

Artificial Intelligence/ Inteligência Artificial

Lecture 6: Knowledge Engineering

(slides baseados em Oliveira, 2017 e Cortez 2014)

Luís Paulo Reis

lpres@fe.up.pt

Director of LIACC – Artificial Intelligence and Computer Science Lab.
Associate Professor at DEI/FEUP – Informatics Engineering Department,
Faculty of Engineering of the University of Porto, Portugal
President of APPIA – Portuguese Association for Artificial Intelligence



Inteligência Artificial e Conhecimento

- Inteligência Artificial e Conhecimento
(C. da Cognição Vs Engenharia do Conhecimento)
 - Inteligência Artificial,
 - Tecnologia da Programação
- A Engenharia do Conhecimento **desenvolve e explicita** métodos de "Representação do Conhecimento" e Estratégias de tratamento computacional desse mesmo Conhecimento
- **Questões principais que se colocam:**
 - 1 O que é o conhecimento?
 - 2 Como pode ser representado?
 - 3 Como pode ser manipulado?

Agentes Baseados em Conhecimento

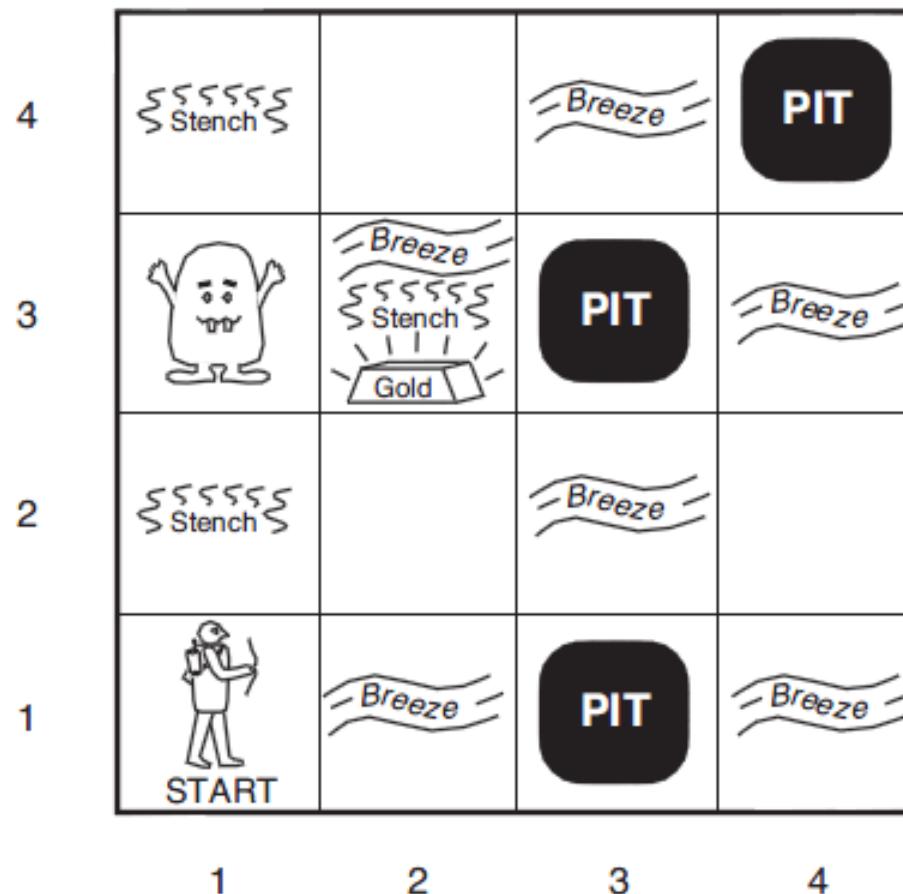
- **Base de Conhecimento (BC) constituída por factos (do domínio)**
- **Mecanismo de Inferência (independente do domínio)**
- **Aproximação Declarativa para Construir um Agente:**
 - Dizer-lhe o que precisa de saber
 - Desta forma ele pode perguntar-se a si mesmo (respostas vêm da Base de conhecimento)
- **Agentes podem ser vistos ao:**
 - Nível do Conhecimento (o que eles sabem sem preocupação com a forma de implementação)
 - Nível da Implementação (estruturas de dados na BC e algoritmos que as manipulam)

Agente Simples Baseado em Conhecimento

- **Agente deve ser capaz de:**
 - Representar estados, ações, etc...
 - Incorporar novas percepções
 - Atualizar as representações internas do mundo
 - Deduzir propriedades escondidas do mundo
 - Deduzir as ações apropriadas

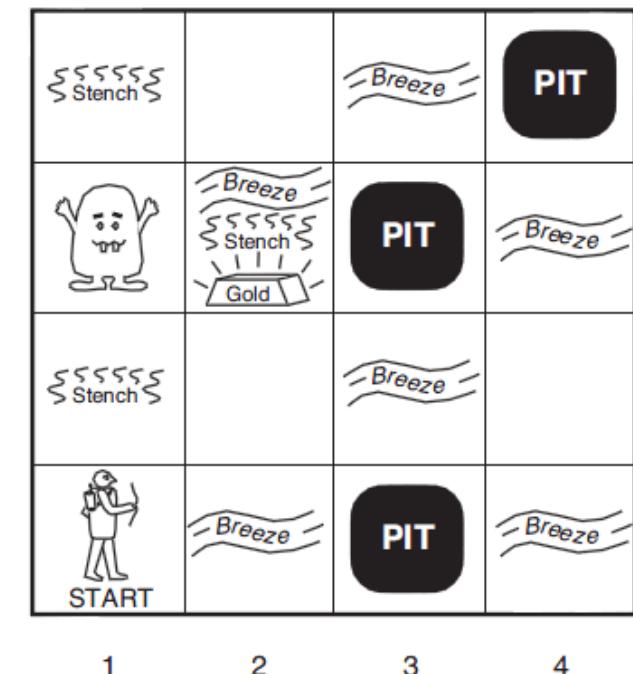
O Mundo do Wumpus

- Jogador deve apanhar o Ouro e regressar ao início sem entrar em nenhum quadrado com buraco ou com o Wumpus



O Mundo do Wumpus

- Perceções: Cheiro, Brisa, Brilho
- Ações: Virar à Direita, Virar à Esquerda, Frente, Agarrar, Largar, Disparar
- Objetivos: Apanhar o Ouro e regressar ao início sem entrar em nenhum quadrado com buraco ou Wumpus
- Ambiente:
 - Quadrados junto ao Wumpus têm cheiro
 - Quadrados junto aos buracos têm brisa
 - Quadrados com Ouro têm brilho
 - Agarrar pega no Ouro se ele estiver no mesmo quadrado
 - Largar deixa o Ouro no quadrado atual
 - Disparar mata o Wumpus se estivermos virados para ele
 - Disparar usa a única seta



Caracterização do Mundo do Wumpus

- **Determinístico?**
 - Sim! Resultado é especificado exatamente!
- **Acessível?**
 - Não! Só existe percepção local!
- **Estático?**
 - Sim! Wumpus, Buracos e Ouro não se movem!
- **Discreto?**
 - Claro que Sim!

Raciocínio e Ação no Mundo Wumpus

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1 A OK	2,1 OK	3,1	4,1

(a)

A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2 P?	3,2	4,2
OK			
1,1 V OK	2,1 A B OK	3,1 P?	4,1

(b)

Raciocínio e Ação no Mundo Wumpus

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

(a)

A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

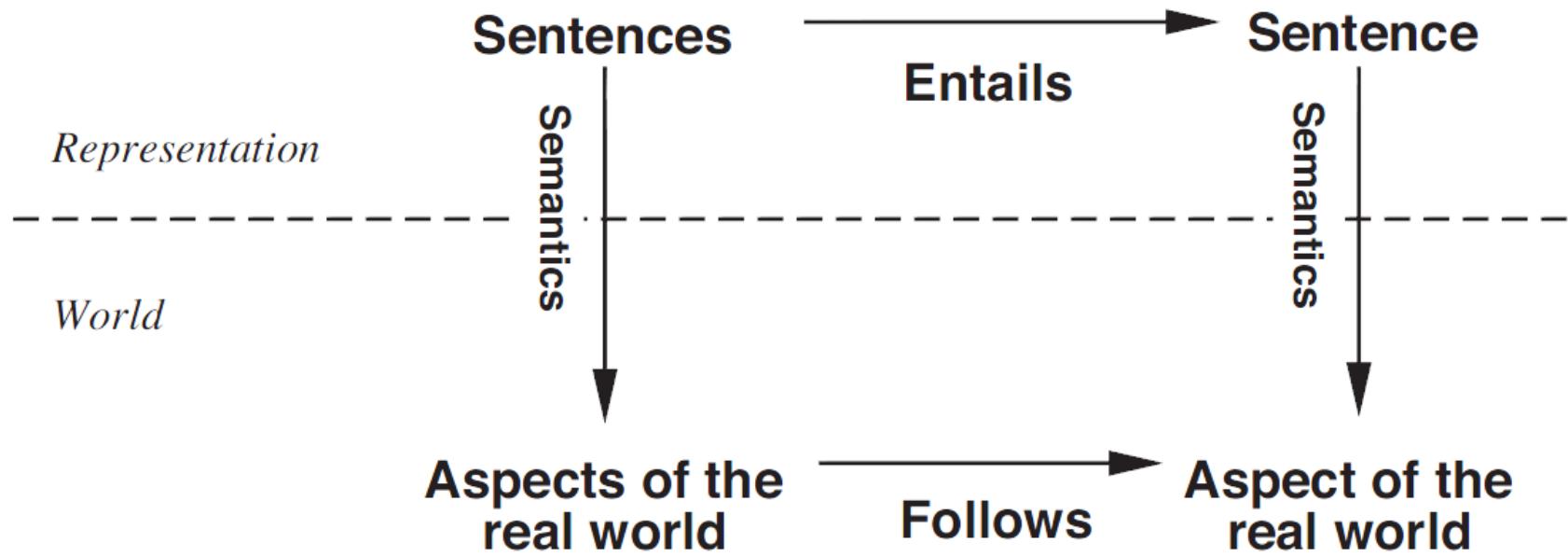
1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A S G B	3,3 P?	4,3
1,2 S V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

(b)

Representação, Raciocínio e Lógica

- **Representação do conhecimento** procura representar o conhecimento de forma a que seja manipulável pelo computador
- **Lógicas** são **linguagens formais** para representar informação de forma a que conclusões possam ser tiradas
- **Sintaxe** define as possíveis frases de uma linguagem
- **Semântica** define os factos do mundo a que as frases se referem (significado da frases)
- Por exemplo em linguagem aritmética:
 - $x+2 > y$ é uma frase; $x2+y>$ não é!
 - $x+2>y$ é verdadeira se o número $x+2$ é maior do que o número y
 - $x+2>y$ é verdadeira num mundo em que $x=5, y=3$
 - $x+2>y$ é falsa num mundo em que $x=0, y=3$

Representação



Lógica de Primeira Ordem (1)

- **Elementos Básicos:**
 - Constantes: D.Dinis, 2, UnivMinho, Paulo, ...
 - Predicados: Irmão, >, <
 - Funções: sen, cos, Pai_de, Perna_Esquerda_de, Sqrt, ...
 - Variáveis: x, y, a, b, ...
 - Conectores: $\wedge \vee \neg \Rightarrow \Leftrightarrow$
 - Igualdade: =
 - Quantificadores: $\forall \exists$
- **Frase atómica= predicado(termo₁, ..., termo_n) ou termo₁=termo₂**
- **Termo = função(termo₁, ..., termo_n) ou constante ou variável**
- **Exemplos:**
 - pai_de(Paulo, Joaquim)
 - >(capacidade(Antas), capacidade (Bessa))

Lógica de Primeira Ordem (2)

- **Frases complexas:**
 - Construídas a partir de frases simples utilizando os conectores:
 $\neg S_1$ $S_1 \wedge S_2$ $S_1 \vee S_2$ $S_1 \Rightarrow S_2$ $S_1 \Leftrightarrow S_2$
- **Exemplos:**
 - $\text{irmão(Paulo, Rui)} \Rightarrow \text{irmão(Rui, Paulo)}$
 - $\text{filho(Paulo, Joaquim)} \Rightarrow \text{pai(Joaquim, Paulo)}$
 - $\text{>(x,2)} \wedge \text{<(x,6)}$
- **Frases são verdadeiras com respeito a um modelo e uma interpretação!**
 - Modelo contém objetos e relações entre eles!
 - Uma frase atómica $\text{predicado(termo}_1, \dots, \text{termo}_n\text{)}$ é verdadeira se os objetos a que se referem $\text{termo}_1, \dots, \text{termo}_n$ estão na relação definida por predicado

Quantificadores Universal (\forall) e Existencial (\exists)

- **Quantificador Universal**
 - \forall (variáveis) (frase)
 - Exemplo: “Todos em Oxford são espertos”:
 - $\forall x \text{ Estuda}(x, \text{Oxford}) \Rightarrow \text{esperto}(x)$
 - Erro comum: Usar \wedge em vez de \Rightarrow
 - $\forall x \text{ Estuda}(x, \text{Oxford}) \wedge \text{esperto}(x)$
 - Significa Todos estão em Oxford e todos são espertos
- **Quantificador Existencial**
 - \exists (variáveis) (frase)
 - Exemplo: “Alguém em Oxford é esperto”:
 - $\exists x \text{ Estuda}(x, \text{Oxford}) \wedge \text{esperto}(x)$
 - Erro comum: Usar \wedge em vez de \Rightarrow
 - $\exists x \text{ Estuda}(x, \text{Oxford}) \Rightarrow \text{esperto}(x)$
 - Verdadeiro se existe alguém que não está em Oxford!

Propriedades dos Quantificadores

- $\forall x \forall y$ é o mesmo que $\forall y \forall x$
- $\exists x \exists y$ é o mesmo que $\exists y \exists x$
- $\forall x \exists y$ é diferente de $\forall y \exists x$
 - $\exists x \forall y \text{Gosta}(x,y)$ significa existe uma pessoa que gosta de toda a gente
 - $\forall y \exists x \text{Gosta}(x,y)$ todas as pessoas têm pelo menos uma que gosta delas
- **Dualidade:**
 - $\forall x \text{Gosta}(x,\text{Chocolate})$ é o mesmo que $\neg \exists x \neg \text{Gosta}(x,\text{Chocolate})$
 - $\exists x \text{Gosta}(x,\text{Couves})$ é o mesmo que $\neg \forall x \neg \text{Gosta}(x,\text{Couves})$

Exercícios

1) Representar as seguintes frases em lógica de primeira ordem:

- a) Existe um aluno que chumbou a História
- b) Só um aluno chumbou a História
- c) Nem todos os estudantes se inscreveram simultaneamente a Sistemas Baseados em Conhecimento e Sistemas Distribuídos
- d) Só um aluno chumbou a História e a Biologia
- e) A melhor nota a História foi mais elevada do que a melhor nota a Biologia
- f) Todos os Governantes são Incompetentes
- g) Todas as pessoas que não gostam de Governantes são espertos
- h) Nenhuma pessoa gosta de um Governante esperto
- i) Existe um Político que gosta de todos os Governantes que não são Incompetentes
- j) Existe um Barbeiro que barbeia toda a gente menos ele próprio
- k) Os políticos podem enganar algumas pessoas todo o tempo ou enganar todas as pessoas algum tempo, mas eles não podem enganar todas as pessoas todo o tempo

Exercícios Resolvidos

a) Existe um aluno que chumbou a História

$$\exists x(\text{Student}(x) \wedge \text{Failed}(x, \text{History}))$$

b) Só um aluno chumbou a História

$$\exists x(\text{Student}(x) \wedge \text{Failed}(x, \text{History}) \wedge \forall y(\text{Failed}(y, \text{History}) \Rightarrow x=y))$$

c) Nem todos os estudantes se inscreveram simultaneamente a Sistemas Baseados em Conhecimento e Sistemas Distribuídos

$$\exists x(\text{Student}(x) \wedge \neg(\text{Take}(x, \text{SBC}) \wedge \text{Take}(x, \text{SD}))) \text{ ou}$$

$$\exists x(\text{Student}(x) \wedge (\neg \text{Take}(x, \text{SBC}) \vee \neg \text{Take}(x, \text{SD})))$$

d) Só um aluno chumbou a História e a Biologia

$$\exists x(\text{Student}(x) \wedge \text{Failed}(x, \text{History}) \wedge \text{Failed}(x, \text{Biology}) \wedge (\forall y((\text{Failed}(y, \text{History}) \wedge \text{Failed}(y, \text{Biology})) \Rightarrow x=y)))$$

k) Os políticos podem enganar algumas pessoas todo o tempo ou enganar todas as pessoas algum tempo, mas eles não podem enganar todas as pessoas todo o tempo

$$\forall \text{Politico}(x) \Rightarrow ((\exists y \forall t(\text{Tempo}(t) \wedge (\text{Pessoa}(y) \Rightarrow \text{Enganar}(x, y, t))) \vee (\exists t \forall y(\text{Tempo}(t) \wedge (\text{Pessoa}(y) \Rightarrow \text{Enganar}(x, y, t))) \wedge \neg \forall x \forall t(\text{Tempo}(t) \wedge (\text{Pessoa}(y) \Rightarrow \text{Enganar}(x, y, t))))$$

Representação do Conhecimento

- “General Problem Solver” Versus Sistema Baseados em Conhecimento
Não determinismo implica métodos de Pesquisa de soluções em Espaços de Estados (Métodos de Resolução de Problemas)
- **Representação** e tratamento do Conhecimento:

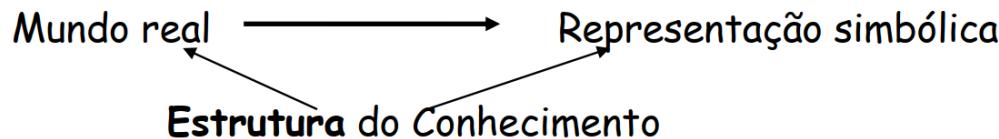
Características: Métodos:

- | | |
|--------------|--|
| – exato | (Processos de inferência da Lógica clássica) |
| – Temporário | (Raciocínio não-monótono) |
| – Incerto | (Teorias probabilísticas frequencistas ou subjetivistas) |
| – Incompleto | (Raciocínio por defeito) |

Representação do Conhecimento

Representação do conhecimento

Tradução computacional do conhecimento e raciocínio



Adequação: para a Retribuição
Inferência
Model(iz)ação

Eficiência: na Inferência

Métodos de representação:

Declarativos

coleção estática de factos
+
procedimentos genéricos de manipulação

Procedimentais

conhecimento codificado
de forma operacional

Representação do Conhecimento

Esquemas:

Descrição mais abstrata de padrões de suporte do conhecimento

- Termos em Lógica

- Enquadramentos (“Frames”)
coleção de atributos e propriedades relacionadas com um **conceito**

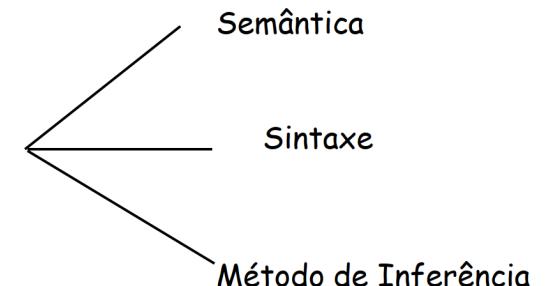
- Guiões (“scripts”)
conjuntos de sequências de **ações** relacionados com eventos típicos

- Modelos de Regras

Associações de pares **Condições – Ações** **Representação**

- Redes Associativas (“Semantic Networks”)

Grafos de objetos (conceitos e instâncias) ligados por **relações**



Representação do Conhecimento

Primitivas Conceptuais do domínio podem representar-se por “triplos”:
Objeto/Entidade Atributos/Propriedades Valores

Conceitos são Classes: descrição de Conjuntos de Objetos parentados
(da mesma classe)

Definição Formal de Sistema de Representação de Conhecimento:

$$S = (Ob, At, \{Val_a\}, f)$$

$$a \in At$$

onde: **Ob**: Conjunto não vazio de Objetos (entidades)

At: Conjunto não vazio de Atributos

{Val_a} Conjunto não vazio de Conj^os de valores para cada Atributo

f: Função de Ob^*At para Val_a $a \in At$

tal que $f(o,a) \in Val_a$, $\forall o \in Ob$, $\forall a \in At$

Val_a é o domínio dos valores do Atributo a

f é a **Função de Informação**

Representação do Conhecimento

ex: ob={Pac1, Pac2, Pac3, Pac4}
At={sexo, idade}
 $Val_{\text{sexo}}=\{\text{ma, fe}\}$
 $Val_{\text{idade}}=\{\text{novo, médio, idoso}\}$

	At	S	I
Pac1		ma	novo
Pac2		ma	médio
Pac3		fe	velho
Pac4		fe	novo

O $\underline{\quad}$ Val

Sistemas de Representação com informação multi-valor

$$S = (Ob, At, \{Val_a\}_{a \in At}, g)$$

g é uma relação tal que se $(o, a, v) \in g \rightarrow V \in Val_a$
e $\forall o \in Ob, \forall a \in At, \forall v \in Val_a : (o, a, v) \in g$

Relações válidas:

$$g1 \equiv (P1, Ling, Fr)$$

$$g2 \equiv (P1, Ling, Pt)$$

$$g3 \equiv (P4, Grau, M) \dots$$

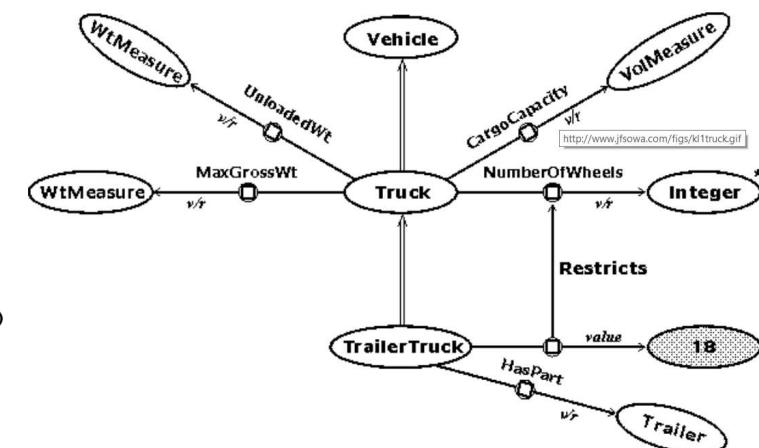
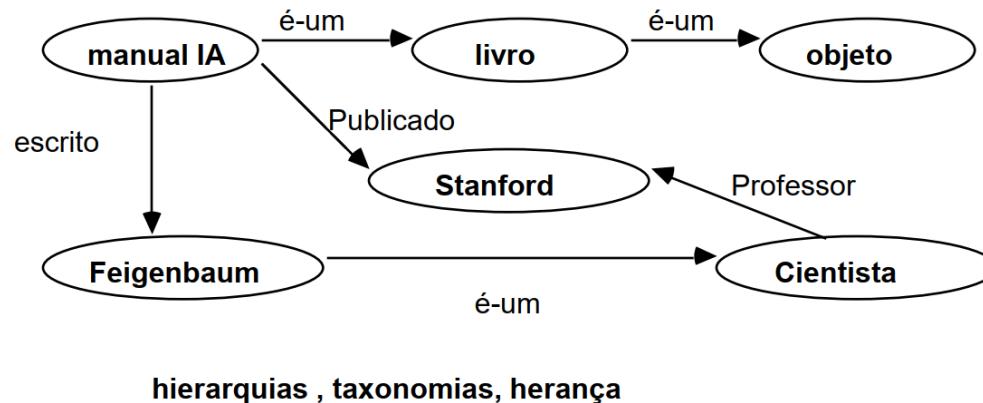
ex: $Ob = \{P1, P2, P3, P4\}$
 $At = \{\text{Ling, Grau}\}$
 $Val_{\text{Ling}} = \{\text{Fr, Ig, Pt, Al, It}\}$
 $Val_{\text{grau}} = \{\text{Bch, Lic, M, D, Pat}\}$

	Ling	Grau
P1	Fr,Pt,Ig	Bch, Lic
P2	Fr, Pt	Bch
P3	Pt	Pat
P4	Fr,Ig,Pt,Al	Lic, M, D

Representação do Conhecimento

REPRESENTAÇÃO DO CONHECIMENTO = ESTRUTURA + ACESSO

- Esquemas lógicos : Lógica de predicados / Lógica proposicional
 - Fáceis de operar por regras de inferência
 - Requerem muita memória
- Redes Associativas: descrever propriedades e relações entre conceitos, objetos, eventos, situações, nós e arcos etiquetados
- Ex: Semantic Networks



hierarquias , taxonomias, herança

Representação do Conhecimento

- Regras de “Produção” (“Produções”):

- Pares Condições – Ação
- Modulares
- Mais fácil expansão
- Ativação pelo estado do sistema
- Próximas do modelo cognitivo

Condições implicam Conclusão

Premissa1 E Premissa2 E ... Premissa n IMPLICA Conclusão

Incluindo Incertezas:

P1 –FC1 E P2-FC2 E ... Pn-FCn IMPLICA C1-FCC

Esquema: IF P THEN C1 ELSE C2... P Função Booleana; Ci Conclusões alternativas

Ex: SE factor (património, Ponto, existe)

OU factor (flora, Ponto, rica)

E factor (acessibilidade, Ponto, boa)

ENTÃO aptidão (lazer, Ponto, grande)

SENÃO aptidão (lazer,Ponto,40% factor1+60%f2).

Representação do Conhecimento

“Frames” (Enquadramentos):

Estruturas de conhecimento para representar objetos ou situações típicas e compostas por:

 nome da “frame”

 campos (“slots”) que podem conter dados ou procedimentos

Frame

Nome: Avião

Tipo: combate, transporte, bombardeiro, abastecimento, vigilância

Fabricante: Boeing, Aerospatial, Embraer, Douglas

Peso: X toneladas

Peso Máximo: 1,6 * Peso

Velocidade: apontador para tabela(tipo de avião, peso máximo)

Tripulação “cockpit”: 1=<T=<3; por defeito 2

Permite hierarquias e procedimentos ligados

Representação do Conhecimento

- **Implementação de um Sistema de Enquadramentos ("Frames") em Prolog.** (Pode ser codificado noutras linguagens)!!
 - Estrutura de Representação:
 - Nome do **predicado**: frame
 - 2 Argumentos**: Nome_frame, Lista das "slots" (campos)

Slot:	Nome do campo-Lista de facetas
Faceta:	Tipofaceta Valor
Tipofaceta:	valor, valordefeito, calculo, add, de

Exemplo:

```
frame (macaco,[ ordem-[val, primata],  
              pelo-[def, curto],  
              peso-[calculo, tab_peso_altura]  
          ]  
      ) .
```

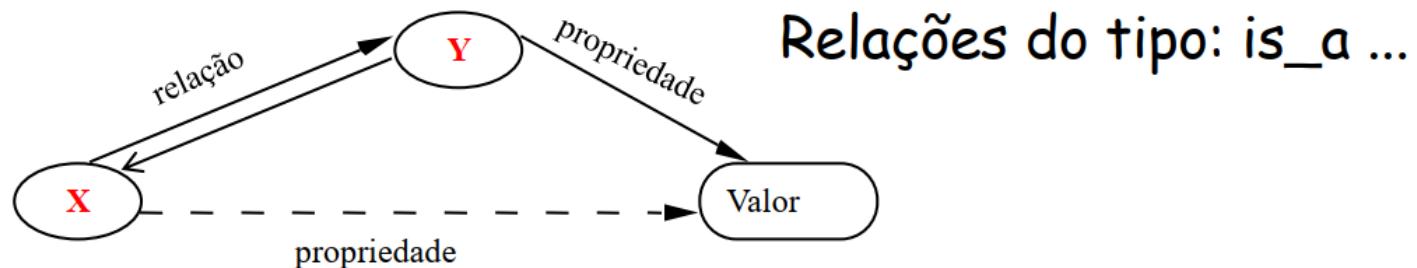
Representação do Conhecimento

- Acesso ao Conhecimento:
getframe (Nome,[A-V|R]).
addframe (Nome,[A-V|R]).
delframe (Nome,[A-V|R]).
- **USO DAS FRAMES**
Vantagem é um natural uso de herança e estruturação do Conhecimento
- Exemplo: Regras sobre identificação de pássaros
info: ordem tubenose, família albatroz, espécie patapreta
frame(tubenose, [nível-[val, ordem], narinas-[val, tubular], habitat-[val, mar], garra-[val, curva]]).
- **frame(albatroz, [tipo-[val, tubenose], nível-[val, familia], tamanho [val, grande], asas-[val, compridas_estreitas]]).**
- **frame(albatroz_vulgar, [tipo-[val, albatroz], nível-[val, especie], cor-[val, branco]]).**
- **frame(albatroz_pata_preta, [tipo-[val, albatroz], nível-[val, especie], cor-[val, preto]])**

Representação do Conhecimento

MECANISMOS DE HERANÇA

Existindo **relacionamento** entre objetos e **propriedades** que podem ser herdadas, pode poupar-se na representação explícita, incluído relações de **herança**:



`propriedade(X,Valor):- relação(X,Y), propriedade(Y,Valor).`

`propriedade(X,Valor):- relação(Y,X), propriedade(Y,Valor).`

Suponhámos que em um sistema de conhecimentos sobre animais sabíamos que as propriedades:

"respiração" e "cobertura" eram herdadas pelos sub-grupos taxonómicos

Representação do Conhecimento

Relação entre objetos que permite **herança**: "é-um"

é-um(truta, peixe).

é-um(leão, mamífero).

é-um(salmão, peixe).

.....

respiração(peixe, guelra).

cobertura(peixe, escamas).

respiração(mamífero, pulmões).

cobertura(mamífero, pelos).

Sendo as propriedades **respiração** e **cobertura** herdáveis:

respiração(**X**,Valor):- é-um(X,Y), respiração(Y,Valor).

cobertura(**X**,Valor):- é-um(X,Y), cobertura(X,Valor).

Devem acrescentar-se as exceções.

Representação do Conhecimento

- O programa de manipulação das “frames” deveria ser tal que permitisse obter valores dadas as seguintes perguntas:
 - ?- getframe (**X**, [cor-preto, asas-compridas_estreitas]).
X= albatroz_pata_preta
 - ?- getframe (**X**, [asas-compridas_estreitas]).
X= albatroz;
X= albatroz_pata_preta;
X= albatroz_vulgar;

Representação do Conhecimento

- Exemplo de Sistema da NASA incluindo vários tipos de Representação de Conhecimento:
 - Regras (“Produções”)
 - Redes Associativas (“Semantic Networks”)
 - Enquadramentos (“Frames”)
- Ex. de Regra (“Produção”):
IF the shuttle power_supply fails **AND** a working power_supply is available **AND** the situation causing the problem no longer exists
THEN switch to the backup power_supply

Representação do Conhecimento

- Ex. de Frame - Airplane Frame:

Type: IFO

range: (fighter, transport, trainer, bomber, light plane, observation)

Manufacturer:

range: (McDonnell-Douglas, Boeing, Airbus,...)

Empty Weight:

range: (500 lbs to 250.000 lbs)

Gross Weight:

range:(500 lbs to 500.000 lbs)

if needed ($1.6 * \text{empty weight}$)

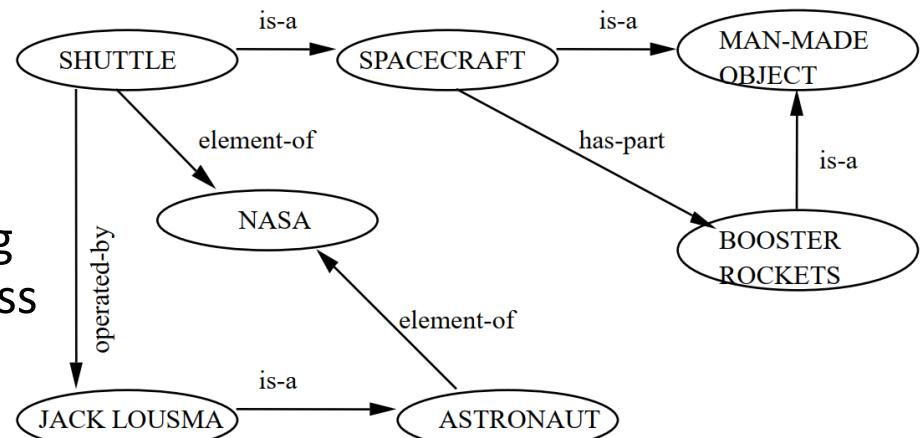
Max Cruising range:

If needed: (lookup in table cruising range appropriate to type and gross weight)

Number of Cockpit Crew:

range: (1 to 3)

default: 2



Sistemas Baseados em Conhecimento

- **Sistemas Baseados em Conhecimento**
 - Programas de computador que utilizam o **conhecimento representado explicitamente** para resolver problemas.
 - Manipulam conhecimento e informação de forma inteligente.
 - São desenvolvidos para resolverem problemas que requerem grandes porções de conhecimento humano e especialização (perícia).
- **CONHECIMENTO + RACIOCÍNIO = RESOLUÇÃO PROBLEMA**

Dados, Conhecimento e Informação

- **DADOS – Elemento puro, quantificável sobre um determinado evento**
 - Ex: Temperatura ambiente (em graus Celsius).
 - Idade de um paciente (em anos).
- **INFORMAÇÃO – Dado analisado (interpretado) e contextualizado.**
 - Ex: A melhor temperatura para os seres humanos é aquela que não dista mais do que 1 grau dos 25 graus
 - É-se adolescente entre os 12 e os 19 anos.
- **CONHECIMENTO – Capacidade de criar um modelo mental que descreva o Objeto e indique as ações a implementar, i.e., as DECISÕES A TOMAR**
 - Ex: Se estiver frio (abaixo dos 15 graus) levar casaco, caso contrário existe a possibilidade de ficar constipado
 - Se for um paciente adolescente de psiquiatria, é necessário ter cuidados especiais com a administração de antidepressivos, caso contrário o paciente pode ter tendências suicidas.

Níveis de Conhecimento

- **FATOS**
 - Relações arbitrárias entre objetos, símbolos, eventos, ...
- **CONCEITOS**
 - Resultam de ideias abstratas, de natureza hierárquica
- **REGRAS**
 - Conjuntos de operações e passos que orientam a ação, desenvolvidas a partir de fatos e conceitos
 - Correspondem à aplicação do conhecimento
- **METAREGRAS**
 - Responsável pela criação e aplicação de novas regras a situações novas
 - Corresponde à geração de conhecimento novo
- ***UMA DECISÃO É OBTIDA PELO USO EXPLÍCITO DO CONHECIMENTO***

Sistemas Baseados em Conhecimento

- Um Exemplo...
 - Computador: Qual o restaurante onde desejas jantar?
 - Humano: Não [Estou indeciso...]
 - Computador: Gostas de alguma comida em particular?
 - Humano: Não [De fato, não sei o que me apetece...]
 - Computador: Gostas de comida picante?
 - Humano: Não [Da última vez, aquilo foi complicado...]
 - Computador: Costumas beber álcool às refeições?
 - Humano: Sim [Vá lá o computador está a acertar...]
 - Computador: Quando bebes vinho, aprecias vinho tinto alentejano?
 - Humano: Sim [É isso mesmo, continua, ajuda-me nisto...]
 - ...
 - Legenda: [Pensamentos...]

SBCs: Exemplos

- **Exemplos...**
 - <http://www.aiinc.ca/demos/index.html>
 - <http://www.myacquire.com/aiinc/spa/>
 - <http://www.myacquire.com/aiinc/stock/>
 - Do you want information about a hot tub or a pool?
 - hot tub.
 - Do you want advice about correcting symptoms you've observed with your hot tub, or advice about adjusting the chemical balance based on the readings of your test strips?
 - troubleshoot symptoms.
 - Check all the symptoms you've observed with your hot tub. You will be asked one or more questions to isolate the cause for each symptom you check. A report will be presented at the end which you can print using the print icon in your browser's toolbar.
 - strange odor and colored water.
 - A strange, musty smell may be due to low sanitizer level. Test the chemical levels to make sure they are in balance and adjust accordingly.
 - Did this eliminate the odor?
 - ...

SBC: Definição

DEFINIÇÃO:

Programas de computador que utilizam o conhecimento, representado explicitamente, para resolver problemas.

- Manipulam conhecimento e informação de forma inteligente
 - São desenvolvidos para resolverem problemas que requerem grandes porções de conhecimento humano e especialização (perícia)
-
- **CONHECIMENTO + RACIOCÍNIO = RESOLUÇÃO PROBLEMA**

SBC: Definição

- **Perspectiva do Conhecimento processável pelo homem**
 - A análise e modelação do método de resolução do problema
- **Perspectiva Simbólica processável pelo computador**
 - A atividade de representar este método através de um formalismo computacionalmente eficiente
- **Capacidade de Raciocínio/Inferência**
 - É a capacidade de definir um conjunto de passos para a resolução eficiente e rápida de um problema
 - O próprio mecanismo de inferência é conhecimento

SBC: Capacidades Intrínsecas

- **CAPACIDADES INTRÍNSECAS**
 - Questionar o utilizador numa linguagem/interface fácil de entender e adquirir informações
 - Estabelecer um raciocínio a partir dessas informações e do conhecimento nele contido
 - Explicar o processo de raciocínio utilizado
 - Admitir o erro (como nos humanos) mas a solução apresentada deverá ser equivalente à apresentada por um especialista humano

SBC: Propriedades

- **PROPRIEDADES**

- Tudo o que se sabe do problema deverá estar explicitamente representado na Base de Conhecimento do Sistema
- Saber interpretar e utilizar o conhecimento contido na Base de Conhecimento → Mecanismo / agente de Inferência
- Tipicamente estes sistemas destinam-se a resolver problemas complexos
- Uso intensivo do conhecimento específico do domínio (em vez do geral)

Diferenças para outros Sistemas

Sistemas Convencionais	Sistemas Baseados em Conhecimento
Estrutura de Dados	Representação de Conhecimento
Dados e Relações entre os Dados	Conceitos, Relações entre Conceitos e Regras
Usam Algoritmos Determinísticos	Pesquisa Heurística
Conhecimento embebido no código do programa	Conhecimento representado explicitamente e separado do programa que o manipula e interpreta
Explicação do raciocínio é difícil	Podem e devem explicar o seu raciocínio

Sistemas Inteligentes, SBCs e SPs



Justificação para Desenvolver SBCs

- **JUSTIFICAÇÃO DO DESENVOLVIMENTO DE SBCs**
 - Vantagem económica - altamente remunerativo
 - Falta de especialistas afetam a eficiência na resolução do problema
 - Especialistas necessários em vários locais (e.g., médicos de uma especialidade) – problema distribuído
 - Domínio bem delimitado – para que o SBC possa resolver bem o problema
 - Necessidade de retenção de conhecimento – evita dependência excessiva dos humanos (e.g., perda de funcionários)

Tipos de Aplicações

- INTERPRETAÇÃO - Análise de dados para determinar o seu significado
- CLASSIFICAÇÃO – Atribuir uma etiqueta a um dado item, definido através de um conjunto de atributos, uma etiqueta, de entre um conjunto de classes pré-definidas
- MONITORIZAÇÃO - Observação contínua de um sistema para agir quando uma situação acontece
- PLANEAMENTO - Determinar conjunto de ações para atingir uma determinada meta
- PROJECTO - Especificações de um objeto de modo a obedecer a um conjunto de requisitos

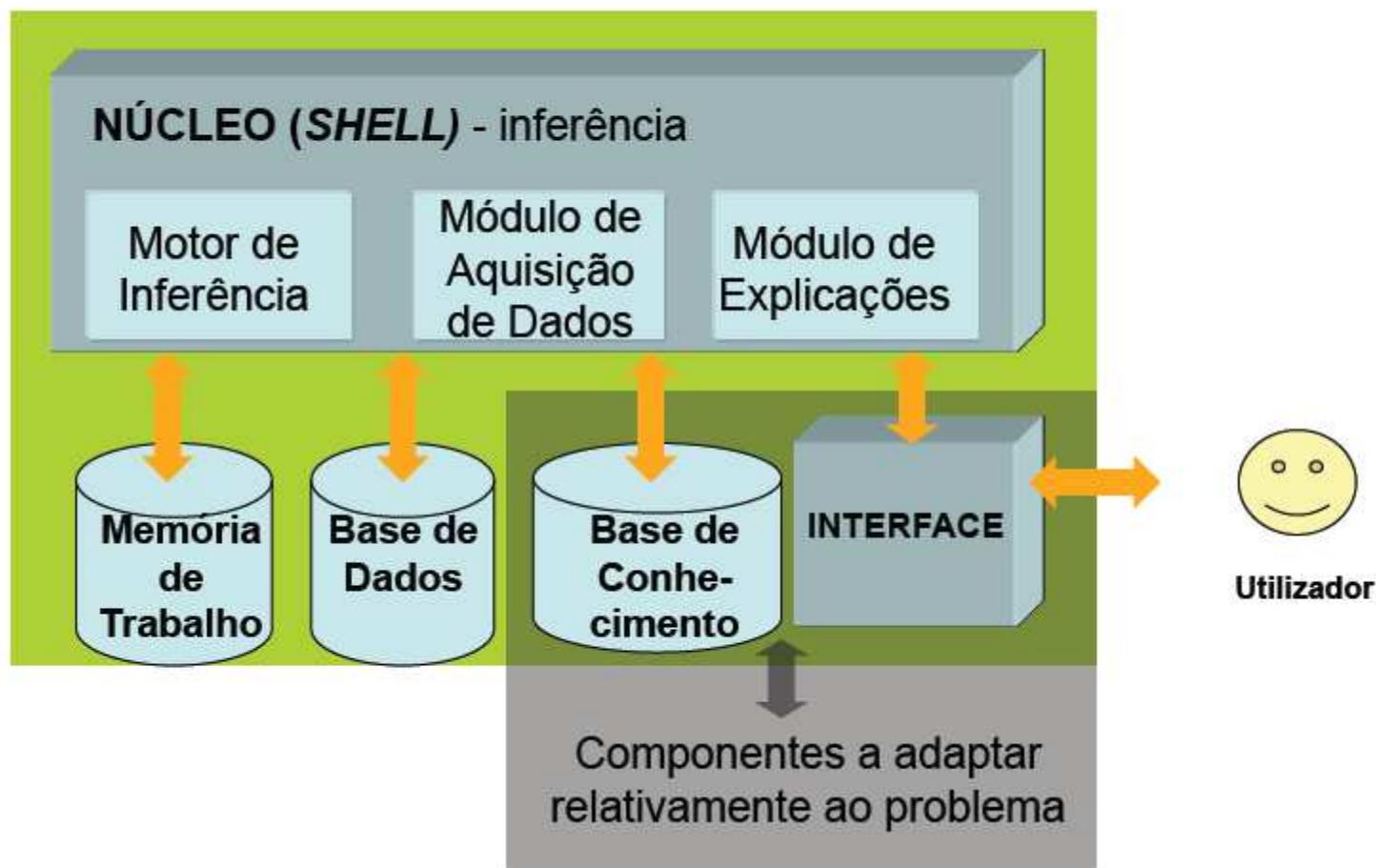
Exercício

Qual o Tipo das seguintes aplicações?

- Interpretação/Classificação/Monitorização/ Planeamento/ Projeto
 - 1. Controlo de tráfego aéreo
 - 2. Guiar um robô de um labirinto
 - 3. Análise de dados de uma imagem de satélite
 - 4. Determinar qual a falha num automóvel a partir de uma avaria
 - 5. Observação contínua duma central nuclear
 - 6. Conceber a estrutura de uma ponte ferroviária
 - 7. Diagnóstico de uma gripe
 - 8. Reconhecimento de fala

Estrutura de um SBC

ESTRUTURA GERAL DE UM SBC

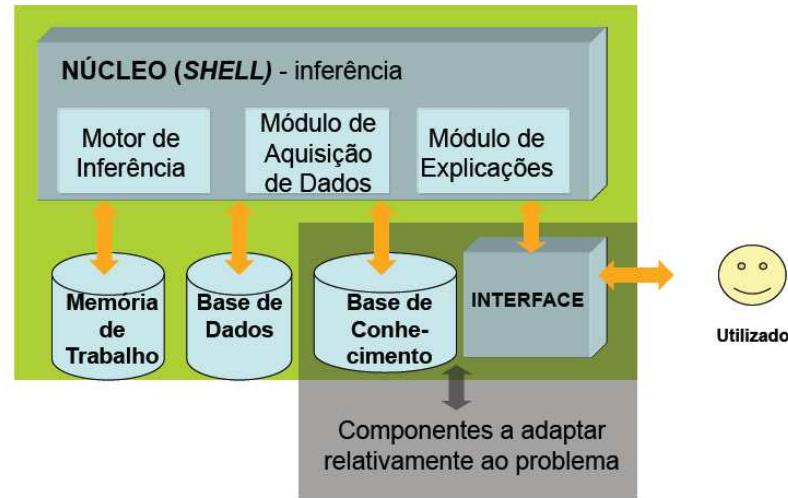


Estrutura de um SBC

- **Núcleo (Shell)**

- Controlo da **interação** com o utilizador
- **Inferência** do conhecimento
- **Explicação** das conclusões

ESTRUTURA GERAL DE UM SBC



- **MOTOR/SISTEMA DE INFERÊNCIA (MI)**

- Desenvolvimento do raciocínio baseado nas informações obtidas pelo MAD e no conhecimento representado na BC.
- Por exemplo, para regras de produção:
- Encadeamento para a frente (*forward chaining*)
- Encadeamento para trás (*backward chaining*)

Estrutura de um SBC

- **MÓDULO DE AQUISIÇÃO DE DADOS (MAD)**

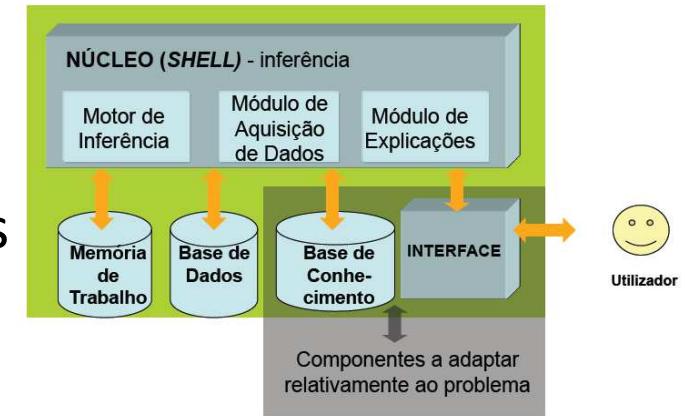
- Interação com o utilizador
- Obtenção de informações sobre o problema (perguntas ao utilizador)
- Verificação da validade das respostas

- **MÓDULO DE EXPLICAÇÃO (ME)**

Justificação das conclusões obtidas:

- Porquê – porque é que o MAD fez a pergunta ao utilizador
- Como – caminho de raciocínio para chegar às conclusões
- Estudo de cenários - O que acontece se alguma informação fornecida pelo utilizador for alterada (*what if*)
- Porque não – explicar porque uma determinada conclusão não foi obtida

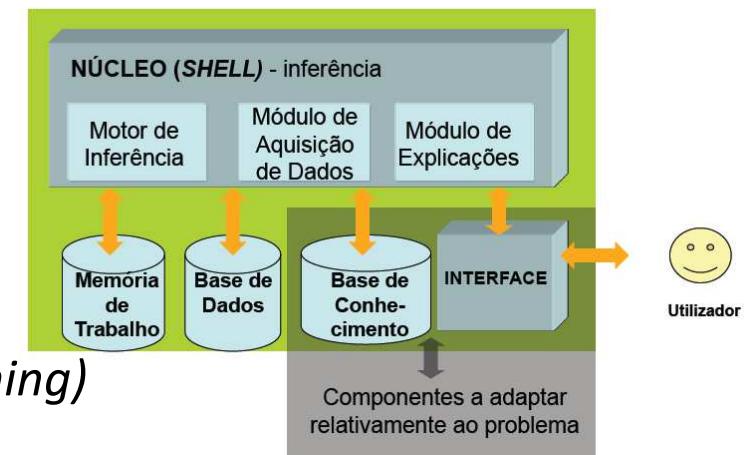
ESTRUTURA GERAL DE UM SBC



Base Conhecimento

- **BASE DE CONHECIMENTO (BC)**
 - Descrição do CONHECIMENTO necessário para a resolução do problema
 - Conjunto de representações de ações e acontecimentos do mundo
 - SENTENÇAS expressas numa determinada LINGUAGEM DE REPRESENTAÇÃO DE CONHECIMENTO:
 - Regras de Produção
 - Redes Semânticas
 - *Frames (Enquadramentos)*
 - Orientado ao objecto
 - Lógica
 - Baseado em Casos (*Case Based Learning*)
 - Híbridas

ESTRUTURA GERAL DE UM SBC



Base Conhecimento

Exemplo de uma frase do tipo CAUSA-EFEITO (regra de produção)

- SE TEMP-PACIENTE > 37.5 ° C ENTÃO PACIENTE-TEM-FEBRE

Exemplo de META-CONHECIMENTO – conduz a procura da solução:

- SE O DOENTE É ALCOÓLICO ENTÃO PROCURAR PRIMEIRO DOENÇAS HEPÁTICAS
- PROCURAR A SOLUÇÃO PRIMEIRO EM CAMINHOS ONDE EXISTEM POUCAS POSSIBILIDADES (heurística)

Atender a problemas de:

- **CONFLITOS / INCONSISTÊNCIA**
 - Teste de consistência
 - Selecção de soluções em conflito
- **INCOMPLETITUDE / INCERTEZA**
 - Métodos Bayesianos (probabilísticos)
 - Teoria da Evidência
 - Teoria da Certeza
 - Lógica Difusa (*Fuzzy Logic*)

Estrutura de um SBC

- **BASE DE DADOS (BD)**
 - Destina-se a conter os dados/informações que caracterizam o problema (fatos)
- **MEMÓRIA DE TRABALHO (MT)**
 - Permite armazenar e fornecer a linha de raciocínio
 - Armazena respostas do utilizador (evita perguntas repetidas)
 - Armazenamento de conclusões intermédias (evita repetição de inferências)

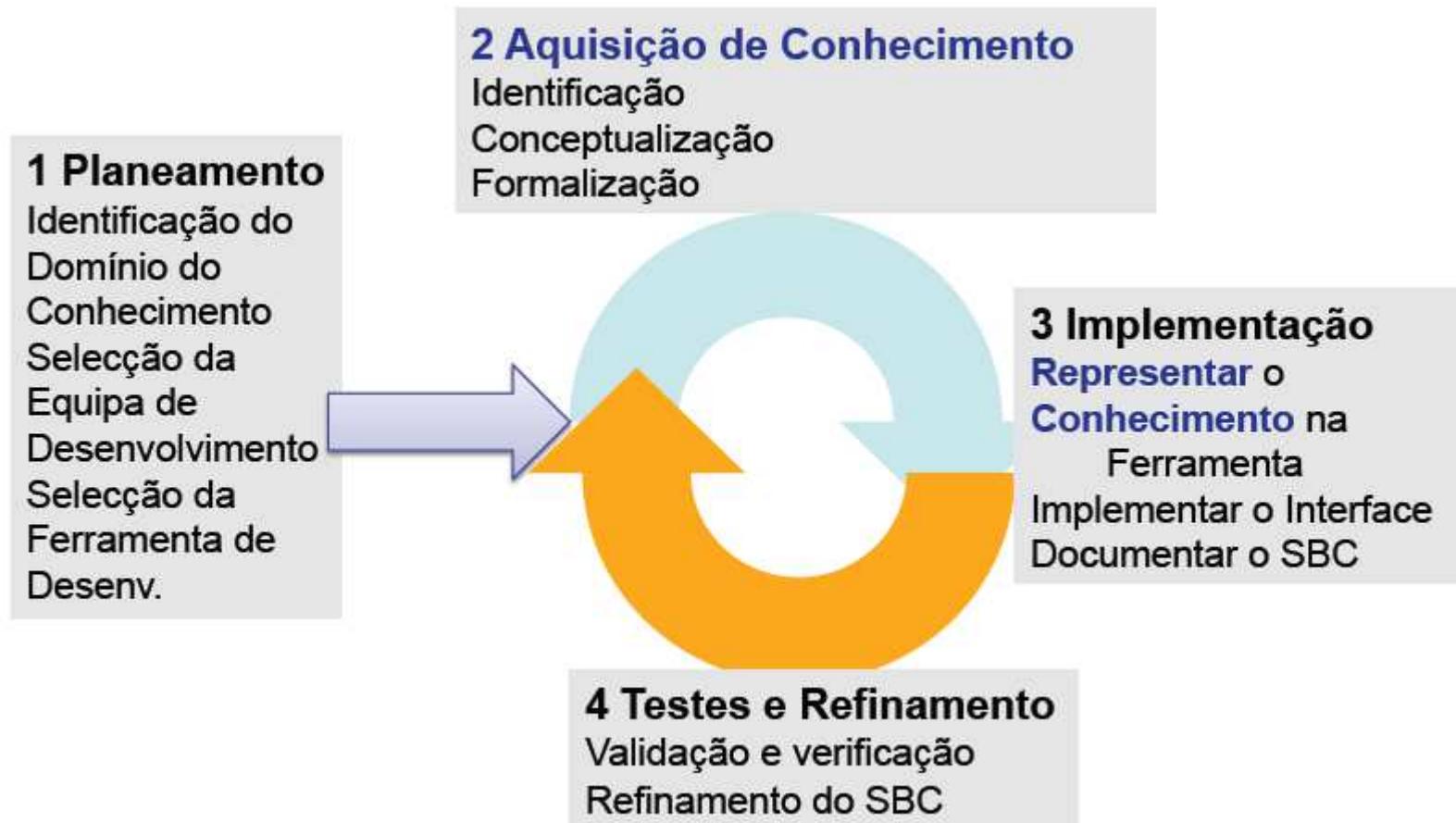
SBC: Interface

INTERFACE

- **Interação entre o SBC e o utilizador**
- **Linguagem difere da utilizada para representar o conhecimento:**
 - Linguagem natural (impraticável)
 - Subconjunto de Linguagem Natural
 - Linguagens Visuais
 - Linguagens Diagramáticas
 - Multimédia
- **Princípios oriundos das teorias cognitivas e semióticas (Human-Computer Interaction - HCI):**
 - Eficiência
 - Dinamismo
 - Desenvolvimento em tempo útil

Metodologia de Desenvolvimento

Metodologia/Processo de Desenvolvimento de um SBC (iterativo):
4 Fases



Ferramentas de Suporte

FERRAMENTAS DE SUPORTE À CONSTRUÇÃO DE UM SBC

- Utilização de linguagens de programação como LISP e PROLOG
- Ferramentas de Apoio – diversos esquemas de representação de conhecimento, motores de inferência, interfaces, etc.
 - ART, Babylon, KEE, Knowledge Craft, Loops, Flex, Elements Environment, ...
- Shells - a interface e estratégia de resolução de problemas é prédefinida:
 - Insight, KES, MED2, M.1, Personal Consultant, S.1, Timm
 - EXSYS CORVID, CLIPS, JESS (Clips em Java), Jlisa (Clips for Java),

Características de Ferramentas

Características de Algumas FERRAMENTAS DE APOIO

	ART	Babylon	KEE	Knowledge Craft	Loops	Flex	Elements Environment
Representação do Conhecimento							
Frames	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Orientado ao Objecto	Lim	Sim	Sim	Sim	Sim	Sim	Sim
Demons	Lim	Sim	Sim	Sim	Sim	Sim	Sim
Regras	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Motor de Inferência							
Forward Chaining	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Backward Chaining	Sim	Sim	Sim	Não	Não	Sim	Sim
Contexto (mundos)	Sim	Não	Sim	Lim.	Não	Não	Não
Manutenção de Verdade	Sim	Não	Sim	Não	Não	Não	Não
Ajuda ao Desenvolvimento							
Editor Interno	Sim	Não	Sim	Sim	Sim	Sim	Sim
Help	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Trace e break point	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Medida de Desempenho	Sim	Não	Não	Não	Sim	Sim	Não
Interface Gráfica	Sim	Limitado	Sim	Sim	Sim	Sim	Sim
Referências Cruzadas	Não	Não	Não	Não	Sim	Não	Sim
Controlo de Versões	Não	Não	Não	Não	Sim	Não	Não
Fornecedor	Inference	GMD	Intelli corp	Carnegie Group	Xerox	LPA Prolog	Neuron Data

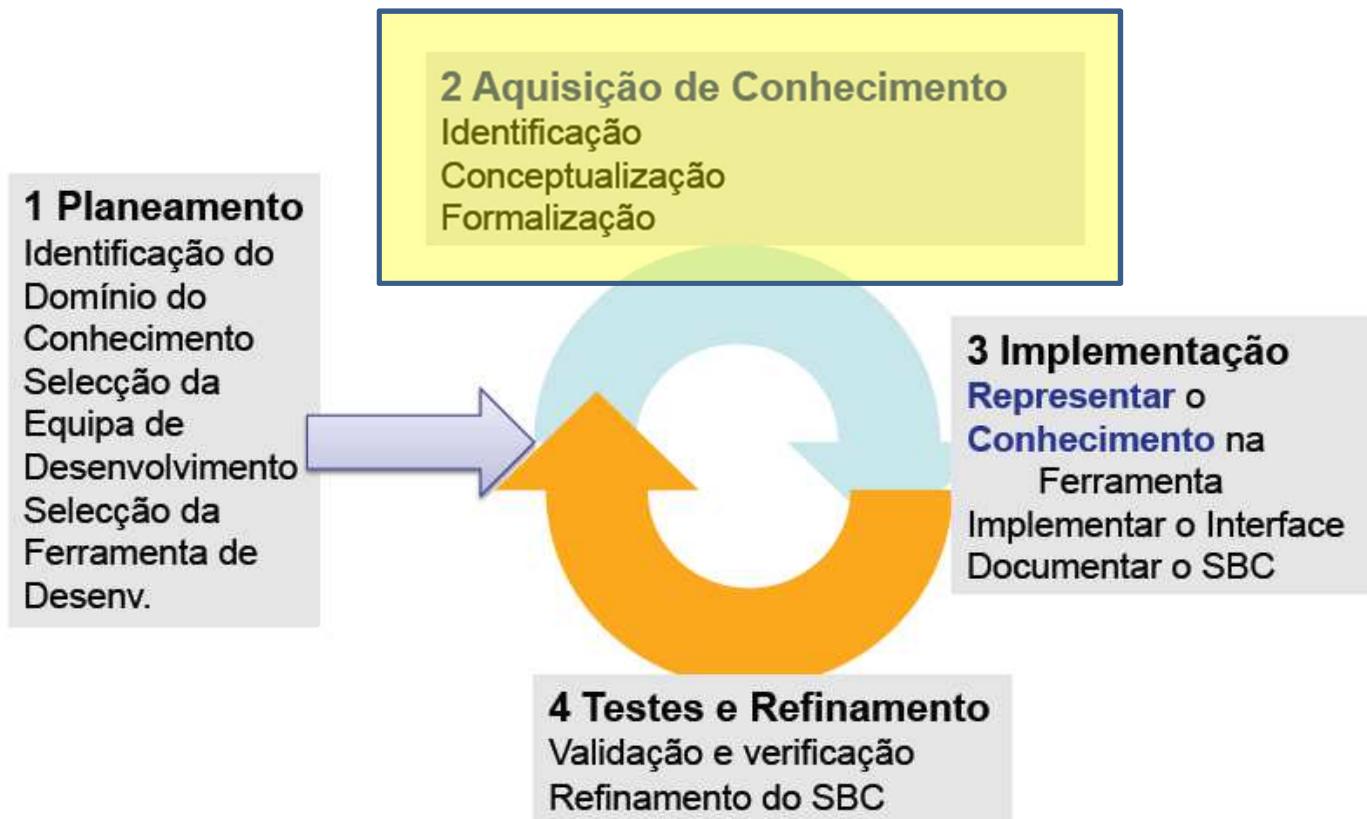
Características de Shells

Características de Algumas SHELLS

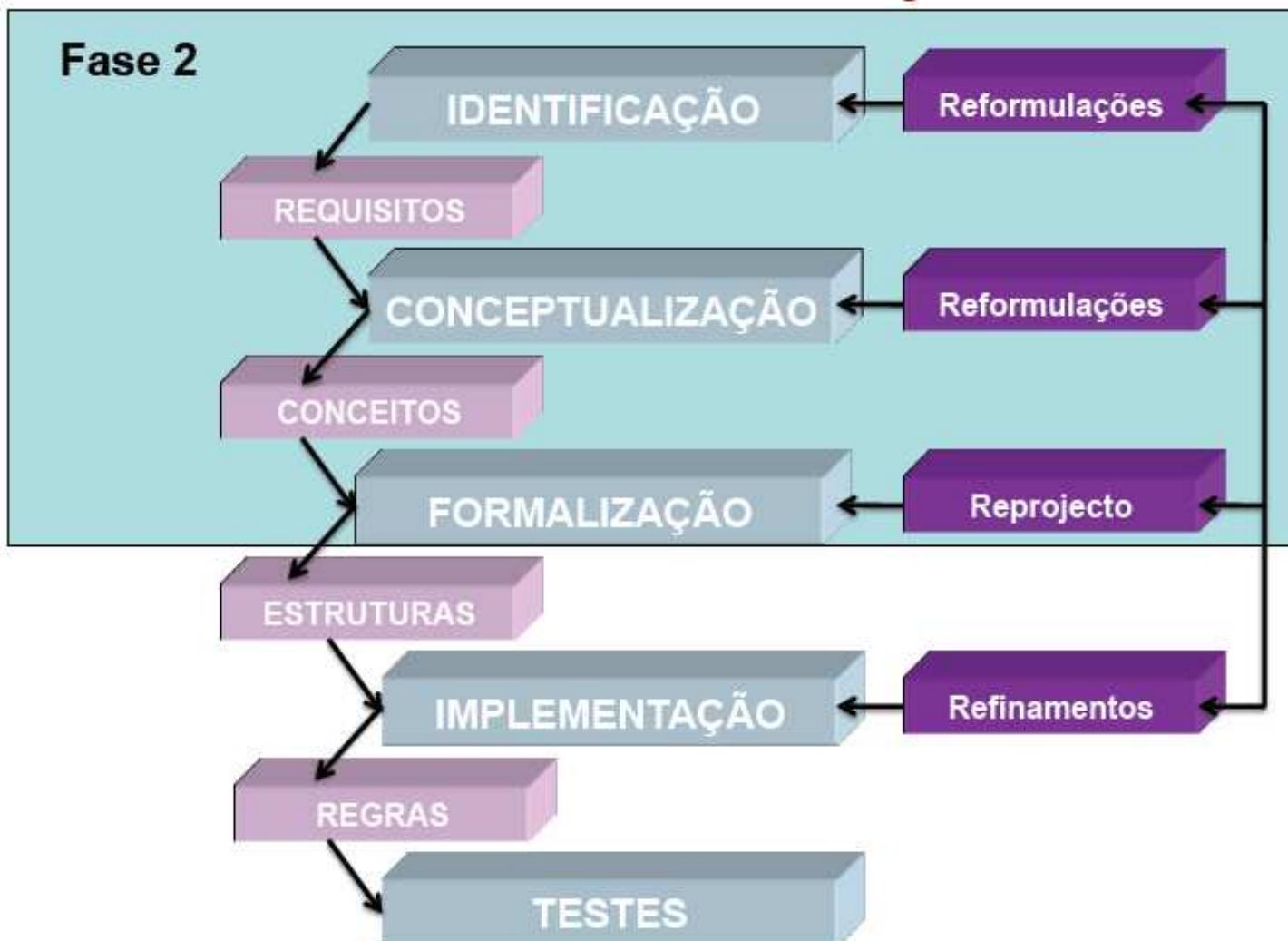
	Insight	KES	MED2	M.1	CLIPS	Personal Consultant	S.1	Timm
Classe do Problema								
Interpretação	Lim	Lim	Lim	Lim	Sim	Lim	Sim	Lim
Diagnóstico	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Monitorização	Não	Não	Lim	Lim	Sim	Não	Não	Sim
Planeamento	Não	Não	Não	Lim	Sim	Não	Não	Não
Projecto	Não	Não	Não	Lim	Lim.	Não	Lim	Não
Representação BC								
Frames	Não	Não	Não	Não	Sim	Sim	Lim	Não
Demons	Não	Não	Não	Sim	Sim	Sim	Sim	Não
Regras	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Motor de Inferência								
Forward Chaining	Sim	Não	Sim	Lim	Sim	Lim	Lim	Sim
Backward Chaining	Sim	Sim	Lim	Sim	Não	Sim	Sim	Não
Apoio ao Desenvolvimento								
Editor da BC	Sim	Não	Lim	Sim	Sim	Sim	Sim	Não
Verif. Consistência	Não	Não	Não	Sim	Não	Não	Sim	Sim
Trace	Sim	Sim	Sim	Não	Sim	Sim	Sim	Sim
Explicação	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Fornecedor	Level-5-r	Soft ware A&E	Inware	Tec kno wle dge	Dom. Públic o	T1	Teckn owledge	GRC

Aquisição de Conhecimento

- Fase 2 da metodologia de desenvolvimento de um SBC



Etapas do Processo de Aquisição



Identificação

Objetivo

- Assemelha-se à Análise de Requisitos em Engenharia de Software
- Procura elementos do domínio que permitam identificar:
 - 1. A classe do problema que o SBC deverá resolver
 - 2. Dados sobre os quais irá operar
 - 3. Critérios para classificar as soluções nos contextos de funcionamento
 - 4. Forma como o problema deve ser resolvido

Metodologia:

- Solicitar bibliografia sobre o domínio aos especialistas (métodos para resolver os problemas)
- Entrevistas não-estruturadas (*brainstorming*) com os especialistas para apurar: Necessidades; Complexidade da tarefa; Terminologia utilizada; e Capacidade de cada especialista em tornar explícito o conhecimento do domínio.
- Entrevistas com utilizadores para formular o modelo de interacção utilizador-sistema



Conceptualização

Objectivo

- Formulação dos conceitos importantes do problema e as relações (causa-efeito, espaço-tempo, ...) entre esses conceitos
- Elaborar uma ONTOLOGIA de forma informal, i.e. Conjunto de termos que designam conceitos e relações, juntamente com as suas definições



Metodologia

- Entrevistas estruturadas (com um guião, perguntas já definidas) a partir do material recolhido anteriormente
- Selecionar casos concretos para modelação e teste do SBC
- Observar o especialista no trabalho

Formalização

- **Objetivo**

- Modelização computacional do problema
- Formalização da ONTOLOGIA através de uma linguagem formal como a LÓGICA de primeira ordem e suas extensões, regras de produção, frames, redes semânticas, ...

- **Metodologia**

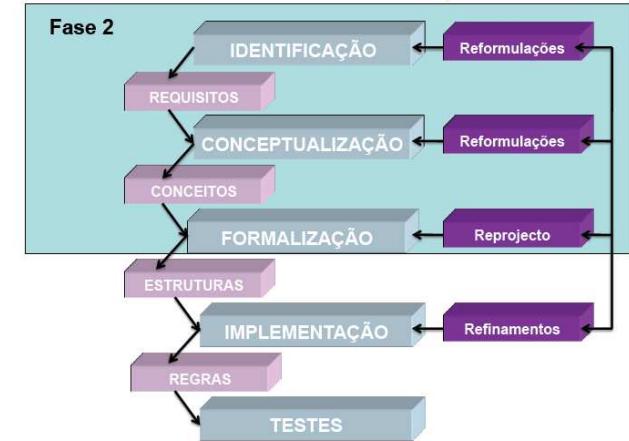
- Definir o modelo de tarefa a ser adotado
- Escolher a linguagem de representação para modelar o sistema
- Definir o espaço de procura do problema
- Definir o espaço de soluções do problema
- Definir os métodos de procura de soluções
- Identificar as limitações do sistema



Implementação

Objectivo

- **Modelação computacional do problema**
- **Metodologia**
 - Seleção de linguagem de programação (e.g., LISP, PROLOG), ferramenta (e.g., ART, ProKappa) ou Shell (e.g. Corvid, Clips)
 - Programação (por programadores)
 - Prototipagem
 - Validação pelo(s) especialista(s)



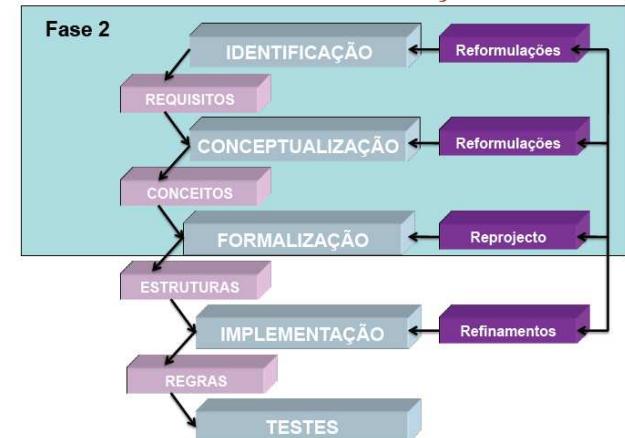
Testes e Refinamento

Objetivo

- Submeter o sistema a um conjunto representativo de casos de teste.
- Manutenção da Base de Conhecimento

Metodologia

- Utilizar o SBC no ambiente real
- Permitir a experimentação durante um período de tempo



Linguagens de Aquisição de Conhecimento

Linguagem Natural

- Fase de IDENTIFICAÇÃO
- Facilidade de comunicação e registo
- Ambiguidade de interpretação (semântica)
- Grande número de vocábulos (combinação explosiva)

Linguagens de Aquisição de Conhecimento

Linguagens Diagramáticas (Fase de CONCEPTUALIZAÇÃO)

- Gestos, imagens, figuras, esquemas e diagramas
- Exemplo de relações do mundo:
 - Composição (como os objectos são compostos)
 - Descrição (descrição de uma entidade e.g., pacote de e-mail)
 - Consequência (causa efeito, e.g., regra)
 - Generalização (relação isa, classes vs subclasses)
 - Fluxo de estados do Mundo (caminho/plano através de estados)
 - Dependência (variáveis dependentes e independentes numa expressão)



Linguagens de Aquisição de Conhecimento

Linguagens Semiformais (Fase de FORMALIZAÇÃO)

Combinação de notações formais com representação informal como a linguagem natural: HTML, XML

Linguagens Formais (Fase de FORMALIZAÇÃO)

Expressam conhecimento de uma forma precisa, consistente e nãoambígua:

- **Lógica**
 - Conjunto de fórmulas (asserções) em lógica de predicados, difusa e modal.
Os mecanismos de inferência correspondem aos métodos dedutivos dos sistemas lógicos.
- **Sistemas de Produção**
 - Regras do tipo SE condição ENTÃO ação.
 - Inferência por encadeamento para a frente (*forward chaining*) ou para trás (*backward chaining*).
- **Estruturados**
 - Grafos onde os nodos e arcos possuem semântica fixa (*frames e objectos*) ou variada (*redes semânticas*).

Técnicas para Aquisição de Conhecimento

Manuais:

- **Baseadas em Descrições**
 - Análise de documentos, textos e referências bibliográficas
- **Baseadas em Entrevistas**
 - Não-Estruturadas
 - Estruturadas
 - Acompanhamento de Casos (para preenchimento de lacunas)
- **Baseadas em Acompanhamento:**
 - Acompanhar o processo de raciocínio do especialista
- **Baseadas em Modelos:**
 - formulação de um modelo geral do conhecimento baseada em descrições estruturadas/componentes reutilizáveis (PROTÉGÉ, KADS)

Técnicas para Aquisição de Conhecimento

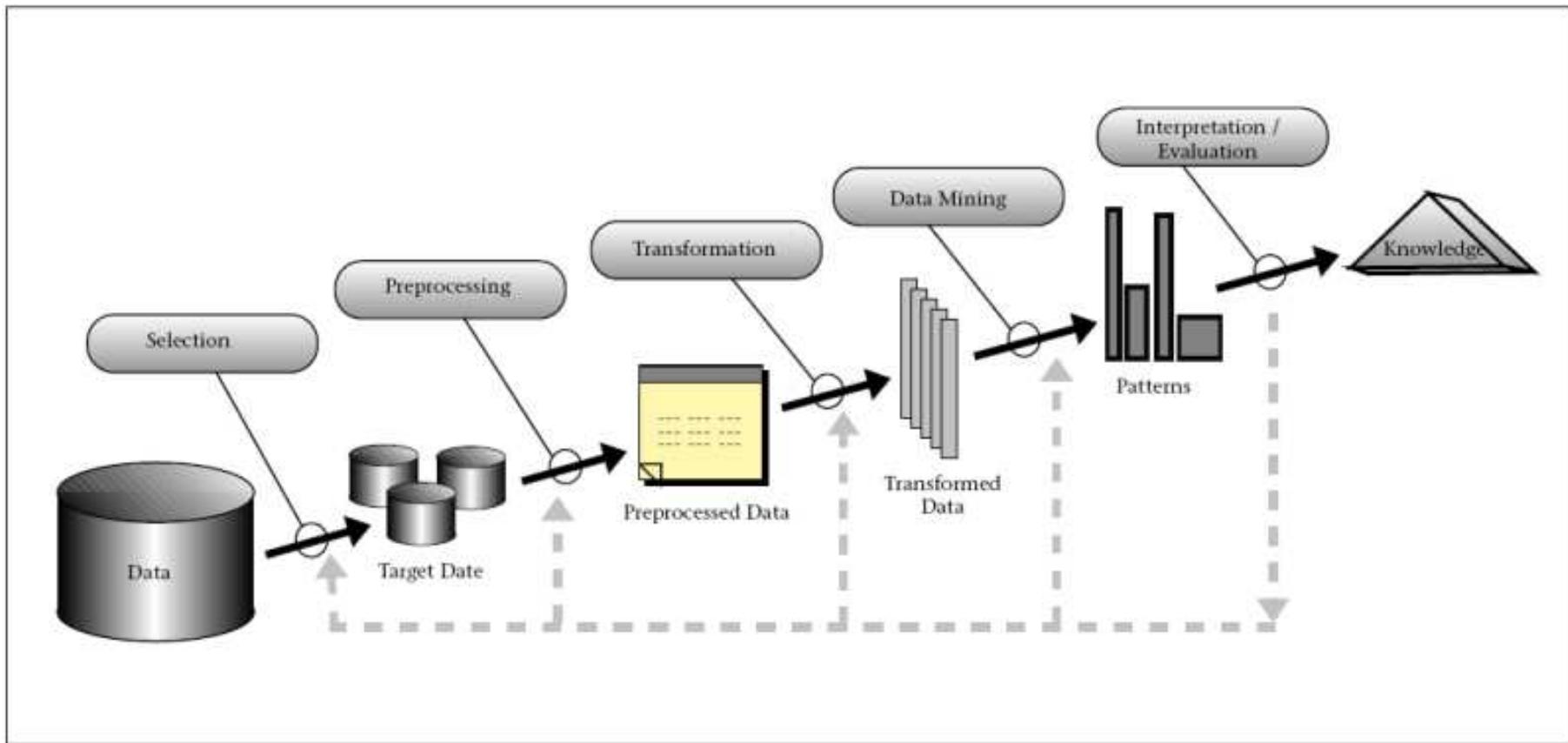
Semi-Automáticas:

- **Baseadas na Reutilização da Representação e dos Mecanismos de inferência**
 - Uso de representações existentes noutros domínios (e.g., EMYCIN – diagnóstico médico)
- **Baseadas na Reutilização do Conhecimento do Domínio**
 - Desenvolvidas para domínios especificados (e.g., OPAL para planear terapia para o cancro) aliviam a carga de aquisição de todo conhecimento (possui já conhecimento sobre medicamentos e química do sangue)
- **Baseadas em Técnicas Oriundas da Psicologia**
 - Análise de Matrizes de Relatórios, baseado na “teoria de construção pessoal” (cada pessoa é vista como um cientista que constrói as sua teorias)
- **Baseadas em Ontologias Reutilizáveis**
 - Linguagens para ontologias e.g., Ontolingua, OCML.
- ...

Técnicas para Aquisição de Conhecimento

Automáticas

- Baseadas na Aprendizagem Automática e *Data Mining*
 - Conhecimento é extraído a partir de dados em bruto!



Regras de Produção

- **Aplicar Regras de Produção para Representar Conhecimento:**
 - Factos e regras,
 - Sistemas de inferência: backward e forward chaining
- Linguagem do tipo: se ... então ...
- Trata-se do formalismo mais popular para representar conhecimento (sistemas periciais)
- Áreas de aplicação: diagnóstico de doenças, detecção de avarias em automóveis, aconselhamento em compra de uma habitação, aconselhamento de serviços de gás/eletricidade, ...

Exemplo:

se o paciente tem febre **e** o paciente tem dores **e**

o paciente não tem infecções detectáveis

então *diagnosticar gripe*

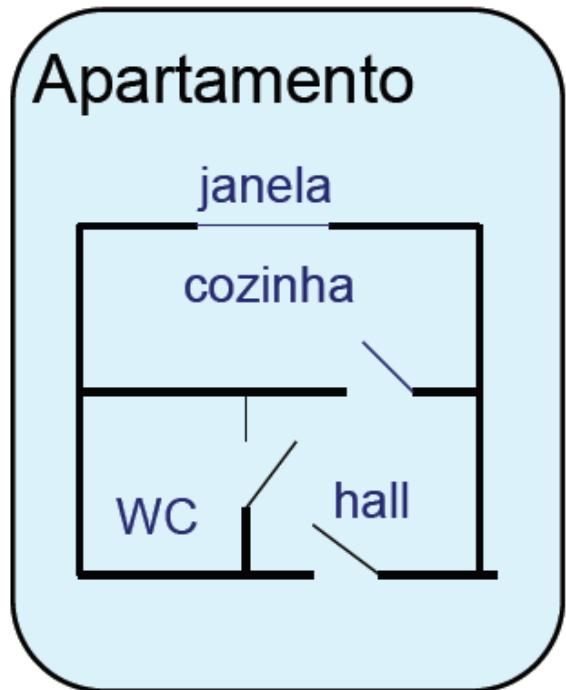
Vantagens das Regras de Produção

- As Regras de Produção oferecem uma forma natural de expressar conhecimento, reunindo vantagens como:
 - Modularidade - cada regra define uma parte do conhecimento, sendo independente
 - Incrementabilidade - novas regras podem ser acrescentadas em qualquer momento
 - Alterabilidade - as regras podem ser alteradas a qualquer momento
 - Independência do sistema de inferência utilizado
 - Facilidade em gerar explicações para uma dada resposta:
 - Como se chegou a uma dada conclusão? (questões como)
 - Porquê é que estamos interessados nesta informação? (questões porquê)

Utilização de Regras de Produção

- A Base de Conhecimento é feita de regras e fatos;
- Considera-se a seguinte sintaxe lógica:
 - se Condição então Conclusão
- Sendo que uma Condição pode ser:
 - um predicado lógico (e.g. Prolog)
 - uma conjunção de duas condições: Cond1 e Cond2
 - uma disjunção de duas condições: Cond1 ou Cond2
- Representar factos (dados): facto(D)
 - D é algo que atualmente é verdadeiro

Caso de Estudo: Diagnóstico de fugas de água num apartamento



Quais as regras e fatos para um SBC de diagnóstico de fugas do apartamento? O que se pode deduzir?

Regras adquiridas via o perito em fugas de água:

- Quando o hall está molhado e a cozinha seca então existe uma fuga na casa de banho
- Por outro lado, quando o hall está molhado mas a casa de banho seca então o problema é na cozinha (existe fuga na cozinha)
- Quando a janela está fechada ou não chove então não entra água pelo exterior
- Se o hall estiver seco e casa de banho molhada então o problema é a torneira aberta
- Existe uma inundação quando há uma fuga na cozinha e a não entra água do exterior

Situação atual:

- o hall está molhado, a casa de banho seca e a janela fechada
 - facto(hall_molhado). facto(banho_seco).
 - facto(janela_fechada).

Representação com Regras

% Regras: base de conhecimento

```
Se hall_molhado e cozinha_seco entao fuga_banho.  
se hall_molhado e banho_seco entao fuga_cozinha.  
se janela_fechada ou nao_chove entao nao_agua.  
se hall_seco e banho_molhado entao torneira_aberta.  
se fuga_cozinha e janela_fechada entao inundacao.
```

% Factos: base de dados, situação actual

```
facto(hall_molhado).  
facto(banho_seco).  
facto(janela_fechada).
```

Sistema de Inferência

- Existem dois sistemas principais de inferência (formas de raciocínio):
 - **Backward chaining:**
 - a partir de uma hipótese (uma questão) o raciocínio retrocede na rede de inferência até aos fatos
 - ou seja, tentamos chegar das conclusões às condições
 - (do então para o se...)
 - **Forward chaining:**
 - a partir dos fatos representados na base, o raciocínio avança na rede de inferência até chegar a uma conclusão
 - Todas as conclusões possíveis de se provar são inseridas na base como fatos derivados
 - (do se... para o então)

Sistema Inferência: Backward Chaining

- Procedimento `demo(T)`, onde T é a teoria a comprovar:

`demo(T) ← facto(T).`

`demo(T) ← se Condição então T, demo(Condição).`

`demo(P1 e P2) ← demo(P1), demo(P2).`

`demo(P1 ou P2) ← demo(P1).`

`demo(P1 ou P2) ← demo(P2).`

Executar o conjunto de passos para demonstrar:

`demo(fuga_na_cozinha) ou seja há fuga na cozinha?`

Sistema Inferência: Forward Chaining

- Procedimento demo, que deriva todos os fatos novos:

demo←

```
novo_facto(P), !,  
write('derivado:'), write(P), nl,  
assert(facto(P)),  
demo.
```

demo ← write('não há mais factos').

```
novo_facto(Ação) ←  
    se Condição então Ação,  
    não(facto(Ação)),  
    facto_composto(Condição).
```

facto_composto(Cond) ← facto(Cond).

facto_composto(C1 e C2) ← facto_composto(C1), facto_composto(C2).

facto_composto(C1 ou C2) ← facto_composto(C1).

facto_composto(C1 ou C2) ← facto_composto(C2).

Exemplo de Forward Chaining:

?- demo.

derivado: fuga_na_cozinha

derivado: não_entra_água

derivado: inundação

não há mais factos

Limitações destes Sistemas de Inferência

- Os sistemas de inferência apresentados anteriormente são muito simples e por isso têm certas limitações:
 - Não é possível ter variáveis nas condições
 - Ex. se idade(X) e X>10 e X<20 então adolescente
 - Como resolver?
 - se id_maior_10 e id_menor_20 então adolescente
 - Não é possível negar uma condição
 - Ex. se não(doente) e quer_medico então maluco
 - Como resolver?
 - se não_doente e quer_medico então maluco e quando quiser ligar um fato à sua negação:
 - facto(não_doente) ← não(demo(doente)).

Regras de Produção em Prolog

Uso de termos em inglês, conforme [Brakto, 1990]:

- **Regras de produção: if, then, and, or, fact, new_derived_fact, composed_fact**

- **if, then, and e or definidos como operadores**

```
: - op( 800, fx, if).
```

```
: - op( 700, xfx, then).
```

```
: - op( 300, xfy, or).
```

```
: - op( 200, xfy, and).
```

- **Tipicamente é necessário definir o “fact” como dinâmico:**

```
: - dynamic(fact/1).
```

← em Prolog será: :-

não em Prolog será: \+ ou not

Regras de Produção em Prolog

`:- dynamic fact/1.`

% regras: base de conhecimento

`if hall_molhado and cozinha_seco then fuga_banho.`

`if hall_molhado and banho_seco then fuga_cozinha.`

`if janela_fechada or nao_chove then nao_agua.`

`if hall_seco and banho_molhado then torneira_aberta.`

`if fuga_cozinha and não_agua then inundacao.`

% factos: base de dados, situação atual

`fact(hall_molhado).`

`fact(banho_seco).`

`fact(janela_fechada).`

Regras de Produção: Backward

% A simple backward chaining rule interpreter

```
:- op( 800, fx, if).
:- op( 700, xfx, then).
:- op( 300, xfy, or).
:- op( 200, xfy, and).
demo( Q) :- fact( Q).
demo( Q) :-
    if Condition then Q,          % A relevant rule
    demo( Condition).           % whose condition is true
demo( Q1 and Q2) :- demo( Q1), demo( Q2).
demo( Q1 or Q2) :- demo( Q1); demo( Q2).
```

Regras de Produção: Forward

demo:-

```
new_derived_fact( P), !, % Simple forward chaining in Prolog  
                          % A new fact
```

```
write( 'Derived: '), write( P), nl,
```

```
assert( fact( P)),
```

```
demo.                      % Continue
```

demo:- write('No more facts'). % All facts derived

new_derived_fact(Concl) :-

```
    if Cond then Concl,                          % A rule
```

```
    \+ fact(Concl),                              % Rule's conclusion not yet a fact
```

```
    composed_fact( Cond).                      % Condition true?
```

composed_fact(Cond) :- fact(Cond). % Simple fact

composed_fact(Cond1 and Cond2) :-

```
    composed_fact( Cond1), composed_fact( Cond2).
```

composed_fact(Cond1 or Cond2) :-

```
    composed_fact( Cond1) ; composed_fact( Cond2).
```

Geração de Explicações

1. Se P é um fato então a árvore de prova é P
2. Se P foi derivado utilizando a regra:
se Cond então P a árvore de prova é:
 $P \leq \text{Prova_de_Cond}$, onde Prova_de_Cond é a
árvore de prova da condição Cond
3. Se P_1 e P_2 são proposições cujas árvores de prova são
 Prova1 e Prova2 , então:
se P é P_1 e P_2 a árvore de prova é Prova1 e Prova2
se P é P_1 ou P_2 a árvore de prova é Prova1 ou Prova2

Interpretador com Geração de Provas (Explicação)

Procedimento demo(T,P), onde T é o teorema a comprovar e P é a prova:

demo(T,T) ← facto(T).

demo(T,T<=ProvaCond) ←
se Condição então T, demo(Condição, ProvaCond).

demo(P1 e P2, Prova1 e Prova2) ← demo(P1,Prova1), demo(P2,Prova2).

demo(P1 ou P2, Prova) ← demo(P1,Prova).

demo(P1 ou P2, Prova) ← demo(P2,Prova).

Exemplo: Passagem a uma disciplina

Regras adquiridas via alunos:

- Sei que passo a uma disciplina se tenho positiva no exame e não tenho faltas
 - Também sei que se estudei então irei ter positiva
 - Por último, se não estudei então tenho negativa
- **Fatos:**
 - o Rui estudou não teve faltas

Exercício:

Elaborar as regras de produção e fatos sobre a disciplina.

Depois, elaborar a questão:

↳ `demo(passo,P)` ou seja como passou o Rui?, determinando o que deve ser P

Incerteza

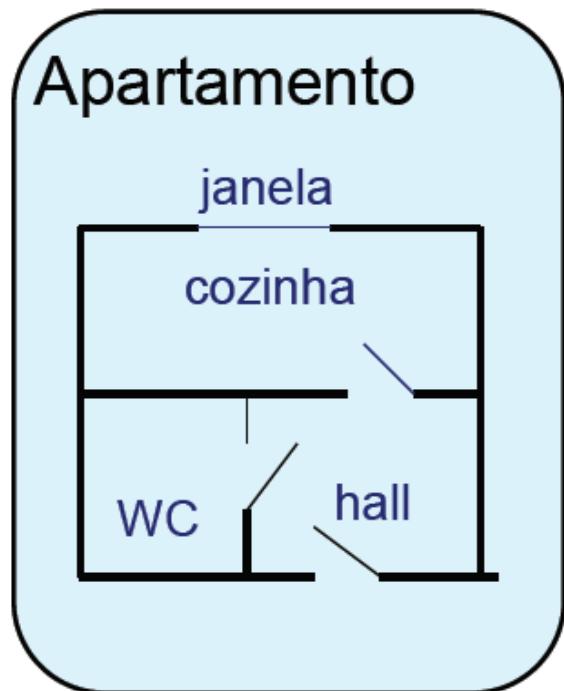
- Nos sistemas anteriores a informação é do tipo verdade ou falso, não existindo valores intermédios (falso, pouco provável, provável, altamente provável, verdadeiro)
- Todavia, muitos sistemas de apoio à decisão do mundo real têm de lidar com incerteza (grau de risco, probabilidades)
- Na nova representação, cada proposição contém um número entre 0 (totalmente falso) e 1 (totalmente verdadeiro), que define a probabilidade de ser verdadeiro:

Para fatos: `facto(Proposição:C).`

Para regras: `se Condição então Acção:C`

sendo C um número entre 0 e 1. (0-probabilidade 0%; 1:probabilidade 100%)

Fugas de água com probabilidades



Novo conhecimento adquirido via um perito em fugas de água:

- julgo que quando o hall está molhado e a cozinha seca, existe então uma fuga com 80% de probabilidade na casa de banho
- penso que deve existir um problema na cozinha se, pelo menos afirmo-o com 90% de certeza, se o hall estiver molhado e a casa de banho seca.

Situação atual:

- o WC está completamente seco, o hall está cerca de 70% molhado.

Exercício:

- Elaborar regras de produção e fatos sobre a fuga de água.
- O que se pode derivar?

Interpretador de Regras de Produção com Incerteza

- Em qualquer representação de incerteza é necessário combinar as certezas das proposições e regras.
- Um esquema simples consiste nas seguintes regras:
 - $P_1 \text{ e } P_2 \rightarrow \min(c(P_1), c(P_2))$
 - $P_1 \text{ ou } P_2 \rightarrow \max(c(P_1), c(P_2))$
 - Se existir uma regra:
 $\text{se } P_1 \text{ então } P_2:C \rightarrow c(P_2) = c(P_1) * C$

Interpretador de Regras de Produção com Incerteza

Procedimento do tipo demo(Proposição,Certeza):

demo(P,C) ← facto(P:C).

demo(Cond1 e Cond2, C) ←
 demo(Cond1, C1), demo(Cond2, C2), min(C1,C2,C).

demo(Cond1 ou Cond2, C) ←
 demo(Cond1,C1), demo(Cond2,C2), max(C1,C2,C).
demo(P,C) ←
 se Cond então P:C1, demo(Cond,C2), C is C1 * C2.

Exercício: Qual a probabilidade de problema na cozinha?
demo(problema_cozinha,X).

Problemas com a Incerteza

- A proposta apresentada pode ser criticada por ser demasiado simples!
- Por exemplo:
 - Imagine que existem os fatos a e b com probabilidades de 0.5 e 0.0. Para a ou b a probabilidade final é 0.5
 - Contudo, se a probabilidade de b aumentar para 0.5, no esquema definido a probabilidade final mantém-se em 0.5!
 - Os seres humanos têm dificuldade em pensar em termos de probabilidades (pelo menos de acordo com as definições matemáticas)
- **Mais à frente vamos ver com mais detalhe o Raciocínio Impreciso e abordagens mais “evoluídas”!**

Regras de Produção em Prolog: Explicações

Regras de Produção em Prolog: Explicações (proof.pl)

% demo(P, Proof), where Proof is a proof that P is true

```
: - op( 800, xfx, <=).
```

```
demo( P, P) :- fact( P).
```

```
demo( P, P <= CondProof) :-
```

```
    if Cond then P, demo( Cond, CondProof).
```

```
demo( P1 and P2, Proof1 and Proof2) :-
```

```
    demo( P1, Proof1), demo( P2, Proof2).
```

```
demo( P1 or P2, Proof) :-
```

```
    demo( P1, Proof); demo( P2, Proof).
```

Regras de Produção em Prolog: Incerteza

Regras de Produção em Prolog: Incerteza (certainty.pl)

% *democ(Proposition, Certainty)*

democ(P, Cert) :- fact(P: Cert).

democ(Cond1 and Cond2, Cert) :-

*democ(Cond1, Cert1), democ(Cond2, Cert2),
Cert is min(Cert1, Cert2).*

democ(Cond1 or Cond2, Cert) :-

*democ(Cond1, Cert1), democ(Cond2, Cert2),
Cert is max(Cert1, Cert2).*

democ(P, Cert) :-

*If Cond then P : C1, democ(Cond, C2),
Cert is C1 * C2.*

Mais sobre Estruturas Hierárquicas

Representação de conhecimento para grandes conjuntos de dados

- Objetivo é compactar os fatos que representam um dado sistema
- Cada entidade pode ser agrupada em classes, partilhando valores para os mesmos atributos
- Com a herança, os fatos de um objeto podem não estar representados ao seu nível mas são reconstruídos a partir de uma inferência sobre as classes superiores
- Como indicado, nos slides iniciais, existem diversos métodos para representar estruturas hierárquicas:
 - Redes semânticas - correspondendo a um grafo, onde os nodos definem entidades (objetos, classes) e os ramos relações entre as mesmas
 - Enquadramentos (Frames) - objetos com um conjunto de slots (atributos), sendo alguns destes utilizados para representar as relações entre classes e super-classes

Mias sobre Redes Semânticas

- **Rede de Entidades e relacionamento entre as Entidades**
- **Corresponde a um GRAFO:**
 - os nodos correspondem às entidades
 - os ramos correspondem às relações e são etiquetados com o nome da relação
- **Tipos de Relações: é_um (isa), tem, desloca-se, respira, etc...**

Redes Semânticas via Agentes e Relações

- **Será baseada em grafos de agentes;**
 - Cada agente possui um conjunto de propriedades, sendo representado num nodo
 - Todos os ramos definem relações isa (é_um)
 - A definição da estrutura hierárquica é realizada através de 2 predicados:

agente(A,P). - lista de propriedades P associadas ao agente A

isa(A,C). - o agente A está inserido na classe C, logo herda todas as suas propriedades

Sistema de Inferência para RS

- As questões deverão ser realizadas através do predicado: `demo(A,Q)` onde A denota o agente e Q a questão;
- O predicado `provar(S,A,Q)` tenta provar a pergunta Q no agente S, se não conseguir, sobe de classe, sendo A o agente atual, e por aí em diante:

`demo(A, Q) ← provar(A, A, Q).`

`provar(S, A, Q) ← agente(A, P), processar(S, Q, P).`

`provar(S, A, Q) ← isa(A, C), provar(S, C, Q).`

Q2.1: Qual a arma da cobra do tipo constrictor?

Q2.2: Qual o revestimento e onde vive o camelo?

Q2.3: Quais as propriedades (todas) herdadas pelo urso?

Mais sobre Enquadramentos (Frames)

- Cada objeto corresponde a um objeto físico ou um conceito mais abstrato como classe de objetos ou situação
- Frame: estrutura de dados cujos componentes se chamam slots
- Os Slots são identificados por nomes e denotam informação de vários tipos: valores simples, referências de outras frames, procedimentos, ...
- Is_a: relação de entre Classe e Superclasse
- Instance_of: relação de membro de uma classe
- Representação: frame(Frame, Slot, Valor).

Exemplo: Passáros



Pássaro	
isa	animal
movimento	voa

Canário	
isa	pássaro
tamanho	20
cor	amarelo

Alberto	
instanceof	canário
tamanho	15

Sistema de Inferência para Frames

- As questões deverão ser realizadas através do predicado:
demo(O,R,V) onde O denota o objeto, R a relação e V o valor:

```
demo(Frame,Slot,Valor) ←  
    frame(Frame, Slot, Valor).  
  
demo(Frame,Slot,Valor) ←  
    super_frame(Frame,Superframe),  
    demo(Superframe,Slot,Valor).  
  
super_frame(Frame,Superframe) ←  
    frame(Frame,isa,Superframe).  
  
super_frame(Frame,Superframe) ←  
    frame(Frame,instanceof,Superframe).
```

- Q4.1 Qual o tamanho do Alberto?**
- Q4.2 O que se pode dizer sobre o Alberto?**
- Q4.3 Quais as superclasses do Alberto?**

Redes Semânticas em Prolog

- É necessário carregar o interpretador e definir como dinâmicas todas as propriedades:
:- [redesem], dynamic(vive/1), ...

- **Sistema de Inferência:**

% demonstrar a questão Q no agente A

demo(A, Q):- provar(A, A, Q).

% S - Self, onde começa a prova, A - Agente atual, Q questão

provar(S, A, Q):- agente(A, P), processar(S, Q, P).

provar(S, A, Q):- isa(A, C), provar(S, C, Q).

Redes Semânticas em Prolog

```
processar(_, Q, _):-      % posso provar já Q?  
    callable(Q),  
    call(Q),!.  
  
processar(S, Q, _):-      % Q é composto (tem mais do que uma questão)?  
    nonvar(Q),  
    Q=(Q1, Q2),           % decompor Q em Q1 e Q2  
    demo(S, Q1), demo(S, Q2).  
  
processar(S, Q, P):- processar_propriedades(S, Q, P).  
processar_propriedades(S, Q, [(Q:-Body)|_]):- demo(S, Body),!.  
processar_propriedades(_, Q, [Q|_]).  
processar_propriedades(S, Q, [_|P]):- processar_propriedades(S,Q,P).
```

Frames em Prolog:

É necessário carregar o interpretador frames.pl

```
:  
- [frames].
```

- Sistema de Inferência:

```
% directo
```

```
demo(Frame,Slot,Valor)
```

```
:  
- frame(Frame,Slot,Valor).
```

```
% herança
```

```
demo(Frame,Slot,Valor)
```

```
:  
- super_frame(Frame,Superframe), demo(Superframe,Slot,Valor).
```

```
% buscar os superframes
```

```
super_frame(Frame,Superframe):-
```

```
    frame(Frame,isa,Superframe).
```

```
super_frame(Frame,Superframe):-
```

```
    frame(Frame,instanceof,Superframe).
```

Raciocínio Incerto/Impreciso

- Qualquer fonte de Conhecimento pode conter incertezas associadas
- É necessário aumentar as capacidades de:
REPRESENTAÇÃO de CONHECIMENTO
com medidas refletindo os graus de evidência e crenças.
- Duas filosofias diferentes:
 - Teoria das Probabilidades (aproximação **Frequencista**, resulta de cálculos sobre experiências)
 - Teorias Probabilísticas **Subjetivistas** (resulta das crenças acumuladas por um Agente)

Raciocínio Incerto

Tipos de incerteza nos Sistemas Baseados em Conhecimento (KBS):

- 1- os factos iniciais tem incertezas associadas
- 2- as regras, mesmo com factos certos, podem dar conclusões incertas
- 3-a conclusão propaga incerteza resultante das dos dois casos anteriores

Podem ainda suceder 2 hipóteses:

- ou a regra não é ativada pois os factos têm certezas demasiado pequenas
 - ou a conclusão obtida com um grau de certeza é igual a outra já conhecida com grau de certeza diferente. Que fazer?
- 4- No processo de propagação pelas regras não há total independência

Raciocínio Incerto

Como exprimir a confiança nas fórmulas ?

1- Multiplos valores e não só dois (V e F)

- ou permitir que V e F sejam definidos em intervalos
- ou assumir graus de pertença aos dois conjuntos o que implica uma lógica própria para propagar o grau de pertença.

2- Atribuir um valor de probabilidade ao valor de verdade da fórmula

A teoria dos “**Fuzzy sets**” (**Conjuntos Difusos**) permite propagar o grau de pertença a um conjunto através das conectivas lógicas.

Raciocínio Incerto

Teoria das Probabilidades (aproximação **Frequencista**)

Representa uma Teoria estatística da evidência baseada na noção de **probabilidades condicionais**

Teorema de Bayes:

$$P(H_i|E) = \frac{P(E|H_i) * P(H_i)}{\sum_{n=1}^k P(E|H_n) * P(H_n)}$$

Simplificável

Aproximação Frequencista

- O **Teorema (ou Regra ou Lei) de Bayes** relaciona a probabilidade **condicional** de um evento com a ocorrência **conjunta** de vários eventos:
 $p(H \cap E) = p(H|E) * p(E)$
 $p(E \cap H) = p(E|H) * p(H)$
- de onde se deriva a fórmula de Bayes relacionando probabilidades condicionais e que na sua forma mais simples se pode escrever:
 $p(H|E) = (p(E|H) * p(H)) / p(E)$
- O teorema de Bayes é uma teoria estatística da evidência baseada na noção das probabilidades condicionais

Aproximação Frequencista

- Para casos simples seria possível ter tabelas de probabilidades conjuntas.
- Exemplo:

		dor	~dor
cavidade	dor	0.04	0.06
	~dor		
~cavidade	dor	0.01	0.89
	~dor		

A soma das probabilidades deve ser = 1

$$P(c)=0.1 \quad p(c \vee d)=0.11 \quad P(d)=0.05 \quad p(c \cap d)=0.04$$

aplicando a fórmula de Bayes: $p(H \cap E) = p(H|E) * p(E)$

$$P(c|d)=p(c \cap d)/p(d)=0.04/0.05=0.8$$

Naive Bayes simplifica de $O(2^n)$
Para $O(n)$

- Mas com muitas variáveis (n sintomas) a representação torna-se impraticável $O(2^n)$. Não se usam as probabilidades **conjuntas** mas sim diretamente as **condicionais**

Aproximação Frequencista

A **regra de Bayes** requer três termos (uma prob. Condicional e duas prob. à priori) para computar uma única prob. Condicional!

$$p(H|E) = (p(E|H) * p(H)) / p(E)$$

É útil tendo essas probs., como acontece em certos casos médicos, onde temos probs. sobre **causas** e queremos Probabilidade de diagnósticos.

prob(cause|effect)

$$p(\text{causa}|\text{efeito}) = p(\text{efeito}|\text{causa}) * p(\text{causa}) / p(\text{efeito})$$

Conhecemos a evidência de um efeito
mas a causa é desconhecida

$p(\text{causa}|\text{efeito})$ quantifica a relação no sentido do **Diagnóstico**

$p(\text{efeito}|\text{causa})$ quantifica a relação no sentido **Causal**

Médicos e outros especialistas conhecem mais facilmente as **relações causais** e querem calcular probabilidades de um diagnóstico ser verdadeiro.

Aproximação Frequentista

A regra de Bayes

$$p(H|E) = (p(E|H) * p(H)) / p(E)$$

Ex: médico conhece meningite → pescoço rígido (70%) $p(pr|m) = 0,7$
prob(meningite) $p(m) = 1/50.000$
prob(pescoço rígido) $p(pr) = 1/100$

prob(meningite|pescoço rígido)

$$p(m|pr) = p(pr|m) * p(m) / p(pr) = 0,0014 \text{ (<1 em 700 pacientes com pr tem m)}$$

$p(pr|m)$ e $p(m|pr)$ são muito diferentes devido à prob. à priori de pr ser muito maior que a de m.

Porquê a necessidade de calcular $p(m|pr)$? Porquê o médico não a conhece como as outras?
Havendo surto de meningite, médicos computando as prob. à priori a partir das estatísticas não sabem como actualizar $p(m|pr)$.

Médicos usando regra de Bayes sabem que crescerá proporcionalmente a $p(m)$.

Aproximação Frequencista

O Problema da dependência das proposições leva à necessidade das Probabilidades condicionais

Seja o problema seguinte de diagnóstico:

- P: Paciente tem pintas
- S: Paciente tem sarampo
- F: Paciente tem febre



Precisamos da probabilidade condicional:

$$P(H|E_1, E_2) = P(H|E_2) * \frac{P(E_2|E_1, H)}{P(E_1) * P(E_2)}$$

O número de probabilidades a calcular aumenta muito com o número das proposições ($O(2^n)$) o que torna o uso de teorema intratável.

Aproximação Frequencista

$$p(H \cap E) = p(H|E) * p(E)$$

Independência

dois acontecimentos H e E são independentes se:

$$p(H|E) = p(H) \quad \longrightarrow \quad p(H \cap E) = p(H) * p(E)$$

se $C = A \cap B \quad p(C) = p(A \cap B) = p(A) * p(B)$

Dependência

 implica:

Aplicação da fórmula de Bayes

H: paciente tem cancro no pulmão

E: paciente fuma

queremos $p(H|E) = p(\text{cancro}|\text{fuma})$

Aproximação Frequentista

dado $p(H)$ probabilidade de **cancro** do pulmão na população
 $p(E)$ probabilidade de se ser **fumador**
 $p(E|H)$ $p(\text{fumador}|\text{cancro})$

isto é, conhece-se de todas as pessoas diagnosticadas com cancro quantas eram fumadoras, então se for dada evidência que um paciente **fuma**, a probabilidade de ter **cancro** modifica-se de $p(H)$ para

$$p(H|E) = \boxed{\frac{p(E|H)}{p(E)}} * p(H)$$

$$> 1 \quad \text{logo } p(H|E) > p(H)$$

se

$$p(E|H) > p(E) \text{ é previsível}$$

Aproximação Frequencista

Limitações da aproximação probabilística:

Problema da **assunção da independência dos factos (leva a incorrecções)**

$$C = A \cap B \quad p(C) = p(A) * p(B)$$

$$D = C \cap A \quad p(D) = p^2(A) * p(B)$$

o que é incorrecto pois C e A já não são independentes

$$p(D) = p(C \cap A) = p(C|A) * p(A) \quad \rightarrow$$

$$p(C|A) = \frac{p(C \cap A)}{p(A)} = \frac{p(A \cap B \cap A)}{p(A)} = \frac{p(A \cap B)}{p(A)} = p(B)$$

$$P(D) = p(C \cap A) = p(A) * p(B) \quad \rightarrow \quad p(D) = p(C)$$

- Assumindo Independência resulta probabilidade inferior $p^2(A) * p(B)$

Outra questão: As probabilidades tem um duplo efeito, isto é, SE E suporta parcialmente H então também deve suportar negativamente a negação da hipótese $\sim H$ pois $p(H|E) + p(\sim H|E) = 1$

Mas os humanos, em contraste, tendem a classificar evidência para suportar ou outras para refutar hipóteses.

Outros Métodos

Três Mecanismos modificam esta teoria para a tornar tratável:

- a) Modelo dos Fatores de Certeza
- b) Rede de Bayes
- c) Teoria de Dempster-Shaffer

- a) **Modelo dos Fatores de Certeza**

As regras (na Representação do Conhecimento) tem Fatores de Certeza associados

$$FC[h|e] = MB[h,e] - MD[h,e]$$

- Os Fatores de Certeza podem ser combinados uma vez que:
 - i) várias **regras** suportam separadamente uma **mesma hipótese**
 - ii) várias **evidências** conjuntas suportam uma **hipótese**
 - iii) o resultado de uma **regra** é a **entrada** em outra

Modelo dos Factores de Certeza

- MODELO DOS FACTORES DE CERTEZA
 - mecanismo relativamente **informal** para quantificar o **grau de crença** numa certa **conclusão** baseado na presença de um conjunto de **evidências**
 - Usado em domínios onde a evidência é incremental
- **Factor de Certeza** é um valor numérico que exprime quanto devemos acreditar numa conclusão dado um conjunto de evidências

$$-1 \leq FC \leq 1$$

- FC é uma quantificação **subjetiva**
- exemplo: diagnóstico de falhanço de retroprojeto

Modelo dos Fatores de Certeza

exemplo: REGRA de diagnóstico de falhanço de retroprojector

SE	o retroprojector está ligado na ficha
E	existe tensão na rede
E	o interruptor do retroprojector está "ON"
E	a lâmpada está boa
ENTÃO	Há uma evidência grande (0.8) que o fusível queimou

Medida de crença $MB[H, E]$ indica como o valor da crença na conclusão C (hipótese H) cresce baseado na presença da evidência E . $0 \leq MB \leq 1$

Medida de descrença $MD[H, E]$

Para a mesma evidência:

$$MB > 0 \rightarrow MD = 0$$

$$MD > 0 \rightarrow MB = 0$$

$$\text{ou } MB = 0 \text{ ou } MD = 0$$

$$FC[H, E] = MB[H, E] - MD[H, E]$$

FACTOR DE CERTEZA DE C DADO E varia entre 1 e -1

Modelo dos Fatores de Certeza

- O Modelo dos Fatores de Certeza foi introduzido por Buchanan, inicialmente para o SP MYCIN.
- O uso deste Modelo pressupõe que:
 - Todas as regras da BC são ou puramente **causais** ou puramente de **Diagnóstico**
 - As evidências são só introduzidas nas raízes das regras
 - As regras são quase sempre simplesmente ligadas
- Nestas circunstâncias o Modelo dos Fatores de Certeza é equivalente à inferência por redes Bayesianas.
- Caso contrário os resultados do Modelo dos Fatores de Certeza podem ser errados

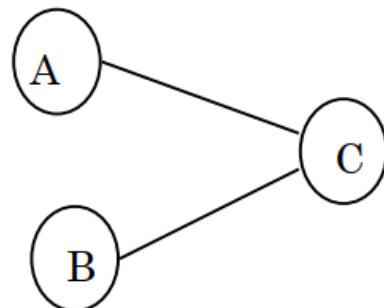
Modelo dos Fatores de Certeza

As fórmulas de combinação deverão verificar os seguintes princípios:

- As funções de combinação devem ser associativas e comutativas (independentes da ordem de aparecimento das evidências)
- Acumulação de evidências devem aumentar MB
- Encadeamento de raciocínios com incertezas associadas devem levar a um progressivo decrescimento na Crença

Modelo dos Fatores de Certeza

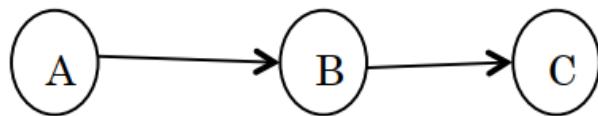
TRÊS FORMAS DE COMBINAÇÃO DE FACTORES DE CERTEZA:



1) Mais que uma regra levam à mesma conclusão



2) Crença numa coleção de fórmulas em conjunto



3) Encadeamento de Regras

Modelo dos Fatores de Certeza

1) Combinação de Evidências

$$\text{SE } MD[h, e1 \& e2] > 0 \quad MB[h, e1 \& e2] = 0$$

Se $MD=0$ então $FC=MB$

$$\text{SENÃO } MB[h, e1 \& e2] = MB[h, e1] + MB[h, e2] * (1 - MB[h, e1])$$

Similar para MD

$$MB(h, e1) \text{ e } MB(H, e2) > 0$$

2) Combinação de Hipóteses

$$MB[h_1 \cap h_2, e] = \min(MB[h_1, e], MB[h_2, e])$$

$$MB[h_1 \cup h_2, e] = \max(MB[h_1, e], MB[h_2, e])$$

3) Combinação da Incerteza na evidência suportando a hipótese com a incerteza na hipótese

$$MB[h, s] = MB'[h, s] * \max(0, FC[s, e])$$

onde $MB'[h, s]$ é a crença em h no caso de s ser certo

Modelo dos Fatores de Certeza

Caso 1)

Ex. Se uma observação inicial confirma h em 0.3 e nada a desconfirma
Então:

$$MB[h,e1] = 0.3 \quad MD[h,e1] = 0 \text{ e } FC[h,e1] = 0.3$$

Se nova evidência $e2$ confirma h com 0.2

Então:

$$MB[h,e2] = 0.2 \quad MD[h,e2] = 0 \quad e$$

$$MB[h,e1 \wedge e2] = 0.3 + 0.2 * (1-0.3) = 0.44 \quad e$$

$$MD[h,e1 \wedge e2] = 0$$

$$\text{Logo: } FC[h,e1 \wedge e2] = 0.44$$

$$0,2 + 0,3 * (1-0,2) = 0,44$$

Modelo dos Fatores de Certeza

Seja agora o caso 2)

Onde temos a considerar um conjunto de hipóteses
(como é o caso das premissas de uma Regra que foi seleccionada)

SE $h_1 \wedge h_2 \wedge (h_3 \vee h_4)$ Então C

$$MB[h_1 \wedge h_2, e] = \min(MB[h_1, e], MB[h_2, e])$$

$$MB[h_3 \vee h_4, e] = \max(MB[h_3, e], MB[h_4, e])$$

Logo

$$MB(C, h_1 \wedge h_2 \wedge h_3 \vee h_4) = \min(MB[h_1, e], MB[h_2, e], \max(MB[h_3, e], MB[h_4, e]))$$

Para MD é análogo

Modelo dos Fatores de Certeza

Consideremos finalmente o caso 3)

Onde as Regras são encadeadas durante o processo de inferência

Temos que considerar as incertezas dos dados de entrada na Regra activada (provenientes de observações com incerteza ou de conclusões de Regras incertas)

Se considerarmos MB' a medida da Crença em h se a observação s fosse Verdadeira, então:

$$MB[h, s] = MB'[h, s] * \max(0, MB[s, e])$$

sendo e a evidência suportando s

Modelo dos Fatores de Certeza

$$MB[h, s] = MB'[h, s] * \max(0, MB[s, e])$$

Exemplo:

(do Sistema Mycin. Os factores de certeza estão instanciados para um caso)

SE tipo_organismo	GRAM POSITIVO	(0.9)
E morfologia	ROD	(0.8)
E aerobicia	AEROBICO	(0.7)
OU aerobicia	DESCONHECIDA	(0.3)
ENTÃO há evidência sugestiva (0.6) de ENTEROBACTERACEAE		

$$\begin{aligned} FC[h, e1 \cap e2 \cap (e3 \cup e4)] &= 0.6 * \max(0, FC[e1 \cap e2 \cap (e3 \cup e4)]) \\ &= 0.6 * \max((0, MB[e1 \cap e2 \cap (e3 \cup e4)] - MD[\dots])) \end{aligned}$$

$$MB[e1 \cap e2 \cap (e3 \cup e4)] = \min(MB[e1], MB[e2], \max(MB[e3], MB[e4]))$$

$$\begin{aligned} FC[Enterobacteraceae, [e1 \cap e2 \cap (e3 \cup e4)]] &= \\ &0.6 * \max(0, \min(0.9, 0.8, \max(0.7, 0.3))) \\ &= 0.6 * 0.7 \\ &= 0.42 \end{aligned}$$

Modelo dos Fatores de Certeza

Dificuldades:

Ex: Suponhamos considerar 3 **regras**

R1: SE organismo gram positivo ENTÃO stafilococcus (0.6)

R2: SE morfologia do organismo é coccus ENTÃO stafilococcus (0.6)

R3: SE conformação de crescimento do organismo é "borrão" ENTÃO stafilococcus (0.6)

Na realidade, esta situação seria compatível com um FC combinado de 0.6 por haver bastante sobreposição das condições

aplicando as fórmulas de 1)

$$MB[h, S_1 \wedge S_2] = 0.6 + (0.6 * 0.4) = 0.84$$

$$MB[h, (S_1 \wedge S_2) \wedge S_3] = 0.84 + (0.6 * 0.16) = 0.936$$

Este resultado afasta-se bastante do resultado do especialista (0.6).
Logo este considerou uma única regra com 3 premissas.

Um dos perigos do modelo

Modelo dos Fatores de Certeza

EXEMPLO DE INFERÊNCIA E PROPAGAÇÃO DOS FACTORES DE CERTEZA:

R1: $P_1 \rightarrow C_1 (0.7)$

P_1, P_3, P_4, P_5 verdadeiros (evidências anteriores)

R2: $C_1 \cap P_2 \cap P_3 \rightarrow C_2 (0.9)$

R3: $P_4 \rightarrow P_2 (0.4)$

R4: $P_1 \cap P_5 \rightarrow C_1 (0.6)$

De **R1** e **R4**, e porque P_1, P_5 são factos, conclui-se C_1

$$MB[C_1] = 0.7 + (1-0.7) * 0.6 = 0.88$$

De **R3** e porque P_4 é verdadeiro concluímos P_2 com **0.4**

R2 pode concluir **C2** com $FC=?$

$$MB[C_2, C_1 \cap P_2 \cap P_3] = MB' [C_2, C_1 \cap P_2 \cap P_3] * \max(0, FC[(C_1 \cap P_2 \cap P_3), Ea])$$

$$MB = 0.9 * \max(0, \min(0.88, 0.4, 1.0)) = 0.36 \rightarrow \text{FC} = 0.36$$



Modelo dos Fatores de Certeza

Violação da **independência das observações** no cenário 3).

Exemplo :

Proposições:

I: o irrigador funcionou na noite passada

H: a relva está húmida

C: choveu a noite passada

Possíveis regras:

R1: SE o irrigador funcionou a noite passada
ENTÃO a relva está húmida esta manhã (0.9)

outra

R2: SE a relva está húmida esta manhã
ENTÃO choveu a noite passada (0.8)

Se fez funcionar o irrigador a noite passada

ENTÃO:

pode concluir-se primeiro "relva húmida" com $FC=0,9$

$MB [H|I] = 0.9$

e depois, conclui-se que "choveu a noite passada"

$MB [C|H] = 0.9 * 0.8 = 0.72$

Incorrecto! Regras de tipo diferente.

Modelo dos Fatores de Certeza

Comparando as aproximações do **Modelo dos Factores de Certeza** e a probabilística (considerando a **independência**) poderemos concluir sobre o grau de confiança de C :

Se considerarmos $C = A \cap B$ (acontecimentos conjuntos):

conf (A)	conf (B)	FC conf(C) = min	Prob considerando independência conf (C) = conf (A) * conf (B)
0	0	0	0
0.1	0.9	0.1	0.09
0.25	1	0.25	0.25
0.25	0.25	0.25	0.06
0.5	1	0.5	0.5
0.5	0.5	0.5	0.25
0.75	1	0.75	0.75
0.75	0.75	0.75	0.56

FC assemelha-se à aproximação **probabilística** considerando **não a independência mas dependencia**

:

$$\begin{aligned} p(C) = p(A \cap B) &= p(A|B) * p(B) \leq p(B) \\ &= p(B|A) * p(A) \leq p(A) \end{aligned}$$

Logo, tem de verificar-se (probabilidade considerando a dependência):

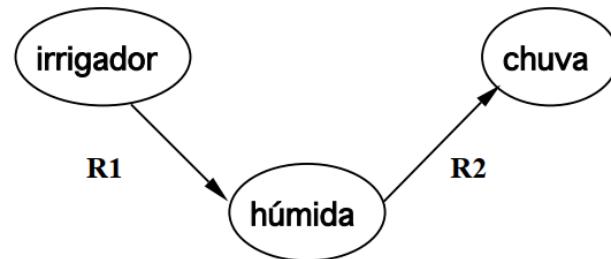
p(A ∩ B) ≤ min{p(B), p(A)} (**FC é = min**) mas superior ou igual à probabilidade considerando

Método das Redes Bayesianas

- **REDES BAYESIANAS como processo de inferir conhecimento incerto**

Temos de descrever apenas grupos de acontecimentos que interagem

Uma Rede (NÃO Bayesiana) que representa o encadeamento que considerado no exemplo anterior é:

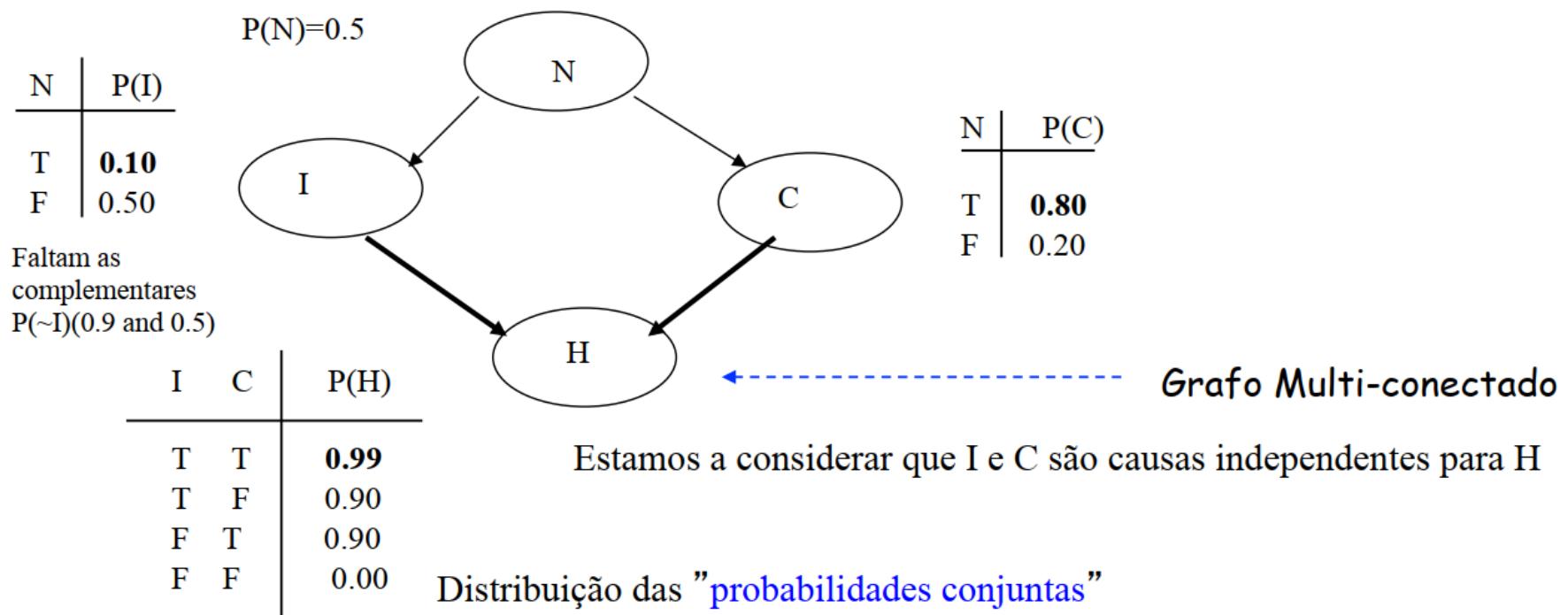


Método das Redes Bayesianas

As Redes Bayesianas **separam** estes dois tipos de influência:

• Um Gráfico aciclico dirigido (GAD) representa só as relações de **Causalidade**.

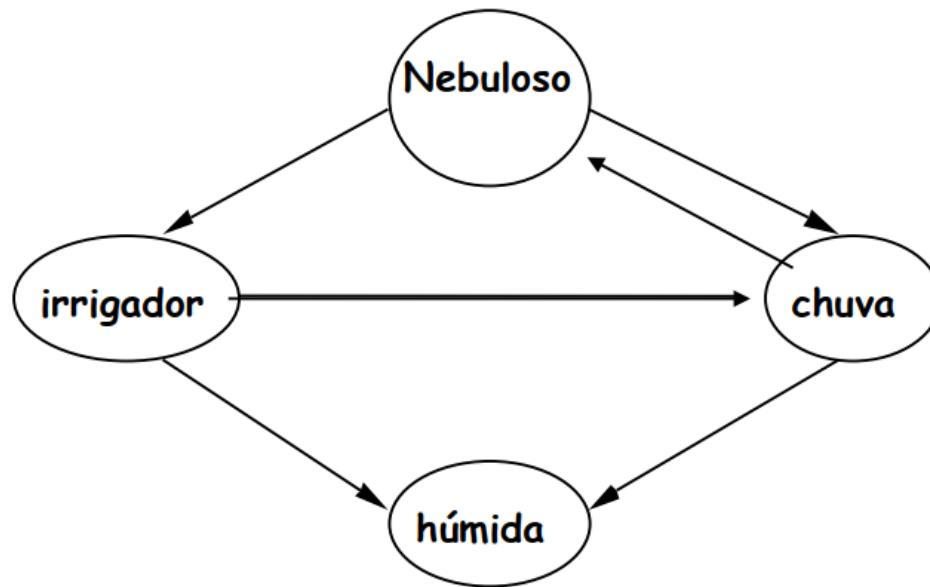
• Ex: Sejam N - muito Nublado; I- Irrigador; C- Chuva; H- relva Húmida:



Método das Redes Bayesianas

Podemos precisar de um mecanismo para computar a influência de um qualquer nodo noutro. Então a direcção dos arcos já não é única.

Por exemplo se choveu a noite passada que concluir acerca da Nebulosidade?



Se, por exemplo considerássemos $p(C|I) = 0$
"irrigador ligado" deixa de ter importância para "chover"

Se H e C afetam I então I e C só são marginalmente independentes
Mas condicionalmente dependentes na presença de H. Usar fórmulas de Bayes

Teoria de Dempster-Schaffer

c) Teoria de Dempster-Shaffer

Par de valores **[Crença, Plausibilidade] ([Bel, Pl])** atribuídos a cada conjunto de proposições:

Crença(S) (ou Bel) vai de 0 (nenhuma evidência) a 1 (certeza).

Plausibilidade(S) = $1 - \text{Bel}(\sim S)$ entre 0 e 1.

Bel(S) indica o grau de confiança que temos numa proposição **S** e a **Plausibilidade ($1 - \text{Bel}(\sim S)$)** indica o valor máximo a que pode chegar aquela confiança.

Crença diferente de probabilidade pois não necessariamente $\text{Bel}(S) + \text{Bel}(\sim S)=1$
1 é apenas o limite superior dessa soma 1

Teoria de Dempster-Schaffer

c) Teoria de Dempster-Shaffer

Par de valores [**Crença, Plausibilidade**] (**[Bel, Pl]**) atribuídos a cada conjunto de proposições:

Crença(S) (ou Bel) vai de 0 (nenhuma evidência) a 1 (certeza).

Plausibilidade(S) = $1 - \text{Bel}(\sim S)$ entre 0 e 1.

Bel(S) indica o grau de confiança que temos numa proposição **S** e a **Plausibilidade (1-Bel(~S))** indica o valor máximo a que pode chegar aquela confiança.

Crença diferente de probabilidade pois não necessariamente $\text{Bel}(S) + \text{Bel}(\sim S)=1$
1 é apenas o limite superior dessa soma 1

Teoria de Dempster-Schaffer

Na Teoria de Dempster-Shaffer distingue-se entre:

Incerteza e

Ignorância

Em vez de computar a crença ou a probabilidade de uma proposição S calcula-se a probabilidade da evidência existente suportar a proposição S.

Este processo pode ajudar a decidir se devemos ou não esperar por mais evidências antes de decidir

Ex: lançamento de moeda usamos $p(\text{ca})=0.5$ e $p(\text{co})=0.5$

Mas se não tiver a certeza total que não está viciada, p.ex. $p(\text{viciada})=0.1$ então $p(\text{ca})$ e $p(\text{co})$ variam entre 0.45 e 0.55

Teoria de Dempster-Schaffer

Exemplo do cálculo e do significado de Plausibilidade:

S: Qual a porção de um conjunto de pessoas/empresas que se crê terem entre 3 e 5 carros?

Pes./Em.	Nº carros	BEL (S) entre 3 e 5
A	1 ou 2	não
B	1	não
C	4 ou 5	sim
D	5 ou 6	possível
E	6	não

$$Bel(S)=1/5$$

$$Pl(S) = 1 - 3/5 = 2/5 \quad \text{é a probabilidade máxima possível para a crença}$$

Teoria de Dempster-Schaffer

Vamos supor hipóteses mutuamente exclusivas para facilitar possível diagnóstico no conjunto Θ

$$\Theta = \{A, G, C, P\}$$

A: alergia

G: gripe

C: constipação

P: pneumonia

Pode ainda acontecer que uma evidência suporte não um elemento do conjunto Θ mas subconjuntos de Θ

Por exemplo Febre suporta $\{G, C, P\}$

Inicialmente apenas sabemos que a verosimilhança ou crença está em $[0,1]$

Não há mais informações

Não vamos usar prob. à priori que seria de 0.33 para cada elemento de $\{G, C, P\}$

Teoria de Dempster-Schaffer

Vamos supor hipóteses mutuamente exclusivas para facilitar possível diagnóstico no conjunto Θ

$$\Theta = \{A, G, C, P\}$$

A: alergia

G: gripe

C: constipação

P: pneumonia

Pode ainda acontecer que uma evidência suporte não um elemento do conjunto Θ mas subconjuntos de Θ

Por exemplo Febre suporta $\{G, C, P\}$

Inicialmente apenas sabemos que a verosimilhança ou crença está em $[0,1]$

Não há mais informações

Não vamos usar prob. à priori que seria de 0.33 para cada elemento de $\{G, C, P\}$

Teoria de Dempster-Schaffer

Consideremos a função **densidade de probabilidade m** para Θ e todos os seus sub-conjuntos.

$m\{p\}$ mede a "quantidade" de crença atribuída ao sub-conjunto p.

A soma de todos os ms não deve exceder 1.

No nosso exemplo suponhamos que **não** temos inicialmente informação então:

$\Theta = \{A, G, C, P\}$ $\{\Theta\}$ tem $m=1$ isto é, não se distinguem os elementos do conjunto.

Conhecemos agora a evidência **Febre** que aponta para o sub-conjunto $\{G, C, P\}$ com 0.6

então

$\{\Theta\}$ fica com $m=0.4$

(notar que 0.4 não está atribuído ao complementar do 1º sub-conjunto)

Teoria de Dempster-Schaffer

Sejam dadas duas funções de crença m1 e m2

Seja X o conjunto de todos os sub-conjuntos de Θ tais que a crença m1, é não zero

Seja Y o conjunto correspondente para m2

m3 combina as crenças m1 e m2 (m1 ⊕ m2) como segue:

$$\frac{m_3(Z) = \sum_{X \cap Y = Z} m_1(X) * m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) * m_2(Y)}$$

Usando o exemplo: Vamos supor primeiro que todas as intersecções de elementos de X e Y são não vazias.

m1 crença depois de observar febre.

{G, C, P}	0.6
Θ	0.4

m2 crença depois de observar “escorrência no nariz”

{A, G, C}	0.8
Θ	0.2

Teoria de Dempster-Schaffer

a função m_3 da combinação \oplus é computada com a ajuda da tabela

		{A, G, C} 0.8	Θ 0.2
m_1	m_2	{G, C, P} 0.6	{G, C, P} 0.12
Θ 0.4		{A, G, C} 0.32	Θ 0.08

$$M_3(Z) = \sum_{(X \wedge Y = Z)} m_1(X) * m_2(Y)$$

Teoria de Dempster-Schaffer

Suponhamos que depois de computado m_3 sabíamos que os sintomas atenuaram quando o doente viajou

$$\begin{array}{ll} m_4 & \{A\} \quad 0.9 \\ & \Theta \quad 0.1 \end{array}$$

ao computar o numerador da fórmula vem:

m3	m4	{A}	0.9	Θ	0.1
{G, C}	0.48	∅	0.432	{G, C}	0.048
{A, G, C}	0.32	{A}	0.288	{A, G, C}	0.032
{G, C, P}	0.12	∅	0.108	{G, C, P}	0.012
Θ	0.08	{A}	0.072	Θ	0.008

Θ tem uma crença de 0.54. Só 0.46 está associado com hipóteses possíveis.
Temos então de re-escalar os valores com o factor $1-0.54 = 0.46$

Então m_5 ($m_3 \oplus m_4$) resulta em:

$$\{G, C\} = 0.048 / 0.46 = 0.104$$

$$\{A, G, C\} = (0.032) / 0.46$$

$$\{G, C, P\} = 0.12 / 0.46$$

$$\{A\} = (0.072 + 0.288) / 0.46 = 0.78$$

$$\Theta = 0.008 / 0.46$$

$$\frac{m_3(Z) = \sum_{X \cap Y = Z} m_1(X) * m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) * m_2(Y)}$$

Teoria de Dempster-Schaffer

Novo Caso: Sejam E_2 e E_3 duas evidências levando à mesma hipótese H_1 da seguinte forma:

$$m_1(E_2) \{H_1\} = 0.8 \\ m_2(E_3) \{H_1\} = 0.4$$

$$m_1(E_2) \{\theta\} = 0.2 \\ m_2(E_3) \{\theta\} = 0.6$$

combinando: $(m_1(E_2) \oplus m_2(E_3)) \{H_1\}$

		$m_2(E_3)$	
		$\{H_1\}=0.4$	$\{\theta\}=0.6$
$m_1(E_2)$	$\{H_1\}=0.8$	$\{H_1\}=0.32$	$\{H_1\}=0.48$
	$\{\theta\}=0.2$	$\{H_1\}=0.08$	$\{\theta\}=0.12$

$$(m_1(E_2) \oplus m_2(E_3)) \{H_1\} = 0.32 + 0.08 + 0.48 = 0.88$$

$$(m_1(E_2) \oplus m_2(E_3)) \{\theta\} = 0.12$$

Usando o modelo dos Factores de Certeza daria:

$$FC(H_1) = MB - MD = MB(H_1) = 0.8 + 0.4 * (1 - 0.8) = 0.8 + 0.4 * 0.2 = 0.88$$

Modelo dos FC é um caso especial da Teoria de Dempster-Schaffer

Teoria de Dempster-Schaffer

Novo Caso: Sejam E_2 e E_3 duas evidências levando à mesma hipótese H_1 da seguinte forma:

$$m_1(E_2) \{H_1\} = 0.8 \\ m_2(E_3) \{H_1\} = 0.4$$

$$m_1(E_2) \{\theta\} = 0.2 \\ m_2(E_3) \{\theta\} = 0.6$$

combinando: $(m_1(E_2) \oplus m_2(E_3)) \{H_1\}$

		$m_2(E_3)$	
		$\{H_1\}=0.4$	$\{\theta\}=0.6$
$m_1(E_2)$	$\{H_1\}=0.8$	$\{H_1\}=0.32$	$\{H_1\}=0.48$
	$\{\theta\}=0.2$	$\{H_1\}=0.08$	$\{\theta\}=0.12$

$$(m_1(E_2) \oplus m_2(E_3)) \{H_1\} = 0.32 + 0.08 + 0.48 = 0.88$$

$$(m_1(E_2) \oplus m_2(E_3)) \{\theta\} = 0.12$$

Usando o modelo dos Factores de Certeza daria:

$$FC(H_1) = MB - MD = MB(H_1) = 0.8 + 0.4 * (1 - 0.8) = 0.8 + 0.4 * 0.2 = 0.88$$

Modelo dos FC é um caso especial da Teoria de Dempster-Schaffer

Teoria de Dempster-Schaffer

Seja agora o caso de E2 confirmar H1 e E3 desconfirmar (isto é confirmar $\sim H1$) da seguinte forma:

$$\begin{aligned}m_1(E2) \{H1\} &= 0.8 \\m_3(E3) \{\sim H1\} &= 0.5\end{aligned}$$

$$\begin{aligned}m_1(E2) \{\emptyset\} &= 0.2 \\m_3(E3) \{\emptyset\} &= 0.5\end{aligned}$$

$$m_1(E2) \oplus m_3(E3)$$

		$m_3(E3)$	
		$\{\sim H1\}=0.5$	$\{\emptyset\}=0.5$
$m_1(E2)$	$\{H1\}=0.8$	$\{\emptyset\}=0.40$	$\{H1\}=0.40$
	$\{\emptyset\}=0.2$	$\{\sim H1\}=0.10$	$\{\emptyset\}=0.10$

$$\begin{aligned}m_1(E2) \oplus m_3(E3) (\{H1\}) &= 0.4/(1-0.4) = \textcolor{red}{0.66} \\m_1(E2) \oplus m_3(E3) (\{\emptyset\}) &= 0.1/0.6 = \textcolor{blue}{0.17}\end{aligned}$$

Inicialmente com $m_1(E2)$ a Crença para $\{H1\}$ potencialmente poderia crescer até 1
Depois com $m_3(E3)$ a Crença para $\{H1\}$ alterou-se para: $[C, PI] = [\textcolor{blue}{0.66}, \textcolor{blue}{0.83}]$

plausibilidade

Teoria de Dempster-Schaffer

Problemas com *Crenças* conflituosas:

1- Ana e Elsa têm as seguintes crenças sobre ver filmes $\{F_1, F_2, F_3\}$:

$$m(\text{Ana}) \{F_1\} = 0.99$$

$$m(\text{Elsa}) \{F_2\} = 0.99$$

$$m(\text{Ana}) \{F_3\} = 0.01$$

$$m(\text{Elsa}) \{F_3\} = 0.01$$

$$m(\text{Ana}) \oplus m(\text{Elsa})$$

$$m(\text{Ana})$$

$$\{F_3\} = 0.01$$

$$m_3(\text{Elsa})$$

$$\{F_2\} = 0.99$$

$$\{F_3\} = 0.01$$

$$\{F_1\} = 0.99$$

$$\{\emptyset\} = 0.9801$$

$$\{\emptyset\} = 0.0099$$

$$\{F_3\} = 0.0099$$

$$\{F_3\} = 0.0001$$

$$m(\text{Ana}) \oplus m(\text{Elsa}) (\{F_3\}) = 0.0001 / (1 - 0.9999) = 1$$

$$m(\text{Ana}) \oplus m(\text{Elsa}) (\{F_1\}) = 0$$

$$m(\text{Ana}) \oplus m(\text{Elsa}) (\{F_2\}) = 0$$

Crença na visão de um filme **em conjunto**: De acordo com a intuição

Teoria de Dempster-Schaffer

Problemas com *Crenças conflituosas*:

2- Doutores A e B têm as seguintes crenças sobre um doente:

$$m(A) \{tumor\} = 0.99$$

$$m(B) \{tumor\} = 0.99$$

$$m(A) \{meningite\} = 0.01$$

$$m(B) \{traumatismo\} = 0.01$$

$$m(A) \oplus m(B)$$

		m(B)	
		{tumor}=0.99	{traumatismo}=0.01
m(A)	{tumor}=0.99	{tumor}=0.9801	{ \emptyset }=0.0099
	{meningite}=0.01	{ \emptyset }=0.00099	{ \emptyset }= 0.0001

$$m(A) \oplus m(B) (\{Tumor\}) = 0.9801 / (1 - 0.0199) = 1$$

Certeza absoluta no diagnóstico. **Contra-intuitivo**

Teoria de Dempster-Schaffer

Ressalta da teoria de Dempster-Schafer que:

a medida da **densidade de probabilidade** não é para distribuir toda enquanto não existirem evidências suficientes que o justifiquem!

Portanto não se deve confundir **Incerteza** com **Ignorância** em cada momento, antes ou no momento da decisão

Lógica dos Conjuntos Difusos

PRINCÍPIOS BÁSICOS DA LÓGICA DIFUSA (FUZZY LOGIC)

Lofti Zadeh (1965- Fuzzy sets)

Assim como existe uma forte relação entre a Lógica de Boole e a Teoria dos Conjuntos, também existe uma relação similar entre **Lógica Difusa e a Teoria dos Conjuntos Difusos (Vagos)**.

Na Teoria Clássica dos Conjuntos um **subconjunto S** de um conjunto **C** é definido como o mapeamento entre os elementos de **C** e os elementos do conjunto **{0,1}**,

$$S : C \longrightarrow \{0,1\}$$

Lógica dos Conjuntos Difusos

PRINCÍPIOS BÁSICOS DA LÓGICA DIFUSA (FUZZY LOGIC)

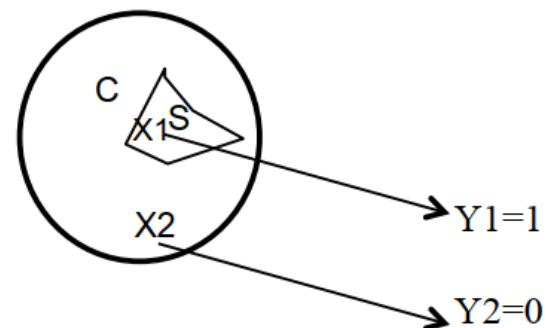
Na T. dos Conjuntos clássica, o mapeamento é representado por um conjunto de **pares ordenados X,Y** em que X é um elemento de C e Y é um elemento do **Conjunto {0,1}**.

A proposição '**a pertence a S**' terá um valor de verdade determinado através do par ordenado X,Y em que Y terá o valor:

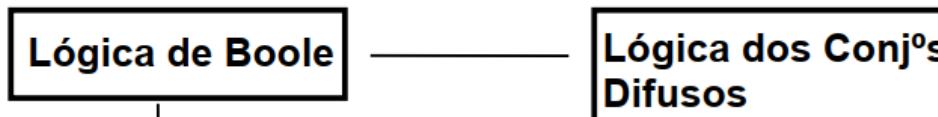
($Y = 1$ e a proposição é **verdadeira**)

ou

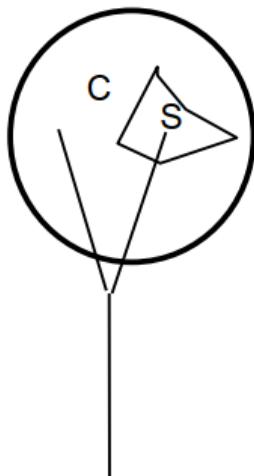
$Y=0$ e a proposição é **falsa**



Lógica dos Conjuntos Difusos



Sub-Conjuntos (*Crisp*)

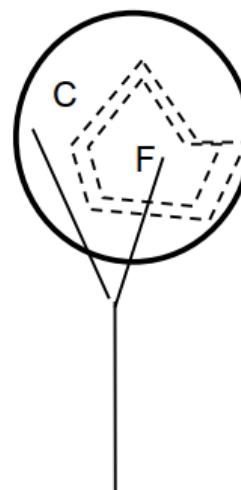


$\{0,1\}$

$S : C \rightarrow \{0,1\}$

O mapeamento é representado por um conjunto de pares ordenados X, Y em que $X \in C$ e $Y \in \{0,1\}$

Sub-Conjuntos Difusos



$[0,1]$

Sub-conjunto F de C é definido como o conjunto de pares ordenados únicos X, Y em que $X \in C$ e $Y \in [0,1]$

intervalo

Lógica dos Conjuntos Difusos

De forma análoga um subconjunto **difuso F** de um conjunto **C** pode ser definido como o conjunto dos **pares ordenados únicos X,Y** em que X é um elemento de C e Y pertence ao **intervalo [0,1]**. O valor 0 representa a completa não pertença de X a F e 1 a completa pertença de X a F. Os valores intermédios representam **GRAUS DE PERTENÇA** de X a F.

O conjunto C é o universo do discurso do conjunto vago F. O mapeamento é chamado **FUNÇÃO DE PERTENÇA** ao conjunto Difuso/Vago F.

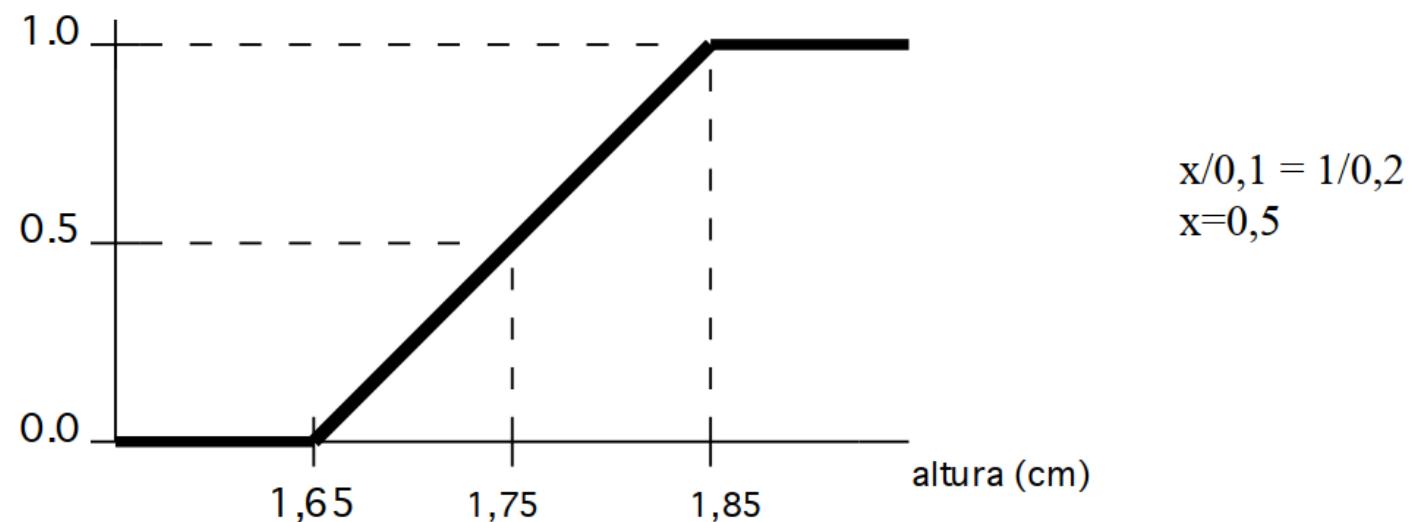
A proposição '**a** pertence a F' tem um valor de verdade dado por Y do par X,Y em que X = **a** e Y está no **intervalo [0,1]**.

Zadeh aplicou estes princípios às **variáveis linguísticas** tais como **ALTO** no universo de discurso das Pessoas. Vamos definir o **subconjunto Difuso F = ALTO** através da seguinte função de pertença:

Lógica dos Conjuntos Difusos

$$\text{alto}(X) = \begin{cases} 0, & \text{se } \text{altura}(X) < 1,65\text{m} \\ \text{altura}(X)-1,65\text{m}/0,2 & \text{se } 1,65\text{m} \leq \text{altura}(X) \leq 1,85\text{m} \\ 1, & \text{se } \text{altura}(X) > 1,85\text{m} \end{cases}$$

O gráfico desta função é:



Lógica dos Conjuntos Difusos

Para esta definição da função de pertença teremos os seguintes exemplos:

<u>Pessoa</u>	<u>altura</u>	<u>grau de pertença ao conjunto "alto"</u>
Elsa	1,50	0.00
Paula	1,70	0.25
Catarina	1,80	0.75
Françoise	1,90	1.00

A expressão '**Catarina é alta**' é interpretada com um grau de verdade de **0.75**.

De notar que as **funções de pertença** normalmente são mais complicadas que esta para '**ALTO**'. Muitas tendem a ser parecidas com **triângulos** apontando para cima.

Muitas vezes, também, a função de pertença não depende de um só critério. Por exemplo, ALTO devia depender também da nacionalidade da pessoa, além da sua altura. Teríamos então uma "**relação difusa**" (Fuzzy relation).

Lógica dos Conjuntos Difusos

Operações Lógicas

Agora que sabemos o que representa em Lógica Difusa uma proposição como X é ALTO, vamos interpretar uma proposição relativa a alturas como:

X é BAIXO E Y é ALTO OU ($\neg Z$ é MÉDIO)

As definições standard das **conetivas** em Lógica Difusa são:

Verdade	$(\neg X) = 1.0 - \text{Verdade}(X)$
Verdade	$(X \wedge Y) = \text{minimo}(\text{Verdade}(X), \text{Verdade}(Y))$
Verdade	$(X \vee Y) = \text{maximo}(\text{Verdade}(X), \text{Verdade}(Y))$

Se usarmos o valor 1 e 0 nestas fórmulas teremos as mesmas **tabelas de verdade** da Lógica de Boole.

Podemos então dizer que os subconjuntos Difusos e a Lógica Difusa são uma verdadeira **generalização** da Teoria Clássica dos conjuntos e da Lógica de Boole.

Todos os subconjuntos(Crisp) são também subconjuntos Difusos e não há nenhum conflito entre os dois métodos.

Lógica dos Conjuntos Difusos

$$\text{alto } (X) = \begin{cases} 0, & \text{se altura}(X) < 1,65\text{m} \\ \text{altura}(X)-1,65\text{m}/0,2, & \text{se } 1,65\text{m} \leq \text{altura}(X) \leq 1,85\text{m} \\ 1, & \text{se altura}(X) > 1,85\text{m} \end{cases}$$

Por exemplo: assumamos a definição dada de ALTO e que a definição de IDOSO é:

$$\text{idoso } (X) = \begin{cases} 0 & \text{se idade}(X) < 18 \text{ anos} \\ (\text{idade}(X)-18)/42 & \text{se } 18 \leq \text{idade}(X) \leq 60 \text{ anos} \\ 1 & \text{se idade}(X) > 60 \text{ anos} \end{cases}$$

seja:

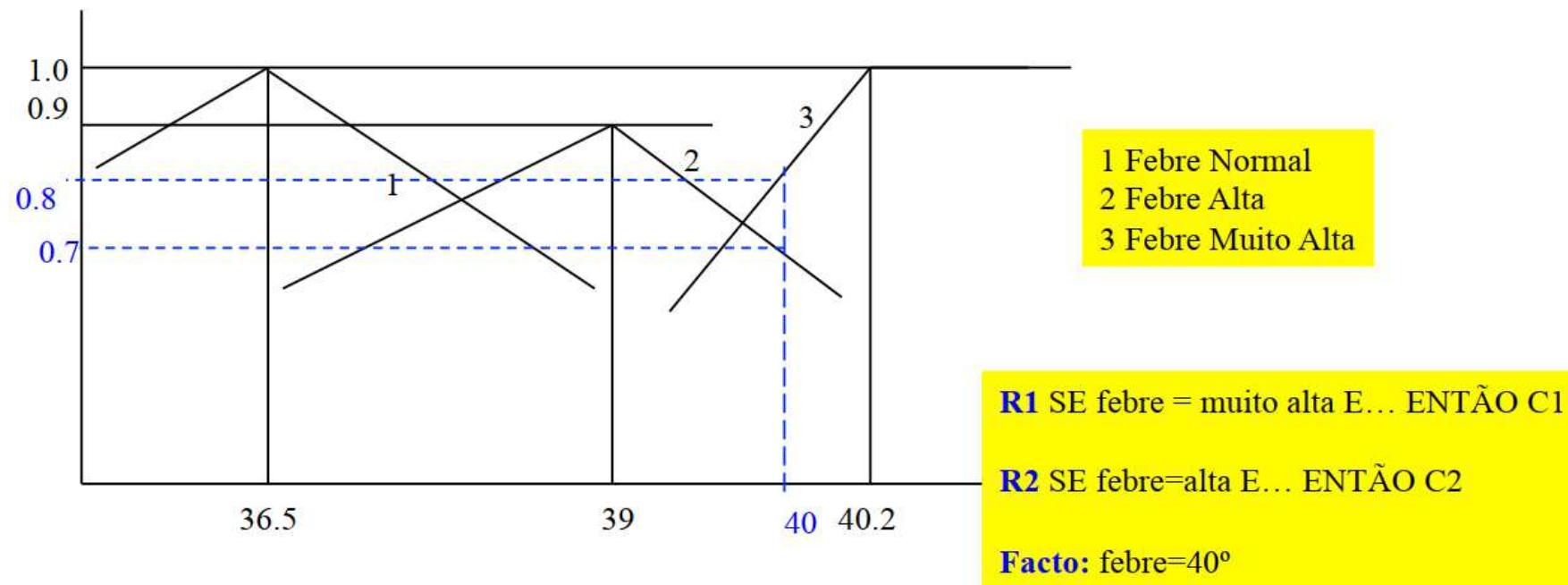
a = X é ALTO E X é IDOSO
b = X é ALTO OU X é IDOSO
c = NÃO X é ALTO

Podemos então computar os seguintes valores:

Alt(X)	id(X)	X é ALTO	X é IDOSO	a	b	c
1,50	65	0.00	1.00	0.00	1.00	1.00
1,70	30	0.25	0.28	0.25	0.28	0.75
1,80	45	0.75	0.64	0.64	0.75	0.25
1,90	45	1.00	0.64	0.64	1.00	0.00
1,50	15	0.00	0.00	0.00	0.00	1.00

Lógica dos Conjuntos Difusos

Febre	Valor Característico	FC
Normal	36.5	1
Alta	39	0.9
Muito Alta	40.2	1



Dependendo das outras condições poderia concluir **C1** e **C2** com diferentes graus de confiança

Lógica dos Conjuntos Difusos

Raciocínio Difuso

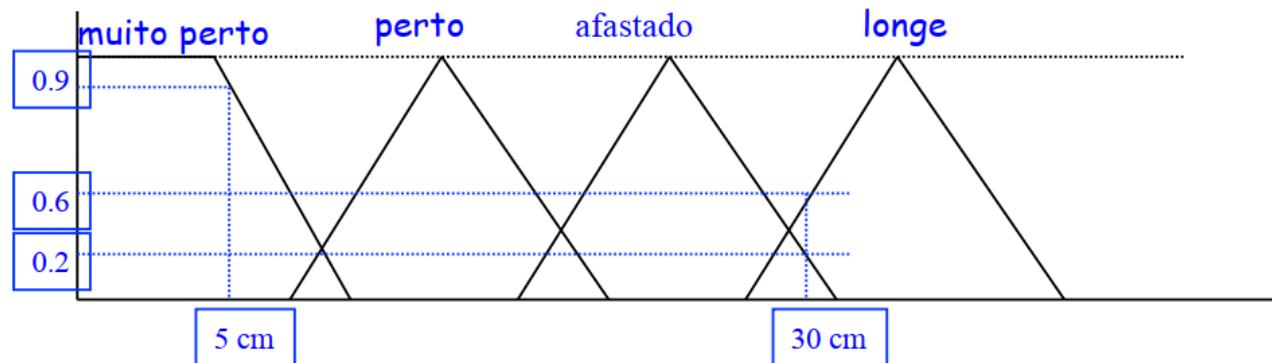
“Desfusificação”: Conversão de saídas “fuzzy” em valores “crisp”

Sejam as Regras **R1** e **R2** de uma BC para controlar um Robô Móvel com sensores S_1 e S_2 reagindo a obstáculos.

R1: SE S_1 =muito perto E S_2 =longe ENTÃO rodar muito esquerda
R2: SE S_1 =muito perto E S_2 =afastado ENTÃO rodar esquerda

DADOS: $S_1=5\text{cm}$ $S_2=30\text{cm}$

FUNÇÕES DE PERTENÇA dos intervalos Difusos para as distâncias medidas pelos sensores:

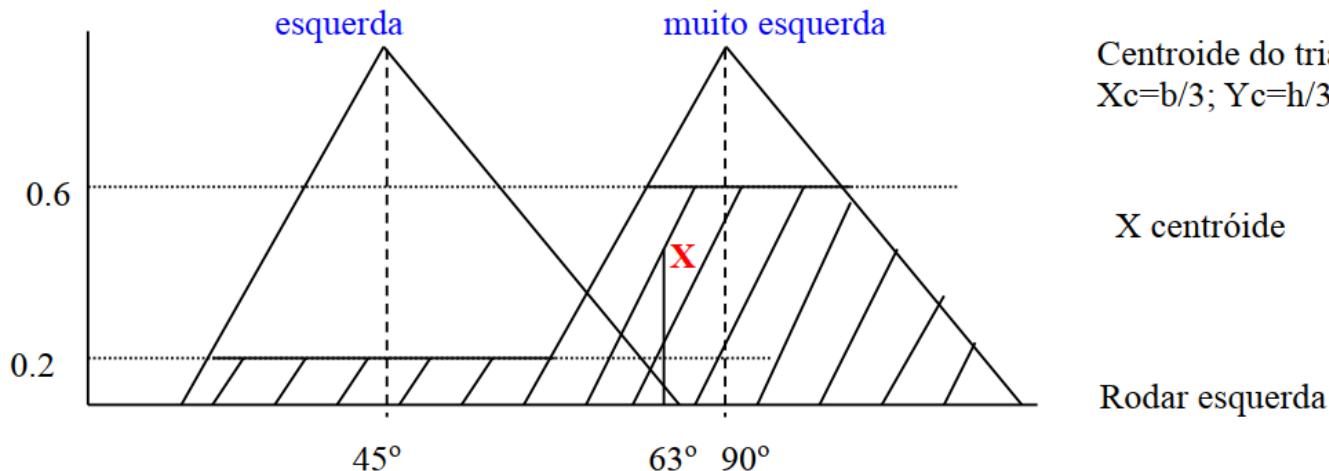


Lógica dos Conjuntos Difusos

Raciocínio Difuso: desfuzificação

Output:

Pela Regra R1	muito esquerda	0.6	$\min(0.6, 0.9)$
Pela Regra R2	esquerda	0.2	$\min(0.9, 0.2)$



Centroide do triângulo retângulo:
 $X_c=b/3$; $Y_c=h/3$

X centróide

Rodar esquerda

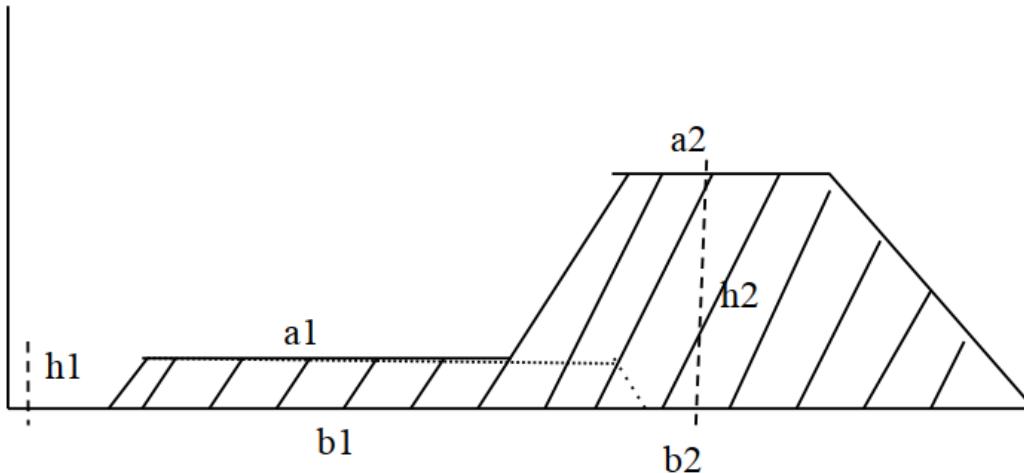
Método do Centróide (aqui simplificado):

Divide-se a área limitada pelos factores de pertença em **áreas elementares** (triângulos e rectângulos). Calculam-se (X_i, Y_i) dos respectivos **centróides**. (X_c, Y_c) do centróide será calculado por: $X_c = (\sum x_i * A_i / Atotal)$ e $Y_c = (\sum y_i * A_i / Atotal)$ em que $Atotal = \sum A_i$ sem sobreposição
Outro método simplificado de “Desfusificação”: $0.2 * 45^\circ + 0.6 * 90^\circ = 9^\circ + 54^\circ = 63^\circ$

Lógica dos Conjuntos Difusos

Cálculo do centróide de trapézios isósceles

$$X_c = h/3 * ((2a+b) / (a+b))$$



Método do Centróide:

Divide-se a área limitada pelos fatores de pertença em áreas elementares (triângulos e rectângulos). Calculam-se (X_i, Y_i) dos respectivos centróides. (X_c, Y_c) do centróide será calculado por: $X_c = (\sum x_i * A_i / A_{total})$ e $Y_c = (\sum y_i * A_i / A_{total})$ em que $A_{total} = \sum A_i$ sem sobreposição

Lógica dos Conjuntos Difusos

Raciocínio Difuso: desfuzificação

Vários métodos podem ser aplicados:

Centro das Somas (COS)

MAX (DOT ou PRODUCT) para quando há um pico.

AVERAGING Média pesada do output das regras

BASEADO NO CENTRÓIDE: que é o mais apropriado pois reflete o peso de cada uma das regras ativadas.

“This method is also known as center of gravity or center of area defuzzification. This technique was developed by **Sugeno** in 1985. This is the most commonly used technique and is very accurate. The centroid defuzzification technique can be expressed as : $x^* = \frac{\int \mu_i(x)x dx}{\int \mu_i(x) dx}$ where x^* is the defuzzified output, $\mu_i(x)$ is the aggregated membership function and x is the output variable. The only disadvantage of this method is that it is computationally difficult for complex membership functions.“

Lógica dos Conjuntos Difusos

Centro das Somas:

(áreas sobrepostas são consideradas múltiplas vezes)

$$X^* = \sum_{i=1..k} A_i * \hat{x}_i / \sum_{i=1..k} A_i$$

A_i representa a área considerada pela regra i

K é o número total de regras ativadas

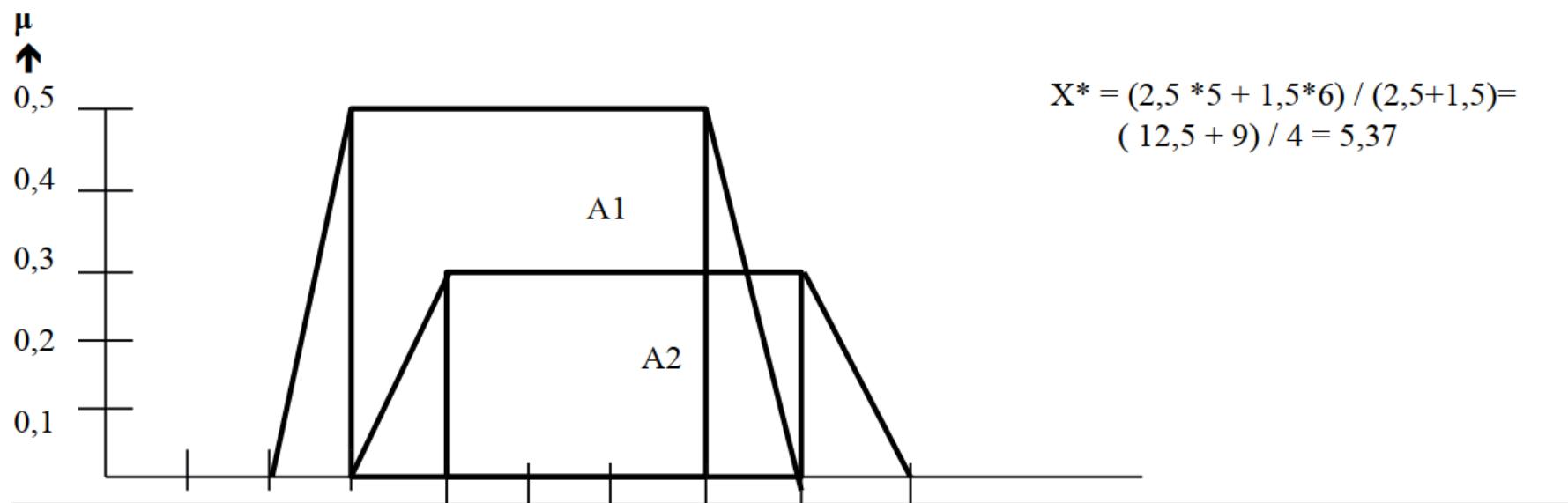
\hat{x} é o centro da área

$$A_1 = 1/2 * [(6+4) * 0,5] = 2,5$$

$$A_2 = 1/2 * [(6+4) * 0,3] = 1,5$$

$$\hat{x}_1 = 5$$

$$\hat{x}_2 = 6$$



Lógica dos Conjuntos Difusos

Centro de Gravidade (COG):

(áreas sobrepostas são consideradas uma única vez)

$$X^* = \sum_{i=1..N} A_i * \hat{x}_i / \sum_{i=1..N} A_i$$

N representa o número de subáreas

A_i representa a área considerada

\hat{x}_i é o centróide da área A_i

centróide:
triângulo retângulo: $b/3, h/3$

$$A_1 = 1/2 * [1 * 0,5] = 0,25$$

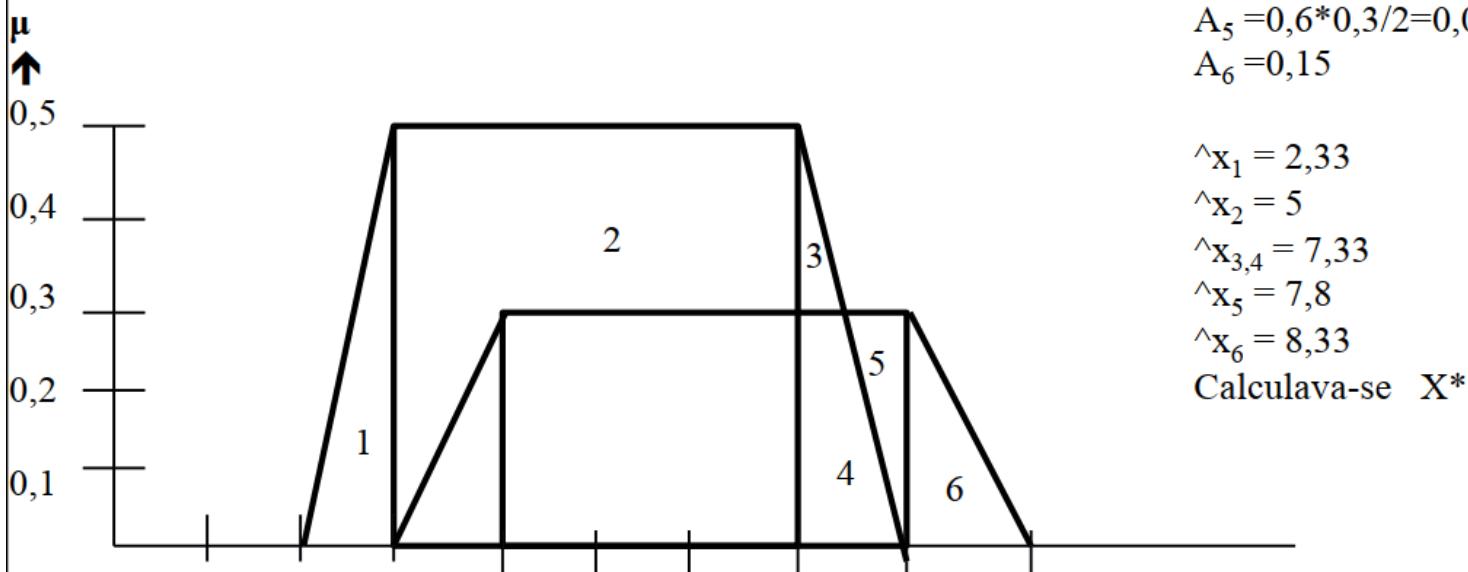
$$A_2 = 4 * 0,5 = 2$$

$$A_3 = 0,4 * 0,2 / 2 = 0,04$$

$$A_4 = (0,25 - 0,04) = 0,21$$

$$A_5 = 0,6 * 0,3 / 2 = 0,09$$

$$A_6 = 0,15$$



Artificial Intelligence/ Inteligência Artificial

Lecture 6: Knowledge Engineering

(slides baseados em Oliveira, 2017 e Cortez 2014)

Luís Paulo Reis

lpres@fe.up.pt

Director of LIACC – Artificial Intelligence and Computer Science Lab.
Associate Professor at DEI/FEUP – Informatics Engineering Department,
Faculty of Engineering of the University of Porto, Portugal
President of APPIA – Portuguese Association for Artificial Intelligence

