Inteligência Artificial

2º Trabalho

Docente: Irene Rodrigues
Trabalho elaborado por:
Marcelo Bábau 30372
Maria Silveiro 33222

Março de 2017

Respostas às perguntas

1. Represente este problema como um problema de satisfação de restrições em prolog. No relatório indique como é que representa os estados, as variáveis (nome, domínio e valor) e as restrições. Também deve indicar no relatório como calcula o estado inicial e apresentar o operador sucessor.

Domino: dominio([1,2,3,4,5,6,7,8,9]).

Restrições: As restrições do quadrado mágico são que no quadrado todos os elementos tem que ser diferente, e a soma das colunas e linhas tem que dar 15 (neste exemplo).

```
Verifica se todos os elementos são diferentes
all_diff([]).
all_diff([v(_,_,V)|Afect]):- member(v(_,_,V),Afect),!,fail.
all_diff([_|Afect]):- all_diff(Afect).

Verifica se as somas são 15
valida_somas(L):-
        findall(V,member(v(n(1,_),_,V), L),L1), linha(L1,6),
        findall(V,member(v(n(2,_),_,V), L),L2), linha(L2, 15),
        findall(V,member(v(n(3,_),_,V), L),L3), linha(L3, 24),

        findall(V,member(v(n(_,1),_,V), L),C1), coluna(C1,12),
        findall(V,member(v(n(_,2),_,V), L),C2), coluna(C2,15),
        findall(V,member(v(n(_,3),_,V), L),C3), coluna(C3,18).

linha( [V1,V2,V3], N ):-!, N is  V1+V2+V3.
linha(_,_).

coluna( [V1,V2,V3], N ):- !, N is V1+V2+V3.
coluna(_,_).
```

2. Resolva o problema com o algoritmo de backtracking.

O backtracking é um algoritmo de pesquisa não informada. O algoritmo utilizado neste projeto, foi o fornecido pela docente em uma aula prática anterior.

3. Resolva o problema modificando o algoritmo anterior para que faça verificação para a frente (forward checking).

O forward checking, tem o objetivo de limitar o domínio das variáveis à medida que as variáveis vão sendo afetadas. Ou seja, quando uma variável é afetada, os domínios das outras variáveis têm os valores que se encontravam no domínio com exceção do valor colocado na variável afetada.

4. Modifique o algoritmo anterior como entender de forma a melhorar a complexidade (temporal e espacial).

Não é possível melhorar a complexidade porque o domínio é sempre igual para todas as variáveis, assim não é possível escolher o que tem menor domínio, de modo a melhorar a complexidade espacial e temporal.

5. No relatório apresente os resultados para exemplos diferentes:

%%pesquisa Forward Checking

[v(n(3,3),[9],9),
v(n(3,2),[8,9],8),
v(n(3,1),[7,8,9],7),
v(n(2,3),[6,7,8,9],6),
v(n(2,2),[5,6,7,8,9],5),
v(n(2,1),[4,5,6,7,8,9],4),
v(n(1,3),[3,4,5,6,7,8,9],3),
v(n(1,2),[2,3,4,5,6,7,8,9],2),
v(n(1,1),[1,2,3,4,5,6,7,8,9],1)]

%%pesquisaBack

[v(n(3,3),[1,2,3,4,5,6,7,8,9],9),
v(n(3,2),[1,2,3,4,5,6,7,8,9],8),
v(n(3,1),[1,2,3,4,5,6,7,8,9],7),
v(n(2,3),[1,2,3,4,5,6,7,8,9],6),
v(n(2,2),[1,2,3,4,5,6,7,8,9],5),
v(n(2,1),[1,2,3,4,5,6,7,8,9],4),
v(n(1,3),[1,2,3,4,5,6,7,8,9],3),
v(n(1,2),[1,2,3,4,5,6,7,8,9],2),
v(n(1,1),[1,2,3,4,5,6,7,8,9],1)]

Conclusão

Este projeto está de acordo com os objetivos esperados. Para testar tem que carregar o ficheiro pesquisaBack.pl no Prolog e depois chamar a função p.

Para alterar de forward checking para backtracking tem que comentar no método back, forwardC e alterar na chamada de função back(E2,So1) para back(E1,So1).