

MC346 - Paradigmas de Programação

Prova Prolog - 04/04/2016

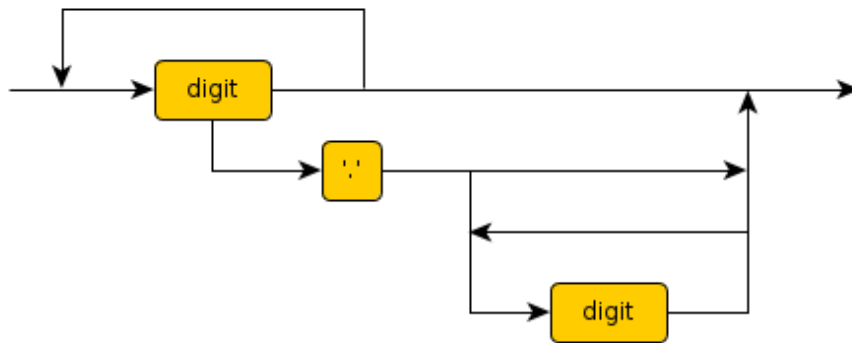
Em cada questão abaixo, faça com que os predicados pedidos falhem em tentativas de ressatisfação, a não ser que haja instruções explícitas em contrário no enunciado da questão.

Questão 1 (Valor 2,5) Escreva um predicado Prolog `palavras(N)` que lê uma linha da entrada padrão e coloca em `N` o número de palavras que há nesta linha. Uma palavra é um bloco de caracteres consecutivos, diferentes de branco e de fim de linha, cercado de ambos os lados por brancos, começo de linha ou fim de linha.

Exemplos de linhas (brancos indicados por `_`):

```
__minha_mae_nao_eh_homem__ (5 palavras)
tudo____pela____vida      (3 palavras)
_____                    (0 palavras)
```

Questão 2 (Valor 2,5) Escreva um predicado `numero(L)` em Prolog que recebe uma lista de caracteres e é satisfeito quando a lista contém um número binário válido de acordo com o seguinte diagrama sintático, onde `digit` só pode ser `'0'` ou `'1'`:



Exemplos:

```
?- numero(['1', '0', '1']).
true
?- numero(['0', '1', '1', '.'']).
true
```

```

?- numero(['1', '1', '0', '.', '1', '1']).
true
?- numero(['1', '3', '5', '.', '7', '2', 'e', '5']).
false

```

Questão 3 (Valor 2,5) Considere a seguinte função:

$$f(n) = \begin{cases} 3n + 1 & \text{se } n \text{ for ímpar} \\ n/2 & \text{se } n \text{ for par.} \end{cases}$$

Dizem que aplicando f sucessivamente a qualquer inteiro positivo chega-se ao número 1. Para ajudar a testar esta afirmação, escreva um predicado `passos(N, P)` que é satisfeito quando N é um inteiro positivo e P é o número mínimo de vezes que f tem que ser aplicada a N para obtermos o número 1. Exemplos:

Exemplos (onde \rightarrow indica uma aplicação de f):

```

?- passos(1, P). % note que 1 já é 1
P = 0
?- passos(2, P) % note que 2 → 1
P = 1
?- passos(3, P). % note que 3 → 10 → 5 → 16 → 8 → 4 → 2 → 1
P = 7

```

Questão 4 (Valor 2,5) Suponha que um fã de séries dramáticas tenha construído uma lista contendo todas as séries de que gosta, juntamente com o ano de início e final de cada série, no formato a seguir:

```

[serie(house_of_cards, 2013, 2016),
 serie(law_and_order, 1990, 2010),
 serie(breaking_bad, 2008, 2013), ...]

```

Escreva um predicado `series(L, A, S)` que recebe uma lista como esta no parâmetro L e um inteiro representando um ano no parâmetro A , retornando em S uma sublista de L contendo apenas **os nomes** das séries que estavam no ar naquele ano.

Boa sorte!