

Alunos: Gabriel Matheus Aguilera Silva e Rafael Yuki Tobaru

USANDO PROLOG NA LÓGICA DE PREDICADOS

O **objetivo** desta tarefa é a utilização de conceitos da lógica de predicados através da linguagem de programação Prolog.

Para realizar essa tarefa, considere o seguinte problema e em seguida, responda aos itens a seguir. A entrega deve ser feita em um documento que relate todos os passos (se houver necessidade, coloque os prints da tela) da tarefa e responda às perguntas feitas aqui. Para isso, a dupla deve criar um documento e compartilhar na entrega da tarefa. Além disso, deixe todo o desenvolvimento no <https://github.com/> e compartilhe o seu projeto com **tckg** (TCamis) e **poste-o na entrega da sua tarefa**.

PROBLEMA:

Suponha que um **banco de dados Prolog** contém as seguintes informações:

```
come(urso, peixe)
come(peixe, peixinho)
come(peixinho, alga)
come(guaxinim, peixe)
come(urso, guaxinim)
come(urso, raposa)
come(raposa, coelho)
come(coelho, grama)
come(urso, veado)
come(veado, grama)
come(lince, veado)
animal(urso)
animal(peixe)
animal(peixinho)
animal(guaxinim)
animal(raposa)
animal(coelho)
animal(veado)
animal(lince)
planta(grama)
planta(alga)
presa(X) <= come(Y, X) e animal(X)
```

Poderíamos, então, ter o seguinte diálogo com o Prolog:

```
?animal(coelho)
sim
?come(lince, grama)
não
?come(X, peixe)
urso
```

guaxinim
?come(X, Y) e planta(Y)
peixinho alga
coelho grama
veado grama
?presa(X)
peixe
peixinho
peixe
guaxinim
raposa
coelho
veado
veado

Note que o peixe é listado duas vezes na resposta à última consulta, pois os peixes são comidos por ursos (fato 1) e por guaxinins (fato 3). Analogamente, veados são comidos por ursos e por lince.

DESENVOLVIMENTO DO TRABALHO:

a. Por que o Prolog é considerado uma definição recorrente ou recursiva?

O Prolog é considerado uma linguagem de programação baseada em lógica de predicados, na qual as definições são frequentemente recursivas. Isso significa que os predicados podem ser definidos em termos de si mesmos, criando assim relações recursivas. Essa característica é útil para representar problemas que envolvem estruturas de dados recursivas, como listas e árvores, de forma concisa.

b. Explique porque são denominados como fatos (fato 1 e fato 3) os itens acima.

Os itens "fato 1" e "fato 3" são chamados de fatos em Prolog porque eles representam informações concretas e incontestáveis sobre o mundo representado pelo programa Prolog. Em outras palavras, eles são declarações que descrevem relações simples entre objetos e são tratados como verdades absolutas dentro do contexto do programa.

Por exemplo, o "fato 1" declara que o urso come peixe, e isso é uma afirmação concreta sobre a relação entre o urso e o peixe que é verdadeira em todas as circunstâncias dentro do programa. O "fato 3" estabelece que o guaxinim come peixe, e essa também é uma verdade inquestionável.

c. **Formule uma regra de Prolog que define o predicado *predador*.**

Uma regra de Prolog que define o predicado *predador*:

```
predador(X) :- come(X, _).
```

Essa regra diz que X é um predador se existe algum _ tal que X come _. Ou seja, qualquer animal que tenha uma relação de alimentação é considerado um predador.

d. **Adicione essa regra ao banco de dados do Exemplo acima e diga qual seria a resposta à consulta.**

?predador(X)

Essa regra diz que X é um predador se existe algum _ tal que X come _. Ou seja, qualquer animal que tenha uma relação de alimentação é considerado um predador.

```
?- predador(X).  
X = urso ;  
X = peixe ;  
X = peixinho ;  
X = guaxinim ;  
X = urso ;  
X = urso ;  
X = raposa ;  
X = coelho ;  
X = urso ;  
X = veado ;  
X = lince.
```

Note que os resultados são duplicados porque algumas relações de alimentação foram declaradas mais de uma vez no banco de dados.

e. **Encontre os resultados da consulta em cada caso no Problema**

?animal(lince)

```
?- animal(lince).  
true.
```

?planta(guaxinim)

```
?- planta(guaxinim).  
false.
```

?come(urso, peixinho).

```
?- come(urso, peixinho).  
false.
```

?come(raposa, coelho).

```
?- come(raposa, coelho).  
true.
```

?come(guaxinim, X).

```
?- come(guaxinim, X).  
X = peixe.
```

?come(X, grama).

```
?- come(X, grama).  
X = coelho
```

?come(urso, X) e come(X, coelho)

```
?- come(urso, X), come(X, coelho).  
X = raposa ;  
false.
```

- f. **Escreva, usando conceitos de Prolog, o significado de uma busca em profundidade.**

Uma busca em profundidade, em termos de Prolog, pode ser entendida como uma forma de explorar recursivamente uma árvore de busca ou grafo. É um algoritmo de busca que começa em um nó raiz e explora o máximo possível ao longo de cada ramo antes de retroceder. Em Prolog, isso pode ser implementado por meio de recursão, onde cada passo da busca gera novas consultas ou ramos para explorar.

O objetivo é encontrar uma solução para um problema ou satisfazer uma condição específica, como encontrar um caminho em um grafo ou encontrar uma combinação que satisfaça um conjunto de restrições.

- g. **Responda: Por que os conceitos de Prolog estão relacionados com a lógica de predicados? Faça uma sistematização com o conceito da regra de *Modus Ponens*.**

Os conceitos de Prolog estão relacionados à lógica de predicados porque a linguagem Prolog é baseada em lógica de predicados, que é uma extensão da lógica formal para representar e raciocinar sobre relações entre objetos e predicados.

A sistematização com o conceito da regra de Modus Ponens pode ser feita da seguinte maneira:

1. A lógica de predicados é uma extensão da lógica formal que lida com predicados, que são proposições que podem ser verdadeiras ou falsas em relação a objetos.
 2. Em Prolog, predicados são representados como relações entre objetos.
 3. A regra de Modus Ponens é uma regra de inferência em lógica proposicional que afirma que se temos uma afirmação "se A, então B" ($A \rightarrow B$) e sabemos que A é verdadeiro (A é um fato), então podemos concluir que B também é verdadeiro.
 4. Em Prolog, essa regra de inferência é usada implicitamente quando consultamos o banco de dados. Por exemplo, se temos o fato "come(urso, peixe)" e consultamos "come(urso, peixe)", o Prolog utiliza a regra de Modus Ponens para inferir que "peixe" é verdadeiro.
 5. Portanto, os conceitos de Prolog estão relacionados à lógica de predicados porque a linguagem Prolog é construída em torno da representação de predicados e do uso da lógica de predicados para inferir informações a partir de fatos e regras.
- h. Por fim, faça uma conclusão do seu trabalho, procure na literatura (atenção para buscas científicas como, por exemplo, o *Google Scholar*) estudos sobre a lógica de predicados e a linguagem Prolog em que vocês entendam como interessantes no referido trabalho. Lembre que esse é o momento de finalizar o que foi desenvolvido pelos autores do trabalho.**

Neste trabalho, exploramos o uso de Prolog para representar e resolver problemas baseados em lógica de predicados. Demonstramos como declarar fatos e regras em Prolog e como fazer consultas para obter respostas com base nas relações definidas no banco de dados. Além disso, mostramos como a recursividade é uma característica importante em Prolog, permitindo a representação de estruturas de dados recursivas.

Prolog é uma linguagem poderosa para a representação de conhecimento e raciocínio automatizado, especialmente em domínios onde as relações entre objetos desempenham um papel fundamental. Sua base na lógica de predicados torna-a adequada para modelar problemas complexos de forma declarativa e eficaz.

Para uma compreensão mais profunda da lógica de predicados e Prolog, recomendamos as seguintes fontes acadêmicas que possuem experimentação prática com problemas e projetos Prolog:

[Introdução à Linguagem Prolog - Silvio Lago](#)

[Lógica de Predicados - Renato Pimentel](#)

Essa combinação de teoria e prática pode ajudar a aprofundar o entendimento desses conceitos e suas aplicações.

Dica!

Se atente para focar o seu trabalho no **objetivo descrito**.

[Material complementar](#)

Sucesso é o acúmulo de pequenos esforços repetidos dia após dia.
Bom trabalho!