

# Programação Lógica (GSI010)

Paulo Henrique Ribeiro Gabriel

Faculdade de Computação / UFU

# Fatos, Regras e Consultas

Nesta unidade, vamos entender como especificar bases de conhecimento em Prolog e como consultá-las

# Base de Conhecimento 1 (BC1)

```
bruxo(harry).  
bruxo(ron).  
bruxo(hermione).  
  
trouxa(tio_vernon).  
trouxa(tia_petunia).  
  
animal(hedwig, coruja).
```

# Consultas que podemos fazer com a BC1

```
?- bruxo(harry).  
true.
```

```
?- animal(hedwig, coruja).  
true.
```

```
?- trouxa(harry).  
false.
```

```
?- bruxo(dumbledore).  
false.
```

```
?- feiticeiro(hermione).  
ERROR: toplevel: Undefined procedure: feiticeiro/1
```

# Mais consultas sobre BC1

```
?- trouxa(X).  
X = tio_vernon ;  
X = tia_petunia.
```

```
?- animal(X, Y).  
X = hedwig ;  
Y = coruja.
```

```
?- animal(X, X).  
false.
```

# Sintaxe Prolog

## Átomos

- Todos os termos que consistem em letras, números e sublinhado ( `_` ), começando com *letra minúscula* são **átomos**:  
`harry` , `tio_vernon` , `ritaSkeeter` , `nimbus2000` , ...
- Todos os termos escritos entre *aspas simples* também são **átomos**:  
`'Professor Dumbledore'` , `'(@ *+'` , ...
- Finalmente, certos símbolos especiais são átomos: `+` , `*` , `-` , `/` , ...

# Sintaxe Prolog

## Variáveis

- Todos os termos que consistem em letras, números e sublinhado (`_`), começando com *letra maiúscula* ou *sublinhado* são **variáveis**:

`X`, `Hermione`, `_ron`

- O sublinhado, isoladamente, é chamado de **variável anônima**

# Sintaxe Prolog

## Termos complexos

- Termos complexos, ou estruturas, são construídos da seguinte maneira: *functor(argumento 1, ..., argumento N)*.
- Functores têm que ser átomos
- Argumentos podem ser de qualquer tipo de termos em Prolog, *inclusive termos complexos!*

```
gosta(ron, hermione). , gosta(harry, X).
```

```
f(a, b, g(h(a)), c)
```



# Sintaxe Prolog

- **Fatos** são termos complexos seguidos de um *ponto final*  
`bruxo(harry).`  
`animal(hedwig, coruja).`
- **Consultas** também são termos complexos seguidos de ponto final; no entanto, são executados no *interpretador* da linguagem

`?- bruxo(hermione).`

## Base de conhecimento 2 (BC2)

```
comendo(dudley).  
infeliz(neville).  
  
feliz(tia_petunia) :- feliz(dudley).  
  
feliz(tio_vernon) :- feliz(dudley), infeliz(harry).  
  
feliz(dudley) :- comendo(dudley).  
feliz(dudley) :- infeliz(harry).
```

## Base de conhecimento 2 (BC2)

```
feliz(tia_petunia) :- feliz(dudley).
```

**se ... então:** Se `feliz(dudley)` é verdadeiro, então  
`feliz(tia_petunia)` é verdadeiro

## Base de conhecimento 2 (BC2)

```
feliz(tio_vernon) :- feliz(dudley), infeliz(harry).
```

**conjunção:** Se `feliz(dudley)` é verdadeiro e `infeliz(harry)` é verdadeiro, então `feliz(tio_vernon)` é verdadeiro

## Base de conhecimento 2 (BC2)

```
feliz(dudley) :- comendo(dudley).  
feliz(dudley) :- humilhando(dudley, harry).
```

**disjunção:** Se `comendo(dudley)` é verdadeiro *ou* `infeliz(harry)` é verdadeiro, então `feliz(dudley)` é verdadeiro

# Consultas que podemos fazer com a BC2

```
?- feliz(dudley).  
true .
```

```
?- feliz(tia_petunia).  
true .
```

```
?- feliz(tio_vernon).  
false.
```

```
?- feliz(X).  
X = tia_petunia ;  
X = dudley ;  
false.
```

# Sintaxe Prolog

## Regras

- Regras são do tipp *cabeça* :- *corpo*
- Assim como fatos e consultas, devem ser seguidas de ponto final
- A *cabeça* é um termo complexo
- O *corpo* é um termo complexo ou uma sequência de termos complexos separados por vírgula

```
feliz(tia_petunia) :- feliz(dudley).  
feliz(tio_vernon) :- feliz(dudley), infeliz(harry).
```

## Base de conhecimento 3 (BC3)

```
pai(albert, james).  
pai(james, harry).  
mae(ruth, james).  
mae(lili, harry).
```

```
bruxo(lili).  
bruxo(ruth).  
bruxo(albert).
```

```
bruxo(X) :-  
    pai(Y, X), bruxo(Y),  
    mae(Z, X), bruxo(Z).
```



## Base de conhecimento 3 (BC3)

- A BC3 pode ser traduzida para algo como:

Para todo  $x$ ,  $y$  e  $z$ , se  $\text{pai}(y, x)$  é verdadeiro e  $\text{bruxo}(y)$  é verdadeiro e  $\text{mae}(z, x)$  é verdadeiro e  $\text{bruxo}(z)$  é verdadeiro, então  $\text{bruxo}(x)$  é verdadeiro

- Em outras palavras, se o pai e a mãe de  $x$  são bruxos,  $x$  será um bruxo
- Note que nossa base possui **3 predicados**:  $\text{pai}/2$ ,  $\text{mae}/2$  e  $\text{bruxo}/1$

# Consultas que podemos fazer com a BC3

```
?- bruxo(james).  
true .  
?- bruxo(harry).  
true .
```

```
?- bruxo(X).  
X = lili ;  
X = ruth ;  
X = albert ;  
X = james ;  
X = harry ;  
false.
```

# Consultas sobre a BC3

```
?- bruxo(X), mae(Y, X), bruxo(Y).  
X = james ;  
Y = ruth ;  
X = harry ;  
Y = lili ;  
false.
```

# Em resumo: Termos em Prolog

- **átomos:** Começam com letra minúscula ou são delimitados por aspas simples

`harry` , `numbus2000` , `tia_petunia` , `'Professor Dumbledore'`

- **Números:** `42` , `314` , `-10`

- **Variáveis:** Começam com letra maiúscula ou sublinhado

`Harry` , `_harry`

- **Termos complexos:** Um átomo (functor) seguido por uma sequência de termos em Prolog separados por vírgula e delimitados por parênteses

`gosta(harry, X)` , `np(det(a), n(pocao))` .

# Agradecimentos

Material desenvolvido com base nos slides da Profa. Kristina Striegnitz