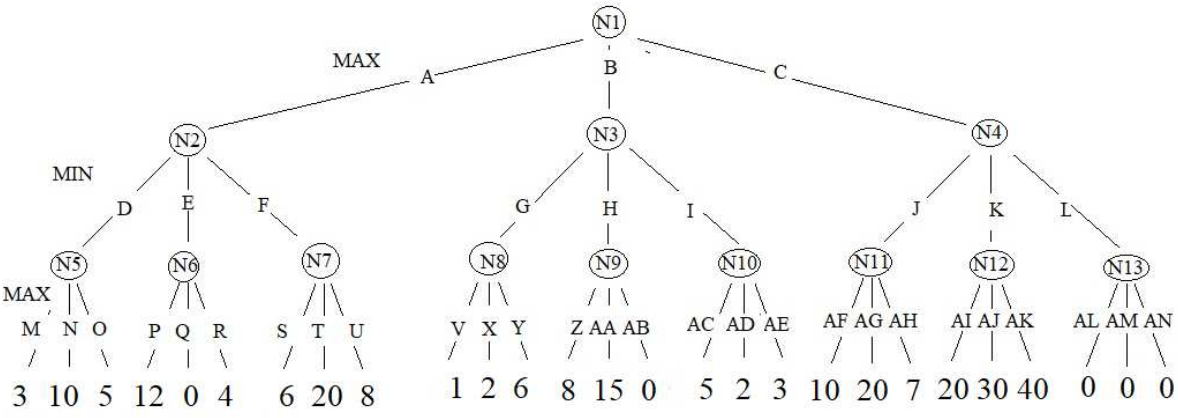
Inteligência Artificial 2019/2020
Ficha de Exercícios 3
Pesquisa com Adversários/Jogos

3. Pesquisa com Adversários/Jogos

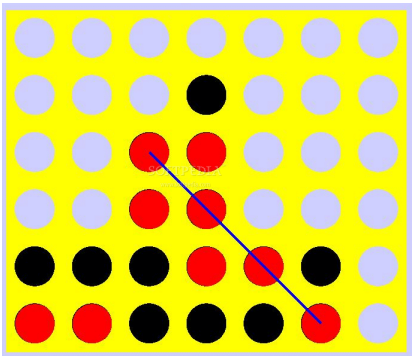
3.1 Funcionamento do Algoritmo Minimax com Cortes Alfa-Beta

Aplique o algoritmo Minimax com cortes Alfa-beta à árvore seguinte que tem um fator de ramificação de 3 no nível superior, 3 no segundo nível e 3 no nível final e com os valores da função avaliação indicados para a linha final. Indique qual o valor final de cada nó e quais os ramos cortados pelos cortes Alfa-Beta.



3.2 Implementação do Jogo "4 em Linha" usando Minimax com Cortes Alfa-Beta

O jogo denominado "Quatro em Linha" ("Connect Four" na versão em língua inglesa) (https://en.wikipedia.org/wiki/Connect_Four) é jogado num tabuleiro vertical de 7x6 casas (i.e. 7 casas de largura e 6 casas de altura), por dois jogadores, aos quais são atribuídas inicialmente 21 peças a cada (um dos jogadores possui peças brancas, e o outro peças pretas ou peças "X" vs peças "O").



Os dois jogadores jogam alternadamente, uma das suas peças. A peça a jogar só pode ser colocada ou na base do tabuleiro, ou numa casa imediatamente acima de outra já ocupada (ver figura anterior). O vencedor será o jogador que conseguir obter uma linha de 4 peças da

sua cor/símbolo na horizontal, vertical ou diagonal. Caso sejam jogadas as 42 peças sem que nenhum jogador consiga uma linha, o resultado final será um empate.

a) Formule este jogo como um problema de pesquisa com adversários, indicando a representação do estado, jogadas/operadores (respetivas pré-condições e efeitos), e o teste objetivo.

b) Implemente uma versão simples do jogo "quatro em linha" numa linguagem de programação à sua escolha.

c) Implemente as seguintes funções:

c1) `int nlinhas4(int Jog)` que dado o estado do tabuleiro calcula o número de linhas com 4 peças (horizontais, verticais, diagonais) de um dado jogador

c2) `int nlinhas3(int Jog)` semelhante à função anterior mas que calcula o número de conjuntos de 4 peças consecutivas que tenham três peças do jogador e uma vazia, i.e. sejam possibilidades de vencer o jogo.

c3) `int central(int Jog)` que atribui 2 pontos a cada peça do jogador Jog na coluna central do tabuleiro (coluna 4) e um ponto a cada peça nas colunas em redor desta (colunas 3 e 5).

d) Implemente um agente para jogar o jogo utilizando o algoritmo minimax com cortes alfa-beta.

e) Compare os resultados de agentes, a jogar 10 partidas deste jogo entre si, utilizando o algoritmo minimax com cortes alfa-beta, com níveis (2, 4, 6 e 8), e as seguintes funções de avaliação:

```
Agente1: fav1 = nlinhas4(1) – nlinhas4(2)
Agente2: fav2 = 100*fav1 + nlinhas3(1) – nlinhas3(2)
Agente3: fav3 = 100*fav1 + central(1) – central(2)
Agente4: fav4 = 5*fav2 + fav3
```

f) Conclua quanto à eficácia de cada uma das funções de avaliação/agentes e quanto ao efeito da profundidade utilizada no Algoritmo Minimax.

g) Como poderia melhorar a função de avaliação para este agente?