eixo definido pelo segmento “ \overline{AB} ”.
- $R_{(A, \theta)}$: que define uma **rotação** de centro “A” e ângulo “ θ ” ($0^\circ \leq \theta \leq 360^\circ$), anti-horário.
- $T_{(\overline{AB})}$: que define uma **translação** de vetor “ \overline{AB} ”.

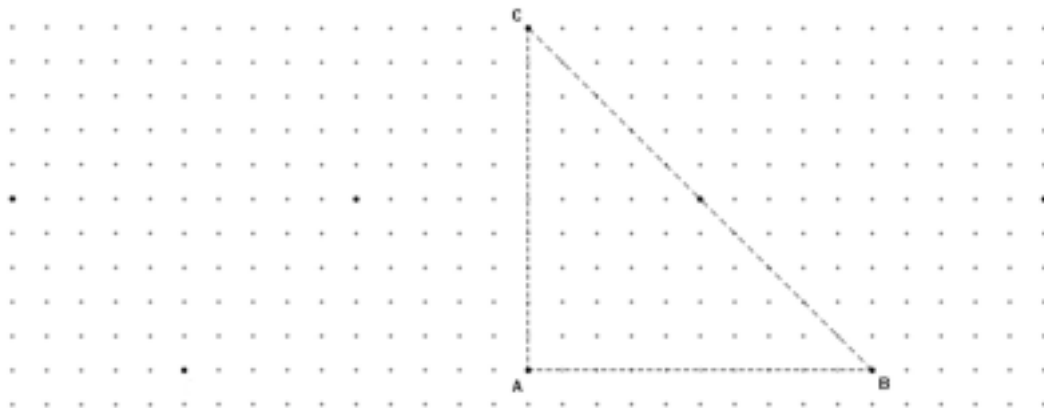
2) Estão descritas abaixo as duas composições de transformações geométricas necessárias para a construção do “*Piso Fractal*”.

$$W_I = T_{(\overline{AB})} \circ S_{(\overline{AC})} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$$

$$W_{II} = T_{(\overline{AF})} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$$

➤ **Desafios:**

(i) Sua tarefa é desenhar os dois primeiros níveis de iterações do fractal e assim determinar a ordem correta das ternas com base nas transformações geométricas dadas acima.



Ordem das ternas :

(ii) Agora, tente criar o *Piso Fractal* a partir das ternas que você determinou, definindo um novo fractal com **3 ternas** no *GeomeTricks*.

Teste os níveis desenhados e construa o “*Piso Fractal*”.

Que relações você percebe, tanto nas composições de transformações como na disposição final das ternas dos fractais “*Cauda do Dragão*” e “*Piso Fractal*” ?

Parte 2 : Fractais no *MicroWorlds LOGO*

Nesta fase da atividade, pretendemos explorar a construção de fractais a partir da linguagem de programação *LOGO*. Esta linguagem irá se basear em **sistemas de funções iteradas** para a construção dos fractais.

➤ *Atividade III* : Construindo o “Piso Fractal”

A construção do **Piso Fractal** com o *LOGO* parte das transformações geométricas já utilizadas no *GeomeTricks*, porém, estas transformações deverão ser traduzidas para um **sistema de funções iteradas** nas qual cada função representa uma dada composição de transformações geométricas.

Este sistema de funções deverá ser gerado por **transformações** representadas por funções algébricas. Portanto, nossa tarefa será a de “traduzir” as **composições de transformações geométricas** para um **sistema algébrico**.

Assim, partiremos das **transformações geométricas** já definidas na *Atividade II*:

$$W_I = T_{(AB)} \circ S_{(AC)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$$

$$W_{II} = T_{(AF)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$$

Inicialmente, iremos utilizar a **forma matricial** para representar esse sistema. Todas as transformações utilizadas serão do tipo $T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, onde $T(\mathbf{u}) = \mathbf{w} \cdot \mathbf{u} + \mathbf{t}$, onde “ \mathbf{u} ” representa uma coordenada (x, y) do \mathbb{R}^2 , “ \mathbf{w} ” é uma composição de transformações lineares e “ \mathbf{t} ” é uma translação.

Visto que todas as transformações utilizam o **ponto A** da figura como referencial, teremos as seguintes representações:

- **Homotetia** : Para a construção de um fractal, a homotetia deverá sempre resultar em **redução de escala** e, visto que essa transformação será representada por um fator “ \mathbf{f} ”, deveremos ter $\mathbf{f} \in]0 ; 1[$.

Portanto, a homotetia $H_{(A, \sqrt{2}/2)}$ irá ser representada por: $\mathbf{f} = \frac{\sqrt{2}}{2}$.

- **Rotação** : Para realizarmos a rotação $R_{(A, 45^\circ)}$, considerando o sentido anti-horário, devemos utilizar a seguinte matriz:

$$\begin{bmatrix} \cos 45^\circ & -\sin 45^\circ \\ \sin 45^\circ & \cos 45^\circ \end{bmatrix}$$

- **Simetria Axial** : Para realizarmos a simetria axial $S_{(AC)}$ (a simetria é aplicada em relação ao eixo y), devemos utilizar a seguinte matriz:

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

- **Translação** : A translação “ t ” será dada por uma matriz coluna com dois elementos que representam, respectivamente, a translação em relação ao eixo x (b_1) e em relação ao eixo y (b_2).

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Os valores de b_1 e b_2 representam uma **proporção** entre os módulos dos vetores que os geram. Dessa forma, como a translação ocorre somente na horizontal (eixo x) e os vetores têm o mesmo módulo com sentidos contrários, teremos:

$$T_{(AD)} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad e \quad T_{(AB)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

A partir dessas informações, é possível traduzir as **composições de transformações geométricas** para um **sistema algébrico de transformações**.

➤ **Desafio** : Determine os **coeficientes** resultantes das **composições**, utilizando as matrizes dadas.

$$W_I = T_{(AB)} \circ S_{(AC)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$$

$$W_{II} = T_{(AF)} \circ R_{(A, 45^\circ)} \circ H_{(A, \sqrt{2}/2)}$$

De posse da forma matricial do **sistema de funções iteradas** que gera o **Piso Fractal**, executamos sua tradução para a representação algébrica tradicional, criando as duas **Regras de cálculo** abaixo:

$$\text{(Regra1)} \quad x_n = -0,5x + 0,5y + 1 \quad ; \quad y_n = 0,5x + 0,5y$$

$$\text{(Regra2)} \quad x_n = 0,5x - 0,5y - 1 \quad ; \quad y_n = 0,5x + 0,5y$$

Finalmente, fazemos a tradução da representação algébrica acima para a sintaxe da linguagem **LOGO**, como segue:

- Representação Algébrica

- Sintaxe na Linguagem LOGO

Regra1 $x_n = -0,5x + 0,5y + 1$

make "xn $-0.5 * :x + 0.5 * y + 1$

$y_n = 0,5x + 0,5y$

make "x :xn

make "y $0.5 * :x + 0.5 * :y$

Regra2 $x_n = 0,5x - 0,5y - 1$

make "xn $0.5 * :x - 0.5 * y - 1$

$y_n = 0,5x + 0,5y$

make "x :xn

make "y $0.5 * :x + 0.5 * :y$

Os procedimentos para a criação do **“Piso Fractal”** já se encontram prontos no arquivo **“piso.txt”**. É possível utilizá-los com as ferramentas **“copiar”** e **“colar”** do próprio Windows, seguindo as instruções abaixo:

1) No **“Bloco de Notas”** do Windows, abra o arquivo **“piso.txt”** e clique em **“Editar”** > **“Selecionar tudo”**.

Feito isso, copie o conteúdo selecionado com o comando **“Copiar”** ou com o atalho **“Ctrl+C”**.

2) No software **“MicroWorlds”**, selecione a página de procedimentos clicando em **“Pages”** > **“Procedures”**.

Feito isso, **“cole”** o conteúdo do arquivo com o comando **“Editar”** > **“Colar”** ou com o atalho **“Ctrl+V”**.

3) Para executar o programa, volte para a **página de desenhos** do software clicando em **“Pages”** > **“Page1”**.

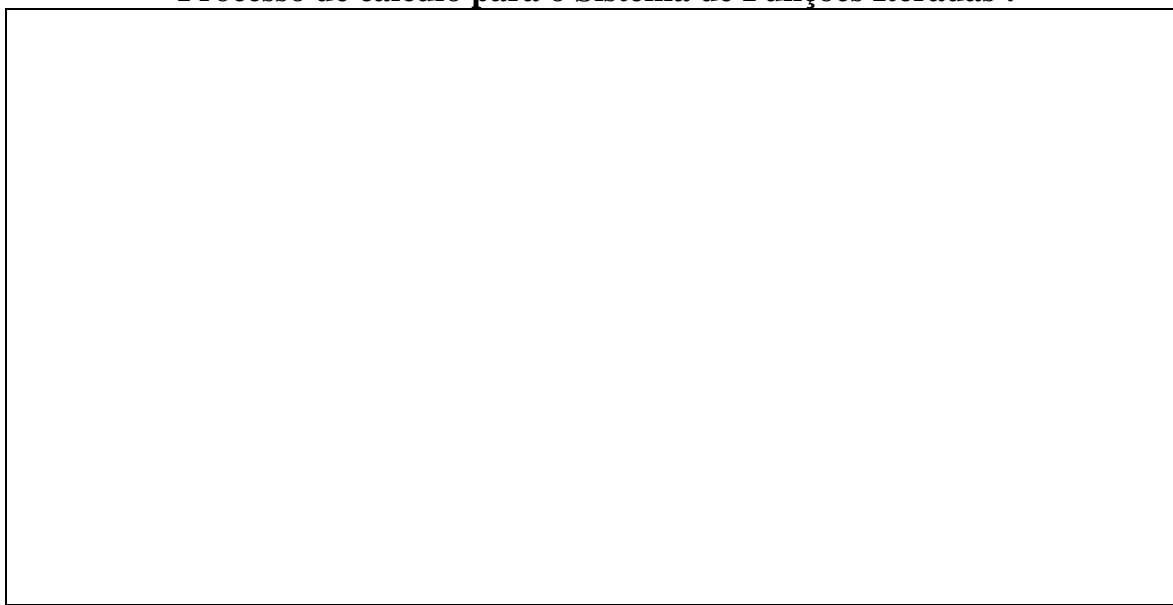
4) Construa o **“Piso Fractal”** digitando o comando **“piso 20000”** na barra de comandos (quadro cinza) situada na parte inferior da tela do **“MicroWorlds”**.

➤ **Atividade IV: Construindo a “Cauda do Dragão”**

1) Determine os dois **sistemas de funções iteradas** necessárias para a construção da “*Cauda do Dragão*”.

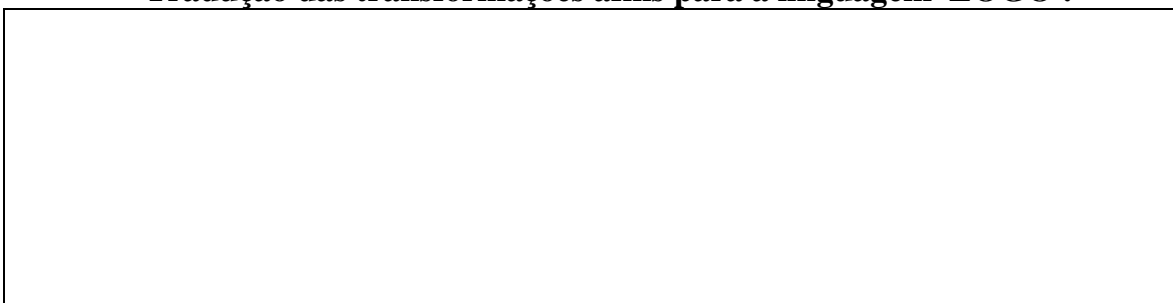
Esta tarefa deve ser descrita passo a passo abaixo, a partir das transformações geométricas já desenvolvidas na *Atividade I*, com o *GeomeTricks*.

Processo de cálculo para o Sistema de Funções Iteradas :



2) Agora, “traduza” o sistema de funções para a linguagem *LOGO*.

Tradução das transformações afins para a linguagem LOGO :



3) Teste o resultado obtido, substituindo os novos coeficientes no programa para a construção do “*Piso Fractal*”.

Utilize os mesmos procedimentos do **exemplo**, apenas substituindo os novos valores que foram determinados acima nos procedimentos de cálculo “**Regra1**” e “**Regra2**”.

4) Construa o fractal “*Cauda do Dragão*” com os novos procedimentos.

Nota: Por tratar-se de uma figura com maior densidade de pontos, serão necessários cerca de **100 000 pontos** para uma boa visualização do fractal.