



Universidade do Minho

@ Paulo Cortez, 2010

Aula 9: Procura num Espaço de Soluções/Estados: gerar e testar

OBJECTIVOS:

- Solucionar Problemas que envolvam uma procura num espaço de soluções/estados via uma estratégia de gerar e testar



Gerar e Testar (Generate and Test)

Existe uma classe de problemas que pode ser resolvida com a técnica **gerar e testar**:

- a técnica consiste em usar o **não determinismo** inerente à linguagem lógica (e.g. Prolog) para resolver problemas cujo espaço de soluções **não seja muito elevado** nem contenha **ramos infinitos**;
- o não determinismo consiste no **mecanismo de procura de soluções e backtracking**, ou seja, percorre **todo o espaço de procura até encontrar uma solução**;



Universidade do Minho

EXEMPLO:

- Qual o caminho mais curto num grafo?



Do Porto, em direcção ao norte, pode-se ir para Famalicão ou Esposende. De Famalicão pode-se ir para Barcelos ou Braga. De Barcelos pode-se ir para Esposende, Braga ou Viana. De Esposende pode-se ir para Barcelos ou Viana. De Braga pode-se ir a Valença. De Valença pode-se ir a Viana ([Estradas do Norte de Portugal](#))

Q1: Liste todos os caminhos entre Porto e Viana e indique qual o que atravessa menos cidades?

(In-Class Teams 1 min)

- Grupos de 3 elementos, o que morar mais perto de Viana é o representante (caneta e papel).



Resolução do Caminho mais curto:

```
:- [caminho]. % aceder ao predicado caminho(X,Y,C)  
                % X e Y são cidades; C é o caminho  
  
% ligações entre as cidades  
arco(porto,famalicao).  
arco(porto,esposende).  
arco(famalicao,braga).  
arco(famalicao,barcelos).  
arco(braga,valenca).  
arco(barcelos,viana).  
arco(barcelos,braga).  
arco(barcelos,esposende).  
arco(esposende,viana).  
arco(esposende,barcelos).  
arco(valenca,viana).  
  
% calcula o comprimento de um dado caminho: comprimento(C,N)  
comprimento([],0).  
comprimento([_|R],N):- comprimento(R,N1), N is N1+1.
```



Caminho mais curto via Gerar e Testar



Q2: Implementar o predicado `caminhocurto(X,Y,C)`, que calcula o caminho mais curto (menos cidades) entre X e Y, via uma estratégia de **gerar e testar.**

(In-Class Teams 3 min)

- Grupos de 3 elementos, o que morar mais perto do Porto é o representante (caneta e papel).

Sugestão: `caminhocurto(X,Y,C):- _____, % gerar,
_____. %testar`



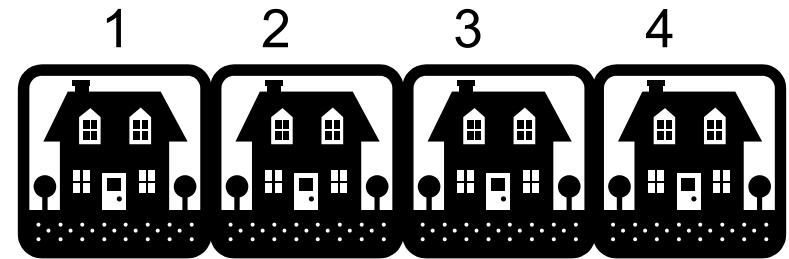
Caminho mais curto via Gerar e Testar

Solução:

```
caminhocurto(X,Y,C):-caminho(X,Y,C), % gerar
                        maiscurto(X,Y,C).% testar
maiscurto(X,Y,C):- comprimento(C,NC),!,
                    \+ menor(X,Y,NC).
menor(X,Y,NC):- caminho(X,Y,C1),
                 comprimento(C1,NC1),
                 NC1<NC.
```



Universidade do Minho



Puzzle Lógico (simplificado)

- Uma rua tem 4 casas em fila, de 1 (esquerda) a 4 (dir.);
- Cada casa foi pintada com uma cor diferente e tem um homem com uma nacionalidade e animal de estimação;
- Sabe-se que:
 - O inglês vive na casa vermelha;
 - O espanhol tem um cão;
 - Na casa verde existe um cavalo e encontra-se numa das extremidades (dir. ou esq.);
 - O português vive na casa mais à direita;
 - A casa laranja fica imediatamente à direita da casa verde;
 - A casa no meio da rua (2 ou 3) tem uma zebra;
 - O português adora a sua casa azul e raposa de estimação;
 - O japonês mora imediatamente à esquerda do espanhol;



Q3: Qual a solução? (In-Class Teams 3 min)

- Grupos de 3 elementos, o que morar mais perto de Braga é o representante (caneta e papel).



Predicado SWI-Prolog SELECT

select(X,L1,L2)

onde: X – é um membro de L1

L1 – é uma lista

L2 – é a Lista L1 excepto X.

▪Exemplos:

```
?- select(2,[1,2,3],L).
```

```
L = [1,3]
```

```
?- select(X,[1,2,3],[1,3]).
```

```
X = 2
```

```
?- select(2,L,[1,3]).
```

```
L = [2,1,3]
```




Solução via Gerar e Testar em Prolog

```
nacionalidades([ingles,espanhol,portugues,japones]).  
cor([vermelha,verde,laranja,azul]).  
animal([cao,raposa,cavalo,zebra]).  
mora([1,2,3,4]).
```

**Representação
do Problema**

```
casas(R):- casa(R), % generate  
           testa(R).% test
```

**Gerar
Testar**

```
casa(R):- nacionalidades(LN),cor(LC),animal(LA),mora(LM),  
          casa(LN,LC,LA,LM,R).
```

```
casa(LN,LC,LA,LM,[c(N,C,A,M)|R]):-  
    select(N,LN,LN2),  
    select(C,LC,LC2),  
    select(A,LA,LA2),  
    select(M,LM,LM2),  
    casa(LN2,LC2,LA2,LM2,R).  
casa([],[],[],[],[]).
```

Gerar



Solução via Gerar e Testar em Prolog

```
testa(R):- member(c(ingles,vermelha,_,_),R),
            member(c(espanhol,_,cao,L1),R),
            member(c(_,verde,cavalo,L2),R),extremo(L2),
            member(c(_,laranja,_,L3),R),direita(L2,L3),
            member(c(_,_,zebra,LZ),R),meio(LZ),
            member(c(portugues,azul,raposa,4),R),
            member(c(japones,_,_,L4),R),esquerda(L1,L4).
```

Testar

% predicados auxiliares, onde X tem de estar instanciado:

meio(2).

meio(3).

extremo(1).

extremo(4).

esquerda(X,Y):- Y is X-1,dentro(Y),!.

direita(X,Y):- Y is X+1,dentro(Y),!.

dentro(X):- X<5,X>0.



Limitações do Gerar e Testar

- Quando o número de soluções possíveis é elevado, esta técnica pode não funcionar em tempo útil;
 - A solução passa por gerar somente soluções que são válidas, mas isto não é generalizável e depende do problema em causa;
 - Existe um ramo da **Inteligência Artificial** chamado **Constraint Logic Programming**, que analisa técnicas sofisticadas para lidar com problemas com restrições.
- **Mais exemplos de exercícios resolvidos de Procura via Gerar e Testar na sebenta a disponibilizar na página da disciplina...**