Capítulo 2: Sintaxe e Significado de Programas Prolog

- Objetos de dados
- Casamento
- Significado declarativo de programas Prolog
- Significado procedural
- Exemplo: macaco e bananas
- Ordem das cláusulas e metas
- A relação entre Prolog e lógica

Objetos de dados

Átomos e números

- Átomos:
  - Começando com letra minúscula: `ana`, `x12`, `x_12`.
  - Só caracteres especiais: `<--->`, `==>`, ...
  - Entre apóstrofos: `'Ana'`, `'Deus'`, `'Brasil'`.
- Números:
  - Inteiros: `0`, `123`, `-33`.
  - Reais: `3.14`, `-0.0021`, `100.0`.

Variáveis

- Começam com letra maiúscula ou "`_`":

  `Res`, `_x12`, `X_12`.

- Variável **anônima**: `_`. Exemplos:

```
hasachild(X) :- parent(X,_).
somebody_has_child :- parent(_,_).
?- parent(X,_).
```

- Escopo léxico de uma variável: uma cláusula.

Estruturas

- **Estruturas** (objetos estruturados): têm um **funtor** e argumentos (que podem ser estruturas).
- Exemplo:

- Sintaticamente, todo objeto de dado é um **termo**.
- Todo objeto estruturado pode ser desenhado como uma *árvore*.

- Outro exemplo:

- Mais um exemplo:

Casamento

Para verificar se 2 termos S e T casam:

1. Se S e T são constantes, eles casam apenas se são o mesmo objeto.
2. Se S é variável, eles casam e S é instanciado para T. Caso contrário, se T é variável, eles casam e T é instanciado para S.
3. Se S e T são estruturas, eles casam somente se:
   (a) S e T têm o mesmo funtor principal, e
   (b) seus argumentos correspondentes casam.

⇒ É determinada a instanciação **mais geral**.

Exemplos:

```
?- date(D,M,2001)=date(D1,may,Y1).

?- triangle(point(1,1),A,point(2,3))
     =triangle(X,point(4,Y),point(2,Z)).
```

- Um exemplo:

```
vertical(seg(point(X,Y),point(X,Y1))).
horizontal(seg(point(X,Y),point(X1,Y))).
```

Consultas:

```
?- vertical(seg(point(1,1),point(X,3))).
X = 1
?- horizontal(seg(point(2,3),P)).
P = point(X,3)
?- vertical(S), horizontal(S).
```

Significado declarativo de programas Prolog

- **Instância** de uma cláusula: obtida substituindo-se suas variáveis por termos.

  Instâncias de `hasachild(X) :- parent(X,Y).`:

  ```
  hasachild(A) :- parent(A,B).
  ```
  (variante)

  ```
  hasachild(ann) :- parent(ann,B).
  hasachild(pat) :- parent(pat,small(carol)).
  ```

- Uma meta G é *verdadeira* se, e somente se:

  (1) há uma cláusula C tal que

  (2) há uma instância I de C tal que

  (a) a cabeça de I é idêntica a G, e

  (b) as metas do corpo de I são verdadeiras.

Significado procedural

Exemplo: macaco e bananas

Estado inicial do “mundo do macaco”:

- Macaco está na porta.
- Macaco está no chão.
- Caixa está na janela.
- Macaco está sem banana.

Estado meta: `state(_,_,_,has)`

Movimentos:

- Formalização: `move(State1,Move,State2)`
- **4 operadores**:

  (1) pegue banana (*grasp*).

  ```
  move(state(middle,onbox,middle,hasnot),
       grasp,
       state(middle,onbox,middle,has))
  ```

  (2) suba na caixa (*climb*).

  ```
  move(state(P,onfloor,P,H),
       climb,
       state(P,onbox,P,H))
  ```

  (3) empurre a caixa (*push*).

  ```
  move(state(P1,onfloor,P1,H),
       push(P1,P2),
       state(P2,onfloor,P2,H))
  ```

  (4) ande (*walk*).

  ```
  move(state(P1,onfloor,B,H),
       walk(P1,P2),
       state(P2,onfloor,B,H))
  ```

A consulta:

```
?- canget(state(atdoor,onfloor,atwindow,hasnot)).
```

O programa:

Ordem das cláusulas e metas

Se a cláusula

```
move(state(P1,onfloor,B,H),
     walk(P1,P2),
     state(P2,onfloor,B,H))
```

for a primeira,

```
?- canget(state(atdoor,onfloor,atwindow,hasnot)).
```

causa *loop*:

(1) `canget(state(atdoor,onfloor,atwindow,hasnot)).`

(2) `move(state(atdoor,onfloor,atwindow,hasnot),M',S2'),`
`canget(S2')`

(3) `canget(state(P2',onfloor,atwindow,hasnot))`

(4) `move(state(P2',onfloor,atwindow,hasnot),M'',S2''),`
`canget(S2'')`

(5) `canget(state(P2'',onfloor,atwindow,hasnot))`

(6) `move(state(P2'',onfloor,atwindow,hasnot),M''',S2'''),`
`canget(S2''')`

(7) `canget(state(P2''',onfloor,atwindow,hasnot))`