

Aula 4

Inteligência Artificial Aplicada

Prof. Dr. Luciano Frontino de Medeiros

Conversa Inicial

- **Manipular símbolos para manifestar comportamento inteligente**
- **Sistemas bem tradicionais dentro da pesquisa da IA**
- **Tem como base a lógica**

▪ **Temas:**

- **Sistemas especialistas**
- **Componentes de um SE**
- **Exemplo de SE**
- **Programação lógica**
- **Cláusulas em Prolog**

Sistemas Especialistas

- **Técnica da IA desenvolvida para resolver problemas em um determinado domínio, cujo conhecimento utilizado é obtido de pessoas que são especialistas naquele domínio**

■ **Primeiros sistemas:**

● **Dendral (1965):**

- ✓ **SE para inferir estruturas moleculares a partir da informação espectrográfica de massa**

(...)

(...)

● **Mycin (1970):**

- ✓ **SE para diagnóstico e tratamento de doenças infecciosas do sangue utilizando 450 regras que permitiam o diagnóstico e a prescrição de tratamentos**

■ **Exemplo de regra do Mycin:**

SE

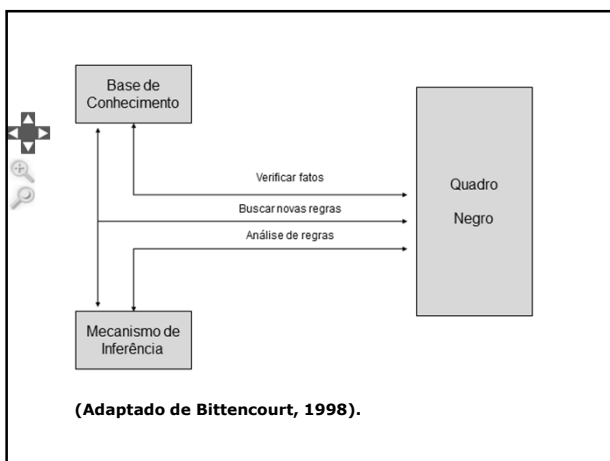
- 1) A infecção é do tipo bacteremia primária E
- 2) O local da cultura é um dos locais estéreis E
- 3) A entrada para o organismo foi pelo trato gastrointestinal

ENTÃO

Existe evidência sugestiva (0.7) que a identidade do organismo seja um bacteroide.

(Adaptado de Bittencourt, 1998).

Componentes de um SE



■ **Atividades do motor de inferência:**

- **A seleção das regras a partir dos dados de entrada**

(...)

(...)

- A resolução de conflitos indicando quais regras serão efetivamente executadas (priorização e ordenação)
- A ação propriamente dita

▪ **Etapas para a construção de um SE:**

- Identificação e definição do domínio do problema
- Aquisição do conhecimento (...)

(...)

- Organização e representação do conhecimento
- Implementação do SE
- Testes e validação

▪ **Organização do conhecimento:**

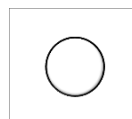
- Fatos são usados para descrever relacionamentos entre estruturas de conhecimento mais complexas e controlar o uso destas durante a resolução de problemas

▪ **Proposição:**

- Em IA e SE, um fato é às vezes referenciado como uma proposição
- Uma proposição é uma declaração que pode ser verdadeira ou falsa

▪ **Exemplo de proposição:**

- Proposição:
"A cor da bola é vermelha"
- Mundo:
- Valor: falso



■ **Notação O-A-V:**

- "A cor da bola é vermelha":
 - ✓ Objeto: bola
 - ✓ Atributo: cor
 - ✓ Valor: vermelha

■ **Regras:**

- Sequências lógicas compostas por premissas (antecedentes) e conclusões (consequentes) (...)

(...)

- É comum o uso da expressão condicional "se-então" para representar regras dentro de um SE

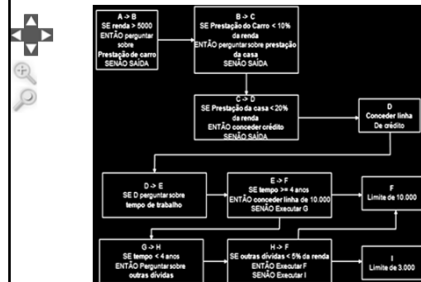
■ **Exemplos de regras:**

SE (distância à frente < 10 cm) **ENTÃO** (vire 180 graus)
SE (temperatura > 50 graus) **ENTÃO** (desligue aquecedor)

- Regras determinísticas
- Regras podem ter ainda um coeficiente de confiança ou probabilidade: regras probabilísticas

Exemplo de SE

■ **Análise de crédito:**

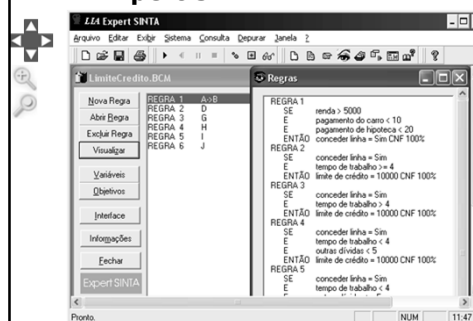


(Adaptado de O'Brien, 2007).

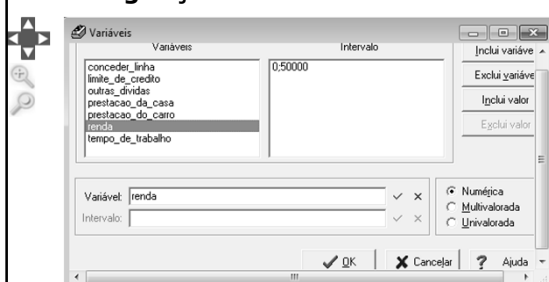
■ Variáveis do problema:

- Conceder linha
- Limite de crédito (objetivo)
- Outras dívidas
- Prestação de imóvel
- Prestação do carro
- Renda
- Tempo no emprego

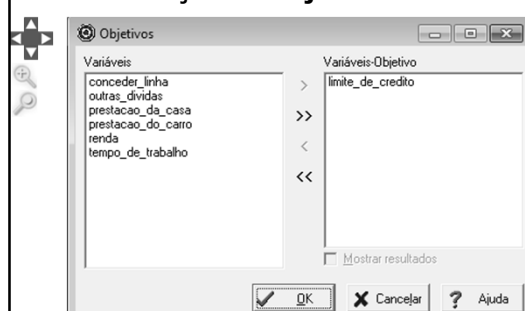
■ Expert SINTA:



■ Digitação das variáveis:



■ Definição do objetivo:



■ Programa do SE:

```

REGRA 1
SE
    renda > 5000
E
    prestacao_do_carro < 10
E
    prestacao_da_casa < 20
ENTÃO
    conceder_linha = Sim CNF 100%
REGRA 2
SE
    conceder_linha = Sim
E
    tempo_de_trabalho >= 4
E
    limite_de_credito = 10000 CNF 100%
ENTÃO
    conceder_linha = Sim
E
    tempo_de_trabalho < 4
E
    outras_dívidas < 5
ENTÃO
    limite_de_credito = 10000 CNF 100%
REGRA 3
SE
    conceder_linha = Sim
E
    tempo_de_trabalho < 4
E
    outras_dívidas < 5
ENTÃO
    limite_de_credito = 3000 CNF 100%
REGRA 4
SE
    conceder_linha = Sim
E
    tempo_de_trabalho < 4
E
    outras_dívidas >= 5
ENTÃO
    limite_de_credito = 0 CNF 100%
REGRA 5
SE
    conceder_linha = Não
OU
    conceder_linha = DESCONHECIDO
ENTÃO
    limite_de_credito = 0 CNF 100%
    
```

Programação em Lógica (Prolog)

- **Representação de um problema expresso por meio de um conjunto finito de um tipo especial de sentenças lógicas denominadas de cláusulas**

- **Lógica x controle:**
 - **Componente lógico:** corresponde à definição do que deve ser solucionado
 - **Componente de controle:** estabelece como a solução pode ser obtida

- **Prolog:**
 - **Na programação em Prolog, o programador precisa somente descrever o componente lógico de um algoritmo, deixando o controle da execução para ser exercido pelo sistema de programação em lógica utilizado**

Cláusulas em Prolog

- **Fatos:** denotam uma verdade incondicional
 - **Regras:** definem as condições que devem ser satisfeitas para que uma declaração seja considerada verdadeira
- (...)

(...)

- **Consulta:** interrogação ao programa para verificar a verdade do conhecimento nele contido

Cláusula	Exemplo	Descrição
Fato	<code>pai(joão, luiz).</code>	Este fato é traduzido como "João é o pai de Luiz".
Regra	<code>filho(X,Y) :- pai(Y,X).</code>	Se X é filho de Y, então Y é pai de X. O sinal ":-" funciona como uma "seta à esquerda", donde a conclusão à direita é obtida a partir das premissas que estão na direita.
Consulta	<code>?- filho(luiz, joão). True</code>	A consulta tenta obter se, a partir dos fatos e das regras existentes na base de conhecimento, Luiz é filho de João. A regra acima deduz um novo fato, a partir do fato e da regra anterior.

■ Componentes de uma cláusula:

- **Corpo:** lista de objetivos separados por vírgulas, que devem ser interpretados por conjunções (...)

(...)

- **Cabeça:** contém o resultado do corpo, inferido a partir dos objetivos

■ Variáveis:

- **X1**
- **Resultado**
- **Objeto2**
- **Lista_Extensa**
- **_var35**
- **_344**
- **_**

■ SWI-Prolog

```

SWI-Prolog (AMD64 Multi-threaded, version 6.2.6)
File Edit Settings Run Debug Help
% library(win_menu) compiled into win_menu 0.00 sec, 29 clauses
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.2.6)
Copyright (c) 1990-2012 University of Amsterdam, Vrije Universiteit
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic) . or ?- apropos(Word) .

1 ?-
1 ?- c:/Users/Moser/Desktop/Aulas/ia/Prolog/programs/fatorial.pl compiled 0.00 sec, 3 clauses
1 ?- factorial(5,X).
X = 120 .
2 ?-

```

