

Relatório

3o. Trabalho de Inteligência Artificial

Jogos de dois jogadores - Jogos com informacao completa e determinísticos

Trabalho realizado por:

Paulo Amaral, nº. 21799
André Mendes, nº. 21659
Marco Águia, nº. 21962

Introdução

O trabalho consiste na representação de um jogo de dois jogadores e a sua escolha como um problema de pesquisa no espaço de estados.

Utilizando como implementação o algoritmo de pesquisa Minimax e AlfaBeta.

A nossa escolha de jogo foi o jogo das Damas, implementado com algumas simplificações. Existe um menor numero de peças que no jogo original, assim como não existem as 'peças damas', podendo assim todas as peças movimentar-se para a frente e para trás.

Apresentamos de seguida a resposta às perguntas colocadas, excepção feita às perguntas 5 e 8 , às quais não conseguimos responder.

Fazendo também um comentário ao código apresentado nos ficheiros damas.pl, jogo.pl, minimax.pl e minimaxcut.pl

Desenvolvimento

1 A estrutura de dados que escolhemos para representação do tabuleiro de damas é uma lista com as posições **ocupadas** do mesmo (assumimos então que posições não representadas estão vazias). Uma posição do tabuleiro representa-se da seguinte forma:

pos(PEÇA,LINHA,COLUNA).

Um estado representa-se guardando a peça a jogar, o tabuleiro actual e o número de peças brancas e pretas existentes.

2 A definição de um estado terminal envolve que ou uma das peças ganhou (ou seja, se o número de peças brancas ou pretas chegou a 0) ou se houve empate (nenhum dos jogadores pode jogar pelo menos uma peça).

3 A função de utilidade está definida no predicado **valor** que devolve 0 em caso de empate, devolve 20 em caso de vitória do **Max** e -20 em caso de vitória do **Min**.

4 Utilizando a implementação dada nas aulas para o caso em que começa o jogador com peça preta a jogar e com um tabuleiro da seguinte forma: [pos(p,1,1),pos(b,2,2)], com 1 peça de cada jogador o resultado é a vitória das peças pretas: jogada(e(b,[pos(p,3,3)],0,1)).

6 A função de avaliação verifica qual é o jogador Max e Min, devolvendo o valor 10 quando o número de peças de MAX é maior que o de MIN e -10 quando acontece o contrário, tendo valor 0 para quando existe empate. Implementamos o Minimax com corte em profundidade 2 no ficheiro Minimaxcut.pl e testámos com o exemplo da pergunta 4.

Como não implementamos a pesquisa Alfa-Beta não nos foi possível comparar os resultados espaço-tempo.

7 O agente inteligente encontra-se implementado no ficheiro jogo.pl para o algoritmo Minimax com corte em profundidade 2 para os estados:

e(p,[pos(p,1,1),pos(p,1,7),pos(b,2,2),pos(b,8,8)],2,2)

e(p,[pos(p,1,1),pos(b,2,2),pos(p,3,3)],1,2)

Sendo o Max as peças pretas e o Min as peças brancas.

Conclusão

Não conseguimos implementar a pesquisa Alfa-Beta, e a pesquisa Minimax não funciona correctamente para estados mais complexos. Assim sendo não respondemos à pergunta 5 e a pergunta 8 não foi também respondida devido à falta de dados (não tínhamos estados suficientes).