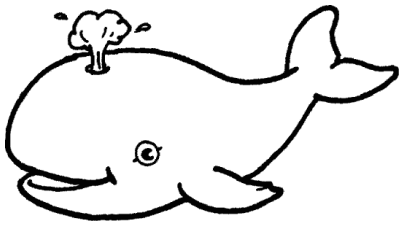
Paradigma Declarativo (Lógico)

Profa. Maria Adelina Raupp Sganzerla

Paradigmas de Linguagem de Programação - Ulbra – Gravataí 2016/2

Modelo de Programação em Lógica

- Base
  - Axiomas lógicos (proposições) que representam o conhecimento e regras de dedução de novos conhecimentos



  - Exemplo:

```
Uma Baleia é um mamífero.
Um mamífero tem sangue quente.
Então, uma Baleia tem sangue quente.
```

Modelo de Programação em Lógica

- Programa
  - Conjunto de FATOS (conhecidos) e REGRAS (de dedução) que formam a base para formular consultas.

Principais Aplicações

- Sistemas Baseados em Conhecimento
  - Sistemas que aplicam mecanismos automatizados de raciocínio para a representação e inferência de conhecimento.
- Banco de Dados "Inteligentes"
  - Sistemas que empregam "agentes" de busca de dados com base em critérios.
- Sistemas Especialistas
  - Sistemas que emulam a especialização humana em algum domínio específico.
- Processamento da Linguagem Natural
  - Usada para desenvolvimento de ferramentas para a comunicação homem-máquina em geral e para a construção de interfaces.

Origens: Lógica Matemática

- O uso da lógica na representação dos processos de raciocínio originou-se nos estudos de:
  - Boole (1815 – 1864)
  - De Morgan (1806 – 1871)
  - Teoria posteriormente denominada de Álgebra de Boole

- O cálculo de predicados é o subconjunto da lógica matemática que formaliza a estrutura lógica e define o significado dos conectivos e, ou, não, se … então entre outros.

Lógica Simbólica

Representação	Significado
$p \vee q$	p ou q
$p \wedge q$	p e q
$p \Rightarrow q$	p implica q
$p \Leftrightarrow q$	p equivale a q
$\neg p$	não p

O Modelo de Programação em Lógica

- Base do Modelo: Lógica Simbólica
  - Axiomas lógicos (proposições) que representam o conhecimento e regras de dedução (inferência) de novos conhecimentos.

O Modelo de Programação em Lógica – Exemplos

- Exemplo de Preposição:

  p: um coelho é um mamífero

  q: um mamífero tem sangue quente

- Exemplo de Dedução:

SE um coelho é um mamífero E um mamífero tem sangue quente

ENTÃO, um coelho tem sangue quente.

Se p $\wedge$ q
Então r

Base do Modelo: Lógica Simbólica

- Uma relação é estabelecida entre objetos e resulta nos valores Verdadeiro e Falso

- Programação em lógica consiste em implementar relações e formular consultas como:

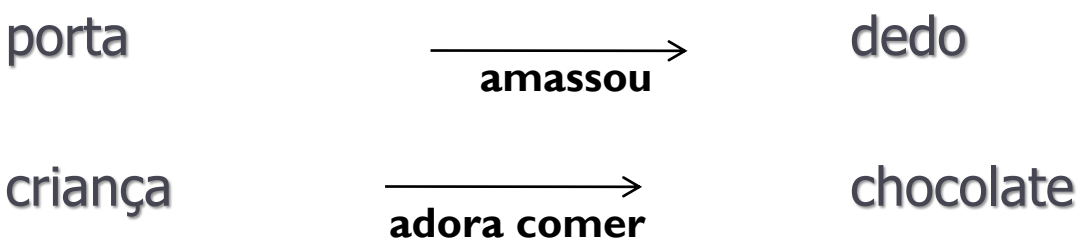
  - Dados a e b, determinar se R(a,b) é verdadeira.
  - Dado a, encontrar todo y tal que R(a,y) é verdadeira.
  - Dado b, encontrar todo x tal que R(x,b) é verdadeira.
  - Encontrar todo x e todo y tal que R(x,y) é verdadeira.

Modelagem de Programas em Lógica

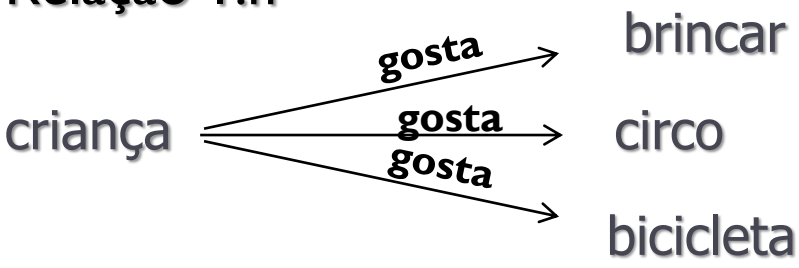
- Especifica o conhecimento a ser usado na solução do problema: FATOS e REGRAS;
- Procedimentos de resolução, baseados em um conjunto de cláusulas e um objetivo: CONSULTA ao programa;
- Projeto de soluções: abordagem de redes semânticas.

Redes Semânticas

- Usadas para representar relacionamentos
- Rede Semântica Simples: $1 \rightarrow 1$

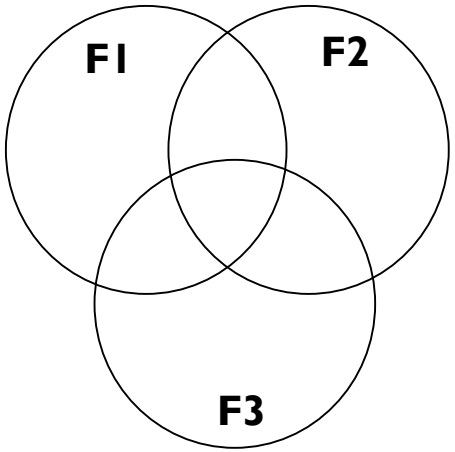


- Relação 1:n



Redes Semânticas

- A abordagem de grupo
- Fatos:
  - F1: Pedro, Maria e Paulo gostam de ler
  - F2: Pedro, Paulo e Vera gostam de cinema
  - F3: Paulo, Maria e Vera gostam de boliche
- Questões:
  - X gosta de ler?
  - Quem gosta de boliche?
  - Quem gosta de boliche e cinema?



Programação em Lógica e Prolog

- Criada em 1972 em Marseille na França
- Programas
  - Cláusulas:
    - Fatos e regras (parâmetro em letras minúsculas)
    - Consultas (parâmetros em letras maiúsculas)

Programas: Fatos e Regras

- Fatos agrupados se constituem na base de dados

  - **Fato 1: Brasília e Porto Alegre são cidades**

```
cidade(porto_alegre).
cidade(brasilia).
```

  - **Fato 2: Brasil é um país.**

```
pais(brasil).
```

  - **Fato 3: Rio Grande do Sul é um Estado (RS).**

```
estado(rs).
```

  - **Fato 4: Brasília é a capital do Brasil.**

```
capital(brasil, brasilia).
```

  - **Fato 5: Porto Alegre é a capital do RS.**

```
capital(rs, porto_alegre).
```

Programas: Fatos e Regras

- Uma regra tem 1 termo (função e argumentos) do lado esquerdo que corresponde ao que se pretende provar (conclusão), seguido do ":-".

- À direita do ":-" podem aparecer outros termos precedidos pelos operadores:
  - `, (E)`
  - `; (OU)`
  - `not (NÃO)`
  - `\+ (Negação de uma Regra)`

- Tal como nos fatos, podemos ter regras com o mesmo nome e mesmo número de argumentos, ou até com o mesmo nome e número de argumentos diferentes.

Programas: Fatos e Regras

- Regras estabelecem relação mais complexas entre objetos

```
capital_pais(X, Y) :-
            pais(X),
            cidade(Y),
            capital(X, Y).

capital_estado(X, Y) :-
      estado(X),
      cidade(Y),
      capital(X, Y).
```

"Y é capital do país X, se X é um País e Y é uma cidade e Y é capital de X".

Operadores Relacionais - Prolog

- Símbolo/Significado

```
>          Maior (X>Y)
<          Menor (X<Y)
>=         Maior Igual (X>=Y)
<=         Menor Igual (X<=Y)
=          Igual (X=Y)
\=         Diferente (X\=Y)
is         Atribuição (X is 5)
```

Operadores Aritméticos - Prolog

- Símbolo/Significado

```
+          Adição X + Y
-          Subtração X - Y
*          Multiplicação X * Y
/          Divisão X / Y
//         Divisão interia X // Y
mod        Resto da divisão inteira X mod Y
^          Potência X ^ Y
```

Escrita - Prolog

- A escrita em Prolog é realizada com o comando write
  - `write('Ola') -> Escreve Ola na tela`
  - `write (A) -> Escreve o conteúdo de A`

- A mudança de linha é feita com o nl (nova linha)

Consultas: Reversibilidade

- Permite a obtenção de respostas alternativas
- Programas com mais de uma finalidade

Exemplo:

```
gosta(maria, peixe).
gosta(maria, vinho).
gosta(pedro, vinho).
gosta(pedro, maria).
```

Perguntas:

- Quem gosta de peixe?

```
?-
```

- Quem gosta de vinho?

```
?-
```

- Pedro e Maria se gostam?

```
?-
```

- Existe algo que Pedro e Maria gostem?

```
?-
```

Mecanismo de Retrocesso (*Backtracking*)

- Quando é efetuada uma consulta e o objetivo não é satisfeito, usa-se o mecanismo *Backtracking* do Prolog para tentar resatisfazer um objetivo, encontrando uma solução alternativa;

- Quando ocorrer conjunções em uma consulta, ela satisfaz na ordem em que foram escritos (da esquerda para direita).

Mecanismo de Retrocesso (*Backtracking*)

Base de Dados

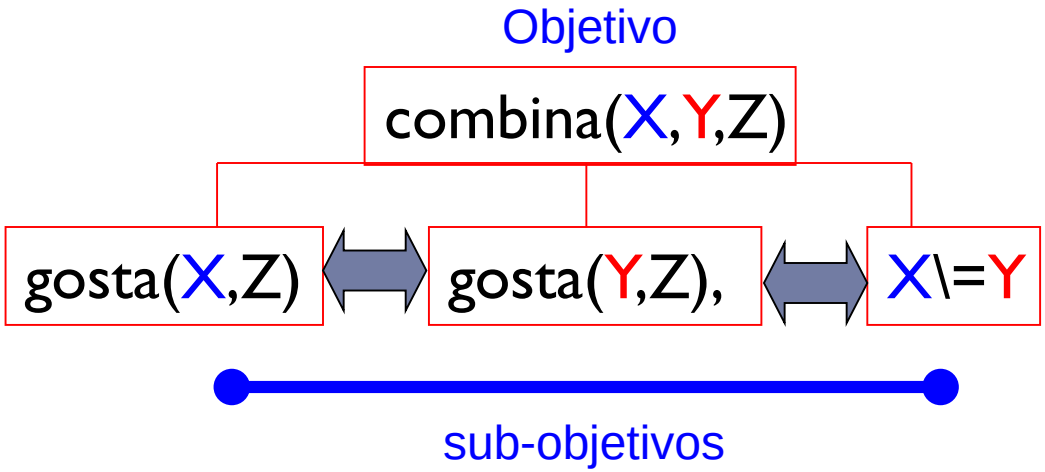
```
gosta(pedro,leitura).
gosta(maria,leitura).
gosta(paulo,leitura).
gosta(pedro,cinema).
gosta(paulo,cinema).
gosta(vera,cinema).
gosta(paulo,boliche).
gosta(maria,boliche).
gosta(vera,boliche).

? - gosta(X,Y).
```

Obs.: Use o ; para visualizar todos os resultados.

Mecanismo de Retrocesso (*Backtracking*)

- /* base de dados*/
- `gosta(pedro,leitura).`
- `gosta(maria,leitura).`
- `gosta(paulo,leitura).`
- `gosta(pedro,cinema).`
- `gosta(paulo,cinema).`
- `gosta(vera,cinema).`
- `gosta(paulo,boliche).`
- `gosta(maria,boliche).`
- `gosta(vera,boliche).`



- /* Existe algo Z que X e Y gostam ? */