

Base de conhecimento dinâmica

Já sabemos que uma base de conhecimento é composta por fatos e regras.

Por exemplo:

```
joga(pelé,futebol).  
joga(guga,tênis).  
esporte(X) :- joga(_,X).
```

Podemos utilizar o predicado **listing** para listar as cláusulas da base.

Por exemplo:

```
? listing(joga)
```

Para adicionar uma nova cláusula dinamicamente, podemos utilizar os predicados **asserta** e **assertz**.

Contudo, para fazer isso, pode ser necessário informar quais predicados serão dinâmicos.

Por exemplo:

```
:- dynamic joga/2.  
joga(pelé,futebol).  
joga(guga,tênis).  
esporte(X) :- joga(_,X).
```

Adicionando dinamicamente:

```
? asserta(joga(romário, futebol)).  
? assertz(joga(nadal, tênis)).  
? listing(joga).
```

Para remover um cláusula dinamicamente, podemos utilizar o predicado **retract**.

```
? retract(joga(X, futebol)).  
? listing(joga).
```

Podemos salvar o estado da base dinâmica, utilizando os predicados **tell** e **told**.

Por exemplo:

```
salvar(predicado, arquivo) :- tell(arquivo),  
                             listing(predicado),  
                             told.
```

```
? salvar(joga, 'base.bd').
```

Podemos carregar uma base que está armazenada em disco, utilizando o predicado **consult**.

Por exemplo:

? consult(arquivo).

Vamos criar um exemplo de programa que caminha por locais informados:

```
:- dynamic estou/1.  
estou(nazare).  
ando(Destino) :- retract(estou(Origem)),  
                  asserta(estou(Destino)),  
                  format('Ando da ~w até a ~w',[Origem, Destino]).
```

Exercício

Questão 01: Implemente os predicados liga, desliga e lampada para que eles funcionem conforme indicado pelos exemplos a seguir:

```
?- liga, lampada(X).  
X = acessa  
Yes  
?- desliga, lampada(X).  
X = apagada
```

Solução

```
:- dynamic lampada/1.  
lampada(apagada).  
liga :- retract(lampada(_)),  
         asserta(lampada(acesa)).  
desliga :- retract(lampada(_)),  
           asserta(lampada(apagada)).
```

Questão 02: O predicado asserta adiciona um fato à base de dados, mesmo que ele já esteja lá. Para impedir essa redundância, defina o predicado memorize, tal que ele seja semelhante a asserta, mas só adicione à base de dados fatos inéditos.

Solução

```
memorize(Predicado) :- \+ Predicado, asserta(Predicado).
```