

UFES - CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA

Prof. Thomas W. Rauber

Estruturas de Dados I – Engenharia de Computação & Ciência da Computação

2º Trabalho 2015/1 – Sistemas de Lindenmayer

Última atualização: 12 de junho de 2015, 10:07

Data de entrega: veja www.inf.ufes.br/~thomas

Linguagem de Programação para Implementação: C

Grupo de até dois alunos

Objetivo: Criação de interpretador de um sistema de Lindenmayer (*L-system*). Visualização do objeto gráfico gerado. Representação e manipulação de informação estruturada por linguagem de programação de alto nível, listas, pilhas e árvores.

Parte I (8 Pontos):

Um sistema de Lindenmayer [3] [6], [4], [1] [2], é uma gramática formal que permite a geração de palavras finais (strings) que podem ser interpretadas como comandos simples de uma sistema gráfico de tartaruga. A sintaxe adotada neste trabalho deve ser o do material¹ de Jennings [5].

Em um primeiro instante o programa deve ler um arquivo que contém a descrição completa da gramática e preencher variáveis o estruturas de dados apropriadas.

A funcionalidade da primeira versão devem incluir todos os comandos do preâmbulo e da produção e os comandos básicos da tartaruga, listados na tabela 1

Preâmbulo	
angle n	Vira a tartaruga $360/n$ graus quando o comando + ou - for acionado
order n	aplique a regras de produção n vezes
rotate α	Rotacione a imagem inteira α antes de começar a desenhar
Produção	
axiom <i>string</i>	A string inicial do L-Sistema (ordem 0)
$p = \text{string}$	Regra de produção. Em cada iteração (ordem) todas as instâncias do símbolo único p serão substituídos pela string <i>string</i> . Note que p obrigatoriamente é um único símbolo, e não pode ser do conjunto de caracteres especiais =, +, -, !, , [,], <, >, @, /, \, -, c, ou espaço. Maiúsculos são diferentes de minúsculos.
Gráficos de Tartaruga	
F	Mover tartaruga com caneta baixada
G	Mover tartaruga com caneta levantada
+	Virar tartaruga por ângulo positivo
-	Virar tartaruga por ângulo negativo
[Empilhar o estado da tartaruga (posição, orientação, comprimento da reta e cor)
]	Desempilhar o estado da tartaruga

Tabela 1: Sintaxe do L-Sistema segundo [5]

As regras de produção devem ser organizadas como lista encadeada.

Todas as produções devem ser organizadas em uma árvore com número de filhos variável. A string final são as folhas dessa árvore. Inicialmente a string que cresce durante a aplicação das regras pode ser implementada como string mesmo, ou seja, sem usar a estrutura árvore, o que facilita a implementação (ou em paralelo). A string final unicamente é composta pelos comandos gráficos da tartaruga da tabela 1, sem serem interpretados pelo sistema de produção da string final. Exemplo: Com o axioma '++FX' e a regra 'X→[-FX]+FX', com ordem 1, a string final deve ser '++F[-FX]+FX' e com ordem 2 a string final deve ser '++F[-F[-FX]+FX]+F[-FX]+FX'.

¹A página contém um Java applet que tem que ser executado para ver o sistema em ação. Algumas versões mais novas do Java não permitem a execução deste applet não certificado. Assim tem que introduzir uma exceção pela interface de configuração do Java para permitir a execução. No Linux é o comando `javaws`, aba 'Security', 'Edit Site List', 'Add' do site citado em [5].

Dicas: Como tem uma quantidade elevada de processamento de cadeia de caracteres (strings), recomenda-se usar as utilidades providenciadas pelo sistema, por exemplo `strstr` para procurar uma string dentro da outra, ou `strcspn` para eliminar caracteres indesejados dentro de uma string.

Parte II (2 Pontos):

Os comandos de empilhar e desempilhar o estado da tartaruga [e], devem ser interpretados do lado do sistema não gráfico. Para atingir este objetivo tem que existir uma pilha que guarda uma estrutura do estado atual da tartaruga (posição (x,y), orientação, comprimento da reta desenhada e cor). Além disso, os comandos de desenhar uma reta (F) e se mover sem desenhar (G) têm que ser quebradas em unidades elementares, pois o retorno para um estado anterior requer a inicialização do caminho atual (`newpath` do Postscript).

Dessa maneira a string final gerada deve ser convertida para uma string de tartaruga composta por comandos básicos com a seguinte sintaxe:

```
n x0 y0 m x1 x2 l s
```

onde (x0,y0) e (x1,y1) são os pontos iniciais e finais de uma reta `n` significa `newpath`, `l` `lineto`, `m` `moveto` e `s` `stroke` e são diretamente interpretado pelo Postscript.

Exemplo que desenha um triângulo:

```
n 0 0 m 100 100 l s
n 100 100 m 200 0 l s
n 200 0 m 0 0 l s
```

A quebra de linha não precisa aparecer na string, portanto a seguinte chamada da função `genpsTurtle`, providenciada na interface, com a combinação de comandos acima desenha o triângulo:

```
genpsTurtle( 4.0, 2, 0.0,
"n 0 0 m 100 100 l s n 100 100 m 200 0 l s n 200 0 m 0 0 l s", "triangulo.ps");
```

Ajuda: É necessário guardar o estado gráfico em uma pilha (Sem mudar a cor e o tamanho da reta, pelo menos a posição (x,y) e orientação). Encontrando o símbolo `+`, a orientação deve ser incrementada pelo ângulo base, `-` decrementa. Quando aparecer `[`, deve empilhar o estado, com `]` desempilhar, ou seja restabelecer a posição e orientação. Quando aparecer `F`, um comando básico da tartaruga deve ser acrescentado ao string da tartaruga. A nova posição deve ser calculada da seguinte maneira

$$x_1 = x_0 + z \cos(\alpha), \quad y_1 = y_0 + z \sin(\alpha),$$

onde z é o comprimento da reta (*stroke length*) e α é a orientação atual. Cuidado: As funções trigonométricas do C exigem os argumentos em radianos.

Parte III (2 Pontos): Como bônus considere a implementação completa da sintaxe de Jennings, i.e. na tabela os comandos da tartaruga 'Drawing', "!", |, @, >, >n, <, <n, cn".

Elaboração: O resultado deve ser um executável que lê um arquivo de entrada com a especificação da gramática e produz dois arquivos de saída Postscript com os resultados, por exemplo `lsystem test.lsy test.ps testTurtle.ps`. Os alunos obrigatoriamente devem fornecer arquivos de exemplos testados. O usuário (neste caso o professor) não deve ter o trabalho de digitar nada, além da linha de comando. Uma interface para o Postscript está sendo fornecido, a função `genps` que gera o arquivo com a string diretamente produzida pelo L-Sistema e a função `genpsTurtle` da parte II do trabalho que gera o arquivo com a string de comandos básicos, convertidos da string original.

O produto final deve ser um arquivo no formato `zip` com a seguinte sintaxe: `aluno1+aluno2.zip`. Atenção: Nenhum executável ou código objeto pode estar dentro do projeto, pois os serviços de e-mail, como, por exemplo, Hotmail recusam o transporte de tais arquivos por razões de segurança. O aluno se responsabiliza pelo envio e recepção correta. Em caso de problemas maiores de tráfego de rede (serviços UFES fora do ar), o aluno deve mandar novamente o arquivo original (encaminhamento da mensagem original) quando o serviço voltar. O arquivo deve conter uma **única** pasta com o nome `aluno1+aluno2`. Duas subpastas devem conter o código fonte (sem código compilado e executável) e a documentação do projeto. O código deve ter um `Makefile` que me permite a compilação facilitada. Devem-se produzir vários arquivos de teste. O arquivo `aluno1+aluno2.zip` deve ser mandado como anexo **exclusivamente** copiando a hiperligação seguinte no browser ou cliente de E-mail:

`mailto:thomas@inf.ufes.br?subject=Estruturas%20de%20Dados%20I:%20Entrega%20de%20trabalho%20`

A documentação deve ser em forma de descrição de projeto com os, preferencialmente gerado por L^AT_EX, contendo os seguintes tópicos:

- Capa do Projeto
 - Título
 - Autoria
 - Data
 - Resumo
- Introdução
- Objetivos
- Metodologia
- Resultados e Avaliação
- Referências Bibliográficas

Rigidez na administração da memória dinamicamente alocada: Recomenda-se fortemente usar a ferramenta **valgrind**, e consequentemente o sistema operacional Unix. Acusa o vazamento de memória o que constitui uma degradação de qualidade do software e se refletirá na avaliação do trabalho.

Além disso, o programa deve ser robusto em relação a erros de sintaxe do arquivo de entrada. Por exemplo uma sintaxe inválida deve causar uma exceção que deve ser tratada. Nesse caso o programa deve acusar esse erro e sair (também nesse caso limpando a memória).

Bom trabalho!

Referências

- [1] Interactive L-system. <http://www.kevs3d.co.uk/dev/lsystems/>, 2015. Último acesso: 2015-05-21.
- [2] L-system-Matlab. http://www.mathworks.com/help/symbolic/mupad_ref/plot-lsys.html, 2015. Último acesso: 2015-05-21.
- [3] Aristid Lindenmayer. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of theoretical biology*, 18(3):280–299, 1968.
- [4] L-system. <http://en.wikipedia.org/wiki/L-system>, 2015. Último acesso: 2015-05-21.
- [5] L-system-Jennings. <http://www.cgjennings.ca/toybox/lsystems>, 2015. Último acesso: 2015-05-21.
- [6] L-Sistema. <http://pt.wikipedia.org/wiki/L-Sistema>, 2015. Último acesso: 2015-05-21.