

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

BRUNO ZANDONA AGUIAR
GABRIEL VARGAS BENTO DE SOUZA
NILSON DEON CORDEIRO FILHO
SAULO DE MOURA ZANDONA FREITAS

Percepções individuais sobre Prolog

Belo Horizonte - MG
2023

1.0. Bruno Zandona

Durante o nosso trabalho de Seminários de Linguagens de Programação, fomos sorteados a tratar da linguagem de Prolog, e essas foram as características dessa linguagem que mais me chamaram a atenção:

Programação Lógica

O modo como a linguagem funciona é fascinante, pois é focada no raciocínio dedutivo, que é uma forma de raciocínio que segue um conjunto de regras lógicas para inferir novas informações a partir de premissas existentes. E no raciocínio probabilístico, que por outro lado, lida com a incerteza, permitindo que o sistema avalie a probabilidade de uma determinada hipótese ser verdadeira com base nas informações disponíveis. E como esperado, os principais projetos baseados em Prolog aproveitam o máximo disso, principalmente a área de Inteligência Artificial.

A inteligência artificial é uma área muito ampla e diversificada, que engloba uma grande variedade de técnicas e tecnologias, e a linguagem Prolog pode ser utilizada em várias delas. Como na programação de sistemas especialistas, que podem fazer chatbots, identificar doenças, planejar projetos, organizar banco de dados e até fazer um eficiente jogo de xadrez!



Em conclusão, foi muito interessante receber essa linguagem como tema, já que comparada as outras, é uma linguagem de programação o qual nosso grupo e provavelmente muitos alunos ainda não conhecem. Mas que depois dos Seminários, podem passar a valorizar sua história e seus destaques na computação.

2.0. Gabriel Vargas

Durante a distribuição dos temas e dos grupos, eu me surpreendi por haver linguagens que nunca tinha visto anteriormente – incluindo o Prolog. Assim, inicialmente, não me chamou muita atenção por acreditar ser uma linguagem que seria antiga e ultrapassada, não sendo nem sequer utilizada. Todavia, a minha expectativa subiu, e muito, durante a pesquisa que realizei.

Nesse sentido, o meu primeiro contato com Prolog foi uma pesquisa no chatGPT, solicitando um código simples na linguagem. Assim, o meu susto inicial foi uma sintaxe completamente diferente de linguagens imperativas, como C e Java. Nessa busca, foi-me fornecido fatos e regras relacionados ao que seria uma árvore genealógica.

A partir disso, comecei a pesquisar melhor as aplicações do Prolog na atualidade, percebendo diferentes áreas, como inteligência artificial, processamento de linguagem natural, robótica, biomedicina. Assim, pude desconstruir o “preconceito” inicial sobre o Prolog, mesmo nunca tendo pesquisado ou entendido antes sobre.

Em seguida, meu passo foi compreender a ideia por trás da linguagem, que foi a questão que mais me fascinou: ser uma linguagem do “Paradigma Declarativo Lógico”. Eu sempre adorei matemática e raciocínio lógico durante minha vida escolar – que foi o aspecto que serviu como atrativo para cursar Ciência da Computação –; então, perceber que Prolog é uma linguagem de cunho matemático com os fatos e regras que geram uma base de conhecimento, a qual pode ser consultada, obtendo uma resposta a partir da busca me surpreendeu positivamente.

Sendo assim, ao pesquisar mais, chamou-me muito minha atenção na correlação entre Prolog e a linguagem, embora não de programação, SQL. Isso ocorre, pois pude perceber que a base de conhecimento gerada em um arquivo Prolog pode ser buscada com a chamada consulta, assemelhando-se à query no SQL. Além disso, pode-se criar predicados e regras dinâmicas em Prolog, de forma a ser possível inserir e deletar fatos da base de informações, tal como o “insert” e o “delete” em um banco de dados.

Além disso, outro aspecto que me surpreendeu são os tipos de dados primitivos. Nesse cenário, em linguagens imperativas, por exemplo, tem-se inteiro, real, caractere, valor lógico e string; todavia, em Prolog, esse não são exatamente os tipos existentes. Nessa linguagem, tem-se como um dos tipos o átomo, que, em linhas gerais, pode-se compreender como um objeto que assume como valor o próprio nome. Ademais, outra questão é o fato de que a palavra variável não tem mesmo significado em linguagens imperativas, visto que ela é tida mais como uma incógnita da matemática do que como a ideia construída de variável em linguagem imperativa. Tal aspecto me fascinou por, tal como comentei, ser um grande fã de exatas e de matemática.

Por fim, minha impressão geral da linguagem e do trabalho é a de que me surpreendi muito positivamente sobre o que encontrei nas minhas buscas. Pude, assim, perceber e compreender que existe um “mundo” de novas linguagens e que, só por pertencerem a um outro paradigma que não estou acostumado ou por não ser comentada diariamente, não significa que seja ruim ou menos importante. Percebi, ainda, que a aplicação da matemática e do raciocínio lógico está bem mais presente, na Computação, do que imaginei e gostei muito.

3.0. Nilson Deon

Prolog é uma linguagem de programação lógica que é muito diferente das linguagens imperativas que eu estava acostumado a usar. Em vez de dizer ao computador exatamente o que fazer passo a passo, em Prolog você define relações entre objetos e faz perguntas ao sistema para obter respostas.

Exemplo:

```
pai(joao, maria).
```

```
pai(maria, ana).
```

```
avo(X, Y) :- pai(X, Z), pai(Z, Y).
```

A primeira linha define que joao é pai de maria. A segunda linha define que maria é mãe de ana. A terceira linha define a relação avo/2 usando a regra: X é avô de Y se X é pai de Z e Z é pai de Y.

No começo foi um pouco desafiador mudar minha maneira de pensar sobre programação, mas depois que entendi os conceitos básicos, tudo começou a fazer sentido. Fiquei impressionado com a facilidade com que Prolog pode resolver problemas complexos com poucas linhas de código.

Além disso, estudar Prolog me deu uma nova perspectiva sobre como os programas podem ser escritos e me ajudou a entender melhor outros paradigmas de programação. Foi uma experiência enriquecedora e estou ansioso para continuar explorando as possibilidades desta linguagem fascinante.

Em resumo, minha experiência estudando Prolog foi extremamente valiosa. A linguagem me desafiou a pensar de maneiras novas e me permitiu resolver problemas complexos de maneira eficiente. Além disso, o conhecimento adquirido me ajudou a entender melhor outros paradigmas de programação.

4.0. Saulo de Moura

O prolog é uma ótima linguagem para inicialização de aprendizagem a linguagem. Os fatos, regras e consultas são ótimos para um auxílio na introdução de consultas em banco de dados ou consultas de uma inteligência artificial em seu banco de conhecimento.

Em banco de dados, pode-se usar como exemplo, o exemplo pronto "movies.pl" da pasta do swi-prolog, que simula uma partição de um banco de dados de filmes em que é possível fazer diversas consultas sobre filmes em geral.

```
42 /* DATABASE
43
44     movie(M, Y) <- movie M came out in year Y
45     director(M, D) <- movie M was directed by director D
46     actor(M, A, R) <- actor A played role R in movie M
47     actress(M, A, R) <- actress A played role R in movie M
48
49 */
50
51 :- disjoint
52     movie/2,
53     director/2,
54     actor/3,
55     actress/3.
56
57 movie(american_beauty, 1999).
58 director(american_beauty, sam_mendes).
59 actor(american_beauty, kevin_spacey, lester_burnham).
60 actress(american_beauty, annette_bening, carolyn_burnham).
61 actress(american_beauty, thora_birch, jane_burnham).
62 actor(american_beauty, wes_bentley, ricky_fitts).
63 actress(american_beauty, mena_suvari, angela_hayes).
64 actor(american_beauty, chris_cooper, col_frank_fitts_usmc).
65 actor(american_beauty, peter_gallagher, buddy_kane).
66 actress(american_beauty, allison_janney, barbara_fitts).
67 actor(american_beauty, scott_bakula, jim_olmeyer).
68 actor(american_beauty, sam_robards, jim_berkley).
69 actor(american_beauty, barry_del_sherman, brad_dupree).
70 actress(american_beauty, ara_celi, sale_house_woman_1).
71 actor(american_beauty, john_cho, sale_house_man_1).
72 actor(american_beauty, fort_atkinson, sale_house_man_2).
73 actress(american_beauty, sue_casey, sale_house_woman_2).
74 actor(american_beauty, kent_faulcon, sale_house_man_3).
75 actress(american_beauty, brenda_wehle, sale_house_woman_4).
76 actress(american_beauty, lisa_cloud, sale_house_woman_5).
77 actress(american_beauty, alison_faulk, spartanette_1).
78 actress(american_beauty, krista_goodsitt, spartanette_2).
79 actress(american_beauty, lily_houtkin, spartanette_3).
80 actress(american_beauty, carolina_lancaster, spartanette_4).
81 actress(american_beauty, romana_leah, spartanette_5).
82 actress(american_beauty, chekeshka_van_putten, spartanette_6).
83 actress(american_beauty, emily_zachary, spartanette_7).
```

Em prolog, também é possível implementar ordenações recursivas facilmente e com poucas linhas. Uma situação não positiva sobre a linguagem é a falta de estruturas de repetição, como o for, por exemplo, ou estruturas de condição, como o if, por exemplo. Mas isso acaba podendo ser resolvido implementando no código uma versão dessas estruturas a partir dos fatos e regras que podemos impor para a criação dessas estruturas.

```
1  % Regras
2  if(Condition,Then,_) :- Condition, !, Then.
3  if(_,_,Else) :- Else.
4
5  paridade(X)      :- if(X mod 2 == 0, write(par), write(ímpar)).
6  múltiplo(_, 0) :- write(não), !.
7  múltiplo(X, Y) :- if(X mod Y == 0, write(sim), write(não)).
```

```
12  % Regras
13  % Condição de parada
14  for(Start, End, _) :- Start >= End, !.
15
16  % Executa a ação, decrementa contador
17  for(Start, End, Do) :-
18      call(Do, Start),
19      Next is Start + 1,
20      for(Next, End, Do).
21
22  % Write contador
23  writeContador(X) :- write(X), nl.
```