

Atividade Avaliativa REO4 : Paradigma Lógico

Descrição

(100 pts.) Considere a seguinte base de dados (com apenas fatos), disposta em duas colunas:

parent(jafe, gomer).	parent(cao, pute).
parent(jafe, magoge).	parent(cao, canaa).
parent(jafe, madai).	parent(cuxe, ninrode).
parent(jafe, javan).	parent(cuxe, seba).
parent(jafe, tubal).	parent(cuxe, havila).
parent(jafe, meseque).	parent(cuxe, sabta).
parent(jafe, tiras).	parent(cuxe, raama).
parent(gomer, asquenaz).	parent(cuxe, sabteca).
parent(gomer, rifate).	parent(misraim, ludim).
parent(gomer, togarma).	parent(misraim, anaquim).
parent(javan, eliza).	parent(misraim, leabim).
parent(javan, tarsis).	parent(misraim, naftum).
parent(javan, quitim).	parent(misraim, patrusim).
parent(javan, dodaim).	parent(misraim, causulim).
parent(cao, cuxe).	parent(raama, seba).
parent(cao, misraim).	parent(raama, deda).

Essa base de dados pode ser salva em um arquivo com extensão '.pl', com o nome 'genealogy.pl', por exemplo. Utilizando o SWI-Prolog, que é uma distribuição muito usada do Prolog, disponível para diversas plataformas, você pode carregar essa base utilizando o predicado `consult`. Veja o exemplo, em que o arquivo `genealogy.pl` está na área de trabalho (Desktop):

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- pwd.
% //
true.

?- consult('/Users/erickmaziero/Desktop/genealogy.pl').
true.

?- parent(X, Y).
X = jafe,
Y = gomer ;
X = jafe,
Y = magoge ;
X = jafe,
Y = madai .

?- parent(gomer, X).
X = asquenaz ;
X = rifate .

?-
```

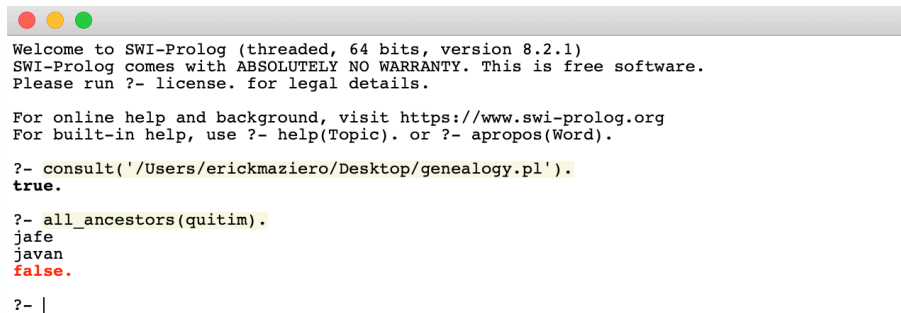
O predicado `pwd` exibe o local em que o SWI-Prolog está sendo executado, isso facilita definir o caminho que você deverá passar para o predicado `consult`, para encontrar o seu arquivo, com a base de dados.

Após o carregamento da base (`genealogy.pl`), você pode realizar várias consultas, com os fatos definidos. Por exemplo, a consulta `parent(X, Y)` retorna, na ordem em que os fatos estão definidos na base, a unificação para as variáveis `X` e `Y`. Caso eu digite `;` (ponto e vírgula) o processo de inferência do Prolog, irá procurar por mais fatos. Caso eu digite `.` (ponto) ele encerra o processo.

Já no exemplo `parent(gomer, X)`, o processo de inferência busca apenas por valores para a variável `X`, pois o valor `gomer`, já deve estar na base.

Você deve, então, criar um predicado, chamado `all_ancestors`, de aridade 1, ou seja, que receba apenas um 'valor de argumento', 'uma variável' e retorne todos os ancestrais do nome passado como argumento. Esse novo predicado deve ser escrito no final do arquivo com a base de fatos (`genealogy.pl`).

Por exemplo, se consultado `all_ancestors(quitim)`, devem ser exibidos `jafe` e `javan`, conforme ilustrado a seguir:



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('/Users/erickmaziero/Desktop/genealogy.pl').
true.

?- all_ancestors(quitim).
jafe
javan
false.

?- |
```

Entrega

A entrega deve ser feita, até o prazo permitido no Campus Virtual, anexando o arquivo `.pl` contendo a base de dados disponibilizada, acrescida da sua resolução.

Então, seu arquivo Prolog (`.pl`) deve conter a base de fatos do início deste documento mais a definição do predicado `all_ancestors`. Como exemplificado, resumidamente, a seguir:

```
parent(jafe, gomer).
parent(jafe, magoge).
...
parent(raama, seba).
parent(raama, deda).

all_ancestors(X) :- ...
```