

GERENCIAMENTO DE CONHECIMENTO E MINERAÇÃO DE DADOS

Representação do Conhecimento Regras, Redes Semânticas e Frames

Huei Diana Lee

Programa de Pós-Graduação em Engenharia Elétrica
e Computação (PGEEC)

1

Regras IF THEN

- Também conhecidas como:
 - Regras de Produção ou
 - Regras de Decisão.
- Um dos formalismos mais comumente usados na representação de conhecimento.

2

Regras IF THEN

- Em geral, são **declarações condicionais**.
- Podem apresentar diversas interpretações, por exemplo:
 - **IF** precondição P **THEN** conclusão C.
 - **IF** situação S **THEN** ação A.
 - **IF** condições C1 e C2 são satisfeitas **THEN** condição C não é satisfeita.

3

Regras IF THEN

- As regras de produção são uma forma natural de representar o conhecimento.
- Exibem as seguintes características desejáveis:
 - **Modularidade**: cada regra define uma pequena parte, relativamente independente, do conhecimento.
 - **Incrementabilidade**: novas regras podem ser adicionadas à base de conhecimento, de modo relativamente independente, das outras regras.

4

Regras IF THEN

- **Capacidade de serem modificadas** (uma consequência da modularidade): regras antigas podem ser modificadas, relativamente independentemente, de outras regras.
- **Transparência**: é possível prover a capacidade para explicar as decisões e soluções.

5

Regras IF THEN

- Regras IF THEN facilitam a resposta de questões típicas de usuários:

“COMO”:

Como você chegou a essa conclusão?

“POR QUE”:

Por que você está interessado nessa informação?

6

Regras IF THEN

- Regras IF THEN geralmente definem **relações lógicas** entre conceitos do domínio do problema.
- Relações puramente lógicas são denominadas como “**Conhecimento Categórico**” (categórico pois sempre supõe-se que são verdadeiros).

7

Regras IF THEN

- Em alguns domínios, tais como o domínio do diagnóstico médico, prevalecem o “**Conhecimento Probabilístico**” (SOFT) (no sentido de que regularidades empíricas são somente válidas até certo ponto, geralmente, porém nem sempre).

8

Regras IF THEN

- No caso anterior, as regras podem ser modificadas acrescentando-se uma **qualificação probabilística** à interpretação lógica.
- Exemplo:
 - IF condição A THEN conclusão B com um grau de certeza F.

9

Regras IF THEN – Exemplo

Regra de produção do sistema especialista médico MYCIN (Shortliffe, 1976):

IF
 “A infecção é bacteriana primária”
 AND
 “O portal suspeito de entrada do organismo é o trato gastro-intestinal”
 THEN
 “Existe uma evidência sugestiva (0.7) de que a identidade do organismo é *bacteróides*”

OBS: 0.7 indica o grau de confiança da regra.

10

Regras IF THEN – Exemplo

Diagnóstico do problema de vazamento de água:

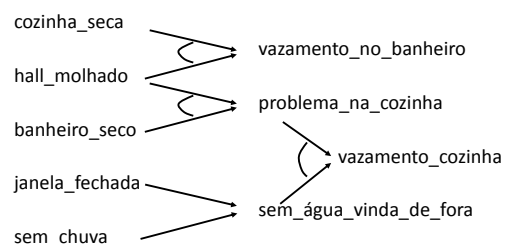
- O problema pode ter sido causado ou por um vazamento na cozinha ou no banheiro.
- Isto causa também um problema no hall (água no chão).
- Essa base assume que o problema somente pode ser causado por uma causa ou outra, nunca as duas ao mesmo tempo.



11

Regras IF THEN – Exemplo

Rede de Inferência - Grafo E/OU:



12

Regras IF THEN – Exemplo

- **Nós:** proposições;
- **Links:** regras na base de dados;
- **Arcos:** conexão conjuntiva entre as proposições correspondentes;

IF hall_molhado **AND** banheiro_seco
THEN problema_na_cozinha.

13

Regras IF THEN

Forward e Backward Chaining

Backward:

- Iniciamos com a hipótese vazamento_cozinha e raciocinamos para trás no grafo;
- Para confirmar a hipótese precisamos que o problema_na_cozinha e sem_água_vinda_de_fora sejam verdade;
- problema_na_cozinha é confirmado se encontrarmos o hall_molhado e banheiro_seco;
- sem_água_vinda_de_fora é confirmado se janela_fechada ou sem_chuva.



14

Regras IF THEN

Forward e Backward Chaining

- *Backward chaining* é o estilo de programação utilizado em Prolog:
 - vazamento_no_banheiro :-
 - hall_molhado,
 - cozinha_seca.
 - problema_na_cozinha :-
 - hall_molhado,
 - banheiro_seco.
- ...

15

Regras IF THEN

Forward e Backward Chaining

- Fatos Prolog:
 - hall_molhado.
 - banheiro_seco.
 - janela_fechada.
- Verificando a hipótese:
 - ?- vazamento_cozinha.
 - yes

16

Regras IF THEN

Forward e Backward Chaining

Forward:

- Uma vez observado que o hall está molhado e que o banheiro está seco, podemos concluir que existe um problema na cozinha;
- Também, sabendo que a janela está fechada, inferimos que a água não veio de fora, o que nos leva a conclusão final de vazamento na cozinha.



17

Redes Semânticas e Frames

- Direcionadas à representação, baseada em uma **estrutura**, de grandes conjuntos de fatos.
- Um conjunto de fatos é estruturado quando os fatos podem ser abstraídos e quando podem ser reconstruídos por meio de inferência.

18

Redes Semânticas

- Consistem de **entidades** e **relações** entre as entidades;
- Usualmente são representados por meio de **grafos**;
- **Nós**: representam as entidades;
- **Arcos**: representam as relações e são rotulados com os nomes dessas relações.

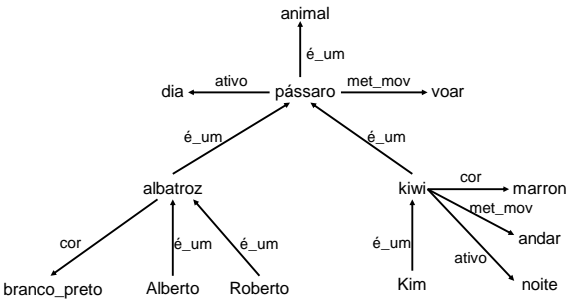
19

Redes Semânticas – Exemplo

- Uma **ave** é um **tipo** de animal;
- **Voar** é o **método** comum de movimentação das aves;
- Um albatroz é um pássaro;
- Alberto é um albatroz, assim como Roberto.

20

Redes Semânticas – Exemplo



21

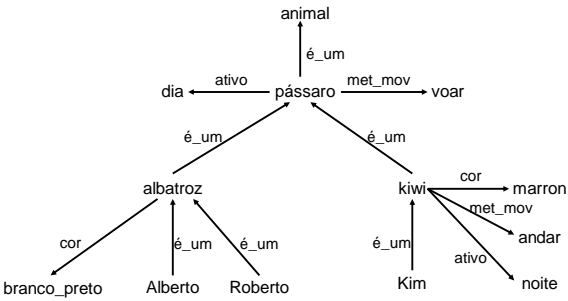
Redes Semânticas – Exemplo



<https://pt.wikipedia.org/wiki/Apteryx>

22

Redes Semânticas – Exemplo



23

Redes Semânticas – Exemplo

- Note que:
 - **é_um** relaciona, algumas vezes, uma classe de objetos a sua superclasse:
 - animal é uma superclasse de pássaro;
 - **é_um** relaciona, outras vezes, uma instância da classe à classe em si:
 - Alberto é um albatroz.

24

Redes Semânticas – Exemplo

- Uma rede como essa pode ser imediatamente traduzida para fatos Prolog:

```
é_um(pássaro, animal).  
é_um(roberto, albatroz).  
met_mov(pássaro, voar).  
met_mov(kiwi, andar).
```

- Além dos fatos, os quais são explicitamente declarados, alguns outros fatos podem ser inferidos por meio da rede.

25

Redes Semânticas – Exemplo

- Uma forma típica é a HERANÇA:
 - O fato “albatroz voa” é herdado de “pássaros voam”.
- Esses fatos são herdados por meio da relação **é_um**. Em Prolog:

```
met_mov(X, Met) :-  
    é_um(X, SuperX),  
    met_mov(SuperX, Met).
```

26

Frames

- Os fatos são agrupados em torno de objetos:
 - Objeto = objeto concreto físico
 - = conceito abstrato, como classe de objetos ou situações
- Bons candidatos para serem representados por frames:
 - Situações típicas de reuniões;
 - Situações de conflitos;
 - ...

27

Frames

- **FRAME**: estrutura de dado cujos componentes são chamados de slots (escaninhos).
- **SLOTS**: têm nomes e acomodam informação de tipos variados:
 - Valores simples;
 - Referências a outros frames ou
 - Procedimentos que podem computar o valor de slot a partir de outras informações.

28

Frames

- Os slots podem ser deixados em branco e serem preenchidos por meio de inferência;
- A forma mais comum de **inferência** é a **herança**.
- Quando um frame representa uma classe de objetos (como albatroz) e outro frame representa a superclasse dessa classe (como pássaro), então a classe frame pode herdar os valores do frame superclasse.

29

Frames - Exemplos

FRAME: pássaro

```
tipo_de: animal  
met_mov: voar  
ativo: dia
```

Pássaro – albatroz e kiwi:

```
FRAME: albatroz  
    tipo_de: pássaro  
    cor: branco_preto  
    tamanho: 115
```

```
FRAME: kiwi  
    tipo_de: pássaro  
    cor: marron  
    tamanho: 40
```

30

Frames - Exemplos

- Albatroz é um tipo comum de pássaro e herda a habilidade de voar e a atividade durante o dia do frame pássaro.
- Porém, kiwi é um pássaro atípico e os valores usuais de método de movimentação e atividade tem que ser sobrepostos (revogados).

31

Frames - Exemplos

- Podemos também ter uma instância particular de uma classe, por exemplo, para um albatroz chamado Alberto:

```
FRAME:  alberto
        instância_de: albatroz
        tamanho: 115
```
- Note a diferença entre `tipo_de` e `instância_de`.
 - `tipo_de`: relação entre classe e superclasse.
 - `instância_de`: relação entre membro da classe e classe.

32

Frames - Exemplos

- A informação nesse exemplo de frames pode ser representada em Prolog como um conjunto de fatos, cada fato para cada valor de slot. Por exemplo:

```
nome_frame(Slot,Valor).
Frame: pássaro           Frame: albatroz
passaro(tipo_de, animal). albatroz(tipo_de, pássaro).
passaro(met_mov, voar).   albatroz(cor, br_pr).
passaro(ativo, dia).       albatroz(tamanho, 115).
```

- Linguagens específicas para frames: KRL, FRL, KRYPTON, THEO, FRAMEKIT.

33

Referências

- Material didático de Monard, M.C. (ICMC-USP)
- Shortliffe, E.H. *Computer-based medical consultations: MYCIN*. New York: American Elsevier, 1976.
- Reiter, J., "AL/X: An Expert System Using Plausible Inference," Intelligent Terminals, Ltd., Oxford, 1980.
- Bratko, I. Knowledge-based problem-solving in AL3.. Machine Intelligence Workshop 10, Cleveland, Ohio, Nov. 1981. Published in Machine Intelligence 10, 1982 (eds. J.Hayes, D.Michie, J.H.Pao), Ellis Hoorwood and Wiley.
- Giarratano, J.C.; Riley, G. Expert Systems, Principles and Programming. Course Technology, 2005
- Outras referências indicadas no curso

34