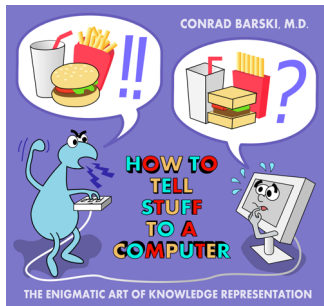


Sistemas Inteligentes, 11-12



Inês Dutra - CRACS & INESC-Porto LA
DCC-FCUP

April 23, 2012

Representação de Dados e de Conhecimento

Refs para este tópico:

- caps. 7, 8 e 12, Artificial Intelligence: a Modern Approach, 3rd ed., by Stuart Russell and Peter Norvig
- part II, Artificial Intelligence, 2nd ed., by Elaine Rich and Kevin Knight
- What is a Knowledge Representation?:
groups.csail.mit.edu/medg/ftp/psz/k-rep.html
- Informal intro to Knowledge Representation and state-of-the-art:
<http://lisperati.com/tellstuff/index.html>
- Decision Support and Expert Systems in Public Health:
<http://www.springerlink.com/content/g518q887221r0027/>
- The Unified Medical Language Systems (UMLS): http://www.nlm.nih.gov/research/umls/new_users/online_learning/OVR_001.htm
- The Cyc Language: <http://www.opencyc.org/doc>
- Prolog: <http://www.dcc.fc.up.pt/~vsc/Yap/>

Representação

- **Conhecimento** x **Dados**?
- **Conhecimento:** “representação simbólica de aspectos de algum universo de discurso”

Exemplos de “conhecimento”

- José é um funcionário da UP
- Todos os funcionários da UP têm salários maiores que 25.000 euros (:-)
- Todos os funcionários da UP sabem que devem ter um bom estilo de vida
- José não acha que tem um bom estilo de vida
- Todos que sabem que ele deveria ter um bom estilo de vida, mas pensa que não tem, estão desapontados

Representação

- **Dados:** “representação simbólica de aspectos **simples** de algum universo de discurso”
- **Dado:** caso especial de “conhecimento”

Exemplos de “dados”

- José é casado com Maria
- José é funcionário da UP
- O salário médio da UP é de 25.000 euros

Representação

- Representação do Conhecimento: expressar **conhecimento** de forma tratável pelo computador.

Diferentes formalismos

| | |
|----------------------|----------------------|
| Linguagem natural | Regras |
| Bases de dados | Árvores de decisão |
| Frames | Lógica |
| Scripts | Ontologias |
| Redes Semânticas | Redes causais |
| Algoritmos genéticos | Redes neuronais |
| Restrições | Orientação a objetos |
| Linguagens | etc! |

Representação em Linguagem Natural

Texto Clínico

“Enviada por densidade assimétrica no QSE da mama esquerda. Esta alteração existe desde 2005 mas a avaliação ecográfica do exterior sugere a necessidade de biópsia. Exame mamário com alteração palpável com cerca de 30 mm no QSE da mama esquerda.”

Desvantagens:

ambígua, redundante, pouca estrutura, sintaxe e semântica não são bem entendidas.

Representação em Bases de Dados

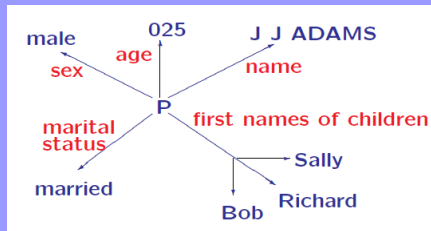
Base de Dados

```
person
record = { name : max 20 characters
           age  : 3 digits in range 000-120
           sex  : male or female
           marital status : married, bachelor,
                           spinster, divorced,
                           widowed, or engaged
           first names of children : up to 10 names
                                   each max 15 characters
           }
```

Representação em Bases de Dados: uma instância

Instância

| |
|-------------|
| J. J. ADAMS |
| 025 |
| male |
| married |
| Sally |
| Richard |
| Bob |



Discussão

- apenas aspectos simples podem ser representados (dados)
- entidades e relações
- Reasoning = lookup

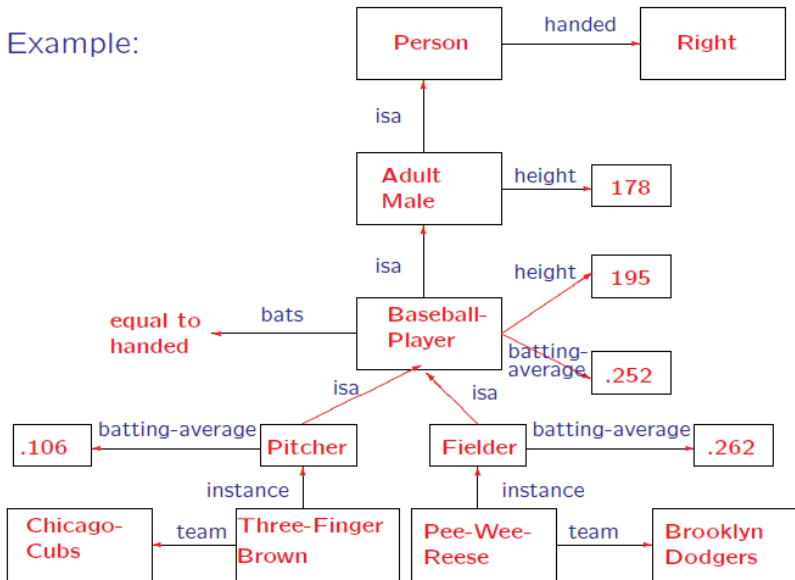
Representação em uma única tabela

Usual: agregar dados em uma única tabela!

| Patient | Location | Size | Date | Calcifications |
|---------|----------|------|----------|----------------|
| P1 | C | 0.1 | 20050403 | F, A |
| P1 | C | 0.2 | 20060412 | F |
| P1 | 9 | 0.1 | 20060412 | A |
| P2 | 12 | 0.3 | 20050415 | M |
| ... | ... | ... | ... | ... |

Representação em Redes Semânticas

Example:



Propriedades de Redes Semânticas

- permite **estruturar** o conhecimento para refletir a parte do universo que está sendo representada
- valores “default”
- sintaxe clara, mas semântica precisa ser trabalhada

Exemplo baseado em Cyc (linguagem)

"Bill Clinton belongs to the collection of U.S. presidents"

```
(#$isa #BillClinton #UnitedStatesPresident)
```

"All trees are plants"

```
(#$genls #Tree-ThePlant #Plant)
```

"Paris is the capital of France."

```
(#$capitalCity #France #Paris)
```

"if OBJ is an instance of the collection SUBSET and SUBSET is a subcollection of SUPERSET, then OBJ is an instance of the collection SUPERSET".

```
(#$implies
  (#$and
    (#$isa ?OBJ ?SUBSET)
    (#$genls ?SUBSET ?SUPERSET))
  (#$isa ?OBJ ?SUPERSET))
```

Frames

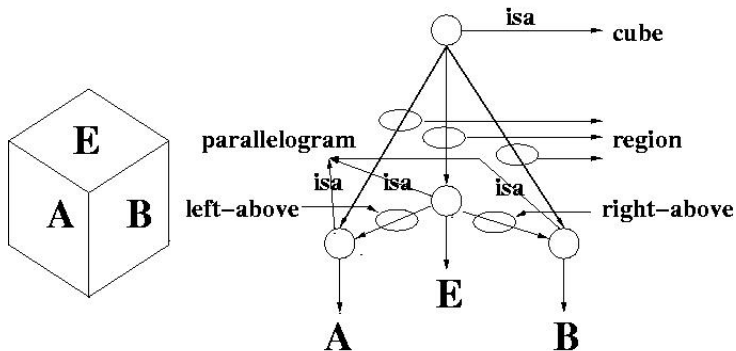
- Um “frame” consiste numa coleção de “slots”, cujo conteúdo pode ser um valor ou um apontador para outro Frame.

| Festa de aniversário | |
|----------------------|--|
| Vestuário: | social desportivo |
| Presente: | deve agradecer o aniversariante deve ser comprado e embrulhado |
| Jogos: | escondidas colocar o rabo no burro |
| Decoração: | balões, brindes, papel crepe |
| Menu: | Bolo, Gelado, Refri, Cachorro quente |
| Bolo: | acender velinhas, assoprar velinhas, fazer um pedido, cantar parabéns |
| Gelado: | napolitano |

Frames

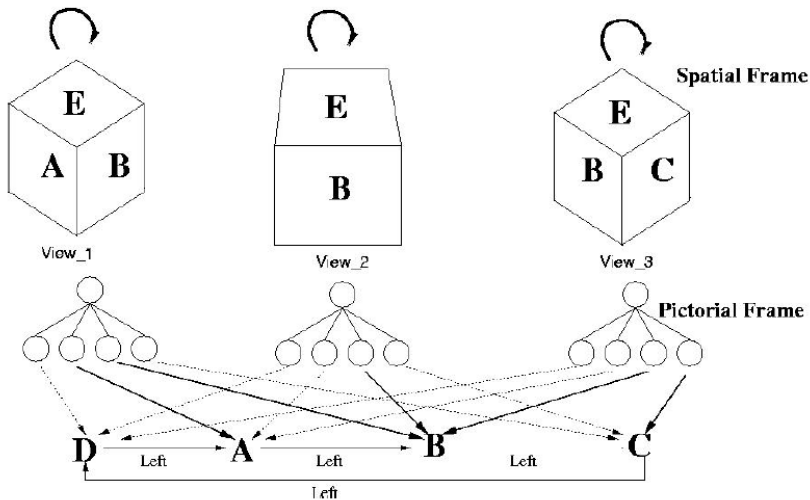
- Um “frame” consiste numa coleção de questões a serem respondidas sobre uma situação hipotética: especifica as questões e os métodos.
 - ▶ O que causou (*agente*)?
 - ▶ Qual é o propósito (*intenção*)?
 - ▶ Quais são as consequências (*efeitos*)?
 - ▶ A quem afeta (*receptor*)?
 - ▶ Como é feito (*instrumentos/métodos*)?

Exemplo



Objeto **composto** por relações.

Exemplo



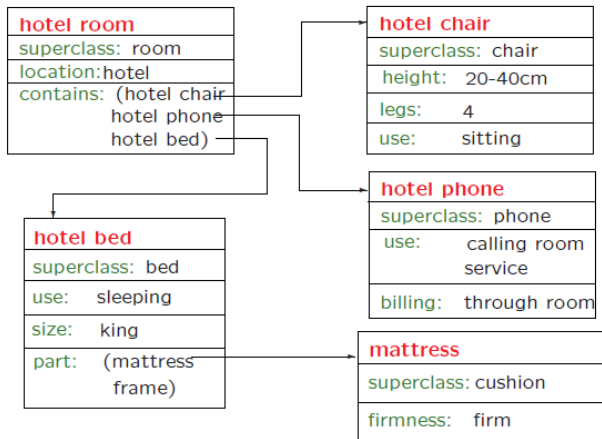
Diferentes aspectos de um cubo.

Frame para um aspecto do cubo

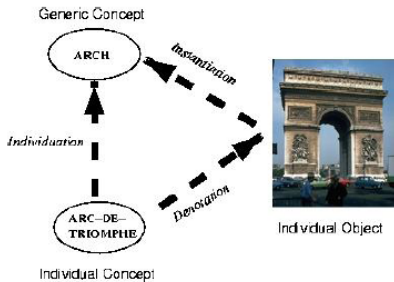
Um aspecto de um cubo usando representação em Frame

| View-of-a-Cube | | |
|----------------|--------|--|
| Slot | Filler | Constraint |
| Name | View_1 | |
| region_of | A | parallelogram & visible |
| region_of | B | parallelogram & visible |
| region_of | C | parallelogram & invisible |
| region_of | D | parallelogram & invisible |
| region_of | E | parallelogram(E) & visible & left-above(E,A) & right-above(E,B) |

Mais um exemplo



Importante!



- distinguir:
 - ▶ **conceitos** (representações) e **objetos** (instâncias)
 - ▶ **conceitos individuais** e **conceitos gerais**

Associação de procedimentos à representação

Procedimentos

| | |
|-------------------|---|
| rectangle | |
| superclass: | polygon |
| Coordinates: | (0cm,0cm) |
| length: | 5cm |
| width: | 2cm |
| area: | procedure(z) length(z) * width(z) |
| perimeter: | procedure(z) 2 * (length(z) + width(z)) |

Scripts

- Um “script” é uma representação estruturada que descreve uma sequência de eventos em um determinado contexto.
 - ▶ Estende os “frames” através de representação explícita de ações e mudanças de estados.
 - ▶ Define primitivas para descrever o universo:
 - PTRANS transferência física de um objeto (“go”)
 - ATRANS transferência de relações (“give”)
 - MTRANS transferência mental (“tell”)

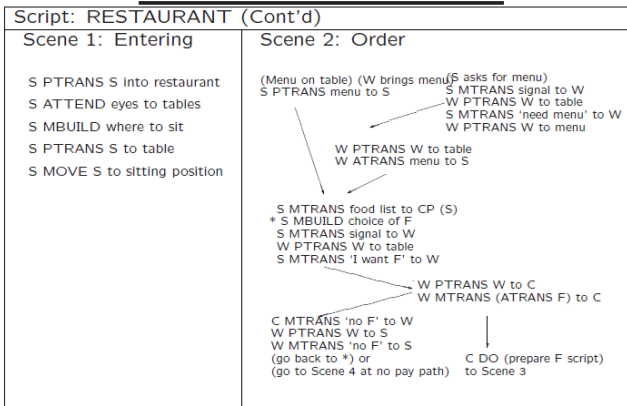
Exemplo de Script

Script para um restaurante

Script: RESTAURANT

| | | | |
|--------|-------------|--------------|------------------|
| Track: | Coffee Shop | Entry cond.: | S hungry |
| Props: | Tables | | S has money |
| | Menu | | |
| | F=Food | Results: | S has less money |
| | Check | | O has more money |
| | Money | | S is not hungry |
| Roles: | S=Customer | | |
| | W=Waiter | | |
| | C=Cook | | |
| | M=Cashier | | |
| | O=Owner | | |

Script para um restaurante (cont.)



Representação procedural ou declarativa

- Como? Procedural
- O que? Declarativo

Propriedades das boas Representações

- objetos importantes e suas relações estão explícitos
- expressam restrições que são naturais
- representam objetos e relações de forma conjunta
- omite detalhes irrelevantes
- transparente: fácil de entender
- completa
- concisa
- de armazenamento e recuperação rápidos
- “computáveis”

Propriedades das boas Representações

- Parte **léxica** que determina quais símbolos devem ser utilizados
- Parte **estrutural** que descreve a forma (restrições) como os símbolos podem ser organizados
- Parte **procedural** que especifica procedimentos de acesso que permitem a criação e modificação de descrições além de permitir fazer perguntas
- Parte **semântica** que estabelece uma forma de associar “significado” à descrição

Propriedades das boas Representações

Por exemplo, redes semânticas:

- Parte **léxica**: nodes, links, link labels
- Parte **estrutural**: grafo dirigido, com arestas etiquetadas
- Parte **procedural**: constructors, readers, writers, erasers (para criar e modificar o grafo)
- Parte **semântica**: significado dos nós e arestas depende da aplicação

Representação Lógica

- Linguagens:
 - ▶ **sintaxe**: descreve as possíveis configurações da linguagem que constituem sentenças válidas.
 - ▶ **semântica**: determina o significado de cada sentença.
- exemplo: $x > y$,
 - ▶ sintaxe: se x é um número e y é um número, então $x > y$ é uma sentença sobre números.
 - ▶ semântica: se $x > y$ retorna verdadeiro, senão retorna falso.

Representação Lógica

- Linguagem com sintaxe e semântica precisas: **lógica**.
- **Mecanismo de inferência**: derivado da sintaxe e da semântica.
- **Importante**: distinguir entre os fatos e sua representação
 - ▶ não podemos colocar todos os fatos do mundo no computador!
 - ▶ neste caso, devemos operar em representações dos fatos (codificação em alguma linguagem)
- **Raciocínio**: processo de construir novas configurações a partir de configurações já existentes.
- Bom raciocínio deve assegurar que as novas configurações representam fatos que se seguem dos fatos já existentes (**lógica monotônica**).

Representação Lógica












- “**Entailment**”: relação entre sentenças tal que novas sentenças geradas são verdadeiras, dado que as anteriores também são.
- $KB \models \alpha$ (**consequência lógica**)
- Mecanismo de inferência:
 - ▶ dada uma base de conhecimento KB, pode gerar novas sentenças que seguem de KB.
 - ▶ dada uma base de conhecimento e uma sentença α , pode dizer se α é consequência lógica de KB.
 - ▶ é **sound** ou **truth-preserving** se somente produzir sentenças que sejam consequência lógica de KB.

Representação Lógica

- **Prova:** procedimento de inferência “sound”.
- Analogia: procurar uma agulha num palheiro.
 - ▶ “entailment”: a agulha está no palheiro.
 - ▶ prova: encontrar a agulha.
 - ▶ palheiro de tamanho finito + procedimento sistemático de busca → agulha vai ser encontrada: procedimento de inferência **completo**.
- Como obter um procedimento “sound”?
 - ▶ passos de inferência devem respeitar a semântica das sentenças já existentes no KB.
 - ▶ derivar novas sentenças que sejam consequência lógica dos fatos já representados no KB.

Representação Lógica

Mundo do Wumpus :-)

| | | | |
|---|--|---|---|
| ST | |  BR |  |
|  | ST /  Gold |  |  BR |
| ST | |  BR | |
|  |  BR |  |  BR |

Representação Lógica

- **Representação:** duas classes de linguagens, programação e natural.
- Vantagens de linguagens de programação:
 - ▶ descrever algoritmos e estruturas de dados concretas.
 - ▶ Ex: `World[2,2] ← Pit.`
- Desvantagem: pouca expressividade. Como representar:
 - ▶ “há um buraco em [2,2] **ou** [3,1]”?
 - ▶ “há um monstro em **algum** quadrado”?

Representação Lógica

- **Lógica proposicional e lógica de primeira ordem** (cálculo de predicados de primeira ordem com igualdade).
- **Lógica proposicional**: símbolos são proposições. Ex: D pode ter a interpretação de que o wumpus está morto. Pode assumir valor falso ou verdadeiro.
 - ▶ símbolos proposicionais combinados através de **conectivos** booleanos formando sentenças mais complexas.
 - ▶ Linguagem bem simples.
- **Lógica de primeira ordem**: **objetos** e **predicados** relacionando objetos.
 - ▶ Admite **quantificadores** (\forall e \exists).
 - ▶ Mais expressiva do que proposicional.

Lógica Proposicional

Sintaxe:

$S \rightarrow AS \mid CS$

$AS \rightarrow \mathbf{True} \mid \mathbf{False} \mid P \mid Q \mid R \mid \dots$

$CS \rightarrow (S) \mid SCS \mid \neg S$

$C \rightarrow \wedge \mid \vee \mid \Leftrightarrow \mid \Rightarrow$

Lógica Proposicional

Exemplo: $S = ((P \vee H) \wedge \neg H) \Rightarrow P$ é uma fórmula válida.

| P | H | $P \vee H$ | $(P \vee H) \wedge \neg H$ | S |
|---|---|------------|----------------------------|---|
| F | F | F | F | T |
| F | T | T | F | T |
| T | F | T | T | T |
| T | T | T | F | T |

- P: wumpus está na posição [1,3].
- H: wumpus está na posição [2,2].
- Se sabemos que $(P \vee H)$ é verdadeiro e $\neg H$ também é verdadeiro, então o wumpus só pode estar na posição [1,3].

Lógica Proposicional

- Agente para o mundo do wumpus!
- B: brisa, S: mau cheiro, W: wumpus.
- $\neg S_{1,1}, \neg S_{2,1}, S_{1,2}, \neg B_{1,1}, B_{2,1}, \neg B_{1,2}$: fatos.

- Regras:

$$R_1 : \neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$$

$$R_2 : \neg S_{2,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$$

$$R_3 : \neg S_{1,2} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$$

$$R_4 : S_{1,2} \Rightarrow W_{1,1} \vee W_{1,2} \vee W_{1,3} \vee W_{2,2}$$

Lógica Proposicional

- Problemas com lógica proposicional:
- muitas proposições para o quadrado 4x4.
- Ex: “não ande para a frente se o wumpus estiver na sua frente” precisa de um conj de 64 regras (16 quadrados x 4 orientações).
- não tem memória do caminho a menos que se represente uma proposição para cada instante no tempo.
- Ex: move para $A_{2,1}$ se torna verdade e $A_{1,1}$ se torna falso. Mas pode ser importante guardar o fato de que o agente esteve em $A_{1,1}$.
- problema: não sabemos o tempo que vai levar para terminar o jogo.

Lógica Proposicional

- Exemplo de proposições adicionais:

$$A_{1,1}^0 \wedge East_A^0 \wedge W_{2,1}^0 \Rightarrow \neg Forward^0$$

$$A_{1,1}^1 \wedge East_A^1 \wedge W_{2,1}^1 \Rightarrow \neg Forward^1$$

$$A_{1,1}^2 \wedge East_A^2 \wedge W_{2,1}^2 \Rightarrow \neg Forward^2$$

⋮

- índice no topo de cada símbolo indica tempo.
- para 100 unidades de tempo: 6400 destas regras, somente para dizer: “não mova para a frente se o wumpus estiver lá”.
- lógica de primeira ordem: reduz as 6400 para apenas 1!

Lógica de Primeira Ordem (First-Order Logic or FOL)

- **objetos e relações** entre objetos, **propriedades, funções**.
- Objetos: pessoas, casas, números, teorias, Fernando Henrique, cores, jogos de futebol, séculos etc.
- Relações: irmão/irmã de, parte de, maior que, tem cor, ocorreu depois, pertence etc.
- Funções: pai de, melhor amigo de, vencedor de, um mais que etc.
- Ex: “quadrados vizinhos ao quadrado do wumpus têm mau cheiro”. Objetos: wumpus, quadrado; Propriedade: mau cheiro; Relação: vizinhança.
- Motivação para o uso de lógica de primeira ordem: formalismo mais estudado e melhor entendido que outras abordagens.

Lógica de Primeira Ordem (First-Order Logic or FOL)

$S \rightarrow AS \mid SCS \mid QVar, \dots S \mid \neg S \mid (S)$

$AS \rightarrow Pred(Term, \dots) \mid Term = Term$

$Term \rightarrow Func(Term, \dots) \mid Const \mid Var$

$C \rightarrow \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow$

$Q \rightarrow \forall \mid \exists$

$Const \rightarrow A \mid X_1 \mid John \dots$

$Var \rightarrow a \mid x \mid s \mid \dots$

$Pred \rightarrow Mother \mid LeftLegOf \mid \dots$

Lógica de Primeira Ordem (First-Order Logic or FOL)

- Agente lógico para o mundo do wumpus.
- três tipos de agentes: **reflexos**, **baseados em modelo** e **baseados em objetivos**.
- 1º passo: definir a interface com o mundo externo
- sentença (interface) típica:
Percept([Maucheiro,Brisa,Brilho,N,N],5), onde:
 - ▶ elem1: percebe ou não percebe mau cheiro,
 - ▶ elem2: percebe ou não percebe brisa,
 - ▶ elem3: percebe ou não percebe brilho,
 - ▶ elem4: percebe ou não percebe parede,
 - ▶ elem5: percebe ou não percebe grito (wumpus sendo morto).
- Ações: Turn(Right), Turn(Left), Forward, Shoot, Grab, Release, Climb.

Lógica de Primeira Ordem (First-Order Logic or FOL)

- Um agente reflexo simples.
- $\forall s, b, u, c, t P([s, b, Brilho, u, c], t) \Rightarrow Action(Grab, t)$
- $\forall b, g, u, c, t P([MauCheiro, b, g, u, c], t) \Rightarrow MauCheiro(t)$
- $\forall s, g, u, c, t P([s, Brisa, g, u, c], t) \Rightarrow Brisa(t)$
- $\forall s, b, u, c, t P([s, b, Brilho, u, c], t) \Rightarrow Ouro(t)$
- $\forall t AtOuro(t) \Rightarrow Action(Grab, t)$

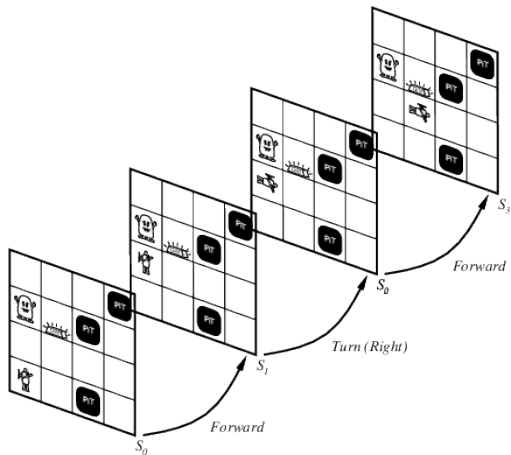
Lógica de Primeira Ordem (First-Order Logic or FOL)

- Limitações de um agente reflexo:
 - ▶ não faz parte da percepção deste tipo de agente saber onde está ou se está com o ouro.
 - ▶ é incapaz de evitar “loops”. Ex: assuma que o agente conseguiu pegar o ouro e está no caminho de volta para casa. Se passar novamente pelo mesmo quadrado visitado na ida, entra em loop.
 - ▶ problema: não está representado neste agente o fato dele estar carregando o ouro e a situação ser diferente da situação da ida.
- precisa de *representação de modificações* no mundo.

Lógica de Primeira Ordem (First-Order Logic or FOL)

- **Representação de modificações:** uma das áreas mais importantes em representação do conhecimento.
- regras *diacrônicas*.
- representação de *situações* e *ações* não é diferente de representação de objetos e relações.
- *Cálculo de Situações:* forma de descrever modificações em lógica de primeira ordem.

Lógica de Primeira Ordem (First-Order Logic or FOL)



Lógica de Primeira Ordem (First-Order Logic or FOL)

- Considera o mundo como uma sequência de **situações**.
- formato: $At(\text{Agente}, \text{posição}, \text{situação})$. Ex:
 $At(\text{Agent}, [1, 1], S_0) \wedge At(\text{Agent}, [1, 2], S_1)$
- cálculo de situações utiliza $Result(\text{action}, \text{situation})$ para representar a situação decorrente da execução de uma ação em situação anterior.
- Ex:
 - ▶ $Result(\text{Forward}, S_0) = S_1$
 - ▶ $Result(\text{Turn}(\text{Right}), S_1) = S_2$
 - ▶ $Result(\text{Forward}, S_2) = S_3$

Lógica de Primeira Ordem (First-Order Logic or FOL)

- Ações: são descritas através de seus efeitos:

Axiomas de efeito

Portable(Ouro)

$\forall s \text{ AtOuro}(s) \Rightarrow \text{Present}(\text{Ouro}, s)$

$\forall x, s \text{ Present}(x, s) \wedge \text{Portable}(x) \Rightarrow \text{Holding}(x, \text{Result}(\text{Grab}, s))$

$\forall x, s \neg \text{Holding}(x, \text{Result}(\text{Release}, s))$

- não suficiente para saber se o agente está segurando o ouro ou continua segurando o ouro.

Lógica de Primeira Ordem (First-Order Logic or FOL)

- necessário: regras para dizer se o mundo continuou o mesmo.

Axiomas de frame

$$\forall a, x, s \text{ Holding}(x, s) \wedge (a \neq \text{Release}) \Rightarrow \text{Holding}(x, \text{Result}(a, s))$$

$$\forall a, x, s \neg \text{Holding}(x, s) \wedge (a \neq \text{Grab} \vee \neg(\text{Present}(x, s) \wedge \text{Portable}(x)))$$

$$\Rightarrow \neg \text{Holding}(x, \text{Result}(a, s))$$

- combinação de axiomas de efeito e de frame:
verdadeiro posteriormente
 \Leftrightarrow [uma ação fez ser verdadeiro \vee já era verdadeiro antes]

Lógica de Primeira Ordem (First-Order Logic or FOL)

Axioma do estado sucessor

$$\forall a, s, x \text{ Holding}(x, \text{Result}(a, s)) \Leftrightarrow [\\ (a = \text{Grab} \wedge \text{Present}(x, s) \wedge \text{Portable}(x)) \vee \\ (\text{Holding}(x, s) \wedge a \neq \text{Release}) \\]$$

Necessário para cada predicado que pode mudar seu valor no decorrer do tempo.

Lógica de Primeira Ordem (*First-Order Logic or FOL*)

- Dedução de “propriedades escondidas”.
 - ▶ $\forall l, s \text{ At}(\textit{Agent}, l, s) \wedge \textit{Brisa}(s) \Rightarrow \textit{Fresco}(l)$
 - ▶ $\forall l, s \text{ At}(\textit{Agent}, l, s) \wedge \textit{MauCheiro}(s) \Rightarrow \textit{MauCheiroso}(l)$
- Regras *sincrônicas* para relacionar propriedades de um estado ao mesmo estado.
 - ▶ **Causais** (sistemas baseados em modelos):
 - $\forall l_1, l_2, s \text{ At}(\textit{Wumpus}, l_1, s) \wedge \textit{Adj}(l_1, l_2) \Rightarrow \textit{MauCheiroso}(l_2)$
 - $\forall l_1, l_2, s \text{ At}(\textit{Buraco}, l_1, s) \wedge \textit{Adj}(l_1, l_2) \Rightarrow \textit{Fresco}(l_2)$
 - ▶ **Diagnósticas** (sistemas baseados em diagnósticos):
 - $\forall l, s \text{ At}(\textit{Agent}, l, s) \wedge \textit{Brisa}(s) \Rightarrow \textit{Fresco}(l)$
 - $\forall l, s \text{ At}(\textit{Agent}, l, s) \wedge \textit{MauCheiro}(s) \Rightarrow \textit{MauCheiroso}(l)$
 - $\forall l_1, s \text{ MauCheiroso}(l_1) \Rightarrow (\exists l_2 \text{ At}(\textit{Wumpus}, l_2, s) \wedge (l_2 = l_1 \vee \textit{Adj}(l_1, l_2)))$

Regras

if Hib1_age_in_months >= 15 and Hib1_age_in_months < 60
then late_start_15_months

if Hib1_age_in_months >= 12 and Hib1_age_in_months < 15
 and (Hib.prior = 1 or Hib12_elapsed_months >= 2)
then late_start_12_months

if Hib1_age_in_months >=15 or
 (Hib2_age_in_months >=15 and Hib12_elapsed_months >= 2) or
 (Hib3_age_in_months >=15 and Hib23_elapsed_months >= 2) or
 Hib4_age_in_months >=12 or
 (Hib.prior = 2 and late_start_12_months) or
 (Hib.prior = 3 and Hib2_age_in_months >= 12 and
 Hib23_elapsed_months >= 2) or
 (Hib.prior = 3 and late_start_7_months) or
 (Hib.prior = 3 and PRPOMP_series and Hib3_age_in_months >=12
 and Hib23_elapsed_months >= 2)
then Hib_completed_valid_dose

Árvores de Decisão

- Simples e fácil de implementar.
- Recebe como entrada um objeto ou situação descrita por um conj de propriedades e produz uma resposta “sim” ou “não”. Representam funções booleanas.
- Exemplo: esperar por uma mesa num restaurante.
- Objetivo: aprender a definição do predicado “VouEsperar”, com a definição expressa por uma árvore de decisão.

Árvores de Decisão

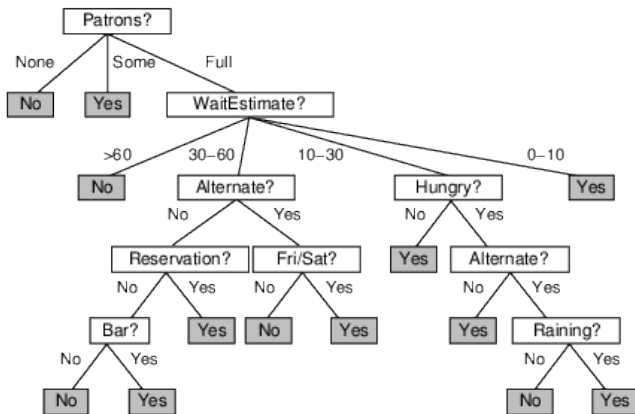
Decidimos (p eqto, pois isto poderia ser decidido pelo algoritmo de aprendizagem) as propriedades ou atributos:

- Alternativo: algum restaurante alternativo perto?
- Bar: restaurante tem uma área de espera?
- Sex/Sab: V se for sexta ou sábado.
- ComFome: estamos com fome?
- Clientes: número de pessoas no restaurante (Nenhuma, Algumas, Cheio).
- Preço: \$, \$\$, \$\$\$.
- Chovendo: está chovendo.
- Reserva: temos reserva?
- Tipo: Francês, Italiano etc.
- EsperaEstimada: 0–10min, 10–30, 30–60, > 60.

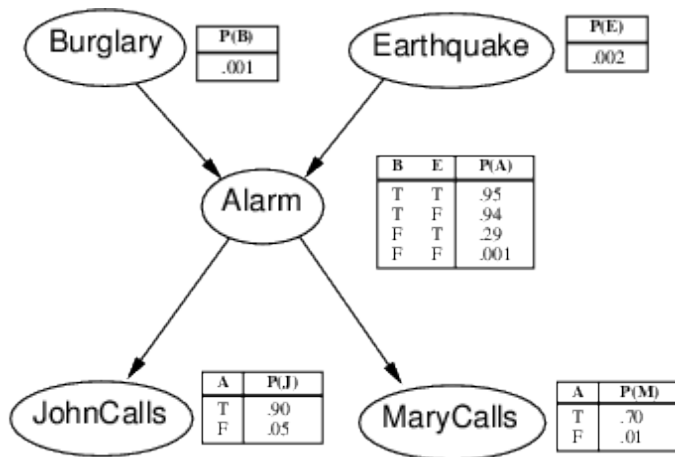
Árvores de Decisão

| Ex | Attributes | | | | | | | | | | Goal WillWait |
|-----|------------|-----|-----|-----|------|--------|------|-----|---------|------|------------------|
| | Alt | Bar | Fri | Hun | Pat | Price | Rain | Res | Type | Est | |
| X1 | Yes | No | No | Yes | Some | \$\$\$ | No | Yes | French | 0-10 | Yes |
| X2 | Yes | No | No | Yes | Full | \$ | No | No | Thai | 0-10 | Yes |
| X3 | No | Yes | No | No | Some | \$ | No | No | Burger | 0-10 | Yes |
| X4 | Yes | No | Yes | Yes | Full | \$ | No | No | Thai | 0-10 | Yes |
| X5 | Yes | No | Yes | No | Full | \$\$\$ | No | Yes | French | 0-10 | Yes |
| X6 | No | Yes | No | Yes | Some | \$\$ | Yes | Yes | Italian | 0-10 | Yes |
| X7 | No | Yes | No | No | None | \$ | Yes | No | Burger | 0-10 | Yes |
| X8 | No | No | No | Yes | None | \$\$ | Yes | Yes | Thai | 0-10 | Yes |
| X9 | No | Yes | Yes | No | Full | \$ | Yes | No | Burger | 0-10 | Yes |
| X10 | Yes | Yes | Yes | Yes | Full | \$\$\$ | No | Yes | Italian | 0-10 | Yes |
| X11 | No | No | No | No | None | \$ | No | No | Thai | 0-10 | Yes |
| X12 | Yes | Yes | Yes | Yes | Full | \$ | No | No | Burger | 0-10 | Yes |

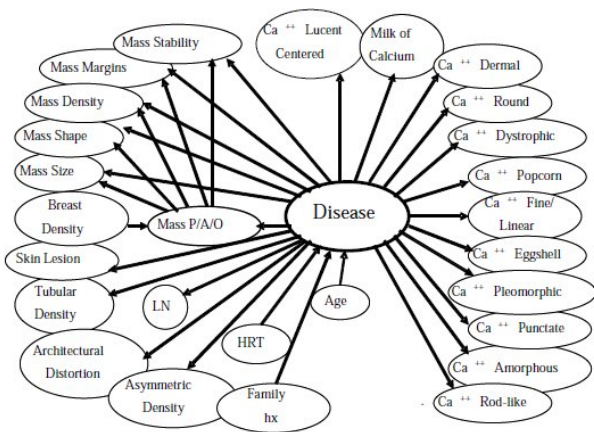
Árvores de Decisão



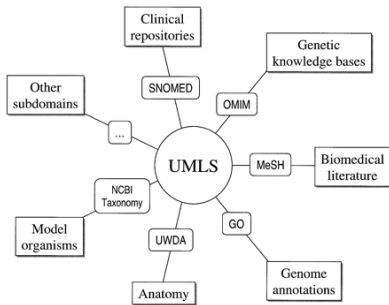
Redes de Bayes



Redes de Bayes



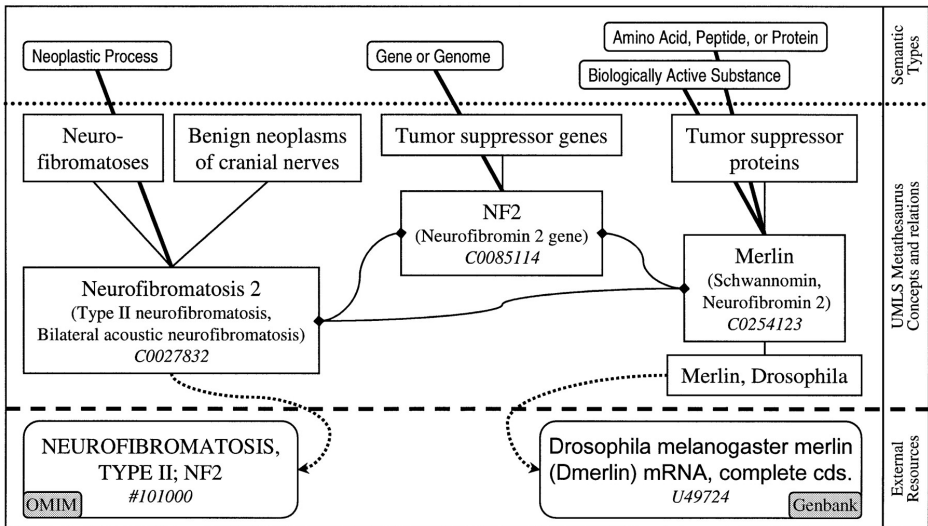
Unified Medical Language System – UMLS



MetaThesaurus

The major component of the UMLS is the Metathesaurus, a repository of inter related biomedical concepts. The two other knowledge sources in the UMLS are the **Semantic Network**, providing high level categories used to categorize every Metathesaurus concept, and lexical resources including the SPECIALIST lexicon and programs for generating the lexical variants of biomedical terms. (Nucleic Acids Research, Oxford Journals)

Unified Medical Language Systems – Example



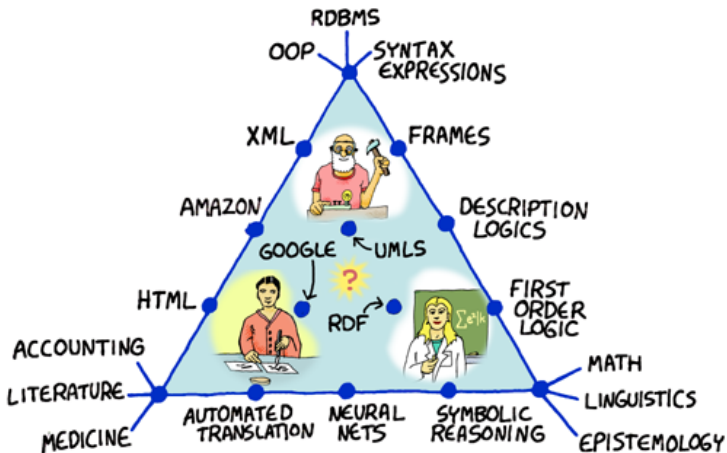
XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Patient id="122">
3   <General>
4     <Weight-kg>35</Weight-kg>
5     <Height-cm>128</Height-cm>
6     <Day>05</Day>
7     <Month>02</Month>
8     <Year>2004</Year>
9     <Sex>1</Sex>
10    <AuscultationPosition>2</AuscultationPosition>
11  </General>
12  <SystemicPressure>
13    <SystemicPressureMethod>1</SystemicPressureMethod>
14    <SystolicSystemicPressure-mmHg>130</SystolicSystemicPressure-mmHg>
15    <DiastolicSystemicPressure-mmHg>90</DiastolicSystemicPressure-mmHg>
16  </SystemicPressure>
17  <PulmonaryPressure>
18    <PulmonaryPressureMethod />
19    <SystolicPulmonaryPressure-mmHg />
20    <DiastolicPulmonaryPressure-mmHg />
21    <CatheterismSimultaneousMeasurement />
22    <EchocardiograSameConsultation />
23  </PulmonaryPressure>
24  <Murmur>
25    <Cycle>2</Cycle>
26  </Murmur>
27  <S1>
28    <S1Status>1</S1Status>
29  </S1>
30  <S2>
31    <S2Status>1</S2Status>
32    <IfAbnormal>0</IfAbnormal>
33    <PulmonaryComponent>1</PulmonaryComponent>

```

Estado da Arte



Exercícios

1. Representar as frases abaixo em lógica proposicional ou de primeira ordem

- a) Cavalos, vacas e porcos são mamíferos.
- b) Um descendente de um cavalo é um cavalo.
- c) Barbazul é um cavalo.
- d) Barbazul é pai de Silver.
- e) descendente e pais são relações inversas.
- f) Todo mamífero tem pais.

Exercícios

2. Seja a Tabela abaixo com a categorização de doenças reumáticas.

| Class | Name | Number of patients |
|-------|---------------------------------------|--------------------|
| A1 | degenerative spine diseases | 158 |
| A2 | degenerative joint diseases | 128 |
| B1 | inflammatory spine diseases | 16 |
| B234 | other inflammatory diseases | 29 |
| C | extra-articular rheumatism | 21 |
| D | crystal-induced synovitis | 24 |
| E | non-specific rheumatic manifestations | 32 |
| F | non-rheumatic diseases | 54 |

Table: Doenças reumáticas

Exercícios

Com base nesta tabelas, grupos de sintomas foram criados por especialistas. Representar estes grupos de sintomas utilizando lógica de primeira ordem.

Dores nas juntas

| | |
|------------|-------------------|
| Joint pain | Morning stiffness |
| no pain | ≤ 1 hour |
| arthrotic | ≤ 1 hour |
| arthritic | > 1 hour |

Dores na coluna

| | |
|-------------|-------------------|
| Spinal pain | Morning stiffness |
| no pain | ≤ 1 hour |
| spondylotic | ≤ 1 hour |
| spondylitic | > 1 hour |

Dores nos homens

| | |
|------|------------|
| Sex | Other pain |
| male | thorax |
| male | heels |

Relações entre dores nas juntas e dores na coluna

| | |
|------------|-------------|
| Joint pain | Spinal pain |
| no pain | spondylotic |
| arthrotic | no pain |
| no pain | spondylitic |
| arthritic | spondylitic |
| arthritic | no pain |
| no pain | no pain |

Exercícios

3. Dadas as variáveis observadas sobre cancro da próstata, qual seria a melhor rede de Bayes para representar estas variáveis?

| Variável | Significado |
|----------|--------------------------------|
| age | idade |
| wt | Peso |
| hx | História familiar de cancro |
| sbp | Pressão arterial sistólica |
| dbp | Pressão arterial diastólica |
| hg | Hemoglobina |
| stage | Estado Clínico |
| dtime | Doubling Time PSA |
| sz | Tamanho da próstata |
| bm | Metástases ósseas |
| status | Status após 5 anos de cirurgia |

PSA=Prostate Specific-Antigen

Doubling Time PSA: tempo decorrido até que os valores de PSA atinjam o dobro.