Roteiro 9

Alexsandro Santos Soares
prof.asoares@gmail.com

Programação Lógica
Faculdade de Computação
Universidade Federal de Uberlândia

12 de fevereiro de 2022

Este roteiro tem por finalidades:

- Praticar o uso de predicados para manipulação de arquivos.
- Exercitar os operadores definidos pelo usuário em definições recursivas envolvendo fórmulas da lógica proposicional.

Ao fazer os exercícios não use qualquer predicado pré definido ou de alguma biblioteca que resolva diretamente o problema pedido.

1 Exercícios envolvendo arquivos

Ex. 1 Escreva um programa que leia um arquivo texto palavra por palavra e insira todas as palavras lidas juntamente com suas frequências na memória do Prolog. Use um predicado dinâmico `palavra/2` para armazenar as palavras, com o primeiro argumento é uma palavra e o segundo é a frequência desta palavra no arquivo. O predicado principal será chamado de `histograma/1` e recebe como argumento o nome do arquivo. O programa deve ignorar sinais de pontuação e espaços em branco.

Como exemplo, suponha que o conteúdo abaixo pertença ao arquivo `arq1.txt`:

```
tinha suspirado, tinha beijado o papel devotamente! Era a primeira vez que
lhe escreviam aquelas sentimentalidades, e o seu orgulho dilatava-se ao
calor amoroso que saía delas, como um corpo ressequido que se estira num
banho tépido; sentia um acréscimo de estima por si mesma, e parecia-lhe que
entrava enfim numa existência superiormente interessante, onde cada hora
tinha o seu encanto diferente, cada passo condizia a um êxtase, e a alma
se cobria de um luxo radioso de sensações!
```

Então, as consultas abaixo deveriam ser verdadeiras para ele:

```
?- histograma('arq1.txt').
true

?- palavra(a, Freq).
Freq=3

?- palavra(acréscimo, Freq).
Freq=1

?- palavra(cada, Freq).
Freq=2

?- palavra(tinha, Freq).
Freq=3

?- palavra(um, Freq).
Freq=4

?- palavra(devotamente, Freq).
Freq=1
```

Ex. 2 Escreva um predicado `copia_arq/2` que copia o conteúdo de um arquivo para outro. O primeiro argumento é o nome do arquivo fonte e o segundo é o nome do arquivo de destino.

Após a consulta abaixo deverão existir dois arquivos: `arq1.txt`, o arquivo fonte; e o arquivo `cópia.txt` que é uma cópia exata de `arq1.txt`.

```
?- copia_arq('arq1.txt', 'cópia.txt').
true
```

Ex. 3 Escreva um predicado `idênticos/2` que compara o conteúdo de dois arquivos e decide se eles são idênticos. Como exemplo, considere `arq1.txt` e `cópia.txt` do exercício anterior. A consulta a seguir deve ser verdadeira, pois um arquivo é uma cópia exata do outro.

```
?- idênticos('arq1.txt', 'cópia.txt').
true
```

Se os arquivos diferirem em algum caracter, o predicado deve falhar. Lembre-se de sempre fechar os arquivos após abrí-los para qualquer que seja o uso.

2 Exercícios envolvendo lógica proposicional

Ex. 4 Seja H uma fórmula da Lógica Proposicional (LP), então:

- H é uma subfórmula de H;
- se H é uma fórmula do tipo $(\neg G)$, então G é uma subfórmula de H;
- se H é uma fórmula do tipo: $(G \vee E)$, $(G \wedge E)$, $(G \rightarrow E)$ ou $(G \leftrightarrow E)$, então G e E são subfórmulas de H;

- se G é subfórmula de H, então toda subfórmula de G é subfórmula de H.

Usando esta definição, escreva um predicado `subfórmula/2` que é verdadeiro sempre que seu primeiro argumento é uma subfórmula de seu segundo argumento.

```
?- subfórmula(~ q, (p => q) <=> (~ q => ~ p) ).
true
?- subfórmula(q, (p => q) <=> (~ q => ~ p) ).
true
?- subfórmula(~ q => ~ p, (p => q) <=> (~ q => ~ p) ).
true
?- subfórmula(r, (p => q) <=> (~ q => ~ p) ).
false
?- subfórmula(~ q => p, (p => q) <=> (~ q => ~ p) ).
false
```

Ex. 5 H é uma *tautologia* se, e somente se, para toda interpretação I, $I[H] = T$.

Escreva um predicado `tautologia/1` que recebe uma fórmula `H` da LP e é verdadeiro sempre que `H` é uma tautologia.

```
?- tautologia( p <=> ~ ~ p ).
true
?- tautologia( p v ~ p ).
true
?- tautologia( p => p v q ).
true
?- tautologia( (p => q) <=> (~ q => ~ p) ).
true
?- tautologia( p & ~ p ).
false
```

Ex. 6 H é uma *contraditória* se, e somente se, para toda interpretação I, $I[H] = F$.

Escreva um predicado `contraditória/1` que recebe uma fórmula `H` qualquer da LP e é verdadeiro sempre que `H` é uma contraditória.

```
?- contraditória( p & ~ p ).
true
?- contraditória( p v ~ p ).
false
```

Ex. 7 Escreva um predicado `equivalente_disjuntivo/2` cujo primeiro argumento é uma fórmula qualquer da LP E e que seja verdadeiro sempre que o segundo argumento E_1 é uma fórmula equivalente a E, que possui apenas os conectivos $\neg$ e $\vee$ e os símbolos proposicionais e de verdade presentes em E.

```
?- equivalente_disjuntivo( ~(p & q), ~p v ~q).
true
?- equivalente_disjuntivo( p & q, ~(~p v ~q) ).
true
```

```
?- equivalente_disjuntivo( p v q, p v q).
true
?- equivalente_disjuntivo( p => q, ~p v q).
true
?- equivalente_disjuntivo( p <=> q, ~(~p v ~q) v ~(p v q)).
true
?- equivalente_disjuntivo( p => q, ~(p & ~q) ).
false
```

3 Sugestões de leitura

- Luiz Gustavo Almeida Martins. *Lógica para Computação.*

  http://www.facom.ufu.br/~gustavo/Logica/Logica.html

- Luiz A. M. Palazzo. *Introdução à programação Prolog*

  http://puig.pro.br/Logica/palazzo.pdf

- Eloi L. Favero. *Programação em Prolog: uma abordagem prática*

  http://www3.ufpa.br/favero

- Wikilivro sobre Prolog em

  http://pt.wikibooks.org/wiki/Prolog

- Patrick Blackburn, Johan Bos and Kristina Striegnitz. *Learn Prolog Now!*

  http://www.learnprolognow.org