

Aula Prática 5

Meta-Programação e Operadores

Objetivos:

- Uso de predicados meta-lógicos
- Declaração e uso de operadores

1. Predicados de ordem superior

- a) Implemente o predicado *map(+Pred, +List1, ?List2)*, com um funcionamento semelhante ao do predicado *maplist/3* da biblioteca de listas. Exemplo:

```
double(X, Y):- Y is X*2.  
| ?- map(double, [2,4,8], L).  
L = [4, 8, 16] ?  
yes
```

- b) Implemente o predicado *fold(+Pred, +StartValue, +List, ?FinalValue)*, com um funcionamento semelhante ao do predicado *scanlist/4* da biblioteca de listas. Exemplo:

```
sum(A, B, S):- S is A+B.  
| ?- fold(sum, 0, [2, 4, 6], F).  
F = 12 ?  
yes
```

- c) Implemente o predicado *separate(+List, +Pred, -Yes, -No)* que recebe uma lista de elementos e um predicado, devolvendo em *Yes* e *No* os elementos de *List* que tornam o predicado *Pred* verdadeiro ou falso, respetivamente.

```
even(X):- 0 == X mod 2.  
| ?- separate([1,2,3,4,5], even, Y, N).  
Y = [2,4],  
N = [1,3,5] ?  
yes
```

- d) Implemente o predicado *ask_execute/o* que lê da consola um objetivo e executa-o. Ex:

```
| ?- ask_execute.  
Insira o que deseja executar  
|: map(double, [2,4,8], L).  
yes
```

2. Functor, Arg e Univ

- a) Implemente os predicados *my_arg/3* e *my_functor/3* (com funcionamento idêntico aos predicados *arg* e *functor*) usando o *..* (univ).
- b) Implemente o predicado *univ/2* com base nos predicados *arg/3* e *functor/3*.

c) Defina *univ/2* como um operador infixo.

3. Associatividade de Operadores

Considere os seguintes operadores:

```
:-op(500, xfx, na).
:-op(500, yfx, la).
:-op(500, xfy, ra).
```

Desenhe a árvore de análise de cada uma das seguintes expressões.

- | | |
|----------------|----------------|
| a) a ra b na c | e) a na b na c |
| b) a la b na c | f) a la b la c |
| c) a na b la c | g) a ra b ra c |
| d) a na b ra c | |

4. Operadores

Considere os seguintes operadores:

```
:-op(550, xfx, de).
:-op(500, fx, aula).
:-op(550, xfy, pelas).
:-op(550, xfx, e).
:-op(600, xfx, as).
```

Desenhe a árvore de análise de cada uma das seguintes expressões.

- aula t de pfl as segundas pelas 11
- aula tp de pfl as tercas pelas 10 e 30
- aula 11 e aula 12 as 4 pelas cinco pelas 6 pelas sete

5. Criação de operadores

Defina operadores que tornem os termos abaixo sintaticamente válidos.

- flight tp1949 from porto to lisbon at 16:15
- if X then Y else Z

Acrescente o código necessário para que os termos da alínea b) sejam não só sintaticamente válidos como também utilizáveis. Exemplo:

```
| ?- if (3<4) then write(menor) else write(maior).
menor
yes
```

6. Operações com Listas

Defina operadores e as relações necessárias de forma a ser possível escrever e usar

- Element exists_in List
- append A to B results C
- remove Elem from List results Result