

CTC-11 Lógica Matemática – Lista de PROLOG -

Professor: Paulo Marcelo Tasinaffo.

Data de Divulgação: primeira semana de aula.

Data de Entrega: até sexta-feira da oitava semana de aula. O atraso na entrega da lista acarretará no desconto de 20% na nota da mesma. Depois das duas semanas de exames a lista de exercícios não será mais aceita pelo professor.

Regulamentos:

1. A lista pode ser resolvida em dupla;.
2. O formato de entrega do relatório com os exercícios resolvidos deve conter sequencialmente: enunciado do problema, listagem completa do código fonte que resolve o determinado exercício e um teste que valide o código. Isto deve ser feito em todas as dez questões que compõe esta lista de laboratório (com exceção da primeira questão).

1. Considere uma base de conhecimento que descreva fragmentos do conhecimento humano, por exemplo, sistemas jurídicos, diagnóstico médico, normas ou regulamentos para obtenção do título de engenheiro pelo ITA, etc. Para o exemplo escolhido estabeleça os sete passos da engenharia do conhecimento para criar um sistema de inferência em Lógica de Primeira Ordem (LPO).

Nota: no final desta lista de exercícios encontra-se um exemplo ilustrativo que pode inspirar a solução do mesmo. Trata-se do exemplo dado em sala de aula para o problema "de normas e regras para retirada de um passaporte".

2. Resolva o exercício 1 utilizando a linguagem Prolog. Utilize nesta resolução o aplicativo SWI-Prolog e faça testes e consultas nas bases de dados consideradas.

3. Exercício de Aquecimento: escolha cinco aplicativos Prolog de seu agrado – entre os exemplos apresentados em sala de aula – e rode-os no aplicativo *SWI-Prolog*. Façam consultas e vejam quais são as respostas obtidas pelo compilador.

4. Ordenar uma lista formada apenas por números. Exemplificando:

- a. `?- sort([1,5,3,9,7],X).`
- b. `X = [1,3,5,7,9]).`
- c. `yes`
- d. `?- sort([5,1,7,3,7,9],[1,3,5,7,7,9]).`
- e. `yes`

5. Encontrar o menor elemento de uma lista. Exemplificando:

- f. ?- min([3,5,1,7,9],X).
- g. X = 1
- h. yes
- i. ?- min([3,5,1,7,9],3).
- j. no
- k. ?- min([3,5,1,7,9],1).
- l. yes

6. Contar o número de vezes que um certo elemento aparece em uma dada lista. Exemplificando:

- m. ?- count([1,3,5,7,3,9],3,X).
- n. X = 2
- o. yes
- p. ?- count([a,b,a,a,b,a,c,a,b],a,X).
- q. X = 6
- r. yes
- s. ?- count([1,3,6,7,9],5,0).
- t. Yes

7. O problema do macaco e a da banana. Há um macaco ao lado da porta em uma sala. No meio da sala uma banana está pendurada no teto. O macaco está zangado e quer obter a banana, mas não pode alcançá-la do chão. Ao lado da janela da sala há uma caixa que o macaco pode utilizar. O macaco pode realizar as seguintes ações: (1) caminhar sobre o chão, (2) escalar a caixa, (3) empurrar a caixa ao redor da sala e (4) agarrar a banana se ele estiver estendido sobre a caixa diretamente sobre a banana. Pode o macaco obter a banana?

Responda as seguintes perguntas com relação a esse problema:

a) o problema resolvido já se encontra listado em uma das transparências do professor desta disciplina. Sendo assim digite-o no editor do Prolog, executando-o com a pergunta: “?-podeobter(estado(sobreporta,sobrechão,najanela,nãotem)).”. Nesse caso, verifique a resposta dada pelo Prolog.

b) O problema resolvido no item (a) possui as “ações” ordenadas desta forma: agarrar, escalar, empurrar e andar. De acordo com essa ordem, as *semânticas procedurais* do Prolog, o macaco preferirá agarrar em vez de escalar, escalar em vez de empurrar, etc. Entretanto, o que aconteceria se a ordem das cláusulas fosse diferente desta? Suponha então, colocar a relação andar em primeiro lugar. Nesta nova ordem refaça a pergunta “?-podeobter(estado(sobreporta,sobrechão,najanela,nãotem)).” e verifique se Prolog entra num *looping infinito*.

c) Se for constatado que o Prolog realmente entra num *looping infinito* dê uma explicação plausível porque esse problema ocorre.

8. Determine se os elementos de uma lista determinam um palíndromo. Um palíndromo é *uma palavra* ou *uma seqüência de caracteres* que podem ser lidos - da mesma forma - tanto da esquerda para à direita, como da direita para à esquerda (e.g., [x, a, m, a, x]).

9. Determine se um dado número inteiro é primo. Exemplo: “?- is _prime(7).”.

Sugestão: se o máximo divisor comum entre um “número dado” e o “próprio um” for o “próprio um”, então o número dado será primo.

10. Problema de *busca em largura* utilizando o *Prolog*. Seja o seguinte problema: duas fichas brancas e duas pretas estão dispostas conforme mostrado na figura abaixo. As fichas brancas estão separadas das pretas por um intervalo vazio. Deseja-se colocar as fichas pretas entre as brancas. Duas operações são permitidas: *escorregar uma ficha para o intervalo vazio* ou *fazê-la saltar sobre a outra e aterrissar no intervalo*. A solução deste problema é extremamente simples, mas deve ser resolvida em *Prolog*. Que pergunta deve-se fazer ao Prolog para resolver esse problema?

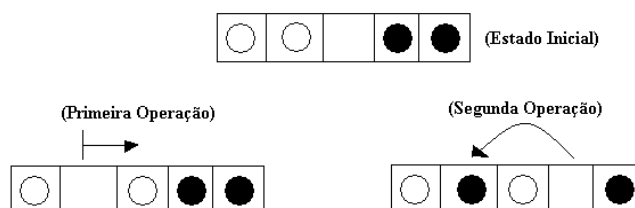


Figura 01 – Operações possíveis deste jogo.

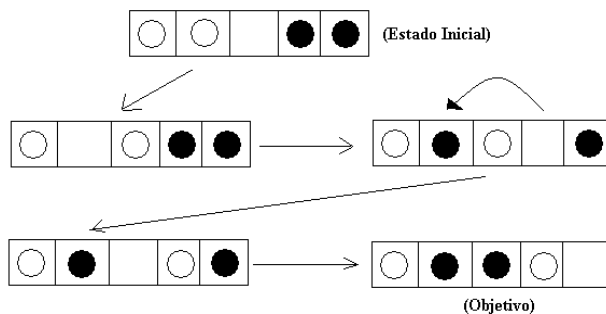


Figura 02 – Uma possível solução.

Boa Sorte ☺!
Prof. Tasinaffo.

Exemplo Ilustrativo Para Resolução da Primeira e Segunda Questões

1. Identificação da tarefa: elaborar uma Base de Conhecimento (BC) em Lógica de Primeira Ordem para emissão de passaportes segundo a legislação brasileira.

2. Agrupar condicionantes relevante:

- (i) Todo brasileiro honesto tem direito a um passaporte.
- (ii) Homens maiores de 18 anos devem possuir título de eleitor e reservista.
- (iii) Todo brasileiro deve estar em dia com o imposto de renda.
- (iv) Estrangeiros não podem ter passaporte do Brasil.

3. Definir um vocabulário de predicados, funções e constantes.

Domínios: cidadãos do mundo.

Predicados Unários

$\text{brasileiro}(x) \equiv x \text{ é brasileiro}$

$\text{honesto}(x) \equiv x \text{ é honesto}$

$\text{ImpostoRenda}(x) \equiv x \text{ está em dia imposto de renda}$

$\text{TítuloEleitor}(x) \equiv x \text{ possui título de eleitor}$

$\text{Reservista}(x) \equiv x \text{ possui reservista}$

Predicados Binários

$\text{homem}(x, y) \equiv \text{o homem } x \text{ possui a característica } y$

$\text{homem}(x, \text{MaiorIdade}) \equiv \text{o homem } x \text{ é maior de idade}$

$\text{passaporte}(x, y) \equiv \text{o indivíduo } x \text{ possui passaporte do país } y$

$\text{passaporte}(x, \text{Brasil}) \equiv x \text{ possui passaporte do Brasil}$

4. Definição das regras correspondentes a todos os termos do vocabulário.

Regra 01 (combinando (i)): $\forall x [(\text{brasileiro}(x) \wedge \text{honesto}(x)) \rightarrow \text{passaporte}(x, \text{Brasil})]$

Regra 02 (combinando (i), (ii) e (iii)): $\forall x [(\text{homem}(x, \text{MaiorIdade}) \wedge \text{Reservista}(x) \wedge \text{TítuloEleitor}(x) \wedge \text{ImpostoRenda}(x)) \rightarrow \text{honesto}(x)]$

Regra 03 (combinando (iv)): $\forall x \neg \text{brasileiro}(x) \rightarrow \neg \text{passaporte}(x, \text{Brasil})$

Regra 04: $\exists x \text{ homem}(x) \rightarrow \text{brasileiro}(x)$

5. Fonte de instâncias de conceitos e elaboração das hipóteses relevantes para α (BC) considerada:

313	1	brasileiro (Mário)	H
323	2	homem (Mário, Maria João)	H
333	3	reservista (Mário)	H
343	4	Título de Eleitor (Mário)	H
353	5	Imposto Renda (Mário)	H
G. Conceitos e inferências α (BC) conceitos e inferências relevantes	3,2,3,4,53	6	homem (Mário, Maria João) \wedge reservista (Mário) \wedge Título de Eleitor (Mário) \wedge Imposto Renda (Mário) 2,3,4,5 (A)
	3,2,3,4,53	7	homos (Mário) 6, Regra 02 (MP6) $\theta = 3x / \text{Mário}$
	3,2,3,4,53	8	brasileiro (Mário) \wedge homos (Mário) 4,7 (A)
		9	passaporte (Mário, Brasil) 8, Regra 01 (MP6) $\theta = 3x / \text{Mário}$

7. Dependence Base de Conhecimento:

Aparentemente os axiomas estão consistentes uma vez que o processo de inferência - pelo menos para o exemplo considerado - foi realizado com sucesso. Evidentemente, mais exemplos testados sem contradição confirmariam ainda mais a consistência dos axiomas considerados. Levando-se isto em consideração, testes importantes também seriam relevantes para verificar a consistência das regras 03 e 04.