

LEIM – Modelação e Simulação de Sistemas Naturais

Exame Segunda Época

6 fevereiro 2017

Grupo I – Sistemas Dinâmicos (5.0 val.)

Considere o seguinte modelo que descreve a dinâmica de um determinado sistema e onde se tem:

$$[TAXA \text{ MÁXIMA CRESCIMENTO}] = 0.4$$

$$[TAXA \text{ DE COLHEITA}] = 0.2$$

$$[CAPACIDADE \text{ MÁXIMA}] = 60$$

$$[\text{crescimento}] = [TAXA \text{ MÁXIMA CRESCIMENTO}] * [\text{Stock}] * (1 - [\text{capacidade relativa}])$$

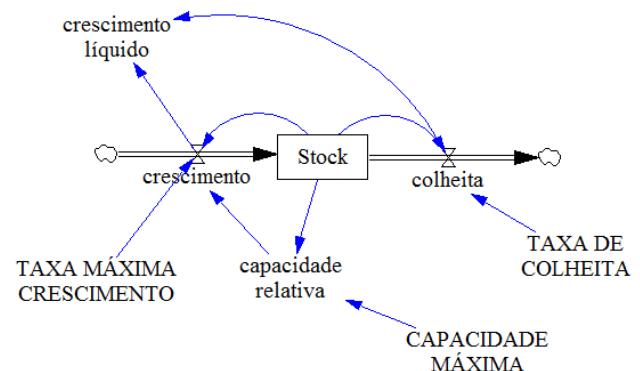
$$[\text{capacidade relativa}] = [\text{Stock}] / [CAPACIDADE \text{ MÁXIMA}]$$

$$[\text{colheita}] = [TAXA \text{ DE COLHEITA}] * [\text{Stock}]$$

$$[\text{crescimento líquido}] = [\text{crescimento}] - [\text{colheita}]$$

Nota: a unidade de tempo é [ano]

- Complete o diagrama marcando cada seta com a polaridade respetiva. Quantos ciclos de feedback contém o sistema? Identifique-o(s) e classifique-o(s). [1.0]
- Descreva o sistema através da respetiva equação diferencial. Quantos pontos fixos tem o sistema? Classifique-os quanto à estabilidade. Justifique. [1.0]
- Considere que o valor inicial ($t=0$) para o Stock é 6. Determine o valor da variável [Stock] para $t=1$ e $t=2$. [1.0]
- Nas condições da alínea anterior determine os respetivos valores da variável [crescimento líquido]. Esboce um gráfico da evolução, ao longo do tempo, desta variável. Justifique. [1.0]
- Descreva um sistema real que possa ser descrito por este modelo apresentado. [1.0]



Grupo II – Autómatos Celulares (5.0 val.)

Responda às seguintes questões relativas a autómatos celulares:

- Considere um autómato celular 2D, ternário, com vizinhança de von Neumann, com raio igual a 1. Para o caso mais geral, qual seria a dimensão da tabela de regra? Quantas seriam as diferentes regras possíveis de definir? Justifique. [1.0]
- Repita a alínea anterior para o caso do autómato ser totalístico externo. Justifique. [1.0]
- Considere a chamada “regra da maioria”. É uma regra totalística? Justifique. [1.0]
- Considere o Jogo da Vida, versão clássica (regra 23/3), com a seguinte configuração inicial. Determine a evolução do autómato. Comente o resultado obtido. [1.0]

				1	1	
	1		1		1	
	1	1				

- Repita a alínea anterior mas desta vez aplicando a regra 245/3 [1.0]

Grupo III – Agentes Autónomos (5.0 val.)

Considere quatro agentes autónomos cujos atributos, para $t=0$, estão expressos na seguinte tabela. Além disso, sabe-se que cada agente tem uma massa igual a 4, ângulo de visão de 180° , com alcance igual a 50, uma velocidade máxima de 5 e uma força máxima de 10.

	Atributos ($t=0$)
Agente 1	Localização: (20,30) Velocidade: (2,2)
Agente 2	Localização: (10,10) Velocidade: (2,2)
Agente 3	Localização: (60, 30) Velocidade: (4, -1)
Agente 4	Localização: (20, 50) Velocidade: (0, 2)

- Represente graficamente, no espaço 2D, as posições e velocidades dos agentes. Assinale também o campo de visão do agente 1. [1.3]
- Determine a força de guiamento, aplicada no agente 1, supondo que ele executa o método FLEE, relativamente ao agente 2. Justifique todos os cálculos realizados. [1.2]
- Determine a força de guiamento, aplicada no agente 1, supondo que ele executa o método ALIGN. Justifique os cálculos realizados. [1.3]
- Nas condições da alínea anterior e supondo um ritmo de simulação de 4 fps, determine a nova localização e velocidade do agente 1, na frame seguinte. Justifique. [1.2]

Grupo IV – Programação (5.0 val.)

Atente ao seguinte código para a classe **MyList** (a classe terá outros métodos não mostrados na tabela) e programa principal.

1	<code>class MyList {</code>
2	<code> ArrayList<Integer> userList;</code>
3	<code> MyList(int n) {</code>
4	<code> for(int i=0;i<n;i++) userList.add(i);}</code>
5	<code> int removeAndSum(int val) {</code>
6	<code> int sum = 0;</code>
7	<code> for(int i=0;i<userList.size();i++) {</code>
8	<code> int element = userList.get(i);</code>
9	<code> if (element == val) userList.remove(i);</code>
10	<code> else sum += element;}</code>
11	<code> return sum;}</code>
12	<code> }</code>
13	<code>void setup(){</code>
14	<code> MyList m = new MyList(6);</code>
15	<code> println(m.removeAndSum(2));}</code>

- O código tem um erro. Identifique-o e corrija-o. Esse erro manifestar-se-ia em tempo de compilação ou em tempo de execução? Justifique. [1.0]
- Tendo corrigido o erro, determine qual a mensagem que é enviada para a consola durante a execução do programa [1.0]
- Acha que o método **removeAndSum** faz aquilo que o programador pretendia ou terá algum erro (semântico)? Justifique a sua resposta e, eventualmente, corrija o erro. [1.0]
- Escreva o código do método **getMax** que retorna o maior valor da lista. [1.0]
- Escreva o código do método **product** que calcula o valor do produto dos elementos contidos na lista e acrescenta esse valor à lista. [1.0]