Universidade do Minho

Aula 5: Representação de Conhecimento via Regras de Produção

OBJECTIVOS:

- **Aplicar** Regras de Produção para Representar Conhecimento: factos, regras, sistemas de inferências backward e forward chaining.

REGRAS DE PRODUÇÃO

- 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, …
- Exemplo:

  - **se** o paciente tem febre **e**
    o paciente tem dores **e**
    o paciente não tem infecções detectáveis
    então **diagnosticar gripe.**

Universidade do Minho

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**)

Universidade do Minho

Como utilizar Regras de Produção?

- **A Base de Conhecimento é feita de regras e factos;**
- **Considera-se a seguinte sintaxe lógica**:

se Condição então Conclusão.

Sendo que uma **Condição** pode ser:

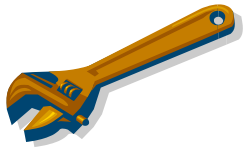
- um predicado lógico (e.g. Prolog)
- uma conjunção de duas condições: **Cond1 e Cond2**
- uma disjunção de duas condições: **Cond1 ou Cond2**

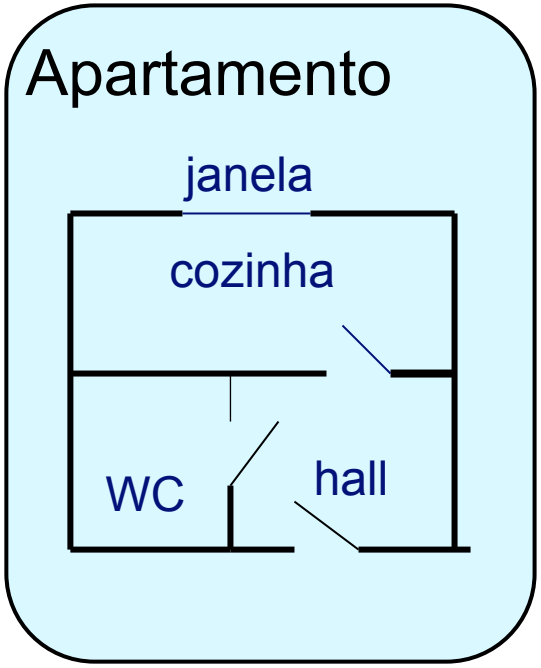
Representar factos (dados): facto(D).

D é algo que actualmente é verdadeiro

Universidade do Minho

Caso de Estudo: diagnosticar fugas de água num apartamento





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;
- 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 um problema na cozinha e a janela está fechada;

Eis a situação actual: o hall está molhado, a casa de banho seca e a janela fechada.

Universidade do Minho

Q1: Quais as regras e factos para um SBC de diagnóstico de fugas do apartamento? Actualmente o que se pode deduzir? (In-Class Teams 3 min)

-Grupos de 3 elementos, o que morar morar mais perto desta sala é o representante (caneta e papel);
-Escrever as regras de produção do problema;
-Escrever os factos;
-Inferir o que se pode deduzir;

Universidade do Minho

Sistemas de Inferência das Regras de Produção

Existem dois sistemas de inferência (formas de raciocínio):

- **Backward chaining** - a partir de uma hipótese (uma questão) o raciocínio retrocede na rede de inferência até aos factos, ou seja, tentamos chegar das conclusões às condições (do **então** para o **se**…)

- **Forward chaining** - a partir dos factos 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 factos derivados (do **se**… para o **então**)

Universidade do Minho

Interpretador (Sistema Inferência) Backward Chaining:

Será utilizado o 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).

Universidade do Minho

Q2: Executar o conjunto de passos para demonstrar: ¬demo(fuga_na_cozinha) ou seja há fuga na cozinha?

(In-Class Teams, 2 min)

-Juntem-se em grupos de 3 elementos

-Descubram quem nasceu mais próximo do dia 1 de Janeiro → representante do grupo (caneta e papel)

-Escrever o conjunto de passos que o interpretador segue

Universidade do Minho

Interpretador (Sistema Inferência) Forward Chaining:

Será utilizado o procedimento **demo**, que deriva todos os factos novos:

```
demo← novo_facto(P), !,
          write('derivado:'), write(P), nl,
          assert(facto(P)),
          demo.
demo ← write('não há mais factos').

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

Universidade do Minho

Exemplo de Forward Chaining:

```
| demo.

derivado: problema_na_cozinha
derivado: não_entra_água
derivado: inundação
não há mais factos
```

Universidade do Minho

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 facto à sua negação:

facto(não_doente)← não(demo(doente)).

Universidade do Minho

Regras de Produção implementadas 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
- É necessário definir **fact** como dinâmico: **:- dynamic(fact/1).**
- ← **:-**
- não **\+**

Universidade do Minho

Regras de Produção em Prolog:

casa.pl:

```
:- dynamic fact/1.

% regras: base de conhecimento
if hall_molhado and cozinha_seco then fuga_banho.
if hall_molhado and banho_seco then problema_cozinha.
if janela_fechada or nao_chove then nao_agua.
if hall_seco and banho_molhado then torneira_aberta.
if problema_cozinha and janela_fechada then inundacao.

% factos: base de dados, situação actual
fact(hall_molhado).
fact(banho_seco).
fact(janela_fechada).
```

Universidade do Minho

Regras de Produção em Prolog:

backward.pl:

```
% 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).
```

Universidade do Minho

Regras de Produção em Prolog:

forward.pl:

```
%   Simple forward chaining in Prolog

demo:- new_derived_fact( P), !,          % 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?
```

Universidade do Minho

Regras de Produção em Prolog:

forward.pl (cont.):

```
composed_fact( Cond)  :-
  fact( Cond).                                  % Simple fact
composed_fact( Cond1 and Cond2)  :-
  composed_fact( Cond1),
  composed_fact( Cond2).                        % Both conjuncts true

composed_fact( Cond1 or Cond2)  :-
  composed_fact( Cond1)
  ;
  composed_fact( Cond2).
```

Para saber mais...

- 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…**