Aula 3: Recursão

- Teoria
  - Introduzir definições recursivas em Prolog
  - Quatro exemplos
  - Mostrar que pode haver discrepâncias entre o significado declarativo e o procedural em um programa Prolog

Definições Recursivas

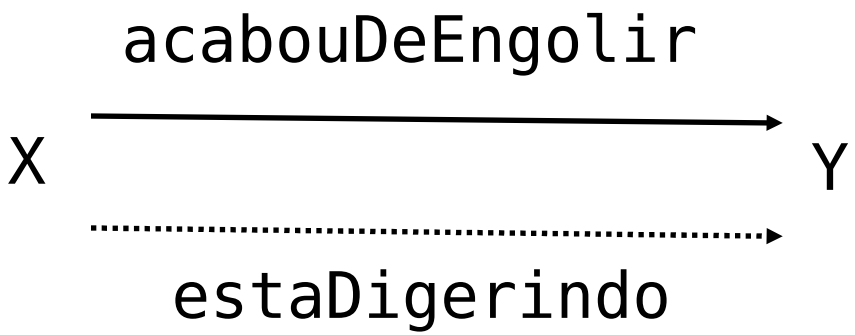
- Os predicados em Prolog podem ser definidos recursivamente
- Um predicado é definido recursivamente se uma ou mais regras em sua definição refere-se a ela mesma.

Exemplo 1: Alimentação

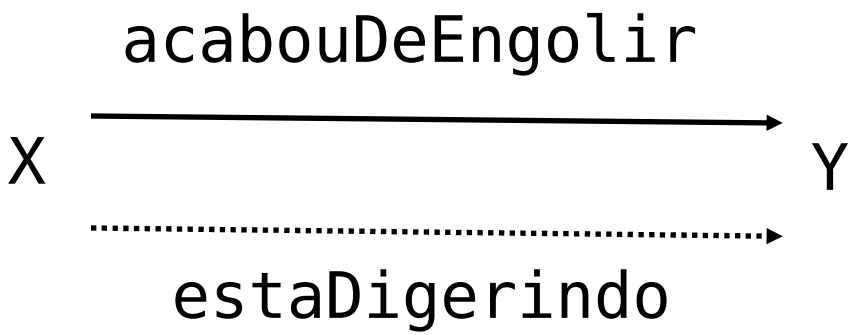
```
estaDigerindo(X,Y):- acabouDeEngolir(X,Y).
estaDigerindo(X,Y):- acabouDeEngolir(X,Z), estaDigerindo(Z,Y).

acabouDeEngolir(mosquito,sangue(joão)).
acabouDeEngolir(sapo,mosquito).
acabouDeEngolir(cegonha,sapo).
```

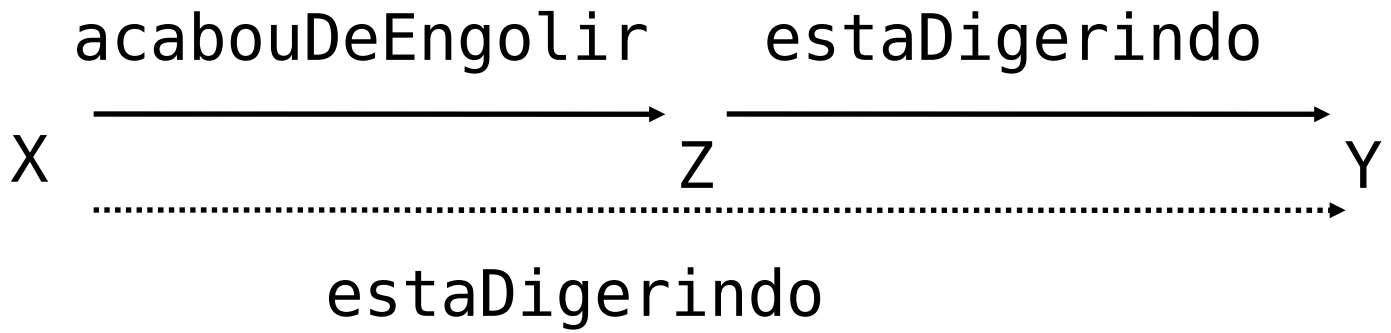
```
?-
```

Retrato da situação



Retrato da situação





Exemplo 1: Alimentação

```
estaDigerindo(X,Y):- acabouDeEngolir(X,Y).
estaDigerindo(X,Y):- acabouDeEngolir(X,Z), estaDigerindo(Z,Y).

acabouDeEngolir(mosquito,sangue(joão)).
acabouDeEngolir(sapo,mosquito).
acabouDeEngolir(cegonha,sapo).
```

```
?- estaDigerindo(cegonha,mosquito).
```

Uma outra definição recursiva

```
p:- p.
```

```
?-
```

Uma outra definição recursiva

```
p:- p.
```

```
?- p.
```

Uma outra definição recursiva

```
p:- p.
```

```
?- p.
ERROR: out of memory
```

Exemplo 2: Descendente

```
filho(brigite,caroline).
filho(caroline,donna).

descende(X,Y):- filho(X,Y).
descende(X,Y):- filho(X,Z), filho(Z,Y).
```

Exemplo 2: Descendente

```
filho(ana,brigite).
filho(brigite,caroline).
filho(caroline,donna).
filho(donna,emilia).


descende(X,Y):- filho(X,Y).
descende(X,Y):- filho(X,Z), filho(Z,Y).
```

Exemplo 2: Descendente

```
filho(ana,brigite).
filho(brigite,caroline).
filho(caroline,donna).
filho(donna,emilia).


descende(X,Y):- filho(X,Y).
descende(X,Y):- filho(X,Z), filho(Z,Y).
```

```
?- descende(ana,donna).
no
?-
```

Exemplo 2: Descendente

```
filho(ana,brigite).
filho(brigite,caroline).
filho(caroline,donna).
filho(donna,emilia).


descende(X,Y):- filho(X,Y).
descende(X,Y):- filho(X,Z), filho(Z,Y).
descende(X,Y):- filho(X,Z), filho(Z,U), filho(U,Y).
```

```
?-
```

Exemplo 2: Descendente

```
filho(ana,brigite).
filho(brigite,caroline).
filho(caroline,donna).
filho(donna,emilia).


descende(X,Y):- filho(X,Y).
descende(X,Y):- filho(X,Z), descende(Z,Y).
```

```
?-
```

Exemplo 2: Descendente

```
filho(ana,brigite).
filho(brigite,caroline).
filho(caroline,donna).
filho(donna,emilia).


descende(X,Y):- filho(X,Y).
descende(X,Y):- filho(X,Z), descende(Z,Y).
```

```
?- descende(ana,donna).
```

Árvore de busca

- Desenhe a árvore de busca para

  ?- descende(ana,donna).

Exemplo 3: Sucessor

- Suponha que usemos o seguinte modo para escrever numerais:
  - **0** é um numeral.
  - Se **X** é um numeral, então **suc(X)** também é um numeral.

Exemplo 3: Sucessor

```
numeral(0).
numeral(suc(X)):- numeral(X).
```

Exemplo 3: Sucessor

```
numeral(0).
numeral(suc(X)):- numeral(X).
```

```
?- numeral(suc(suc(suc(0)))).
true
?-
```

Exemplo 3: Sucessor

```
numeral(0).
numeral(suc(X)):- numeral(X).
```

```
?- numeral(X).
```

Exemplo 3: Sucessor

```
numeral(0).
numeral(suc(X)):- numeral(X).
```

```
?- numeral(X).
X=0;
X=suc(0);
X=suc(suc(0));
X=suc(suc(suc(0)));
X=suc(suc(suc(suc(0))))
```

Exemplo 4: Adição

```
?- soma(suc(suc(0)), suc(suc(suc(0))), Resultado).
Resultado=suc(suc(suc(suc(suc(0)))))
true
```

Exemplo 4: Adição

```
soma(0,X,X).                    %%% cláusula base
```

```
?- soma(suc(suc(0)), suc(suc(suc(0))), Resultado).
Resultado=suc(suc(suc(suc(suc(0)))))
true
```

Exemplo 4: Adição

```
soma(0,X,X).                    %%% cláusula base

soma(suc(X),Y,suc(Z)):-         %%% cláusula recursiva
    soma(X,Y,Z).
```

```
?- soma(suc(suc(0)), suc(suc(suc(0))), Resultado).
Resultado=suc(suc(suc(suc(suc(0)))))
true
```

Árvore de busca

- Desenhe a árvore de busca

Prolog e Lógica

- Prolog foi a primeira tentativa razoável de criar uma linguagem de programação baseada na lógica
  - O programador fornece uma especificação declarativa do problema, usando a linguagem da lógica
  - O programador não deveria ter que dizer ao computador o que fazer
  - Para obter informação, o programador simplesmente faz uma consulta.

Prolog e Lógica

- Prolog dá passos importantes nesta direção, mas, no entanto, Prolog não é uma linguagem de programação lógica completa!
- Prolog possui um modo específico de responder consultas:
  - Pesquisar a base de conhecimento de cima para baixo
  - Processar cláusulas da esquerda para a direita
  - Retroceder para se recuperar de escolhas ruins

descende1.pl

```
filho(ana,brigite).
filho(brigite,caroline).
filho(caroline,donna).
filho(donna,emilia).

descende(X,Y):- filho(X,Y).
descende(X,Y):- filho(X,Z), descende(Z,Y).
```

```
?- descende(A,B).
A=ana
B=brigite
```

descende2.pl

```
filho(ana,brigite).
filho(brigite,caroline).
filho(caroline,donna).
filho(donna,emilia).

descende(X,Y):- filho(X,Z), descende(Z,Y).
descende(X,Y):- filho(X,Y).
```

```
?- descend(A,B).
A=ana
B=emilia
```

descende3.pl

```
filho(ana,brigite).
filho(brigite,caroline).
filho(caroline,donna).
filho(donna,emilia).

descende(X,Y):-  descende(Z,Y), filho(X,Z).
descende(X,Y):- filho(X,Y).
```

```
?- descende(A,B).
ERROR: Out of local stack
```

descende4.pl

```
filho(ana,brigite).
filho(brigite,caroline).
filho(caroline,donna).
filho(donna,emilia).

descende(X,Y):- filho(X,Y).
descende(X,Y):-  descende(Z,Y), filho(X,Z).
```

```
?- descend(A,B).
```

Resumo desta aula

- Nesta aula foram introduzidos os predicados recursivos
- Também foi visto as diferenças entre o significado declarativo e procedural dos programas Prolog
- Identificou-se algumas das deficiências do Prolog como uma linguagem de programação em lógica

Próxima aula

- Introdução às **listas** em Prolog
  - Estutura de dados recursiva muito importante em programação Prolog
  - Definição do predicado membro/2, uma ferramente fundamental do Prolog para se trabalhar com listas
  - Discutir a ideia de recursão sobre listas