PROLOG[1]

Capítulo 1: Introdução a Prolog

- Definindo Relações por meio de fatos
- Definindo Relações por meio de regras
- Regras recursivas
- Como Prolog responde a consultas
- Significado declarativo e procedural de programas

[1]Baseado no livro: Bratko, I. *PROLOG: Programming for Artificial Intelligence*, 3rd edition, Addison-Wesley, 2001.

Definindo Relações por meio de fatos

- *Instância* (*relacionamento*) $\Rightarrow$ **cláusula**.
- Exemplo de cláusula: `parent(pam,bob).`

- Exemplos de consultas:

```
?- parent(bob,pat).
yes

?- parent(liz,pat).
no

?- parent(tom,ben).
no

?- parent(X,liz).
X = tom

?- parent(bob,X).
X = ann;
X = pat;
no

?- parent(X,Y).
X = pam
Y = bob;
X = tom
Y = bob;

⋮
```

- Algumas consultas mais sofisticadas:

  ```
  ?- parent(Y,jim),parent(X,Y).
  ```

  ```
  ?- parent(X,Y),parent(Y,jim).
  ```

  ```
  ?- parent(tom,X),parent(X,Y).
  ```

  Ann e Pat são filhos de uma mesma pessoa?

- Cada cláusula de uma consulta é uma **meta**.
- Meta satisfeita: **sucesso**
- Meta não satisfeita: **falha**.

Definindo Relações por meio de regras

- Exemplo de **Regra**:

  Para todo X e Y,
  Y é filho(a) de X se
  X é pai ou mãe de Y.

- Em Prolog:

```
offspring(Y,X) :- parent(X,Y).
```

- Outro exemplo:

```
mother(X,Y) :- parent(X,Y), female(X).
```

- Alguns novos fatos:

```
female(pam).
male(tom).
male(bob).
female(liz).
female(pat).
female(ann).
male(jim).
```

- Algumas regras representadas por grafos:

- Consultas:

```
?- sister(ann,pat)
?- sister(X,pat)
```

Alguns pontos importantes:

- Programas podem ser estendidos adicionando-se cláusulas.
- Existem 3 tipos de cláusulas: *fatos*, *regras* e *consultas*.
- Uma cláusula tem uma *cabeça* e um *corpo*.
- Um fato tem uma cabeça e o corpo é vazio.
- Um regra tem uma cabeça e corpo não é vazio.
- Uma consulta tem apenas corpo.
- Variáveis são quantificadas universalmente.

  Exemplo:

```
hasachild(X) :- parent(X,Y)
```

Regras recursivas

- Exemplo: a relação *predecessor*:

```
predecessor(X,Z) :- parent(X,Z).
predecessor(X,Z) :-
  parent(X,Y),
  predecessor(Y,Z).
```

Quais as respostas para a consulta a seguir?

```
?- predecessor(pam,X).
```

- **Comentário**: após % ou entre /* e */.

Como Prolog responde a consultas

- Para processar uma consulta: mostrar que ela é *consequência lógica* dos fatos e regras.
- As cláusulas são usadas na *ordem em que são escritas*.
- As metas dos corpos (da consulta ou de uma regra) são processadas *da esquerda para a direita*.
- Se uma meta não é satisfeita, ocorre *backtracking*.

Um exemplo:

```
?- predecessor(tom,pat)
```

1. Tenta a regra pr1:

2. Ao falhar, tenta regra pr2:

3. Completando:

Significado declarativo e procedural de programas

- Significado *declarativo*: programa visto como um conjunto de fórmulas lógicas (definindo relações).
- Significado *procedural*: programa visto como uma sequência de procedimentos, avaliados de determinada forma.
- O significado declarativo determina *quais* são as respostas; o procedural determina *como* as respostas são obtidas.
- A programação pode ser feita em 2 fases:
  1. Especificação (programação) declarativa.
  2. Rearranjo (e talvez modificações) tendo em vista aspectos de controle e eficiência.