

## Sintaxe e Semântica de Prolog

Prof. Dr. Carlos Lopes

## Sintaxe

- Termos
  - Variáveis: X, Y, C, ABC, Input
  - Constantes: prolog, a, 123, 3.14, 'Uberlandia'
  - Estruturas (termos compostos): data(10,10,2003).
  - Caracteres especiais: +, -, \*, /, <, >, =, :, ., &, \_ ~
- Símbolos especiais: :-, ;,
- Comentários
  - linha: % isto é um comentário
  - texto: /\* este também é um comentário \*/

## Observações sobre Variáveis

- Quando uma variável aparece uma única vez numa cláusula não é necessário darmos um nome a ela:
  - Exemplo: *temFilho(X) :- pai(X,Y)* pode ser reescrito como: *temFilho(X) :- pai(X,\_)*. O caractere `_` representa uma variável anônima.
- O escopo léxico de uma variável é a cláusula onde ela se encontra. O que isto significa?

## Estruturas

- Representa uma composição de componentes simples ou estruturados.
- A Composição é realizada por um *functor*, aplicado a seus *termos*.
- Exemplo de estrutura: data(19,10,2003)
  - Identificação do functor: data
  - Identificação dos termos: são três – 19, 10 e 2003.
- Os componentes podem ser constantes, variáveis ou até mesmo estruturas.

## Unificação

- A operação mais importante relacionada a termos é a *unificação*.
- A unificação é o processo que, dados dois termos como dados de entrada verifica se eles se "casam".
- Se os dois termos se casam dizemos que o processo *fracassa*.
- Se os dois termos se casam dizemos que o processo *obtem sucesso*. O processo instancia as variáveis em ambos os termos para valores de modo a fazer com que os termos fiquem idênticos.

## Unificação (cont.)

- Exemplo: considere a unificação de duas datas. Esta operação é comunicada ao Prolog pela seguinte consulta usando o operador `'=`':  
*?- data(D,M,2003)=data(D1,setembro,Ano).*  
A resposta é:  
*D = D1*  
*M = setembro*  
*Ano = 2003*

## Unificação (cont.)

- Entretanto existem outras instanciações possíveis. Uma delas é:
  - D = 1
  - D1 = 1
  - M = setembro
  - Ano = 2003
- Esta última instanciação é menos geral que a primeira.
- A unificação em Prolog resulta sempre na instanciação mais geral.

## Unificação (cont.)

Dois termos unificam se

- Forem idênticos
  - Constantes numéricas ou alfanuméricas
- Envolverem variáveis que, ao serem instanciadas, passem a ser idênticas, i.e., a ter o mesmo valor
- `data(D,maio,Ano) = data(25,maio,1983)`
  - D=25; Ano=1983
- `data(D,maio,2002) = data(25,maio,1983)`
  - Não unifica: constantes numéricas *distintas*
- `data(25,maio,1983) = data(25,maio,1983)`
  - Unifica: todos os valores constantes e *iguais*

## Unificação (cont.)

Termo 1	Termo 2	Resultado da Unificação
henrique	henrique	unificam
eduardo	henrique	não unificam
X	par(a,b)	unificam (X = par(a,b))
2.35	Var	unificam (Var=2.35)
data(2,outubro,1999)	date(2,outubro,1999)	não unificam
pai_de(X,eduardo)	pai_de(henrique,Y)	unificam (X=henrique; Y=eduardo)
pai_de(X,maria)	pai_de(eduardo,X)	não unificam
pai_de(X,maria)	pai_de(X,filhos(eduardo,maria))	não unificam

## Unificação (cont.)

Se S e T forem constantes, S e T unificam somente se forem o *mesmo objeto*

- Se S for uma variável e T for qualquer termo, S é instanciado com T
  - Variáveis unificam com qualquer coisa
  - Variáveis unificam com outras variáveis também
- Se S e T forem *estruturas*, elas unificam somente se
  - S e T tiverem o mesmo funtor
  - Cada um de seus termos unificarem
    - Ordem dos termos na estrutura deve ser preservada

## Significado Declarativo e Procedimental de Programas

- Exemplo: `p :- q,r.`
  - Significado declarativo:
    - "P é verdade se q e r são verdades"
    - Alternativa: q e r implicam p.
  - Significado procedimental:
    - Para resolver p resolva q e depois resolva r
    - Para satisfazer p primeiro satisfaça q e depois r.

## Significado Declarativo de Programas Prolog

- Determina se uma dada meta é verdadeira e para quais valores de variáveis ela é verdadeira.
- Para entender o significado declarativo é necessário entender dois conceitos:
  - Instância de uma cláusula C
  - Variante de uma cláusula C

### Significado Declarativo de Programas Prolog

- Uma instância de uma cláusula C é a cláusula C com cada uma de suas variáveis substituídas por algum termo.
- Uma variante de uma cláusula C é uma instância da cláusula C onde cada variável é substituída por outra variável.
- Seja a seguinte cláusula:  
temFilho(X) :- pai(X,Y).  
Variantes desta cláusula?  
Instâncias desta cláusula?

### Significado Declarativo de Programas Prolog

- Dado um programa e uma meta G, o significado declarativo especifica: *uma meta G é verdadeira (ou satisfatível ou uma consequência lógica do programa) se e somente se :*
  1. Existe uma cláusula C no programa tal que
  2. Existe uma instância I de C tal que
    - a) A cabeça de I é idêntica a G, e
    - b) Todas as metas no corpo de I são verdadeiras

### Significado Procedimental de Programas Prolog

- O significado procedimental especifica como Prolog responde consultas.
- Responder a uma consulta significa tentar satisfazer uma lista de metas.
- A lista de metas pode ser satisfeita se as variáveis que aparecem nas metas podem ser instanciadas de tal forma que as metas tornem-se *consequências lógicas* do programa.
- Desta forma o significado procedimental de Prolog é um procedimento para *executar* uma lista de metas com respeito a um dado programa.

### Significado Procedimental de Programas Prolog (cont.)

- 'executar metas' significa tentar satisfazê-las.
- O processo de execução tem como:
  - Dados de entrada: um programa e uma lista de metas;
  - Dados de saída: um indicador de sucesso/fracasso e uma instancição de variáveis.

### Significado Procedimental de Programas Prolog

- Significado dos dados de saída:
  - O indicador de sucesso/fracasso é 'yes' se as metas são satisfeitas e 'no' caso contrário.
  - Uma instanciação de variáveis é somente produzida no caso de uma terminação com sucesso.
- Exemplo para ilustrar o significado procedimental de Prolog.

### Exemplo

```
grande(urso).      % clausula 1
grande(elefante).  % clausula 2
pequeno(gato).     % clausula 3
marrom(urso).      % clausula 4
preto(gato).       % clausula 5
cinza(elefante).   % clausula 6
escuro(Z) :- preto(Z). % clausula 7
escuro(Z) :- marrom(Z). % clausula 8
Consulta:
?- escuro(X),grande(X).
```

### Exemplo(cont.) - depurando

1. Lista de metas iniciais: **escuro(X),grande(X)**.
2. Percorra o programa (base de fatos e regras) de cima para baixo procurando por uma cláusula cuja cabeça unifique-se com a primeira meta **escuro(X)**. A cláusula encontrada é **escuro(Z) :- preto(Z)**. Substituindo a primeira meta pelo corpo da cláusula encontrada obtém-se um nova lista de metas: **preto(X),grande(X)**.
3. Percorra o programa para encontrar uma cláusula que unifique com **preto(X)**. A cláusula encontrada é **preto(gato)** que não tem corpo. Desta forma a nova lista de metas torna-se **grande(gato)**.

### Exemplo(cont.) - depurando

4. Percorrer o programa para a meta **grande(gato)**. Nenhuma cláusula é encontrada. Portanto retrocede ("backtrack") ao passo 3 e desfaz a instanciación **X=gato**. Assim a lista de metas torna-se **preto(X), grande(X)**. Continua a percorrer o programa abaixo da cláusula 5. Nenhuma cláusula é encontrada. Portanto retrocede ("backtrack") ao passo 2 e continua percorrendo abaixo da cláusula 7. A cláusula 8 é encontrada: **escuro(Z) :- marrom(Z)**. Substitui a primeira meta na lista por **marrom(X)** gerando **marrom(X), grande(X)**.

### Exemplo(cont.) - depurando

5. Percorre o programa para unificar **marrom(X)**. Encontra **marrom(urso)**. Esta cláusula não tem corpo de tal forma que a lista de metas torna-se **grande(urso)**.
6. Percorre o programa e encontra a cláusula **grande(urso)**. Esta cláusula não tem corpo e a lista de metas torna-se vazia. Isto indica terminação com sucesso e a instanciación da variável correspondente torna-se **X=urso**.