Lógica Computacional

LEI, 2023/2024
FCT UNL

Aula Prática 11

Prolog

Pergunta 1. Considere a seguinte base de conhecimento em Prolog[1].

```
pessoa(maria). pessoa(joao). pessoa(tomas). gosta(maria, comida). gosta(maria, vinho).
gosta(joao, vinho). gosta(joao, maria). gosta(tomas, futebol). gosta(tomas, comida).
estuda(tomas, medicina). estuda(maria, matematica). estuda(joao, informatica).
estuda(pedro, informatica). mora(tomas, lisboa). mora(maria, almada). mora(joao, loures).
mora(pedro, almada).
```

a) Adicione regras que correspondam às seguintes afirmações:

1. O João gosta de tudo o que a Maria gosta.
2. O João gosta de qualquer pessoa que goste de vinho.
3. O João gosta de qualquer pessoa que esteja a estudar.
4. Quem gosta de comida gosta de vinho.

b) Considerando a base de conhecimento dada juntamente com as regras da alínea anterior, escreva interrogações que correspondam às seguintes questões:

1. O Tomás gosta de vinho?
   ?-gosta(tomas,vinho). true
2. Que pessoas, das que o João gosta, moram em Lisboa?
   ?-pessoa(x), gosta(joao,X), mora(X,lisboa). X=tomas X=tomas
3. Quais as pessoas de quem o Tomás gosta?
   ?-pessoa(X), gosta(tomas,X). false
4. Quais as pessoas de quem o João gosta e que gostam de futebol?
   ?-pessoa(X), gosta(joao,X), gosta(X,futebo). X=tomas X=tomas
5. Quem gosta de comida e vinho?
   ?-pessoa(X), gosta(x,comida), gosta(X,vinho). X=maria (x2) X=joao (x4) X=joao (x1)
6. Quais os gostos comuns do João e do Tomás?
   ?-gosta(joao,X), gosta(tomas,X). X= vinho X=comida X=vinho (x3)

c) Para as seguintes interrogações, desenhe a árvore de procura correspondente.

1. ?- gosta(maria, vinho).
2. ? - gosta(maria,X).
3. ?- mora(X, almada), estuda(X, informatica).
4. ?- gosta(tomas, comida).

[1]Pode experimentar a versão online do SWI Prolog, disponível em `https://swish.swi-prolog.org`

Pergunta 2. Considere a seguinte base de conhecimento em Prolog

```
homem(jose). homem(carlos). homem(miguel). homem(tomas). homem(rodrigo).
mulher(catarina). mulher(joana). mulher(sofia).
filho(carlos, jose). filho(joana, jose). filho(miguel, carlos). filho(catarina, carlos).
filho(tomas, carlos). filho(sofia, joana). filho(rodrigo, sofia).
```

a) Adicione regras que definam os predicados binários:
Irmão, Mãe, Pai, Tia, Tio, Avó, Avô, Primo

b) Considerando a base de conhecimento dada juntamente com as regras da alínea anterior, escreva interrogações que correspondam às seguintes questões:

1. O Rodrigo é primo da Catarina? falso
2. Quem são os tios da Sofia?
3. Quem são os netos do José?
4. Que pessoas têm irmãs?
5. Quem tem pelo menos um irmão e uma irmã?

c) Para as interrogações 1. e 3. acima, desenhe a árvore de procura correspondente.

Pergunta 3. Defina os seguintes predicados sobre listas.

a) **somaLista(L,N)** que verifica se N é a soma dos elementos da lista L de inteiros.
Exemplo: somaLista([7,-1,3],9)

b) **concat(L1,L2,L3)** que verifica se L3 é a concatenação das listas L1 com L2.
Exemplo: concat([a,b,c], [d,e], [a,b,c,d,e])

c) **palindrome(L)** que verifica que a lista L é igual à lista L invertida. Pode usar o predicado **concat** definido anteriormente.
Exemplo: palindrome([s,a,l,a,s])

d) **prefixo(L1,L2)** que verifica se L1 é um prefixo (a parte inicial) de L2. Pode usar o predicado **concat** definido anteriormente.
Exemplo: prefixo([7,a,joao], [7,a,joao,a,b,2])

e) **sufixo(L1,L2)** que verifica se L1 é um sufixo (a parte final) de L2. Pode usar o predicado **concat** definido anteriormente.
Exemplo: sufixo([a,b,2], [7,a,joao,a,b,2])

f) **subLista(L1,L2)** que verifica se L1 é uma sublista (alguma parte) de L2. Pode usar os predicados definidos anteriormente.
Exemplo: subLista([joao,a], [7,a,joao,a,b,2])

g) **ultimo(X,L)** que verifica se X é o último elemento da lista L.
Exemplo: ultimo(2, [7,a,joao,a,b,2])

Pergunta 4. Suponha que é dada uma base de conhecimento com os seguintes factos:

```
trd(um, one). trd(dois, two). trd(tres, three). trd(quatro, four). trd(cinco, five). trd(seis, six).
trd(sete, seven). trd(oito, eight). trd(nove, nine).
```

Escreva um predicado **listaTrd** que traduza uma lista de nomes de números em Português para os correspondentes nomes dos números em Inglês. Por exemplo a interrogação:

```
?- listaTrd([dois, sete, quatro], X).
```

deveria resultar em:

X=[two, seven, four].

Pergunta 5. Escreva uma base de conhecimento que defina o predicado fProp/1 das fórmulas proposicionais sobre um conjunto de símbolos proposicionais definido por sProp/1. Pode assumir que apenas tem o conjunto $\{p, q, r, s\}$ de símbolos proposicionais, e use **not/1, or/2, and/2, imp/2** como representação dos conectivos, e **bot** para o falso.

Pergunta 6. Suponha que é dada a seguinte base de conhecimento:

```
soma(0,N,N).
soma(s(N),M,s(Z)):- soma(N,M,Z).
```

Escreva interrogações que correspondam às seguintes questões:

1. Qual a soma de s(s(0)) com s(0)?
2. Que número é que somado com s(0) dá s(0)?
3. Quais os pares de números que somados dão s(s(0))?

Para cada uma das interrogações acima, desenhe a árvore de procura correspondente.