**PROLOG**

O prolog é uma linguagem declarativa, significando que em vez de o programa estipular a maneira de chegar a solução, passo a passo, limita-se a fornecer uma descrição do problema que se pretende computar. Prolog utiliza um arquivo onde são guardadas dados (condições básicas) e utiliza “argumentos” para definir novos dados/argumentos. Assim podemos compreender o porquê de o prolog ser uma definição recursiva, pois elas devem ser permitidas a fim de tornar a linguagem útil para muitas aplicações. Pelas regras recursivas devemos ter necessariamente ter, no mínimo, uma definição não recursiva. Se isto não acontecer, a definição é logicamente mal-formada e o programa ficaria em laço infinito. Além disso, é necessário tomar cuidado com a ordem na qual unificações são procurados para objetivos.

Supondo que o banco de dados do nosso sistema 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)

Nisso definimos que, animal recebe uma variável, guardando a como um animal, como no exemplo “animal(urso)”, urso é armazenado como um animal, portanto come recebe 2 variáveis, guardando a relação das duas, então em: “come(urso, peixe)”, urso e peixe são armazenados dentro de uma mesma relação e por fim em presa(X) <= come(Y,X) e animal(X), declaramos uma regra que diz que: presa é um X que se relaciona secundariamente com um Y em come (come(Y,X)) e é um animal em animal(X), ou seja presa é um animal que é comida por um Y qualquer.

Então se quisermos descrever o que seria um predador, a relação ficaria predador(X) <= come(X,Y) e animal(Y), primeiro declaramos que predador é um X que come um Y e Y é um animal, ou seja predador é todo aquele que come outro animal. Nesse caso se escrevêssemos em nossa busca predador(alga), a resposta seria falso, pois a alga não come nenhum animal, o mesmo seria para o peixinho, que apenas come alga, mas se tentássemos com o urso, a resposta seria verdadeira em 3 casos, no caso que ele come o guaxinim, a raposa e o veado.

Avançando mais um pouco nos deparamos com outro “problema”, quando efetuamos a pesquisa “?presa(X)” buscamos por tudo que se enquadra em “presa”, e esses seriam os resultados:

?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 linces.

Mas, onde está o “problema”? Fica perceptível que tanto o peixe quanto o veado são citados mais de uma vez, e por que isso ocorre? Bom é simples, eles simplesmente se enquadram como presa em dois casos, no exemplo dos peixes eles são presas de ursos e guaxinins, então durante a pesquisa, devido a ordem do registro o peixe aparece como presa do urso (Fato 1), depois o peixinho aparece como presa do peixe (Fato 2) e finalmente o peixe aparece de novo como presa do guaxinim (Fato 3) e o mesmo se aplicaria ao veado nos fatos 7 e 8.

*Então como visto no exemplo acima, na busca de* “?presa(X)”, encontramos repetidas vezes a mesma presa, e se fizermos o mesmo e buscarmos um predador? *Então a resposta seria:*

*?predador(X)*

*urso*

*peixe*

*guaxinim*

*urso*

*urso*

*raposa*

*urso*

*lince*

*E novamente alguns dados foram repetidos, e porque isso acontece? Bom, desta vez vamos usar o exemplo do urso que aparece quatro vezes, e é muito simples, do mesmo jeito que em* “?presa(X)” o peixe se repetia, em “*?predador(X)” o urso se repete simplesmente pois ele se encaixa como predador em quatro casos, casos esses: peixe, guaxinim, raposa e veado.*

*Agora vamos testar outros tipos de busca no sistema e ver se encontramos mais “problemas”.*

*?animal(lince)*

***true***

*?planta(guaxinim)*

***false***

*?come(urso, peixinho)*

***false***

*?come(raposa, coelho)*

***true***

*Nesses casos nada muito complicado acontece, simplesmente o sistema verifica se as variáveis indicadas se encaixam aos dados que a elas são inseridos e responde se é* ***verdadeiro*** *ou* ***falso****. Algumas perguntas podem ter mais de uma resposta, portanto se pesquisarmos “?come(guaxinim, X)”, encontramos X = peixe, mas quando perguntamos “?come(X, grama)”, encontramos as respostas X = coelho e X = veado, nada muito complexo, apenas foi identificado que o guaxinim come apenas peixe e grama é comida por coelho e veado. Agora para uma busca mais complexa como: “?come(urso, X) e come(X, coelho)”, o X seria raposa, pois a questão seria a busca de um X que é comido pelo urso e ao mesmo tempo come o coelho, e a raposa é o único animal que no nosso banco de dados se encaixa nisso.*

Partindo agora para um outro tópico, podemos afirmar o significado da busca em profundidade, que seria como pedir: “?come(X,Y),come(Y,Z)”, a fim de descobrir animais “Z” que estão em uma terceira posição ou maior na cadeia alimentar (maior sendo a profundidade na cadeia).

Em Prolog, identificadores de relacionamentos são denominados predicados e identificadores de objetos são denominados átomos. Tanto predicados quanto átomos devem iniciar com letra minúscula. A partir do momento que uma regra é declarada, a dedução por *modus ponens* é aplicada automaticamente, tendo o intérprete usará a informação futuramente para isso. Como por exemplo: Ao declarar que X come chocolate em seu tempo livre, por dedução, o interpretador Prolog responde que sim se for perguntado se “X come chocolate”, mesmo não havendo nenhum fato declarado sobre isso.

Contextualizando novamente o Prolog, ele é dito como uma linguagem de programação que se enquadra no paradigma de Programação em Lógica Matemática. É uma linguagem de uso geral que é especialmente associada com a inteligência artificial e linguística computacional. Seguindo com a lógica de predicados ,ela tem como base modelar objetos através de variáveis, relações entre objetos, funções, permitir composições de atributos, permitir derivação de relacionamentos de forma mais simples e consegue tratar de problemas de quantificação através de quantificadores. Sendo os elementos básicos da lógica de predicados símbolos que se referem a objetos, funções e relações.

Conhecendo agora as funcionalidades básicas de Prolog, conseguimos deduzir o quão simples ele é e a importância dele com relação a vários aspectos, tanto como a programação recursiva citada no começo, podemos dizer a facilidade em que resolvemos os problemas em Prolog usando ela, encontrando uma simplificação adequada para o problema em questão e descobrir como a solução obtida recursivamente pode ser usada para construir a solução final. Continuando nessa linha, descobrir os quesitos de quem “come” quem e quais seus “predadores” e “presas” acaba se tornando muito fácil, pelo fato da recursividade.

Agora relacionando Prolog com a lógica de predicados, e tendo consciência de que o principal objetivo do estudo da lógica de predicados é a generalização da lógica proposicional, com finalidade de obter um sistema lógico mais amplo, que seja capaz de expressar sentenças mais complexas, podemos comparar ambos no quesito de como elas são modeladas, ou seja, como as regras são feitas e deduzidas em conjunto pelos objetos, funções e relações citadas na lógica de predicados. Além disso, podendo até utilizar os quantificadores e as variáveis, porque indubitavelmente sem eles isso não se tornaria tão expressivo, em outras palavras, essa expressividade se dá aos conectivos lógicos citados anteriormente, possibilitando a formação de sentenças complexas pelo mais básico.

Por fim, concluímos que a linguagem Prolog está diretamente ligada a lógica de predicados, é uma linguagem um tanto quanto simples, porém pouco estudada atualmente, ela permite generalizar a lógica proposicional, e é a mais popular nessa vertente, mas mesmo assim ainda há uma alta demanda de profissionais encarregados dessa linguagem, sendo ela a mais popular em quesitos de lógica.

**Membros:**

Nicolas Afonso P de Mello - Aluno de Ciência da Computação - 221041048

Antonio Bruschi Gimenez - Aluno de Ciência da Computação - 221041007

Filippo Buschi Grassano - Aluno de Ciência da Computação - 222041001

**Referências:**

[Github](https://github.com/arerorero/prolog)

[Lógica de Predicados](https://wwwp.uniriotec.br/cristinabicharra/wp-content/uploads/sites/16/2018/11/Lo%CC%81gica-de-Predicados-Unificac%CC%A7a%CC%83o-Skolemizac%CC%A7a%CC%83o-e-ProLog-1.pdf)

[Prolog](http://walderson.com/site/wp-content/uploads/2014/08/ia-5-prolog.pdf)