

# Inteligência Artificial

## PROLOG

José Luis Seixas Junior

# Índice

- Introdução.
- Divisões:
  - Átomo.
  - Número.
  - Variáveis.
  - Termos compostos;
  - Listas;
- Bases:
  - Fatos;
  - Regras;



# Introdução

- Prolog é uma linguagem de programação:
  - Programação lógica;
  - Associada diretamente com a Inteligência Artificial;
- Representada por fatos e regras;
- Tem contexto de formação linguística e nasceu para processamento de linguagem natural:
  - Onde uma linguagem pudesse ser codificada à PROLOG;

# Introdução

- Aplicações:
  - Comprovação de teoremas;
  - Sistemas especialistas;
  - Atualmete:
    - Busca em bancos de dados;
    - Sistemas por controle vocal;
    - Templates;
    - Código intermediários.



# Átomo

- Termo geral, sem algum sentido aplicado;
- Uma palavra;

# Número

- Aceita:
  - Inteiros;
  - Flutuantes;
- String;
  - Sequência de caracteres rodeado por “aspas” é equivalente a uma lista de código de caracteres;



# Termo Composto

- Uma série de termos que realizam uma determinada expressão;
- Função, parâmetros, testes;
- Átomo pode ser dito um termo composto com parâmetros zero;

# Lista

- O átomo [] é uma lista;
- Um termo composto com nome . (ponto) e dois parâmetros recebe sua própria lista como parâmetro;
- Separadores:
  - .(1, .(2, .(3, [])))
  - [1 | [2 | [3 | ]]]
  - [1, 2, 3]



# Bases

- Prolog descreve relações por meio de cláusulas;
- Regras:
  - *Head* :- *Body*;
- *Head* será verdadeiro se *Body* for verdadeiro;
-

# Bases

- Uma regra onde o corpo é vazio, é um fato:
  - Ou seja, uma fato é um contexto de regra onde é especificado um valor verdadeiro;
  - Pois o corpo vazio não será testado;
- Assim:
  - `cat(tom) :- true.`
  - `cat(tom).`



# Fatos

- Dizer:
  - $X$  is  $3+2$ .
- Resulta em verdadeiro para o teste:
  - $?- X = 5$ .

# Regra

- Uma relação que deriva de uma relação pré-existente;
- Exemplo:
  - `relacao(X, Y) :-  
outraRelacao(Z, X),outraRelacao(Z, X).`
  - `outraRelacao(X, Y) :- dependencia(X, Y).`



# Recursão

- Prolog executa recursão em pilha;
- Cuidados:
  - Cláusulas podem não acontecer no tempo correto do teste gerando um resultado incoerente;
  - Recursão deve ser mantida a direita caso haja;
  - O símbolo ! previne qualquer recursão após ele, bloqueando recursões que poderiam causar problemas;
  - Analisar a recursão acaba não sendo simples.

# Variável Anônima

- Representada por `_`
  - Underline;
  - Underscore;
- Aceitar qualquer valor da regra e não inseri-lo em qualquer variável;



# Negação

- Símbolo  $\backslash$
- Nega o teste a seguir;
- $X \backslash = Y$ .
- Exemplo:
  - $\text{primos\_entre\_si}(X, Y) :-$   
   $\backslash + \text{divisorComum}(X, Y),$   
   $X \backslash = Y.$

# Funções

```
sentence(S)          --> statement(S0), sentence_r(S0, S).
sentence_r(S, S)     --> [].
sentence_r(S0, seq(S0, S)) --> statement(S1), sentence_r(S1, S).

statement(assign(Id,E)) --> id(Id), [=], expression(E), [;].

expression(E) --> term(T), expression_r(T, E).
expression_r(E, E) --> [].
expression_r(E0, E) --> [+], term(T), expression_r(plus(E0,T), E).
expression_r(E0, E) --> [-], term(T), expression_r(minus(E0, T), E).

term(T)          --> factor(F), term_r(F, T).
term_r(T, T)     --> [].
term_r(T0, T)    --> [*], factor(F), term_r(times(T0, F), T).

factor(id(ID))    --> id(ID).
factor(digit(D))  --> [D], { (number(D) ; var(D)), between(0, 9, D) }.

id(a) --> [a].
id(b) --> [b].
```



# Linguagens

- Ciao;
- GNU Prolog;
- Jekejeke Prolog;
- SIC Stus Prolog;
- SWI-Prolog (ProDT);
- Visual Prolog;
- XSB (ProDT);
- YAP-Prolog;

# Exercicio para Entregar

- Existem diversos tipos de gemas preciosas. Para simplificar, somente serão classificadas um conjunto reduzido de gemas, segundo as regras:
  - O berilo é caracterizado pelo facto de ser duro e também por ser um mineral;
  - O berilo é uma pedra preciosa, sendo que uma qualquer outra gema que contenha óxido de alumínio também é uma preciosa;
  - Uma esmeralda é uma gema preciosa com um tom verde;
  - Se uma gema for preciosa e tiver cor avermelhada, então é do tipo rubi;
  - Simplificando, podemos admitir que uma safira é uma gema que é preciosa cuja tonalidade não é verde nem avermelhada.
- Represente este conhecimento através de regras de produção em Prolog.
- Admita o seguinte cenário: tem um mineral que contém óxido de alumínio, cuja cor não é verde nem vermelha. Represente esta informação via factos.
- Entrega dia 28/05;