



Universidade do Minho

@ Paulo Cortez & Manuel Filipe Santos, 2009

Aula 6: Representação de Conhecimento via Regras de Produção: explicação e incerteza

OBJECTIVOS:

- **Aplicar** Regras de Produção para Representar Conhecimento: explicação e incerteza



GERANDO EXPLICAÇÕES

Existem 2 tipos de explicações: o “**como**” e o “**porquê**”

- Quando um **Sistema Inteligente** obtém uma resposta, o utilizador pode perguntar: Como encontraste a resposta?
- A explicação típica consiste em **apresentar o caminho** que levou à obtenção da resposta;
- Este tipo de explicação é uma **árvore de prova** sobre como uma conclusão foi obtida a partir dos factos e das regras;



Geração de Explicação

1. Se **P** é um **facto** 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 <= Prova_de_Cond**, onde
Prova_de_Cond é a árvore de prova da condição **Cond**
3. Se **P1** e **P2** são proposições cujas árvores de prova são
Prova1 e **Prova2**, então:
se **P** é **P1 e P2** a árvore de prova é **Prova1 e Prova2**.
se **P** é **P1 ou P2** a árvore de prova é **Prova1 ou Prova2**.



Interpretador (Sistema Inferência) com Geração de Provas (Explicação):

Será utilizado o procedimento **demo(T,P)**, onde T é o teorema a comprovar e P é a prova:

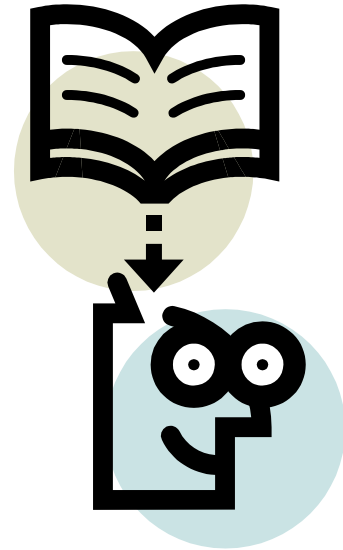
```
demo(T,T) ← facto(T).  
demo(T,T≤ProvaCond) ← se Condição então T,  
demo(Condição,ProvaCond).
```

```
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).
```



Universidade do Minho

Exemplo: Passagem a uma disciplina



Regras adquiridas via alunos:

- Sei que passo a uma disciplina se tenho positiva no exame e estou sem faltas;
- Também sei que se estudei então irei ter positiva;
- Por último, se não estudei então tenho negativa;

Eis os factos: o Rui estudou e está sem faltas.



Universidade do Minho

Q1: Elaborar as regras de produção e factos sobre a disciplina. Depois, elaborar a questão:
1 demo(passo,P) ou seja como passou o Rui?, determinando o que deve ser P

(In-Class Teams, 2 min)

- Juntem-se em grupos de 3 elementos
- Descubram quem nasceu mais próximo do dia 1 de Agosto
→ representante do grupo (caneta e papel)
- Resolver o exercício





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:

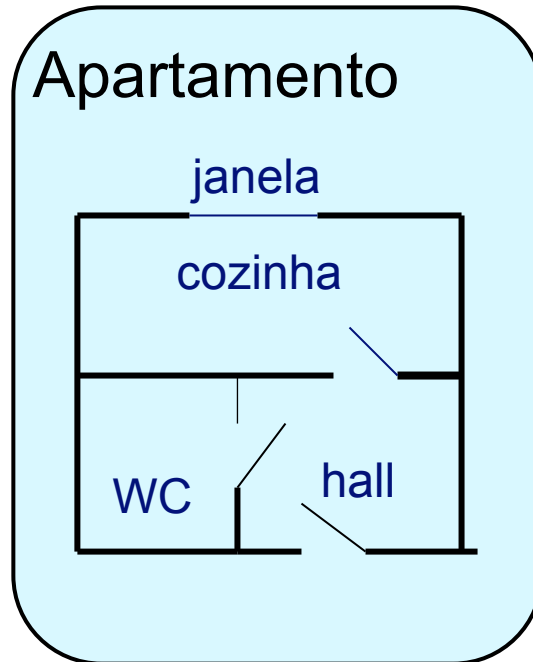
Para **factos**: **facto(Proposição:C)**.

Para **regras**: **se Condição então Acção:C**

sendo **C** um número entre 0 e 1.



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.

Eis a situação actual: o WC está completamente seco, o hall está cerca de 70% molhado.

Método



Activo

Q2: Elaborar regras de produção e factos sobre a fuga de água.

O que se pode derivar?

(Think-Pair-Share, 1min+1min)



Interpretador de Regras de Produção para Incerteza

Em qualquer representação de incerteza é necessário combinar as certezas das proposições e regras.

Um esquema simples consiste nas seguintes regras:

- $P1 \text{ e } P2 \rightarrow \min(c(P1), c(P2))$
- $P1 \text{ ou } P2 \rightarrow \max(c(P1), c(P2))$
- Se existir uma regra:

se $P1$ então $P2:C \rightarrow c(P2)=c(P1) * C$



Interpretador de Regras de Produção para Incerteza

Procedimento do tipo $\text{demo}(\text{Proposição}, \text{Certeza})$:

$\text{demo}(\text{P}, \text{C}) \leftarrow \text{facto}(\text{P}:\text{C}).$

**$\text{demo}(\text{Cond1 e Cond2}, \text{C}) \leftarrow$
 $\text{demo}(\text{Cond1}, \text{C1}), \text{demo}(\text{Cond2}, \text{C2}), \text{min}(\text{C1}, \text{C2}, \text{C}).$**

**$\text{demo}(\text{Cond1 ou Cond2}, \text{C}) \leftarrow$
 $\text{demo}(\text{Cond1}, \text{C1}), \text{demo}(\text{Cond2}, \text{C2}), \text{max}(\text{C1}, \text{C2}, \text{C}).$**

**$\text{demo}(\text{P}, \text{C}) \leftarrow$
se Cond então $\text{P}:\text{C1}$, $\text{demo}(\text{Cond}, \text{C2})$, $\text{C is } \text{C1} * \text{C2}.$**



Universidade do Minho

Q3: Qual a probabilidade de problema na cozinha? demo(problema_cozinha,X).

(Think-Pair-Share, 30s+30s)

-Resolver o exercício





Problemas com a Incerteza

A proposta apresentada pode ser criticada por ser demasiado simples. Por exemplo:

- Imagine que existem os factos **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).



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

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.
```



Para saber mais...

Universidade do Minho

- Consultar o Capítulo 14, Prolog – [Bratko, 1990]: Bratko, Ivan, Prolog – Programming for Artificial Intelligence, Longman, 1990.
- **Mais exemplos de exercícios resolvidos com Regras de Produção na sebenta a disponibilizar na página da disciplina...**