INSERT IMAGEM UNI AQUI

Disciplina de Inteligência artificial

Jogo do Galo

INSERT IMAGEM AQUI

1. Jogo do Galo

1.1. Questões

1.1.1. Escolha uma estrutura de dados para representar os estados dos dois jogos.

A estrutura utilizada para representar o estado do jogo do galo é um tuplo com uma lista de posições tabuleiro onde cada posição contem uma referencia (um valor não instanciado), e um caracter “x” ou “o” que indica uma jogada e a ultima peça que foi jogada.

1.1.2. Defina o predicado terminal (estado) que sucede quando o estado é termina para cada jogo.

Um estado é terminal se há uma linha, coluna ou diagonal completa, seja ela com x’s ou com o’s, ou no ultimo dos casos quando todas as posições estão preenchidas, ou seja, quando há um empate. Para o segundo jogo ainda não está definido.

1.1.3. Defina uma função de utilidade que para um estado terminal que deve retornar o valor do estado (ex: -1 perde, 0 empata, 1 ganha), para cada jogo.

A função de utilidade verifica a profundidade na arvora de pesquisa e os casos em que o estado é terminal, com a exceção de empate. Os valores devolvidos pela mesma podem ser 1, 0 ou -1 sendo que 0 representa empate, 1 ganha e -1 perde.

1.1.4. Use a implementação da pesquisa minimax dada na aula prática para escolher a melhor jogada num estado.

Com o algoritmo minimax as jogadas são sempre óptimas, logo se jogarmos contra o computador, no melhor dos casos conseguimos um empate.

1.1.5. Implemente a pesquisa Alfa-Beta e compare os resultados (tempo e espaço) em exemplos com os dois jogos.

Através dos resultados observados, podemos concluir que o minimax, apesar de demorar mais tempo a efetuar as jogadas, faz sempre uma jogada ótima, enquanto que o corte alfa-beta não faz a melhor jogada ótima, mas leva menos tempo a efetuar a mesma.

INSERT TABLES HERE

1.1.6. Defina uma função de avaliação que estime o valor de cada estado do jogo, use os dois algoritmos anteriores com corte em profundidade e compare os resultados (tempo e espaço), com exemplos dos dois jogos.

Dunno what to write here

1.1.7. Implemente um agente inteligente que joga os dois jogos, usando a pesquisa definida na alínea anterior.

Implementamos um agente inteligente que joga contra o jogador num ciclo até que se verifique o terminal.

1.1.8. Apresente uma tabela com o numero de nos expandidos para diferentes estados dos 2 jogos (10 estados de cada jogo no mínimo) com os vários algoritmos

Nothing to see here

2. Outro jogo

2.1. Nothing to see here yet