

Algoritmos MINIMAX

Considerações gerais

O algoritmo MINIMAX é um procedimento recursivo direto, que toma por base três procedimentos auxiliares, específicos do jogo que está sendo jogado:

- GERMOV(Posição, Jogador): o gerador de movimentos plausíveis, o qual retorna uma lista de nós que representam os movimentos que podem ser feitos por *Jogador* em *Posição*. Chamamos os dois jogadores de JOGADOR-UM e JOGADOR-DOIS;
- ESTÁTICA (Posição, Jogador): a função de avaliação estática, que retorna um número que representa a qualidade de *Posição* do ponto de vista de *Jogador*;
- PROFUNDO-O-SUFICIENTE (Posição, Jogador): pode apenas avaliar a profundidade, retornando verdadeiro se está numa profundidade n desejada.

Algoritmo: MINIMAX (Posição, Profundidade, Jogador)

1. Se PROFUNDO-O-SUFICIENTE (Posição, Profundidade), então retorne a estrutura

VALOR = ESTÁTICA (Posição, Jogador)

CAMINHO = nil

Isso indica que não há nenhum caminho saindo desse nó e que seu valor é aquele determinado pela função de avaliação estática.

2. Caso contrário, gere mais uma camada da árvore chamando a função GERMOV (Posição, Jogador) e atribuindo a SUCESSORES a lista que ela retornar.
3. Se SUCESSORES estiver vazio, então não há nenhum movimento a fazer. Retorne a mesma estrutura que teria sido retornada se PROFUNDO-O-SUFICIENTE retornasse VERDADEIRO.
4. Se SUCESSORES não estiver vazio, então examine cada elemento e mantenha registro do melhor. Isto é feito assim:

Inicialize MELHOR-CONTAGEM com o valor mínimo que ESTÁTICA consiga retornar. Ele será atualizado para refletir o melhor placar que pode ser atingido por qualquer elemento de SUCESSORES.

Para cada elemento SUC de SUCESSORES, faça o seguinte:

- (a) Atribua a RESULTADO-SUC MINIMAX (SUC, Profundidade + 1, OPOSTO(Jogador)). Esta chamada recursiva na verdade executará a exploração de SUC.
 - (b) Atribua a NOVO-VALOR – VALOR(RESULTADO-SUC). Isso refletirá os méritos da posição do ponto de vista oposto ao do próximo nível mais baixo.
 - (c) Se NOVO-VALOR > MELHOR-CONTAGEM, então encontramos um sucessor melhor que qualquer outro que já tenha sido examinado até o momento. Registre este fato:
 - I. Atribua a MELHOR-PLACAR NOVO-VALOR.
 - II. O melhor caminho conhecido agora é de CORRENTE para SUC e depois para o caminho apropriado abaixo de SUC, conforme determinado pela chamada recursiva de MINIMAX. Portanto, atribua MELHOR-CAMINHO o resultado da inclusão de SUC à frente de CAMINHO (RESULTADO-SUC).
5. Agora que todos os sucessores foram examinados, sabemos o valor de Posição e também do caminho que deve ser seguido para chegar a este parâmetro. Então retorne a estrutura:
VALOR = MELHOR-CONTAGEM
CAMINHO = MELHOR-CAMINHO

Algoritmo: MINIMAX-A-B(Posição, Profundidade, Jogador, Usar-Limite, Passar-Limite)

1. Se PROFUNDO-O-SUFICIENTE (Posição, Profundidade), então retorne a estrutura
VALOR = ESTÁTICA(Posição, Jogador);
CAMINHO = nil
2. Caso contrário, gere mais uma jogada da árvore chamando a função GERMOV(Posição, Jogador) e atribuindo a SUCESSORES a lista retornada.
3. Se SUCESSORES estiver vazio, não há movimentos a serem feitos; retorne a mesma estrutura que seria retornada caso PROFUNDO-O-SUFICIENTE tivesse retornado VERDADEIRO.
4. Se SUCESSORES não estiver vazio, então examine cada elemento e acompanhe o melhor deles. Isto é feito da seguinte maneira:
Para cada elemento SUC de SUCESSORES:
 - (a) Atribua RESULTADO-SUC MINIMAX-A-B(SUC, Profundidade + 1, OPOSTO(Jogador), -Passar-Limite, -Usar-Limite).
 - (b) Atribua NOVO-VALOR – VALOR(RESULTADO-SUC).
 - (c) Se NOVO-VALOR > Passar-Limite, então encontramos um sucessor melhor que qualquer outro examinado até o momento. Registre o fato fazendo o seguinte:
 - I. Atribua Passar-Limite NOVO-SUC
 - II. O melhor caminho conhecido é agora de CORRENTE para SUC e depois para o caminho apropriado, conforme determinado pela chamada recursiva de MINIMAX-A-B. Ajuste MELHOR-PERCURSO para o resultado da inclusão de SUC à frente de CAMINHO(RESULTADO-SUC).
 - (d) Se Passar-Limite (refletindo o melhor valor atual) não for melhor que Usar-Limite, então, deveremos parar de examinar essa ramificação. Mas tanto os limites quanto os valores foram invertidos. Portanto, se Passar-Limite = Usar-Limite, então retorne imediatamente com o valor
VALOR = Passar-Limite
CAMINHO = MELHOR-CAMINHO
5. Retorne a estrutura
VALOR = Passar-Limite
CAMINHO = MELHOR-PERCURSO