

# Lição de Síntese

# Os desafios da Modelação Procedimental de

# **Ambientes Urbanos**

António Fernando V. C. C. Coelho Professor Auxiliar do DEI/FEUP



Sunrise by the Ocean - Vladimir Kush

"O desenvolvimento de um organismo pode ser considerado como a execução do 'programa de desenvolvimento' presente no ovo fertilizado... A principal tarefa da Biologia do Desenvolvimento é descobrir o algoritmo subjacente..."

Aristid Lindenmayer

Sumário pormenorizado da Lição de Síntese no âmbito das provas de agregação no ramo de conhecimento em Engenharia Informática, e de acordo com o Decreto-Lei no 239/2007, de 19 de junho, que aprova o regime jurídico do titulo académico de agregado.

# Resumo

As ferramentas de modelação procedimental podem ter um papel muito relevante para um conjunto diversificado de profissionais que operam sobre ambientes urbanos. Desde o planeamento urbano à educação, passando pelo turismo e pela indústria do entretenimento, estas ferramentas trazem maior produtividade.

As ferramentas de modelação procedimental geram, de forma automática, modelos tridimensionais através de processos computacionais que ampliam a informação base segundo um determinado algoritmo. De forma a não ser necessário reprogramar os sistemas de modelação procedimental, recorre-se a métodos matemáticos que operam sobre descrições simbólicas parametrizáveis que, de forma flexível, são capazes de modelar diferentes tipos de objetos. Os métodos matemáticos mais utilizados são a geometria fractal e as gramáticas formais, particularmente os sistemas L e as gramáticas de formas.

Desde as primeiras abordagens que os métodos e ferramentas têm vindo a evoluir no sentido de gerar modelos tridimensionais mais detalhados e com maior nível de fidelidade visual. Os sistemas L geoespaciais e as gramáticas CGA foram evoluções das primeiras abordagens à modelação procedimental de ambientes urbanos que, posteriormente, deram lugar a ferramentas comerciais. No entanto, apresentam limitações em alguns dos elementos constituintes desses ambientes, fomentando o desenvolvimento de novas soluções específicas para a modelação dos terrenos, da vegetação, da rede viária e dos edifícios. Algumas destas soluções já se encontram integradas nestas ferramentas comerciais, no entanto há ainda problemas e limitações que deixam em aberto algumas linhas de investigação.

# Índice

E	nquadı	ramento	1
1	Intr	odução	2
2		Fundamentos	
_	2.1	Geometria Fractal	
	2.2	Sistemas L	
	2.2.1		
	2.2.2		
	2.2.3	,	
	2.2.4		
	2.3	Gramáticas de Formas	
3	Mod	delação Procedimental de Ambientes Urbanos	12
J	3.1	Modelação do Terreno	
	3.2	Modelação da Vegetação	
	3.3	Modelação da Rede Viária	
	3.4	Modelação de Edifícios	
4	Ten	dências para o futuro	26
5	Cor	- nclusões	27
R	eferênc	cias	28
A	nexo A	- Classes de Sistemas L	32
A1. Sistemas D0L  A2. Sistemas L Sensíveis ao Contexto			
	A4. Sis	35	
	A5. Sis	stemas L Paramétricos	37

# Enquadramento

Esta lição de síntese foca-se sobre a Modelação Procedimental de Ambientes Urbanos, um dos tópicos de investigação que tem vindo a ser desenvolvido pelo autor, desde a sua tese de doutoramento. Esta lição enquadra-se no âmbito de uma unidade curricular de 2º ciclo na área da Engenharia Informática, em particular na área científica e disciplinar da Computação Gráfica ou no âmbito de uma unidade curricular de seminário do 3º ciclo.

A opção pelo tema da lição derivou dos diversos trabalhos de investigação desenvolvidos e supervisionados pelo autor e que têm grande relevância na atualidade para os profissionais e investigadores, da área da Computação Gráfica. Efetivamente esta lição vai servir de introdução para o tópico de Modelação Procedimental da unidade curricular Modelação Gráfica e Procedimental do Mestrado em Computação Gráfica, oferecido conjuntamente pela Universidade Aberta, pela Universidade do Porto e pela Universidade de Coimbra, a iniciar no ano letivo 2018/19. Por esta razão optou-se pelo formato lição de síntese em vez de seminário.

De salientar que a síntese que é feita nesta lição deriva de uma linha de investigação que contou com as contribuições das teses de doutoramento desenvolvidas por Pedro Silva, Carlos Campos e Diego Jesus, bem como os projetos de investigação 3D4LBMS, 3DWikiU e ERAS liderados pelo colega Augusto de Sousa. Pelo que ao longo da lição estas contribuições irão ser evidenciadas.

Pretende-se que no final desta lição o estudante seja capaz de:

- Identificar as áreas de aplicação da modelação procedimental de ambientes urbanos e as vantagens da sua utilização;
- Descrever os conceitos base e os fundamentos matemáticos da modelação procedimental;
- Explicar o processo de modelação dos sistemas L e das gramáticas de formas;
- Discutir e relacionar as diferentes abordagens e métodos desenvolvidos para a modelação procedimental de ambientes urbanos e de partes destes ambientes, como o terreno, a vegetação, a rede viária e os edifícios;
- Identificar as principais tendências de desenvolvimento.

# A lição apresenta a seguinte estrutura:

- A secção 1 apresenta a introdução da lição, focando-se na motivação para a modelação procedimental de ambientes urbanos e a sua definição;
- A secção 2 explica os fundamentos matemáticos que servem de base aos principais métodos de modelação procedimental, abordando a Geometria Fractal e as gramáticas formais, tais como os Sistemas L e as Gramáticas de Formas;
- A secção 3 descreve os métodos de modelação procedimental utilizados para os ambientes urbanos, classificando os métodos desenvolvidos de acordo com os principais elementos constituintes dos ambientes urbanos: terreno, vegetação, rede viária e edifícios;
- Finalmente a secção 4 apresenta as tendências futuras e conclusões.

# 1 Introdução

Os ambientes urbanos são um ecossistema de importância fundamental para a evolução da sociedade. A utilização de modelos tridimensionais destes ambientes urbanos é essencial para o desenvolvimento de diversas atividades como a gestão e o planeamento, mas também para fomentar outras como o Turismo, a Educação ou o Entretenimento. No entanto, a grande extensão e a complexidade inerente a estes ambientes tornam impraticável (ou excessivamente dispendiosa¹) a sua modelação rigorosa de forma não automática.

A modelação procedimental tem origens na década de 80, quando a quantidade de memória disponível nos computadores era insuficiente para o nível de complexidade de modelação tridimensional que se pretendia. São exemplos os jogos "Elite" e "The Sentinel". Daí ter surgido a necessidade de se criarem algoritmos que gerassem de forma automática, representações tridimensionais de modelos virtuais em tempo de execução do programa, surgindo o termo "procedimental".

Pode-se assim definir Modelação Procedimental como a geração automática de modelos tridimensionais através de processos computacionais que ampliam a informação base segundo um determinado algoritmo. Estes algoritmos implementam conjuntos de regras ou são parametrizáveis, integrando frequentemente aleatoriedade.

A modelação procedimental surge como uma necessidade para ultrapassar dificuldades inerentes ao processo de modelação tradicional, proporcionando diversas vantagens:

- Maior produtividade, tanto pela redução de custos com a utilização de equipas de especialistas mais reduzidas, como pela redução do tempo associado às tarefas de modelação de modelos extensos. A um outro nível é potenciador de pequenas empresas (ex. startup), possibilitando que sejam capazes de produzir produtos (ex. jogos) para os quais não teriam capacidade sem estas ferramentas.
- Maior iteração, através da possibilidade de obter de forma automática várias soluções para um determinado problema (ex. planeamento urbano) com um esforço adicional reduzido. Também se inclui a capacidade de utilizar novas formas de expressividade, potenciando o alargamento desta área a profissionais de outras áreas.
- Compressão de informação, possibilitando criar aplicações de menor dimensão que gerem dinamicamente os modelos tridimensionais à medida que são necessários (ex. níveis de um jogo). É também possível criar "mundos infinitos" que vão sendo gerados de forma ilimitada à medida que o jogador vai evoluindo no jogo.
- Integração com bases de informação diversificadas, através da conversão dos formatos em que essa informação está definida e integrando-a no processamento algorítmico. Um exemplo típico é a utilização de informação geográfica, através da integração com SIG (Sistemas de Informação Geográfica).

A evolução da área da modelação procedimental tem uma base consolidada em teorias e modelos matemáticos, que são descritos de seguida, e tem tido uma evolução muito ampla no sentido de tentar responder às complexidades inerentes aos ambientes urbanos. Esses modelos têm dado origem a diversos métodos que possibilitam uma modelação integrada das cidades. Ao longo desta lição descrevem-se esses métodos e os fatores que fomentaram a sua evolução, numa perspetiva histórica para o entendimento do estado da arte. Irão também ser abordadas de forma mais específica, soluções que têm vindo a ser desenvolvidas para os principais

<sup>&</sup>lt;sup>1</sup> No caso da indústria dos videojogos, há exemplos como o *Grand Theft Auto V*, em que estes modelos tridimensionais têm sido desenvolvidos com grande intervenção humana, embora com custos de várias dezenas de milhões de USD.

elementos que compõem uma cidade, como o terreno, a vegetação, a rede viária e o edificado. Finalmente, abordam-se as principais tendências no âmbito da modelação procedimental.

# 2 Fundamentos

A modelação procedimental assenta num conjunto de ferramentas e métodos matemáticos para elevar o nível de flexibilidade da definição dos processos, evitando a programação recorrente dos algoritmos. Os primeiros trabalhos de modelação procedimental assentaram sobre a geometria fractal, particularmente na modelação de terrenos, e sobre gramáticas formais, sobretudo na modelação de plantas. Abordaremos com maior detalhe os Sistemas L, fazendo posteriormente a abordagem às Gramáticas de Formas, como base dos processos de modelação procedimental mais utilizados na atualidade.

## 2.1 Geometria Fractal

A geometria fractal é o ramo da matemática que estuda as propriedades e comportamento dos fractais. Surgiu com o objetivo de descrever situações que não podem ser explicadas pelas definições tradicionais da geometria euclidiana. Estas situações ocorrem mais frequentemente em fenómenos naturais. A título de exemplo, se tentarmos medir o contorno de uma montanha vista ao longe e fizermos novamente essa medição com a utilização de binóculos, onde podemos observar melhor as irregularidades do terreno, obtemos comprimentos diferentes, o que contraria a geometria euclidiana.

Um fractal é um objeto geométrico que pode ser fragmentado em partes, e cada uma destas partes tem algum tipo de semelhança com o objeto fragmentado. O termo foi definido por Benoît Mandelbrot, matemático francês que descobriu a geometria fractal na década de 70. O conjunto de Mandelbrot é um exemplo típico de um fractal que exibe semelhança (Figura 1).

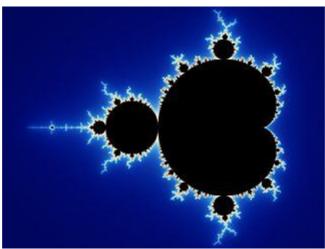


Figura 1 - Conjunto de Mandelbrot

Outro conceito fundamental é o da dimensão fractal, que é superior à dimensão topológica do objeto. Isto quer dizer que um objeto fractal escala de forma diferente de um objeto euclidiano. Por exemplo, sendo uma linha um objeto unidimensional, quando dividido em três partes cada uma terá 1/3 do seu comprimento. No entanto uma linha fractal, como a curva "floco de neve" (Figura 2) também é fragmentada, neste caso em 4 segmentos de reta, cada um com 1/3 de comprimento da linha original. Ou seja, a cada iteração, a curva tem um comprimento de 4/3 do segmento original. Isto quer dizer que o perímetro deste objeto tende para infinito, enquanto que a sua área tende para um valor finito.

Os fractais podem ser gerados através de diversos processos, desde funções recursivas a sistemas L, que iremos abordar na secção seguinte.

### 2.2 Sistemas L

Os sistemas L (acrónimo de Sistemas Lindenmayer - *L-systems* no original) surgiram em 1968 num artigo intitulado "*Mathematical models for cellular interaction in development*" [Lin68], publicado por um botânico holandês de nome Aristid Lindenmayer, no *Journal of Theoretical Biology*. Nesse artigo propõe-se um sistema para a modelação do processo de desenvolvimento das plantas, baseado numa nova classe de sistemas de produção desenvolvido por Lindenmayer.

O conceito que esteve na base do desenvolvimento dos sistemas L foi a teoria morfogénica<sup>2</sup> da emergência na vida multicelular [Tay92]. O conceito de emergência reflete um processo de desenvolvimento, no qual um conjunto de unidades interactuantes adquire, de forma qualitativa, novas propriedades que não poderiam ser deduzidas através de uma simples sobreposição das suas contribuições individuais. Da mesma forma que, por exemplo, partindo de uma simples semente, se obtém uma estrutura tão complexa como uma árvore, as diferentes células que constituem um determinado ser vão-se multiplicando e especializando, de acordo com a sua disposição, dando lugar aos diferentes órgãos que formam o seu corpo.

O ponto forte dos sistemas L é assim o seu potencial para a ampliação de dados [Smi84], devido ao facto de a partir de um conjunto inicial de dados simples ser possível obter-se um conjunto de dados com uma complexidade muito superior.

#### 2.2.1 Conceitos Fundamentais

As gramáticas formais são um conjunto de regras de produção numa linguagem formal. As regras de produção descrevem como formar cadeias válidas a partir do alfabeto da linguagem e de acordo com a sintaxe da linguagem. Aplicações das gramáticas formais podem ser encontradas nas Ciências da Computação, linguística ou Matemática. A título de exemplo, a gramática BNF (Backus-Naur Form) é utilizada para a definição de linguagens de programação.

A abordagem de Aristid Lindenmayer para a descrição morfológica das plantas baseou-se num tipo de gramáticas formais, designados por sistemas de reescrita de cadeias de caracteres, tendo utilizado caracteres para a descrição das diversas partes constituintes das plantas. Neste tipo de sistemas de reescrita são utilizadas regras de produção para operar cadeias de caracteres pertencentes a um determinado alfabeto. Partindo da cadeia de caracteres inicial e aplicando iterativamente estas regras ao longo de uma sequência de passos, é possível modelar a essência do desenvolvimento a nível modular que ocorre nas plantas.

Os conceitos de reescrita e de desenvolvimento modular são centrais à definição de sistemas L, sendo, portanto, fundamentais para a compreensão do seu processo de funcionamento, pelo que se apresentam de seguida.

O mecanismo de reescrita é uma técnica que possibilita a definição de objetos complexos através da sucessiva substituição de partes de um objeto inicial, utilizando um conjunto de regras de reescrita, as regras de produção. A relevância dos sistemas de reescrita foi reconhecida com o trabalho de Chomsky sobre gramáticas formais [Cho56]. A diferença fundamental entre os sistemas L e as gramáticas de Chomsky reside na metodologia de aplicação das regras de produção. Nas gramáticas de Chomsky, as regras de produção são aplicadas sequencialmente, enquanto que no caso dos sistemas L estas são aplicadas em paralelo e simultaneamente a todos os símbolos da cadeia inicial. Esta diferença deve-se ao facto de as gramáticas de Chomsky se basearem no conceito da máquina de Turing, que

<sup>2</sup> A morfogenia é um conjunto de leis biológicas que regulam a produção da forma dos seres e dos seus órgãos, ao longo do seu desenvolvimento.

pressupõe que todo o processamento seja efetuado em série, mantendo apenas um nó ativo a cada momento. Pelo contrário, os sistemas L baseiam-se na teoria morfogénica da emergência da vida multicelular, considerando que a evolução é efetuada paralelamente em diversos nós.

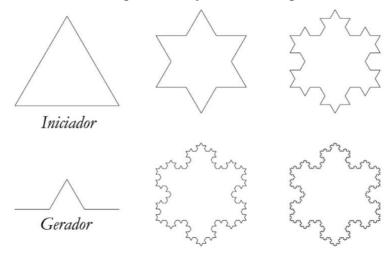


Figura 2 - Construção da curva "floco de neve" utilizando um sistema de reescrita

A Figura 2 ilustra graficamente o princípio de funcionamento de um sistema de reescrita, aplicado à construção da curva "floco de neve"<sup>3</sup>. O sistema inicia-se com dois objetos geométricos: o iniciador e o gerador. A cada passo de desenvolvimento, cada aresta do objeto geométrico em desenvolvimento, a começar com o iniciador, é substituída pela sequência de arestas que compõem o objeto gerador. O processo vai-se repetindo recursivamente até se atingir o resultado gráfico pretendido.

Outro dos conceitos fundamentais dos sistemas L é o de módulo, designando uma unidade construtiva discreta que é repetida à medida que o sistema se desenvolve. Um exemplo concreto deste conceito são os ramos e as folhas no caso da modelação de uma árvore.

A essência do desenvolvimento ao nível modular é captada através de um sistema de reescrita paralela que substitui individualmente determinados módulos, denominados predecessores, por configurações de outros módulos, denominados sucessores. Todos os módulos pertencem a um alfabeto finito de tipos de módulos e todo o processo de desenvolvimento modular é especificado através de um conjunto, também finito, de regras de reescrita (ou de produção).

No seu caso mais simples, os sistemas L de contexto livre, cada regra de produção substitui um módulo predecessor, por zero, um ou mais módulos sucessores. No caso dos sistemas L sensíveis ao contexto, a aplicabilidade de uma determinada regra de produção não depende apenas do módulo predecessor, mas também da sua vizinhança, ou seja, dos módulos posicionados à sua esquerda ou à sua direita na cadeia modular. As regras de produção são aplicadas em paralelo, sendo reescritos todos os módulos simultaneamente, a cada passo de derivação. A sequência de cadeias obtida em passos de derivação consecutivos denomina-se sequência de desenvolvimento.

## 2.2.2 Definição de um Sistema L

Um sistema L é definido por um axioma, definido como sendo a cadeia de caracteres inicial, e por um conjunto de regras de produção que definem como o sistema irá evoluir ao longo da sequência de desenvolvimento.

<sup>&</sup>lt;sup>3</sup> A curva "floco de neve", proposta por von Koch em 1905, é utilizada frequentemente na literatura para ilustrar o princípio de funcionamento dos sistemas de reescrita e é também um fractal.

A título exemplificativo, a Figura 2Figura 3 apresenta um sistema L definido por um axioma, constituído por sete caracteres, e por uma regra de produção. A regra de produção substitui cada ocorrência do carácter "F" por uma sequência de oito caracteres.

Axioma:

$$F - - F - - F$$

Regras de produção:

1: 
$$F \rightarrow F+F--F+F$$

Figura 3 – Definição de um sistema L

A Figura 4 ilustra os dois primeiros passos da sequência de desenvolvimento do mesmo sistema. Como se pode observar, a sequência de desenvolvimento resultante pode iterar indefinidamente dado que, a partir do primeiro passo de derivação, a regra de produção é sempre aplicável, devido ao facto de existir sempre um carácter "F" na cadeia de caracteres. Na implementação de um sistema L deve assim ser definido um número máximo de passos de derivação, para que a sua sequência de desenvolvimento seja finita.

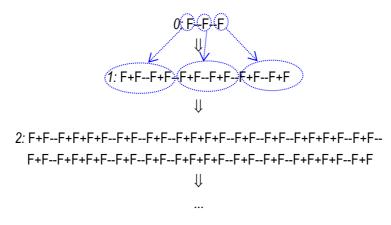


Figura 4 – Sequência de desenvolvimento de um sistema L

# 2.2.3 Interpretação Gráfica de um Sistema L

De acordo com o exemplo anterior, um sistema L desenvolve um determinado axioma, produzindo uma cadeia de caracteres final que representa o estado evolutivo definitivo de um determinado organismo ou entidade.

Ao nível da Computação Gráfica, esta cadeia modular final apenas terá relevância se puder ser traduzida para uma representação gráfica. O crescimento da utilização dos sistemas L nesta área deu-se com o trabalho de Szilard e Quinton [Szi79], onde se demonstra a aplicabilidade da utilização de uma linguagem "tartaruga" ao estilo LOGO [Abe82], para a interpretação de sistemas L.

A ideia base da interpretação gráfica de sistemas L utilizando uma linguagem "tartaruga", é a de que cada carácter seja interpretado como um comando desta linguagem. Desta forma, um cursor (geralmente apelidado de "tartaruga") associado a uma "caneta", é controlado através de um conjunto de comandos que o deslocam sobre um plano, desenhando figuras geométricas.

Considere-se para o exemplo anterior (Figura 4) a seguinte interpretação gráfica dos três caracteres ("F", "+" e "-") utilizados:

- "F" Mover a tartaruga para a frente desenhando uma linha de comprimento unitário.
- "+" Rodar a tartaruga para a esquerda 60°.
- "-" Rodar a tartaruga para a direita 60°.

A figura seguinte apresenta o resultado da interpretação gráfica dos dois primeiros passos do sistema L definido na listagem 1. Observa-se que o axioma é interpretado como um triângulo equilátero e que a sequência de desenvolvimento do sistema L produz a sequência de curvas "floco de neve" que foram apresentadas na Figura 2, a menos de um fator de escala de três vezes, por passo de derivação. Este fator de escala deve-se ao facto de a cadeia de caracteres que substitui cada aresta ter um comprimento três vezes maior.

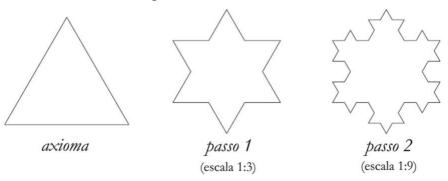


Figura 5 - Interpretação gráfica de um sistema L

De forma a corrigir o problema do fator de escala, é necessário que o comprimento da linha desenhada pelo cursor gráfico possa ser controlado ao longo da sequência de desenvolvimento, através da definição de um parâmetro associado ao carácter "F". Este tipo de mecanismo é possível através da utilização de sistemas L paramétricos, tal com definido na Figura 6. O carácter "F" passa a ser considerado um módulo com um parâmetro associado "F(x)", cujo valor pode ser manipulado pelas regras de produção, e cuja interpretação gráfica cria uma linha com comprimento igual ao valor desse parâmetro. O axioma é composto por três módulos "F(1)" com comprimento unitário, e a regra de produção passa a criar módulos com um terço do comprimento do módulo predecessor, por forma a que o conjunto de quatro arestas resultantes originem um avanço unitário.

Axioma: F(1) --F(1) --F(1) Regras de produção:  $1: F(x) \rightarrow F(x/3) +F(x/3) --F(x/3) +F(x/3)$ 

Figura 6 -Definição de um sistema L paramétrico

A Figura 7 ilustra a sequência de desenvolvimento do sistema L paramétrico definido anteriormente, observando-se claramente a diminuição do parâmetro a cada passo de derivação, obtendo-se assim o resultado desejado, representado na Figura 2.

Figura 7 – Sequência de desenvolvimento de um sistema L paramétrico

#### 2.2.4 Classes de Sistemas L

Os sistemas L têm sido utilizados em inúmeras áreas de aplicação. No processo de adaptação às características intrínsecas dos problemas abordados, têm originado o desenvolvimento de diversas extensões do conceito original. De acordo com o tipo de conceito introduzido em cada uma destas extensões, existem diversas classes de sistemas L.

Nesta lição apresentam-se de forma breve, com base em simulações em tempo real, as classes mais básicas de sistemas L, que servem de suporte às restantes classes. Incluem-se nesta classificação os sistemas D0L, os sistemas L sensíveis ao contexto, os sistemas L parentéticos, os sistemas L estocásticos e os sistemas L paramétricos. No entanto, este texto de suporte apresenta em maior detalhe estas classes de sistemas L no anexo A, para possibilitar ao estudante um estudo mais aprofundado destes conceitos.

Os sistemas L na sua versão base denominam-se Sistemas D0L, denotando a sua natureza determinística (D) e de contexto livre (0). As cadeias de caracteres operadas pelos sistemas L representam estruturas complexas compostas por elementos diversos, aspeto que se reflete na diversidade de módulos que as compõem. A evolução de determinado módulo pode ser dependente de módulos vizinhos, tal como no caso das flores que crescem apenas nas extremidades dos ramos de uma árvore. Os Sistemas L Sensíveis ao Contexto permitem que a aplicabilidade das regras de produção seja dependente, não só do carácter predecessor, mas também dos caracteres situados à sua esquerda ou direita, na cadeia modular.

Para a simulação de estruturas complexas é necessário que o mecanismo de interpretação da cadeia de caracteres, obtida como resultado final de uma sequência de desenvolvimento, seja bastante flexível. Nem todos os objetos de estudo visados por soluções baseadas em sistemas L podem ser representados por estruturas lineares. Os Sistemas L Parentéticos, permitem a definição de estruturas hierárquicas em forma de árvore sobre a cadeia de caracteres, através da introdução de dois novos símbolos (os parêntesis retos) para delimitação de cada um dos ramos da estrutura.

Os sistemas determinísticos geram sempre a mesma solução a partir de um mesmo axioma, pelo que não são os mais adequados para lidar com determinados problemas, como por exemplo a modelação de plantas, onde a complexidade do meio ambiente em que se desenvolvem origina que cada espécime tenha um aspeto único. Os Sistemas L Estocásticos abordam esta problemática, permitindo que seja introduzida aleatoriedade na sequência de desenvolvimento. Desta forma, mesmo partindo dos mesmos axioma e regras de produção, pode obter-se uma grande diversidade de cadeias de caracteres como solução final.

Os Sistemas L Paramétricos associam uma sequência de parâmetros a cada carácter, operando assim sobre módulos paramétricos, e possibilitando a utilização dos valores dos seus argumentos, quer para o condicionamento da aplicabilidade das regras de produção, quer para a parametrização dos comandos de interpretação da cadeia modular obtida no final da sequência de desenvolvimento.

Os sistemas L possuem uma grande flexibilidade para simular os processos de desenvolvimento, no entanto têm grandes limitações para incorporar a natureza geométrica dos objetos a processar. As gramáticas de formas superam esta limitação através do processamento de formas geométricas em vez de cadeias de caracteres.

## 2.3 Gramáticas de Formas

Uma gramática de formas (*shape grammar*) é um tipo de sistema de produção que gera formas geométricas. Este conceito foi definido em por George Stiny e James Gips [SJ72] e consiste num conjunto de regras que operam com formas e um mecanismo de geração que seleciona e processa essas regras. Uma regra de forma define como uma forma existente pode ser transformada numa outra forma.

Uma forma (*shape*) é um arranjo limitado de linhas retas definidas num sistema de coordenadas cartesiano com eixos reais e uma métrica euclidiana associada. Os pontos que compõem as formas podem associar-se a símbolos, criando as denominadas formas etiquetadas (*labelled shapes*). Uma regra de formas (*shape rule*) é definida por  $\alpha \rightarrow \beta$ , tal como numa regra de produção, sendo  $\alpha$  e  $\beta$  formas etiquetadas. A forma  $\alpha$  é substituída pela forma  $\beta$  quando a regra é aplicada [Sti80].

Uma gramática de formas consiste, no mínimo, em três regras de formas: uma regra de início, pelo menos uma regra de transformação e uma regra de terminação. A regra inicial é necessária para iniciar o processo de geração de formas. A regra de terminação é necessária para que o processo de geração de formas termine. A maneira mais simples de interromper o processo é por meio de uma regra de forma que remove o símbolo. As regras de formas podem ser aplicadas em série (como as gramáticas de Chomsky) ou em paralelo (como os sistemas L).

Na figura seguinte podem observar-se as regras de formas (a), e a forma inicial (b) de uma gramática de formas. A regra 1 inscreve um quadrado num quadrado inicial com símbolo associado ao ponto médio do lado superior. A regra 2 elimina o símbolo, terminando a derivação dessa forma.

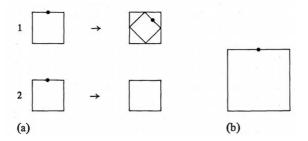


Figura 8 – a) Regras de formas e b) forma inicial [Sti80]

Um sistema de gramática de formas possui adicionalmente uma área de trabalho onde a geometria criada é exibida. O mecanismo de geração verifica se a geometria existente, muitas vezes referida como forma de trabalho atual (CWS – *Current Working Shape*), corresponde ao predecessor das regras de forma. As regras de formas com predecessores correspondentes são elegíveis para utilização. Se mais de uma regra se aplicar, o mecanismo de geração terá que escolher qual a regra aplicar.

A figura seguinte apresenta um exemplo de aplicação da gramática de formas definida anteriormente. A aplicação duas vezes da regra 1 cria dois quadrados inscritos no quadrado inicial. A aplicação da regra 2 termina o processo, gerando a forma final.

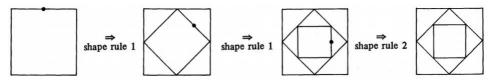


Figura 9 - Exemplo de aplicação da gramática de formas definida anteriormente [Sti80]

Uma gramática de formas pode originar muitas formas distintas, como representado na figura seguinte:



Figura 10 - Formas possíveis de gerar com a gramática de formas definida [Sti80]

As gramáticas de forma foram originalmente apresentadas para pintura e escultura, mas foram estudadas em particular na arquitetura (projeto arquitetónico auxiliado por computador), pois fornecem um formalismo para criar novos projetos. Apresentam assim um potencial elevado para a modelação procedimental de ambientes urbanos.

# 3 Modelação Procedimental de Ambientes Urbanos

O primeiro artigo de modelação procedimental que abordou o conceito de ambiente urbano, de forma integrada, foi publicado por Parish & Muller em 2001 [PM01], e foi desenvolvido com base na utilização de Sistemas L (Figura 11). Este foi a semente que deu lugar a uma das ferramentas de modelação procedimental mais relevantes no mercado: o CityEngine<sup>4</sup>.

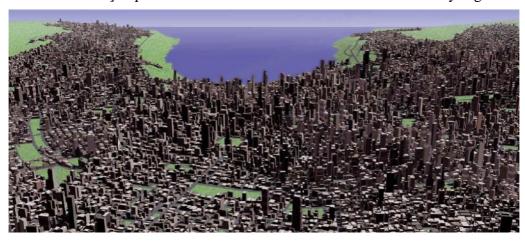


Figura 11 – Modelação de ambientes urbanos utilizando sistemas L [PM01]

Sendo um artigo seminal para a modelação procedimental de ambientes urbanos, os resultados deixaram muitas linhas de investigação em aberto. Os Sistemas L apresentaram várias limitações, particularmente na modelação de edifícios. Mesmo na modelação da rede viária houve a necessidade de desenvolver extensões, denominadas Sistemas L Abertos para controlar a expansão dos sistemas L de forma a construir redes viárias coerentes e funcionais.

As limitações dos Sistemas L para a modelação dos ambientes urbanos deram lugar ao desenvolvimento de dois modelos matemáticos inovadores. Um dos desenvolvimentos passou pela integração das gramáticas de formas, desenvolvendo o que foi denominado de Gramática de Divisão (*Split Grammar*) [WWSR03]. A outra abordagem, desenvolvida pelo autor na sua tese de doutoramento, passou pela integração nos Sistemas L de contextualização espacial, dando origem aos Sistemas L Geoespaciais [CBSF07]. Uma outra abordagem mais genérica, denominada sistemas L funcionais (FL-Systems) [MPB05] foi também desenvolvida para a modelação de malhas tridimensionais, tendo também sido utilizada para a modelação procedimental de ambientes urbanos. Explicaremos com maior detalhe as duas primeiras contribuições: as gramáticas de divisão e os sistemas L geoespaciais.

Uma gramática de divisão é um novo tipo de gramática paramétrica baseada no conceito de forma [Sti80]. A inovação desta nova gramática reside no conjunto de regras de produção permitidas, baseadas no conceito de divisão (*split*) de formas geométricas e na sua parametrização. Um sistema de correspondência de parâmetros permite especificar várias metas de design de alto nível e controlar a aleatoriedade para garantir uma saída consistente. No entanto, é necessária uma outra gramática de controlo, uma gramática de contexto livre, para lidar com a distribuição espacial. A figura seguinte apresenta as características das regras de divisão (operação de *split*).

<sup>&</sup>lt;sup>4</sup> http://www.esri.com/software/cityengine

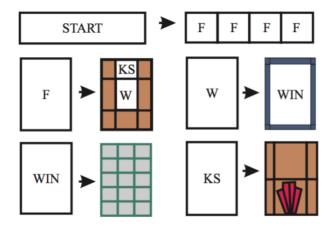


Figura 12 - Gramática de Divisão [WWSR03]

A necessidade de uma gramática de controlo é uma limitação desta abordagem e 3 anos depois surge uma nova contribuição para a ultrapassar – as gramáticas CGA (*Computer Generated Architecture*) [MWH\*06]. Estas gramáticas estendem o conceito de forma, adicionando uma nova entidade – o *scope* – que é a *bounding box* da forma, denominando-se *CGA shape* associado a um sistema de coordenadas locais (Figura 13). Desta forma é possível aplicar diversas operações espaciais segundo os eixos do *scope*. A título de exemplo, pode-se dividir um edifício em diversos pisos ao longo do eixo Y.

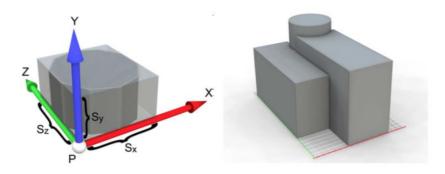


Figura 13 – Representação gráfica de uma CGA shape [MWH\*06]

Evoluindo de forma paralela, foram desenvolvidos os Sistemas L Geoespaciais [CBSF07] que assentam no conceito de cadeia geoespacial, que é uma cadeia modular composta por módulos geoespaciais. Os módulos geoespaciais são módulos paramétricos associados a um objeto geoespacial (Figura 14), possibilitando desta forma que, por exemplo, um módulo geoespacial representando uma estrada possa ser associado à linha que descreve o seu eixo de via, ou ainda, que um módulo geoespacial representando um edifício, possa ser associado a um polígono caracterizando a sua projeção no solo. A cadeia geoespacial é operada por regras de produção que integram operadores espaciais. Estes atuam sobre objetos geoespaciais, individualmente, ou sobre coleções destes objetos. A título de exemplo, uma operação espacial de intersecção entre um determinado módulo representando um edifício e uma coleção de objetos representando o terreno, permite identificar a altitude a que a base desse edifício se encontra.

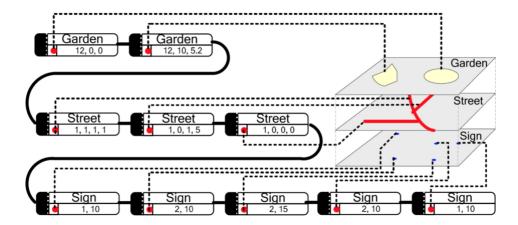


Figura 14 - Representação gráfica do uma cadeia geoespacial [CBSF07]

Isto é uma vantagem relativamente às gramáticas CGA pois estas apenas conseguem trabalhar a contextualização espacial internamente ao *scope*, enquanto que os sistemas L geoespaciais conseguem operar operações geométricas e topológicas com qualquer elemento espacial e conseguem incorporar a semântica da informação geográfica, como se pode verificar na figura seguinte.



Figura 15 - Modelação procedimental de um ambiente urbano com sistemas L geoespaciais [CBSF07]

Os sistemas L geoespaciais possibilitam assim uma maior flexibilidade para modelar procedimentalmente um ambiente urbano complexo, enquanto as gramáticas CGA possibilitam maior detalhe na modelação de edifícios. Estas características levaram ao desenvolvimento de uma nova gramática, denominada PG3D [SC11], no âmbito da tese de doutoramento de Pedro Silva, que expande e integra os conceitos CGA shape, através da definição da PG3DShape, com as cadeias geoespaciais através dos conceitos de PG3DLayer e PG3DBoundary. Esta gramática adiciona também maior capacidade de tratamento semântico da informação com o conceito de PG3DTag (Figura 16). A gramática PG3D deu origem, posteriormente, à comercialização da ferramenta Sceelix<sup>5</sup>.

<sup>&</sup>lt;sup>5</sup> https://www.sceelix.com/

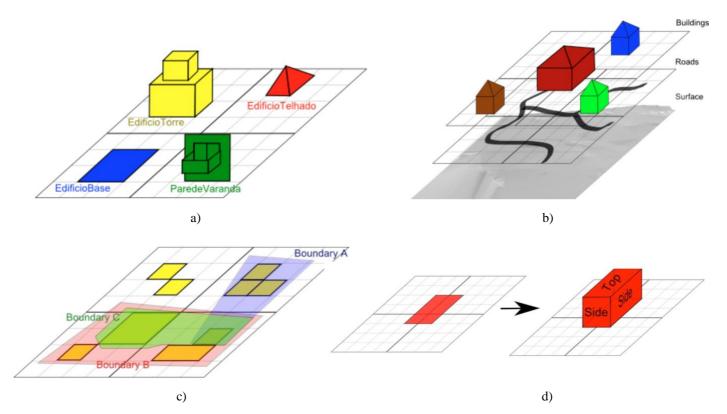


Figura 16 - Representação gráfica de PG3DShape (a), PG3DLayer (b), PG3DBoundary (c) e PG3DTag (d).

A informação SIG possui uma grande relevância para a modelação procedimental e torna-se necessário uniformizar esta informação. Através do projeto 3DWikiU, foi possível desenvolver uma ontologia urbana para o desenvolvimento de ambientes virtuais que possibilitassem uma interação direta com esta informação [MSCS12]. Desta forma, surgiu um dos primeiros conceitos de SIG 3D, em que a informação se encontra associada a modelos 3D e não apenas a mapas planos.

No âmbito da modelação procedimental de ambiente urbanos, têm surgido diversas contribuições que procuram responder a elementos específicos desses ambientes, que analisaremos nas seguintes secções. Uma revisão bibliográfica mais completa do que a abordada nesta lição pode se consultada em [STBB14].

# 3.1 Modelação do Terreno

A modelação do terreno é a base de qualquer ambiente virtual, sendo o suporte para os demais modelos tridimensionais. Existem três abordagens principais à modelação de terrenos: utilização de fractais, simulação da erosão física e síntese de terrenos a partir de imagens ou nuvens de pontos.

Esta última abordagem é a mais simples para a representação da realidade, particularmente com a utilização de DEM (*Digital Elevation Model*). Os modelos são assim detalhados mas estão limitados à representação da realidade.

Uma das primeiras abordagens à modelação de terrenos baseou-se em fractais, sendo pioneiros os trabalhos de Mandelbrot [Man82] na definição dos fractais ou de Fournier [FFC82] propondo a técnica de deslocamento do ponto médio para a modelação de superfícies fractais. Este método foi desenvolvido posteriormente por Voss [Vos85] e posteriormente por Miller [Mil86], tendo este último descrito várias variantes deste método. A ideia principal consiste

em iniciar com um retângulo, onde todos os cantos recebem um valor de altura inicial e adicionar iterativamente um novo ponto central, cuja elevação é calculada pela soma da média dos valores dos seus cantos com um deslocamento aleatório (Figura 17). O intervalo do deslocamento diminui em cada iteração, com base em um parâmetro que define a rugosidade pretendida do terreno.

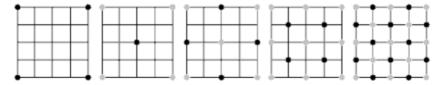


Figura 17 - Desenvolvimento do algoritmo do ponto médio ao longo de 4 iterações (esquerda para a direita)

Um outro método baseado em fractais, que é utilizado com muita frequência, é o ruído de Perlin [Per85]. Este método consiste em criar um conjunto inicial de valores aleatórios, que são depois interpolados com ruído coerente. É possível compor várias camadas de ruído juntas para criar terrenos mais naturais [SKG\*09]. Várias ferramentas comerciais já incorporam estas técnicas na modelação de terrenos (Figura 18).



Figura 18 - Terreno desenvolvido com Terragen

De forma a incrementar o realismo dos modelos de terreno, os trabalhos pioneiros de [KMN88] e [MKM89] combinam fractais com simulação dos fenómenos físicos que simulam a erosão. Embora com resultados mais realistas, o custo computacional é elevado. Trabalhos mais recentes propõem abordagens em tempo real [NWD05] e [Ols04].

Devido ao elevado nível de aleatoriedade e falta de controlo, estes métodos apresentam limitações à sua utilização efetiva em cenários específicos, em que o terreno tem que ser adequado aos diversos usos urbanos (edificado, rede viária, áreas de lazer, hidrografia, etc). Stachniak e Stuerzlinger [SS05] propõem um método que utiliza restrições (na forma de máscaras de imagem) no processo de geração de terreno. O algoritmo pesquisa um conjunto aceitável de operações de deformações para aplicar ao terreno de forma a que se obtenha um resultado que esteja de acordo com as restrições especificadas. Zhou et al. [ZSTR07] descreve uma técnica que usa um exemplo de modelo de terreno como entrada e permite que o utilizador desenhe as características desejadas. Os recursos são extraídos do exemplo e comparados com o esboço e fundidos para formar o resultado final (Figura 19). De Carpentier e Bidarra [CB09] introduziram o conceito de pincéis procedimentais (*procedural brushes*) para os utilizadores poderem pintar o mapa de elevações diretamente em 3D.

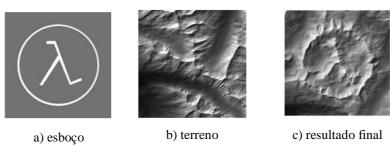


Figura 19 - Geração do terreno por exemplo [ZSTR07]

Uma das limitações de trabalhar com mapas de elevação é a impossibilidade de definir terrenos com grutas ou outro tipo de características morfológicas mais complexas. A Figura 20 apresenta os resultados do trabalho de Peytavie et al. [PGGM09] que através de uma estrutura mais elaborada possibilita a modelação de diferentes camadas de materiais, sendo capaz de modelar rochas, arcos, saliências e cavernas.



Figura 20 – Modelação de estruturas orográficas complexas [PGGM09]

# 3.2 Modelação da Vegetação

Após a publicação, em 1984, do trabalho de Alvy Ray Smith [Smi84], seguindo a ideia inicial que presidiu ao desenvolvimento dos sistemas L (o processo de desenvolvimento das plantas), a linha de investigação primordial ao nível da Computação Gráfica foi precisamente a síntese de imagem de modelos de plantas.

Como culminar de diversos trabalhos seus no âmbito desta linha de investigação, Przemyslaw Prusinkiewicz e Aristid Lindenmayer publicam em 1990 o livro "*The Algorithmic Beauty of Plants*" [Pru90]. Este livro tornar-se-ia a referência para a aplicação dos sistemas L na área da Computação Gráfica, e foi o percursor dos grandes avanços ocorridos ao longo da década de 90.

Até ao início da década de 90, os trabalhos desenvolvidos na área da Computação Gráfica abordavam o desenho de fractais e de plantas isoladas, apresentando um desenvolvimento estrutural aceitável, mas com uma apresentação visual ainda pouco realista.

Alguns trabalhos foram então desenvolvidos com o intuito de obter modelos cada vez mais realistas de plantas. [Ham95] apresenta uma metodologia para a criação de modelos tridimensionais de plantas, baseada em sistemas L e incorporando dados biológicos. Cria um modelo qualitativo de acordo com observações do crescimento e da forma da planta e, posteriormente, efetua a medição das suas características fundamentais. Partindo de uma análise estatística destes dados, define funções descritivas do crescimento da planta.

Em [Pru01] apresenta-se um conjunto de algoritmos para a modelação de plantas com elevado nível de realismo visual. Estes algoritmos têm em atenção aspetos considerados fundamentais por parte de artistas, tais como: a postura, a variação gradual das características e a progressão gradual do processo de desenho desde a silhueta geral aos detalhes locais. Estes aspetos são incorporados nos algoritmos através da informação posicional, que se define como sendo a posição dos componentes da planta ao longo dos eixos dos ramos e caules que os suportam. Os sistemas L capturam o desenvolvimento das plantas ao longo do tempo e a simulação termina quando o organismo atinge uma idade terminal predefinida, correspondente a um dado número de passos de derivação. A interpretação gráfica da cadeia obtida é efetuada através de geometria tartaruga diferencial permitindo obter modelos mais realistas de plantas, contemplando configurações mais curvilíneas. A Figura 21 apresenta duas fotografias de plantas reais e os respetivos modelos tridimensionais obtidos utilizando estes algoritmos.



Figura 21 – Ilustração de modelos realistas de duas plantas [Pru01]

Um processo diferente, abordado em outros trabalhos, é a animação do desenvolvimento de plantas, ou seja, a produção de sequências de imagem e animações que reproduzam o processo de crescimento das plantas ao longo do tempo. Um dos trabalhos mais marcantes neste contexto, levou à definição dos sistemas L diferenciais [Prus93a]. Um exemplo de resultados obtidos no trabalho referido apresenta-se na Figura 22.

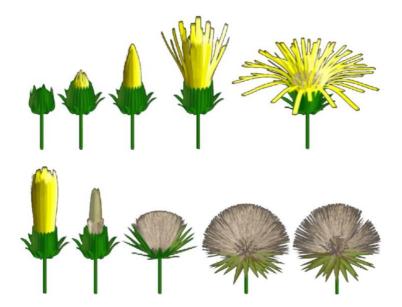


Figura 22 – Ilustração dos diferentes estádios de crescimento de uma planta [Prus93a]

Num artigo publicado em 1994 apresentam-se os sistemas L sensíveis ao ambiente [Prus94], que foram desenvolvidos para possibilitar a simulação do processo de poda sintética (Figura 23). Estes sistemas têm a capacidade de limitar o seu crescimento a um volume envolvente.



Figura 23 – Processo de poda sintética baseado em sistemas L sensíveis ao ambiente [Pru94]

Em [Mec96] são apresentados exemplos de aplicação de sistemas L abertos ao desenvolvimento e propagação de plantas clonais, crescimento de raízes e ao crescimento das árvores influenciado pela competição pela luz. A Figura 24 apresenta o exemplo de uma trepadeira que cresce à volta do tronco de uma árvore.

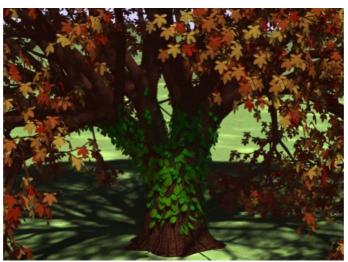


Figura 24 – Exemplos de aplicação dos sistemas L abertos [Mech96]

O potencial da síntese de imagens realistas de plantas para a indústria cinematográfica motivou o desenvolvimento de trabalhos de simulação de ecossistemas naturais completos. Em [Deu98] descreve-se um sistema desenvolvido para a visualização realista de ecossistemas de plantas. O processo de modelação inicia-se com a modelação do terreno de forma interativa, ou partindo de fontes de dados diversas, tais como os SIG. Em seguida, é especificada a distribuição inicial da localização das plantas, quer de forma explícita (interactivamente ou a partir de dados reais), quer de forma procedimental, sendo geradas distribuições de acordo com propriedades estatísticas desejadas. O ecossistema é posteriormente simulado utilizando sistemas L abertos,

de forma a obter-se uma nova especificação da distribuição das localizações das plantas, de acordo com o modelo ecológico definido. O modelo global do ecossistema é obtido pela instanciação de modelos tridimensionais detalhados das plantas, selecionados aleatoriamente a partir de um conjunto de espécimenes semelhantes, gerados também por sistemas L (Figura 25).





Figura 25 - Exemplos de aplicação dos sistemas L abertos [Deu98]

Em [LP02], o realismo da simulação de ecossistemas de plantas é levado um pouco mais além (Figura 26). Utilizando sistemas L paramétricos com coleções simulam-se os processos de interação e competição entre plantas, típicos de qualquer ecossistema natural, obtendo-se uma especificação da distribuição da localização das plantas.



Figura 26 – Simulação de um ecossistema de plantas [LP02]

Atualmente, também já existem soluções comerciais para a modelação de plantas, tais como o Xfrog ou o speedtree (ver Figura 27).





Figura 27 - Vegetação desenvolvida com Xfrog (esquerda) speedtree (direita)

# 3.3 Modelação da Rede Viária

A modelação de redes viárias surgiu já no artigo seminal de Parish and Muller [PM01] utilizando sistemas L. A abordagem é semelhante ao desenvolvimento de plantas, numa

perspetiva global, começando com um segmento de rua, vai-se derivando novas ruas conectadas com esta, de forma iterativa. No entanto, para assegurar a conectividade da rede viária, é necessário seguir este primeiro passo pela aplicação local de um conjunto de restrições que corrige determinadas situações, evitando cruzamentos sem nó de conectividade e "becos sem saída". Estas gramáticas são denominadas de *self-sensitive L-systems*. Esta abordagem produz bons resultados no traçado das ruas quando a cidade segue modelos padronizados (ver Figura 28).



Figura 28 - Utilização de Sistemas L para a modelação procedimantal do traçado da rede viária [PM01]

Este tipo de abordagem peca por não permitir controlo local nos padrões da rede viária. Para ultrapassar este problema, foi desenvolvida uma técnica baseada em campos tensoriais (*tensor fields*) [CEW\*08] para definir o padrão de geração da rede viária localmente. Esta técnica tem a vantagem adicional de ser interativa, permitindo ao utilizador alterar a rede viária através da aplicação de um "pincel", modificando a orientação dos tensores à volta dessa localização (Figura 29).



Figura 29 – Aplicação de um campo tensorial para a geração da rede viária. [CEW\*08]

Outra sistema, deminado CityGen [KM07] possibilita ainda maior nível de controlo. O utilizador define interativamente os nós principais que formam a rede viária primária e o sistema adapta o traçado das estradas ao terreno. A partir da rede de estradas primária, são geradas células urbanas onde vão ser definidas as estradas secundárias através de um sistema L, num processo semelhante a [PM01].

A definição do traçado e a posterior modelação da rede viária de forma visualmente apelativa pode não ser suficiente para determinadas aplicações. Na criação de um ambiente virtual para simulação de condução é essencial que as estradas cumpram as regras definidas pela

Engenharia Civil para a sua construção e sigam a geometria específica do projeto de estradas. O trabalho de Galin [GPMG10] apresenta um algoritmo para gerar estradas, dados dois pontos, adaptando-se às características de uma determinada cena de entrada. Isto é conseguido calculando o caminho mais curto entre esses dois pontos, minimizando uma função de custo que leva em conta a inclinação do terreno, rios, lagos, florestas, etc. Este caminho é primeiro definido discretamente, e só depois convertido num conjunto de splines clotóides para descrever a geometria da trajetória da estrada. Esses segmentos são então divididos em estradas de superfície, túneis e pontes (Figura 30). O controlo é assim realizado através de funções de custo parametrizadas e controláveis que operam com base nos parâmetros da cena.



Figura 30 –Traçado de estradas com geometria clotóide. [GPMG10]

De forma a otimizar a modelação procedimental para a condução em simulador, foi realizado um mapeamento das regras de construção de estradas para o processo de modelação procedimental, desde a geração do trajeto, de acordo com os parâmetros do utilizador, até à criação do perfil da estrada [CLC15] (ver Figura 31). Este trabalho, desenvolvido na tese de doutoramento de Carlos Campos, permite obter um comportamento dinâmico dos veículos equivalente ao comportamento real, para além de obter resultados visualmente muito realistas.

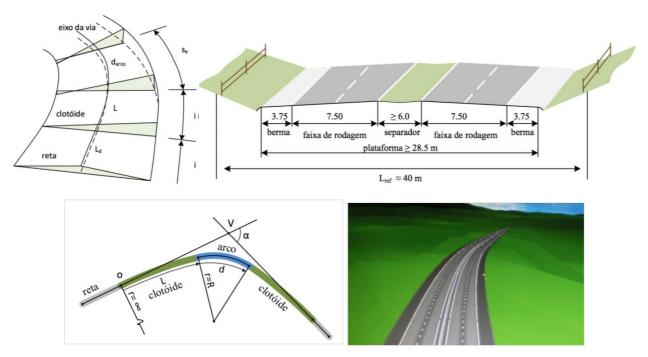


Figura 31 - Definição da geometria das vias rodoviárias [CLC15]

# 3.4 Modelação de Edifícios

A modelação de edifícios é um dos elementos mais complexos no processo de modelação procedimental de ambientes urbanos, mas também um dos mais relevantes na sua caracterização e nível de fidelidade visual. Os sistemas L foram utilizados numa primeira abordagem [PM01] criando a geometria dos edifícios através de extrusões (Figura 32).



Figura 32 – Extrusão de edifícios com sistemas L [PM01]

Para o detalhe das fachadas foram aplicadas texturas geradas através de composição de outras texturas, controladas por meio de ondas retangulares que geram as bandas verticais e pisos da fachada do edifício (Figura 33).

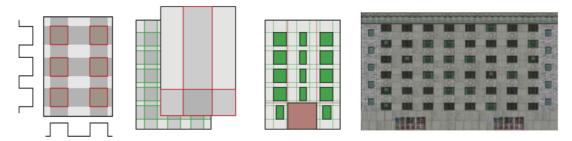


Figura 33 - Geração das texturas das fachadas [PM01]

Os Sistemas L Geoespaciais [CBSF07] resolvem vários dos problemas da modelação de fachadas, uma vez que a contextualização espacial permite determinar a fachada virada para a rua (onde deve estar a porta) ou as fachadas que têm edifícios vizinhos (e que não devem ter janelas ou varandas), bem como a decomposição fachada em pisos e bandas verticais, posicionando janelas e varandas de forma coerente. A Figura 34 apresenta um exemplo da geração procedimental de fachadas de edifícios.



Figura 34 - Modelação procedimental de fachadas com sistemas L geoespaciais [CBSF07]

O desenvolvimento das gramáticas CGA [MWH\*06] teve como objetivo a modelação detalhada das fachadas dos edifícios. Embora não possua capacidade de contextualização espacial dos sistemas L geoespaciais, apresenta um nível de detalhe e flexibilidade superior no desenvolvimento dos edifícios individualmente (Figura 35).



Figura 35 - Modelação procedimental de edifícios com uma gramáticas CGA [MWH\*06]

Numa abordagem distinta, denominada de modelação procedimental inversa, captura-se informação de imagens de fachadas reais para reconstruir as regras de produção que modelam essas fachadas [MZWvG07]. Num primeiro passo é reconhecida a estrutura da fachada (pisos e bandas verticais), seguindo-se o processo de identificação dos elementos constituintes dessa fachada (janelas, portas e varandas) numa base de modelos 3D. Na terceira e última fase são criadas as regras de produção que geram essa fachada (Figura 36).



Figura 36 - Modelação procedimental inversa de fachadas [MZWvG07]

Fachadas mais complexas, com ornamentos por exemplo, requerem outros processos de modelação procedimental como o sistema desenvolvido por Finkenzeller [Fin08]. A primeira etapa do processo requer que o utilizador crie uma representação plana do piso térreo, consistindo em um ou mais polígonos convexos, denominado *floor plan module* (fpm). Os andares superiores são modelados com base no piso inferior, mantendo, removendo ou dividindo o fpm existente.

O sistema gera, de seguida, uma representação dos vários elementos da fachada. Uma parede é gerada para cada borda externa do fpm, que é subdividida verticalmente, de acordo com o estilo arquitetônico escolhido, criando partições. Cada partição, por sua vez, contém outros elementos arquitetônicos e, no máximo, uma porta ou janela (Figura 37).

O último passo gera os telhados levando em conta a estrutura do edifício. O algoritmo define o aspecto de todo o telhado com base na estrutura individual de cada fpm. Para isso, são gerados telhados individuais para cada fpm no topo do edifício, que depois são combinados, adaptando-os nas zonas de intersecção.

É possível, a qualquer instante, modificar o parâmetro de qualquer elemento e, portanto, o aspecto do edifício. Isso permite que o utilizador altere o estilo de todo o edifício, de um piso ou até mesmo de janelas ou portas individualmente. Como consequência, o sistema permite a geração de estruturas mais complexas num período de tempo inferior.

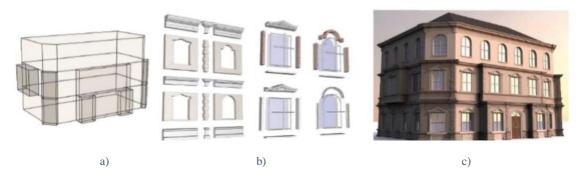


Figura 37 – Modelação detalhada de fachadas [Fin08]

A grande variabilidade arquitetónica das fachadas dos edifícios levou ao desenvolvimento de uma solução mais flexível, as *Layered shape grammar* [JCS16], no âmbito da tese de doutoramento de Diego Jesus. Esta gramática estende as gramáticas CGA com um conjunto de operações adicionais que permitem a especificação tanto das *layers* que compõem uma fachada, como definição vetorial de formas em 2D. Esta metodologia é assim capaz de gerar layouts não-triviais, indo além da estrutura regular de grelha e integrando de forma transparente elementos arquitetónicos mais complexos, como por exemplo portas e janelas arqueadas.

O sistema baseia-se num algoritmo que combina as diferentes *layers* com os elementos que compõem a fachada, tendo em conta o recorte e a oclusão das formas, de acordo com a sua profundidade dentro das *layers*, resultando num novo conjunto de superfícies independentes. Este processo denomina-se de normalização planar das formas (Figura 38).



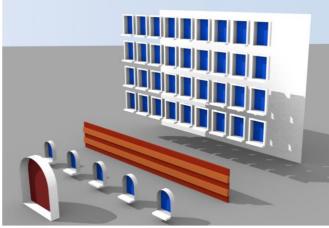


Figura 38 - Composição de uma fachada (à esquerda) pela normalização planar das formas das diferentes layers (à direita) [JCS16]

# 4 Tendências para o futuro

Ao longo desta lição foram apresentados vários métodos, e até ferramentas comerciais, que possibilitam a modelação procedimental de ambientes urbanos de forma integrada. Quando se pretende obter maior detalhe em determinados elementos urbanos pode-se recorrer a técnicas específicas.

Embora o estado atual das técnicas de modelação procedimental já consiga superar muitos dos desafios dos ambientes urbanos, há ainda um conjunto de problemas em aberto:

- Modelação procedimental inversa: Uma vez que muitas das abordagens apresentadas utilizam gramáticas formais, um dos desafios é o de conseguir extrair as regras de produção com base em exemplos. Uma abordagem para a modelação de fachadas de edifícios foi já desenvolvida [MZWvG07], mas esta área ainda tem desafios em aberto.
- Gramáticas baseadas em linguagem natural: Existem diversos textos, literários e técnicos, que descrevem ambientes urbanos. Esta linha de investigação procura extrair regras de produção com base nesses textos. Tal abordagem já começou a ser explorada no âmbito de uma tese de doutoramento orientada pelo autor [RCR10].
- Linguagens visuais: A definição de regras de produção baseadas em linguagem matemática no processo de modelação procedimental tem a desvantagem de não ser intuitivo, levando a curvas de aprendizagem longas. A utilização de linguagens visuais torna o processo mais intuitivo, particularmente para utilizadores da área da arquitetura ou do design. O trabalho proposto por Lipp et al. [LWW08] apresenta um sistema onde a edição de regras é feita diretamente no modelo gerado e não há necessidade de escrever as regras de produção. O trabalho de Pedro Silva [SEBC15] apresenta uma linguagem visual baseada em grafos procedimentais de conteúdo, que utiliza os nós para descrever os procedimentos de criação, transformação, análise ou filtro de conteúdos e os arcos para descrever o fluxo dos mesmos entre os nós. O trabalho mais recente de Diego Jesus, apresenta as bases para a utilização de uma modelação baseada em esboços (sketch-based) para a definição das regras de produção [JPCS18].
- Informação Semântica: Por um lado, a informação semântica presente em muitas fontes de dados (como os SIG) permite maior expressividade nos processos de modelação procedimental ao incrementar o controlo do processo por parte do utilizador. Por outro lado, a representação visual dos modelos 3D nem sempre é suficiente para todas as aplicações, como nos jogos digitais ou simuladores de condução. Nestes casos é necessária informação adicional para as mecânicas de jogo, a sinalização das vias ou as trajetórias dos veículos. A integração de informação semântica foi abordada previamente por Tutenel et al. [TSL\*11], tendo sido expandida pelos trabalhos de doutoramento de Pedro Silva, Carlos Campos e Diego Jesus [SEBC15; CLC15; JPCS18].

# 5 Conclusões

Ao longo desta lição identificaram-se as vantagens da utilização de modelação procedimental, bem como as áreas de aplicação nos ambientes urbanos. As ferramentas de modelação procedimental aumentam a produtividade das equipas no desenvolvimento de projetos de modelação de cenas extensas, reduzindo os custos e o tempo de execução do projeto. Permitem também que determinados profissionais (planeamento urbano ou arqueologia, por exemplo) possam explorar virtualmente diferentes soluções para um determinado problema e integrar com sistemas de informação como os SIG.

A modelação procedimental baseia-se em métodos matemáticos como a geometria fractal ou as gramáticas formais, destacando-se os sistemas L e as gramáticas de formas. Devido à complexidade inerente aos ambientes urbanos têm vindo a ser desenvolvidas vários métodos e técnicas para solucionar determinados problemas, tanto na modelação integral dos ambientes urbanos como de determinados elementos que os compõem. Atualmente há já um conjunto de ferramentas comerciais que possibilita a modelação procedimental de ambientes urbanos.

No entanto, ainda há alguns problemas que ainda não têm solução definitiva, deixando linhas de investigação para desenvolvimento futuro. Um dos problemas fundamentais é a dificuldade inerente à especificação das regras de produção, tanto pela falta de controlo direto sobre os modelos gerados como pela dificuldade de definir essas regras em linguagem matemática. A utilização de imagem de edifícios reais ou descrições textuais dos edifícios para gerar as regras de produção é uma das soluções, mas também a utilização de paradigmas de interação mais intuitivos, como as linguagens visuais, a definição baseada em exemplos ou em esboços. Há também ainda uma limitação na utilização da informação semântica ao longo do processo de modelação procedimental.

A modelação procedimental de ambientes urbanos é já uma realidade em muitas empresas, e evolui para um conceito mais amplo, o da geração procedimental de conteúdos, muito relevante para a indústria dos jogos digitais, por exemplo.

# Referências

[Abe82]	Abelson H., diSessa A.: Turtle Geometry. M.I.T. Press, 1982.
[CB09]	Carpentier G, Bidarra R.: Interactive GPU-based procedural heightfield brushes. In Proceedings of Fourth International Conference on the Foundations of Digital Games (2009), pp. 55-62.
[CBSF07]	Coelho A., Bessa M., Sousa A., Ferreira F.: Expeditious modeling of virtual urban environments with geospatial l-systems. Computer Graphics Forum 26, 4 (2007), 769–782.
[CEW*08]	Chen G., Esch G., Wonka P., Müller P., Zhang, E.: Interactive procedural street modeling. ACM Transaction on Graphics (SIGGRAPH '08) 27, 3 (2008).
[Cho56]	Chomsky N.: Three models for the description of language. IRE Trans. On Information Theory 2 (1956), $113-124$ .
[CLC15]	Campos C., Leitão J. M., Coelho A.: Procedural Generation of Road Paths for Driving Simulation. International Journal of Creative Interfaces and Computer Graphics 6, 2 (2015), 37-55.
[Deu98]	Deussen O., Hanrahan P., Lintermann B., Mech R., Pharr M., Prusinkiewicz P.: Realistic modeling and rendering of plant ecosystems. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH '98), ACM (1998), pp. 275-286.
[Fin08]	Finkenzeller D.: Detailed Building Facades. Computer Graphics and Applications, IEEE (2008), 58–66.
[FFC82]	Fournier A., Fussell D., Carpenter L.: Computer rendering of stochastic models. Communications of the ACM 25, 6 (1982), 371–384.
[GPMG10]	Galin E., Peytavie A., Maréchal N., Guérin, E.: Procedural generation of roads. Computer Graphics Forum 29, 2. (2010), 429–438.
[Ham95]	Hammel M., Prusinkiewicz P., Remphrey W., Davidson C.: Simulating the development of Fraximus pennsylvanica shoots using L-systems. In Proceedings of the Sixth Western Computer Graphics Symposium (1995), pp. 49-58.
[JCS16]	Jesus D., Coelho A., Sousa A. A.: Layered shape grammars for procedural modelling of buildings. Visual Computer (33rd Conference on Computer Graphics International (CGI) 32, 6 (2016), 933-943.
[JPCS18]	Jesus D., Patow G., Coelho A., Sousa A. A.: Generalized Selections for Direct Control in Procedural Buildings. Computers & Graphics 72, Elsevier (2018), 106-121.
[KMN88]	Kelley A. D., Malin M. C, Nielson G. M.: Terrain simulation using a model of stream erosion. In Proceedings of the 15th annual conference on Computer graphics and interactive techniques (SIGGRAPH '88), Richard J. Beach (Ed.). ACM (1988), 263-268.
[LP02]	Lane B., Prusinkiewicz P.: Generating Spatial Distribuitions for Multilevel Models of Plant Communities. In Proceedings of Graphics Interface (2002), pp. 69-80.

[Lin68] Lindenmayer A.: Mathematical models for cellular interaction in development Parts I and II. Journal of Theoretical Biology, 18 (1968) 280-[LWW08] Lipp M., Wonka P., Wimmer, M.: Interactive visual editing of grammars for procedural architecture. ACM Transactions on Graphics 27, 3 (2008). [Man82] Mandelbrot B.: The fractal geometry of nature. Freeman, 1982. [Mec96] Mech R., Prusinkiewicz P.: Visual Models of Plants Interacting with Their Environment. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96). ACM, (1996), pp. 397-410. Gavin S P Miller: The definition and rendering of terrain maps. In [Mil86] Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96) 20, 4 (1986), pp. 39-48. Musgrave F. K., Kolb C. E., Mace R. S.: The synthesis and rendering of [MKM89] eroded fractal terrains. In Proceedings of the 16th annual conference on Computer graphics and interactive techniques (SIGGRAPH '89), ACM (1989), 41-50.[MPB05] Marvie J., Perret J., Bouatouch K.: The FL-system: a functional L-system for procedural geometric modeling. The visual computer 21, 5 (2005) 329-[MSCS12] Martins T., Silva P., Coelho A., Sousa A. A.: An Urban Ontology to Generate Collaborative Virtual Environments for Municipal Planning and Management. In Proceedings of GRAPP 2012 - 7th International Conference in Computer Graphics Theory and Applications (2012), pp. 507-510. [MWH\*06] Müller P., Wonka P., Haegler S., Ulmer A, van Gool L: Procedural modeling of buildings. ACM Trans. Graph. 25, 3 (2006), 614-623. [MZWvG07] Müller P., Zeng G., Wonka P., van Gool L.: Image-based procedural modeling of facades. ACM Transactions on Graphics 26, 3 (2007). [NWD05] Neidhold B., Wacker M., Deussen, O.: Interactive physically based fluid and erosion simulation. Natural Phenomena (2005), 25-32. [Ols04] Jacob O.: Realtime procedural terrain generation. Journal of the Department of Mathematics And Computer Science, IMADA University of Southern Denmark (2004). [Per85] Perlin K.: An image synthesizer. In Proceedings of the 12th annual conference on Computer graphics and interactive techniques (SIGGRAPH '85), ACM (1995), 287-296. [PGGM09] Peytavie A., Galin E., Grosjean J., Merillou, S.: Arches: A framework for modeling complex terrains. Computer Graphics Forum, 28, 2 (2009), 457-[PM01] Parish Y., Muller P.: Procedural Modeling of Cities. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01), ACM (2001), 301-308. [Pru01] Prusinkiewicz P., Muendermann L., Radoslaw K., Lane B.: The Use of Positional Information in the Modeling of Plants. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques

(SIGGRAPH '01), ACM (2001), 289-300.

[Pru90]	Prusinkiewicz P., Lindenmayer A.: The Algorithmic Beauty of Plants, Springer-Verlag, 1990.
[Pru94]	Prusinkiewicz P., James M., Mech R.: Synthetic Topiary. In Proceedings of the 21st annual conference on Computer graphics and interactive techniques (SIGGRAPH '94). ACM (1994), 351-358.
[Pru96]	Prusinkiewicz P., Kari L.: Subapical bracketed L-systems. Cuny J., Ehrig H., Rozenberg G. (eds.), Lecture Notes in Computer Science (Grammars and their Application to Computer Science), 1073 (1996), Springer-Verlag, 550-564.
[RCR10]	Rodrigues R., Coelho A., Reis L.: Procedural Modelling of Monumental Buildings from Textual Descriptions. In Proceedings of GRAPP 2010 - Fifth International Conference in Computer Graphics Theory and Applications (2010), pp.130-133.
[SC11]	Silva P., Coelho A.: Procedural Modeling for Realistic Virtual Worlds Development. Journal of Virtual Worlds Research, 4, 1 (2011).
[SEBC15]	Silva P. B., Eisemann E., Bidarra R., Coelho A.: Procedural Content Graphs for Urban Modeling. International Journal of Computer Games Technology (2015).
[SJ72]	Stiny, G., Gips, J.: Shape grammars and the generative specification of painting and sculpture. Information Processing, 71 (1972), 1460–1465.
[SKG*09]	Smelik R. M., Kraker K. J. Saskia, Groenewegen A., Tutenel T., Bidarra R.: A survey of procedural methods for terrain modelling. In Proceedings of the CASA'09 Workshop on 3D Advanced Media in Gaming and Simulation (2009), pp. 25-34.
[Smi84]	Smith A.: Plants fractals and formal languages. In Proceedings of the 11th annual conference on Computer graphics and interactive techniques (SIGGRAPH '84), ACM, 18, 3 (1984), 1-10.
[SS05]	Stachniak S., Stuerzlinger W.: An Algorithm for Automated Fractal Terrain Deformation. In Proceedings of Computer Graphics and Artificial Intelligence (2005), pp. 64–76.
[STBB14]	Smelik R., Tutenel T., Bidarra R., Benes B.: A survey on procedural modeling for virtual worlds. Computer Graphics Forum 33, 6 (2014), 31-50.
[Sti80]	Stiny G.: Introduction to shape and shape grammars. Environment and Planning B (1980),343-361.
[Szi79]	Szilard A., Quinton R.: An interpretation for DOL systems by computer graphics. The Science Terrapin 4 (1979) 8-13.
[Tay92]	Taylor, C. E.: "Fleshing out" Artificial life II. In C. G. Langton, C. Taylor, J. Farmer, & S. Rasmussen (Eds.), Artificial life II (1992), 25-38.
[TSL*11]	Tutenel T., Smelik R. M., Lopes R., Kraker K. J., Bidarra R.: Generating consistent buildings: a semantic approach for integrating procedural techniques. IEEE Transactions on Computational Intelligence and AI in Games, 3, 3 (2011), 274-288.
[Vos85]	Voss, R. F.: Random Fractal Forgeries. Fundamental Algorithms for Computer Graphics SE 34, 17 (1985), 805–835.
[WWSR03]	Wonka P., Wimmer M., Sillion F., Ribarsky W: Instant architecture. ACM Trans. Graph, 22, 3 (2003), pp. 669–677.

[ZSTR07]

Zhou H., Sun J., Turk G, Rehg. J. M.: Terrain Synthesis from Digital Elevation Models. IEEE Transactions on Visualization and Computer

Graphics 13, 4 (2007) 834–848.

KM07 Kelly G, McCabe H.: Citygen: An Interactive System for Procedural City

Generation. In Proceedings of GDTW 2007: The Fifth Annual International Conference in Computer Game Design and Technology (2007), pp. 8–16.

# Anexo A - Classes de Sistemas L

Neste anexo apresentam-se com maior detalhe as classes mais básicas de sistemas L. Incluem-se nesta classificação os sistemas D0L, os sistemas L sensíveis ao contexto, os sistemas L parentéticos, os sistemas L estocásticos e os sistemas L paramétricos.

#### A1. Sistemas D0L

Os Sistemas D0L são a classe mais simples de sistemas L, operando sobre cadeias de caracteres. Caracterizam-se por serem determinísticos e de contexto livre, havendo uma só regra de produção aplicável a cada carácter e não sendo a sua aplicabilidade dependente dos caracteres que se situam na sua vizinhança. Para cada carácter de um determinado vocabulário existe uma e só uma regra de produção em que esse carácter aparece como predecessor.

Seja V um alfabeto não vazio e finito,  $V^*$  o conjunto de todas as palavras em V, e  $V^+$  o conjunto de todas as palavras não vazias em V. Um sistema 0L é um terno ordenado  $G=(V, \omega, P)$ , em que V é o alfabeto do sistema,  $\omega \in V^+$  é uma palavra não vazia denominada axioma e  $P \subset V \times V^*$  é um conjunto finito de regras de produção.

Uma regra de produção correspondente ao par  $(A,\chi) \in P$  representa-se como  $A \to \chi$ . O carácter A e a palavra  $\chi$  denominam-se respetivamente, predecessor e sucessor da regra de produção. Assume-se que, para cada carácter  $A \in V$ , existe pelo menos uma palavra  $\chi \in V^*$  tal que  $A \to \chi$ . Se nenhuma regra de produção for definida explicitamente para um determinado predecessor, a regra de produção identidade  $a \to a$  é assumida como pertencente ao conjunto de regras de produção. Um sistema 0L é determinístico (denominando-se sistema D0L) se e só se, para cada  $A \in V$  existe exatamente um  $\chi \in V^*$  tal que  $A \to \chi$ .

Seja agora  $\mu = A_1...A_m$  uma palavra arbitrária em V. Diz-se que a palavra  $v = \chi_1...\chi_m$  é derivada diretamente de (ou gerada por)  $\mu$ , representando-se por  $\mu \Rightarrow \nu$ , se e só se  $A_i \to \chi_i$  para todos os i = 1,...,m. Uma palavra  $\nu$  é gerada por G numa derivação de comprimento n se existe uma sequência de desenvolvimento de palavras  $\mu_0, \mu_1, ..., \mu_n$  tal que  $\mu_0 = \omega, \mu_n = \nu$  e  $\mu_0 \Rightarrow \mu_1 \Rightarrow ... \Rightarrow \mu_n$ .

A simulação do desenvolvimento de um fragmento multicelular, tal como encontrado nas cianobactérias<sup>6</sup> como a *Anabaena Catenula* [Pru90], ilustra o funcionamento dos sistemas D0L. Este tipo de processo de desenvolvimento é definido pelo sistema L da Figura 39.

Axioma:

 $a_r$ 

Regras de produção:

1:  $a_r \rightarrow a_l b_r$ 

 $2: a_l \rightarrow b_l a_r$ 

 $3: b_r \rightarrow a_r$ 

 $4: b_l \rightarrow a_l$ 

Figura 39 – Sistema L para simulação da cianobactéria Anabaena Catenula

Os símbolos "a" e "b" representam estados citológicos das células (tamanho e prontidão para divisão) e os índices "l" e "r" indicam a polaridade da célula ("left" ou "right"). Apenas as

<sup>&</sup>lt;sup>6</sup> As cianobactérias são um tipo de algas com função fotossintética aeróbica, que libertam oxigénio e fixam o azoto nos oceanos, tendo sido preponderantes para a formação da Biosfera. Estas bactérias possuem uma estrutura multicelular em forma de filamento. [Wikipedia]

células do tipo "a" têm tamanho suficiente para a subdivisão, que ocorre através das regras de produção 1 e 2, enquanto as células do tipo "b" crescem para células do tipo "a" (regras de produção 3 e 4) a cada passo de derivação. A partir de uma única célula  $a_r$  (o axioma), derivase a sequência de palavras (Figura 40) que representam o crescimento do filamento multicelular de *Anabaena Catenula*, que pode ser observado graficamente na Figura 41.

```
a_r
a_lb_r
b_la_ra_r
a_la_lb_ra_lb_r
b_la_rb_la_ra_rb_la_ra_r
```

Figura 40 - Sequência de palavras representando o filamento de Anabaena Catenula

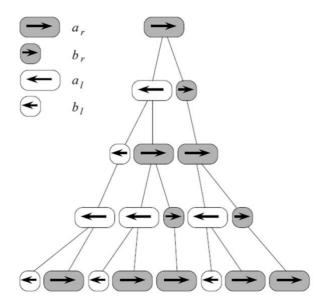


Figura 41 – Desenvolvimento do filamento de Anabaena Catenula utilizando um sistema DOL [Pru90]

#### A2. Sistemas L Sensíveis ao Contexto

Ao contrário dos sistemas L de contexto livre (sistemas 0L), em que as regras de produção têm apenas em consideração um único carácter (o predecessor), nos Sistemas L Sensíveis ao Contexto (*Context-sensitive L-systems*) [Prus90a], a aplicabilidade de uma regra de produção depende adicionalmente dos caracteres situados à sua esquerda ou à sua direita. Este facto traduz a natureza contextualizada do processo de desenvolvimento de alguns organismos, como por exemplo as plantas, onde o fluxo de nutrientes e hormonas influencia a interação entre diferentes partes da planta e onde a floração ocorre geralmente na extremidade dos ramos.

Em termos da sensibilidade ao contexto, os sistemas L podem ser classificados em sistemas 0L, sistemas 1L ou sistemas 2L, consoante sejam respetivamente, de contexto livre, sensíveis ao contexto num dos lados ou em ambos. Todos estes sistemas L pertencem à classe mais vasta dos sistemas IL, também denominados sistemas (k,l), cujo contexto esquerdo é uma palavra de comprimento k e o contexto direito é uma palavra de comprimento k.

Os Sistemas IL são uma extensão aos sistemas 0L pelo que a sua definição é baseada nestes, exceto no que respeita à aplicabilidade e definição das regras de produção.

Considerando  $\phi = \tau A \upsilon$ , em que  $\tau, \upsilon \in V^+ e A \in V$ , as regras de produção correspondem ao par  $(\phi, \chi) \in P$ , representando-se como  $\tau < A > \upsilon \to \chi$ . O carácter A é designado predecessor estrito e pode ser aplicável na produção da palavra  $\chi$  se e só se o carácter A é precedido à esquerda pela palavra  $\tau$  e sucedido à direita pela palavra  $\upsilon$ . Sendo k o comprimento da palavra  $\tau$  e l, o comprimento da palavra  $\upsilon$  diz-se estar na presença de um sistema (k,l).

Assume-se que as regras de produção sensíveis ao contexto têm precedência sobre as restantes com o mesmo predecessor estrito. Em consequência, se uma regra de produção sensível ao contexto e outra de contexto livre forem aplicáveis, a primeira deve ser selecionada. Se nenhuma regra de produção for aplicável, este carácter deve ser substituído por si próprio, através da regra de produção identidade  $A \rightarrow A$ , tal como o assumido para os sistemas 0L.

Na Figura 42 define-se um sistema L sensível ao contexto para a simulação da propagação de um sinal. Como se pode constatar por observação da Figura 43, o sinal (representado pela letra "b") vai-se propagando pelo meio de propagação (representado pela letra "a"), ao longo da sequência de desenvolvimento.

Axioma:

baaaaaaaa

Regras de produção:

 $1: b < a \rightarrow b$ 

 $2 \colon b \to a$ 

Figura 42 - Sistema L sensível ao contexto

baaaaaaaa

abaaaaaaa

aabaaaaaa

aaabaaaaa

aaaabaaaa

Figura 43 - Exemplo da sequência de desenvolvimento de um sistema L sensível ao contexto

# A3. Sistemas L Parentéticos

Os Sistemas L Parentéticos (*Bracketed L-systems*) foram desenvolvidos no sentido de permitirem a modelação de estruturas hierárquicas em forma de árvore. O conceito fundamental, tal como definido em [Pru90] e [Pru96], é o da utilização de cadeias de caracteres com parêntesis. Os símbolos "[" e "]" delimitam os ramos da estrutura hierárquica, sendo posteriormente interpretados, respetivamente, como operações *push* e *pop* sobre uma pilha de estado.

Considerando V um alfabeto não vazio e finito e os parêntesis ("[" e "]") como dois símbolos fora de V denominados delimitadores de ramo, denota-se respetivamente  $V_E = V \cup \{[,]\}$  como extensão de V. Considere-se ainda V\* o conjunto de todas as palavras em V, e V+ o conjunto de todas as palavras bem formadas em  $V_E$ .

Uma palavra em  $V_E$  é bem formada se e só se puder ser especificada por um número finito de aplicações das seguintes regras:

- Cada palavra  $\mu \in V^*$  é bem formada;
- Se  $\tau$ ,  $\upsilon \in V_E^*$  são bem formadas então [ $\tau$ ] e  $\tau \upsilon$  também são bem formadas;

Uma palavra  $[\omega] \in V_E^*$ , tal que  $\omega$  é bem formada, denomina-se ramo. A composição padrão de um ramo  $[\omega] \in V_E^*$  é uma palavra na forma  $[\omega] = [\chi_1[\alpha_1]\chi_2[\alpha_2]...\chi_n[\alpha_n]\chi_{n+1}]$ , em que as palavras  $\chi_1, \chi_2, ..., \chi_{n+1} \in V^*$  não contêm parêntesis e as palavras  $\alpha_1, \alpha_2, ..., \alpha_n \in V_E^*$  são bem formadas. A palavra  $\chi_1\chi_2...\chi_n\chi_{n+1}$  é denominada o eixo principal, as palavras  $\chi_1, \chi_2, ..., \chi_n$ , são denominadas entrenós e a palavra  $\chi_{n+1}$  é denominada de ápice. As palavras  $[\alpha_1], [\alpha_2], ..., [\alpha_n]$  são denominadas ramos laterais (de 1ª ordem) de  $[\omega]$ .

Define-se assim um Sistema D0L Parentético (BD0L - *Bracketed D0L-system*) como  $G=(V_E,[\omega],P)$ , em que o axioma  $[\omega]$  é uma palavra bem formada sobre o alfabeto  $V_E$ , e P o conjunto de regras de produção  $P \subset V_E \times V_E^*$ .

Cada regra de produção tem a forma  $A \to \chi$ , onde  $a \in V$ ,  $\chi \in V_E^*$  e  $\chi$  é bem formada; aos parêntesis correspondem as regras de produção identidade, respetivamente  $[\to]$  e  $]\to]$ .

Exemplificando a aplicação dos sistemas L parentéticos, a Figura 44 apresenta uma cadeia definindo uma estrutura em forma de árvore. A estrutura representada por  $[\omega]$  é composta por um eixo principal, constituído pelos símbolos *abefjlm*, aos quais estão ligados os ramos laterais cd, k e gi, contendo este último um ramo lateral de  $2^a$  ordem h.

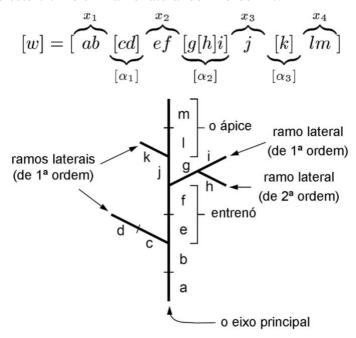


Figura 44 - Estrutura hierárquica definida por meio de sistemas L parentéticos [Pru96]

# A4. Sistemas L Estocásticos

Uma grande variedade de sistemas, tais como os ecossistemas naturais, apresenta um aspeto diversificado denotando a aleatoriedade que afeta o seu desenvolvimento. Observando um bosque não é possível encontrar duas árvores iguais, não obstante a sua semelhança. Os sistemas L determinísticos não conseguem captar esta essência e são assim inadequados para simular este tipo de sistemas.

Os Sistemas L Estocásticos (*Stochastic L-systems*) [Pru90] incorporam um carácter aleatório que permite que várias regras de produção possam ser aplicáveis a um mesmo símbolo, e que a seleção da regra de produção a aplicar seja feita de forma estocástica (aleatória), baseada na probabilidade associada a cada uma delas.

Um sistema 0L estocástico é uma quadra ordenada  $G = (V, \omega, P, \pi)$ , onde o alfabeto V, o axioma  $\omega$  e o conjunto de regras de produção P são definidas tal como num sistema 0L. A função  $\pi$ :  $P \to J0, IJ$  é denominada distribuição de probabilidade e mapeia o conjunto de regras de produção no conjunto de probabilidades das regras de produção. Assume-se que, para qualquer carácter  $A \in V$ , a soma das probabilidades de todas as regras de produção com predecessor A é igual a 1.

A derivação  $\mu \Rightarrow \nu$  denomina-se derivação estocástica se, para cada ocorrência do carácter A na palavra  $\mu$ , a probabilidade de aplicação da regra de produção P com predecessor A seja igual a  $\pi(p)$ . Desta forma, diferentes regras de produção com o mesmo predecessor podem ser aplicadas a diversas ocorrências do mesmo carácter da cadeia de caracteres, a ser operada num determinado passo de derivação.

Um exemplo simples que ilustra o princípio de funcionamento de um sistema L estocástico é o utilizado para geração de arbustos, sendo apresentado na Figura 45. O sistema L é composto por três regras de produção estocásticas com igual probabilidade, que definem três formas distintas de fazer evoluir cada ramo do arbusto: uma primeira regra cria duas bifurcações do ramo, enquanto que as duas regras seguintes apenas criam uma bifurcação, em orientações diferenciadas. A seleção aleatória da regra a aplicar gera resultados distintos para cada uma das sequências de desenvolvimento efetuadas.

Axioma:

F

Regras de produção:

$$1\colon F \stackrel{0,33}{\longrightarrow} F[+F]F[-F]F$$

$$2: F \xrightarrow{0,33} F[+F]F$$

$$3: F \xrightarrow{0,34} F[-F]F$$

Figura 45 – Sistema L estocástico

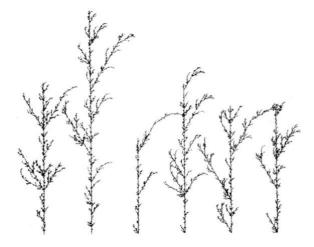


Figura 46 – Figura representando arbustos gerados através de um sistema L estocástico [Pru90]

A Figura 46 ilustra o resultado da aplicação deste sistema L estocástico, mostrando diversos arbustos, com aspecto distinto, que foram gerados. Dado que existe sempre um carácter "F" na cadeia de caracteres a cada passo de derivação, o conjunto de regras de produção pode ser aplicado indefinidamente. De forma a que o sistema termine em tempo útil com o resultado pretendido, é definido um limite máximo para o número de passos de derivação da sequência de desenvolvimento que gera cada um destes arbustos.

### A5. Sistemas L Paramétricos

Os Sistemas L Paramétricos (*parametric L-systems*), tal como descritos em [Pru90], estendem o conceito base dos sistemas L, associando atributos numéricos a cada carácter, no que se designa por módulos. O objetivo principal desta extensão dos sistemas L é o de potenciar que cada carácter possa ter interpretações distintas, consoante os parâmetros associados. Desta forma, um sistema L paramétrico opera, não sobre uma cadeia de caracteres, mas sim sobre uma cadeia modular paramétrica.

Os sistemas L paramétricos operam com palavras paramétricas, ou seja, cadeias de módulos consistindo em caracteres com parâmetros associados. Os caracteres pertencem a um determinado alfabeto V não vazio e finito, e os parâmetros pertencem ao conjunto dos números reais  $\mathcal{R}$ . Um módulo com carácter  $A \in V$  e parâmetros  $a_1, a_2, ..., a_n \in \mathcal{R}$  é representado por  $A(a_1, a_2, ..., a_n)$ . Todos os módulos pertencem ao conjunto  $M = V \times \mathcal{R}^*$ , em que  $\mathcal{R}^*$  representa o conjunto de todas as sequências finitas de parâmetros. O conjunto de todas as cadeias de módulos e o conjunto de todas as cadeias não vazias denotam-se respetivamente por  $M^* = (V \times \mathcal{R}^*)^*$  e  $M^+ = (V \times \mathcal{R}^*)^+$ .

Os argumentos de valor real que aparecem nas palavras correspondem a parâmetros usados na especificação das regras de produção dos sistemas L. Sendo  $\Sigma$  um conjunto de parâmetros, então  $C(\Sigma)$  denota uma expressão lógica com parâmetros de  $\Sigma$ , e  $E(\Sigma)$  uma expressão aritmética com parâmetros do mesmo conjunto. Ambos os tipos de expressões consistem em parâmetros formais e constantes numéricas, combinados pela utilização de operadores aritméticos, relacionais e lógicos, definidos na linguagem de programação C e ainda pelo operador exponenciação e parêntesis. Os conjuntos de todas as expressões aritméticas e lógicas construídas corretamente com parâmetros de  $\Sigma$  denotam-se  $\zeta(\Sigma)$  e  $\Xi(\Sigma)$ .

Um sistema 0L paramétrico é definido como uma quadra ordenada  $G = (V, \Sigma, \omega, P)$ , onde V é o alfabeto do sistema,  $\Sigma$  é o conjunto de parâmetros formais,  $\omega \in (V \times R^*)^+$  é uma palavra paramétrica não vazia, denominada axioma, e  $P \subset (V \times \Sigma^*) \times \mathcal{G}(\Sigma) \times (V \times \Xi(\Sigma))^*$  é um conjunto de regras de produção finito.

Os símbolos ":" e "\rightarrow" são usados para separar os três componentes de uma regra de produção: o predecessor, a condição e o sucessor.

A aplicabilidade de uma regra de produção depende da veracidade simultânea das seguintes condições:

- O carácter do módulo e o carácter do predecessor da regra de produção são os mesmos;
- O número de argumentos do módulo é igual ao número de parâmetros do predecessor da regra de produção;
- A condição é avaliada como verdadeira ao instanciar os parâmetros com o valor dos argumentos do módulo a avaliar.

Quando as três condições se verificam num determinado módulo, a regra pode ser aplicada, substituindo-o pela cadeia de módulos definida pelo sucessor. Os valores dos argumentos do módulo são utilizados para instanciar os parâmetros dos módulos sucessores, de acordo com a sua posição. A definição da sequência de desenvolvimento é a mesma dos sistemas D0L.

A Figura 6, apresentada no início da secção 2.2.3, descreve um sistema L paramétrico muito simples para a definição da curva "Floco de neve" (Figura 2). A Figura 47 ilustra um outro sistema L paramétrico, composto por dois pares de regras de produção, cada qual aplicável ao

mesmo módulo paramétrico. A seleção da regra de produção a aplicar a cada módulo é efetuada com base nas condições associadas, que neste caso particular são mutuamente exclusivas.

Axioma:

B(2) A(4, 4)

Regras de produção:

1:  $A(x, y) : y \le 3 \rightarrow A(x^2, x+y)$ 

2:  $A(x, y) : y > 3 \rightarrow B(x) A(x/y, 0)$ 

3: B(x): x<1 → C

 $4: B(x): x >= 1 \to B(x-1)$ 

Figura 47 – Definição de um sistema L paramétrico

Na Figura 48 pode observar-se a evolução dos valores dos parâmetros dos módulos ao longo dos primeiros passos da sequência de desenvolvimento do sistema L paramétrico definido.

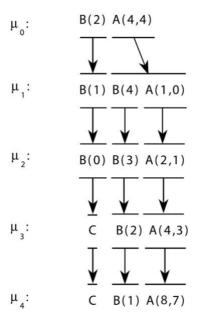


Figura 48 - Sequência de desenvolvimento de um sistema L paramétrico