

# Técnicas para Implementação de Jogos

Solange O. Rezende  
Thiago A. S. Pardo

## Considerações gerais

### Aplicações atrativas para métodos de IA

- ✧ Formulação simples do problema (ações bem definidas)
- ✧ Ambiente acessível
- ✧ Abstração (representação simplificada de problemas reais)
- ✧ Sinônimo de inteligência
- ✧ Primeiro algoritmo para xadrez foi proposto por Claude Shannon na década de 50

## Considerações gerais (cont.)

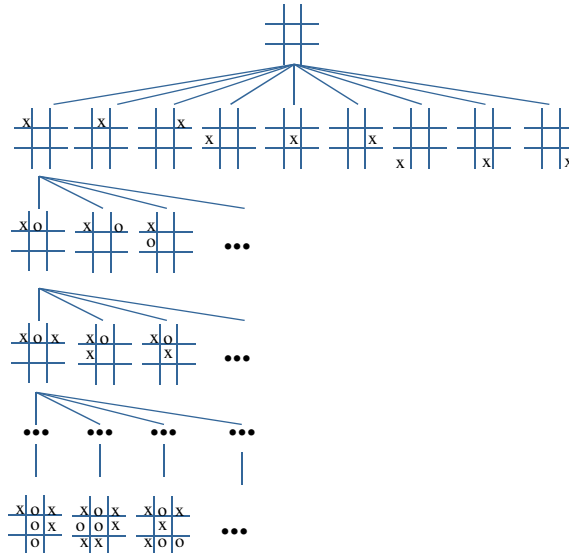
### Problema **desafiador**

- ✧ Tamanho + limitação de tempo ( $35^{100}$  nós para xadrez)
- ✧ Restrições sobre recursos → difícil encontrar a meta
- ✧ Adversário “imprevisível” → solução é ter um plano de contingência
  - agente deve agir antes de completar a busca

## Tipos de jogos

	determinístico	sorte
informações perfeitas	xadrez, damas, go, othello	gamão Banco Imobiliário
informações imperfeitas		bridge, pôquer, scrabble guerra nuclear

## Jogo da velha



## Técnicas para implementação de jogos

- ◆ **Problema pode ser formulado como um tipo de problema de busca**
  - ✧ **Estado inicial:** posições do tabuleiro e indicação do jogador (*de quem é a vez*)
  - ✧ **Estado final:** posições em que o jogo acaba
  - ✧ **Operadores:** jogadas legais
  - ✧ **Função de utilidade:** valor numérico do resultado (pontuação)

## Técnicas para implementação de jogos (cont.)

### ◆ Busca: algoritmo **minimax**

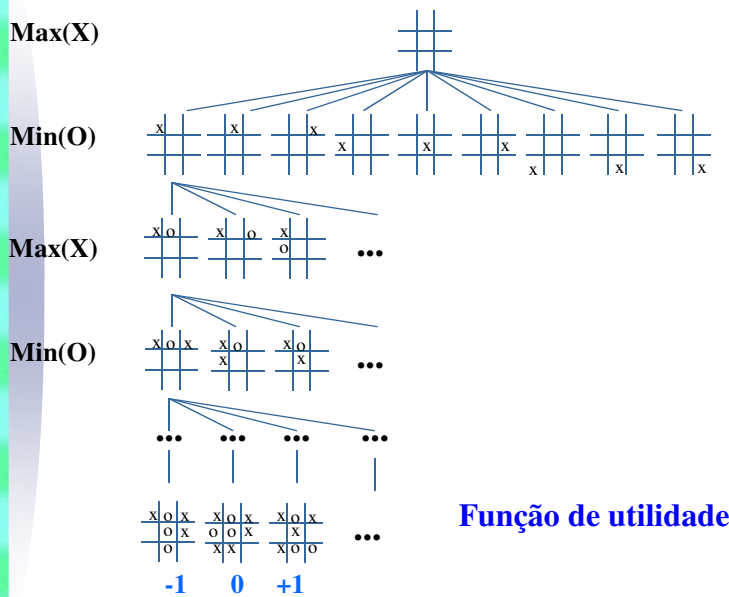
✧ **Ideia:** maximizar a avaliação supondo que o adversário vai tentar minimizá-la

- iniciar no estado atual
- gerar o conjunto de possíveis estados sucessores
- aplicar a função de avaliação a esses estados
- escolher o melhor

✧ Minimax faz **busca cega em profundidade**

✧ O agente é **MAX** e o adversário é **MIN**

## Jogo da velha: **max** vai iniciar





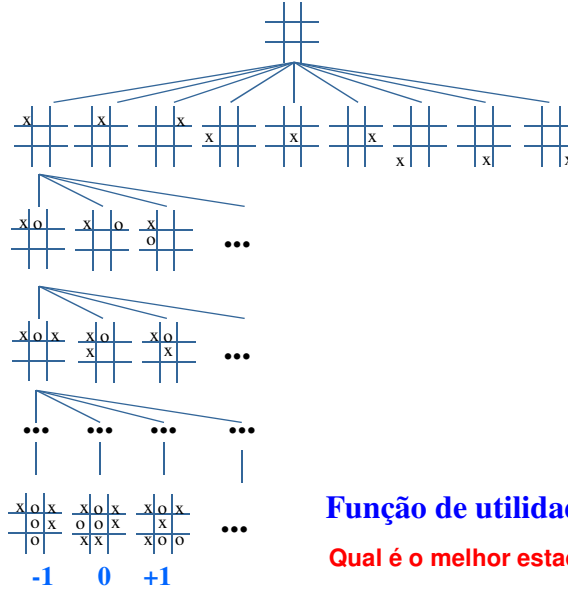
# Jogo da velha: **max** vai iniciar

Max(X)

Min(O)

Max(X)

Min(O)



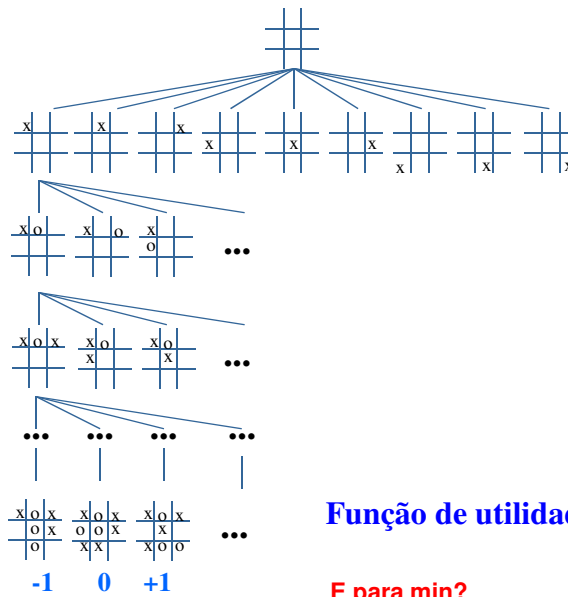
# Jogo da velha: **max** vai iniciar

Max(X)

Min(O)

Max(X)

Min(O)



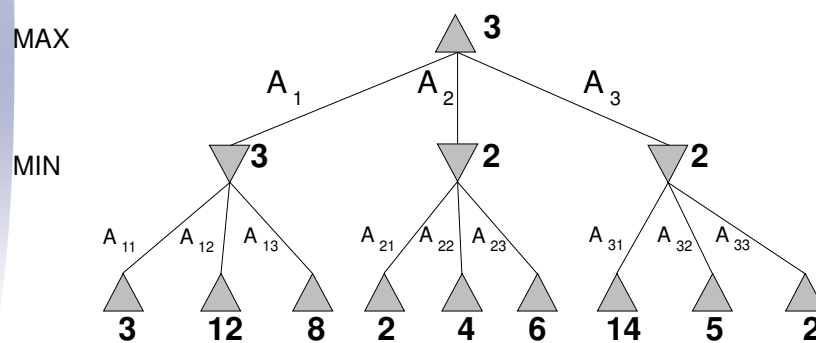
# Algoritmo minimax

## Algoritmo básico

- ✧ Gerar a árvore inteira até os estados terminais
- ✧ Aplicar a função de avaliação nas folhas (nós terminais)
- ✧ Propagar os valores dessa função subindo um nó na árvore até o nó raiz
- ✧ Determinar qual o valor que será escolhido por MAX

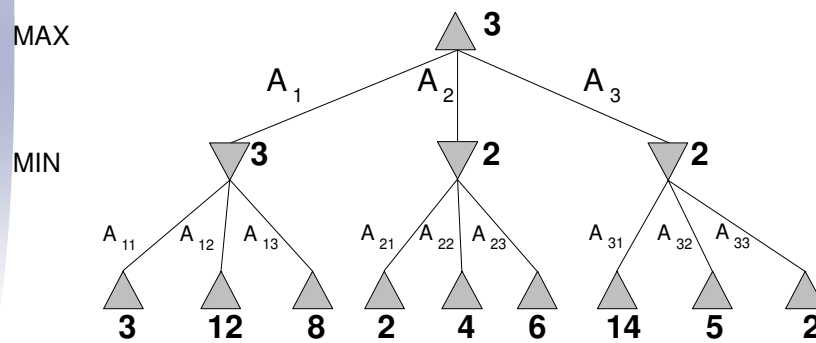
## Jogada perfeita para jogos determinísticos, com informações perfeitas

- ♦ Ideia: escolher movimento para posição com valor minimax mais alto = **melhor retorno possível contra melhor jogada possível**
  - ✧ Max deve buscar o maior valor
  - ✧ Min deve buscar o menor



## Jogada perfeita para jogos determinísticos, com informações perfeitas

- ♦ Ideia: escolher movimento para posição com valor minimax mais alto = **melhor retorno possível contra melhor jogada possível**
  - ✧ Max deve buscar o maior valor
  - ✧ Min deve buscar o menor valor



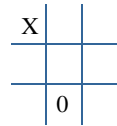
E se min não escolher a melhor jogada?

## Funções de avaliação

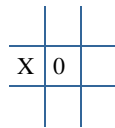
### ♦ Exemplos de funções?

- ✧ Jogo da velha
- ✧ Xadrez
- ✧ 8-puzzle
- ✧ Etc.

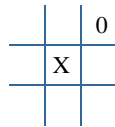
## Função de avaliação para o jogo da velha



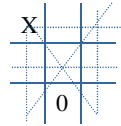
$$H = 6 - 5 = 1$$



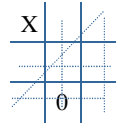
$$H = 4 - 6 = -2$$



$$H = 5 - 4 = 1$$

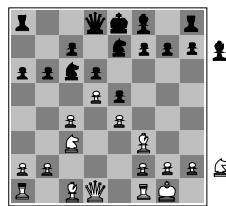


X tem 6 possibilidades

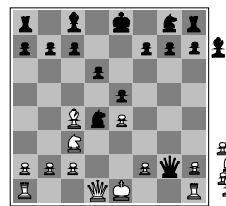


O tem 5 possibilidades

## Funções de avaliação para jogos de xadrez



Veza do Preto  
Branco um pouco melhor



Veza do Branco  
Preto ganhando

Em geral calcula-se uma soma *linear* com pesos de características

$$\text{Aval}(s) = w_1 f_1(s) + w_2 f_2(s) + \dots + w_n f_n(s)$$

por ex.,  $w_1 = 0.8$  com

$$f_1(s) = (\text{no. rainhas brancas}) - (\text{no. rainhas pretas})$$



## Propriedades do minimax

- ♦ Completeza? Sim, se árvore é finita
- ♦ Admissibilidade? Sim, contra um adversário ótimo
- ♦ Complexidade de tempo:  $O(b^m)$
- ♦ Complexidade de espaço:  $O(bm)$

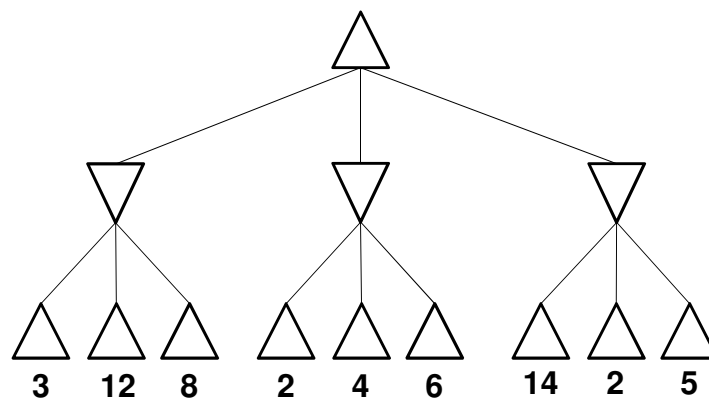
## Críticas

- ♦ **Problemas**
  - ✧ Tempo gasto é totalmente impraticável, porém o algoritmo serve como base para outros métodos mais realísticos
    - **Muitos possíveis estados a explorar**, que piora de acordo com a complexidade do jogo
- ♦ **Para melhorar**
  - 1) **Limitar a profundidade** e usar uma boa função heurística
  - 2) Podar a árvore onde a busca seria irrelevante: **poda alfa-beta**

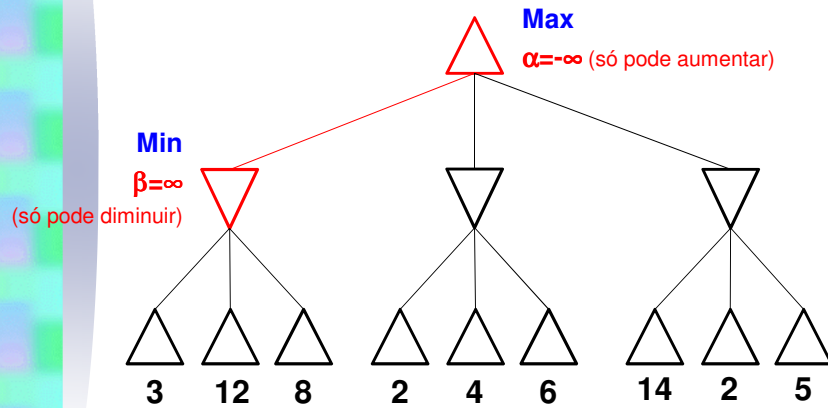
## Poda Alfa-Beta

- ◆ **Função:** Não expandir desnecessariamente nós durante o minimax
- ◆ **Ideia:** não vale a pena piorar, se já achou algo melhor
- ◆ **Mantém 2 parâmetros (origem do nome)**
  - ◇  $\alpha$  - melhor valor (no caminho) para MAX
  - ◇  $\beta$  - melhor valor (no caminho) para MIN
- ◆ **Teste de expansão**
  - ◇  $\alpha$  não pode diminuir (não pode ser menor que um ancestral)
  - ◇  $\beta$  não pode aumentar (não pode ser maior que um ancestral)
    - ◇ Supondo-se que a função de avaliação é melhor se max vai bem (nada impede que se modele de forma inversa)

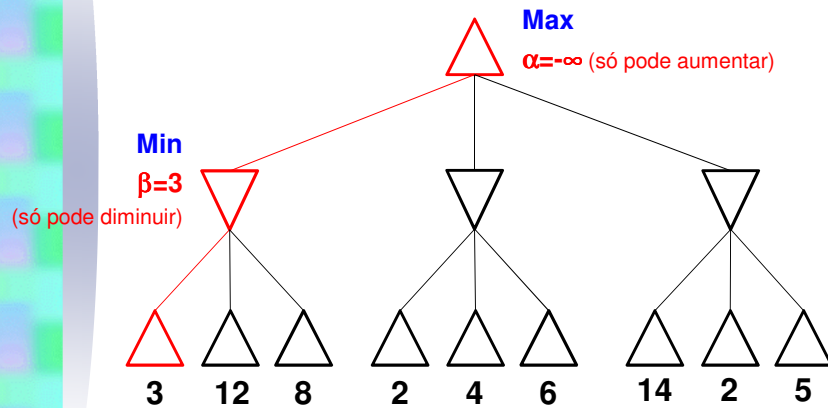
## Exemplo: 2 turnos



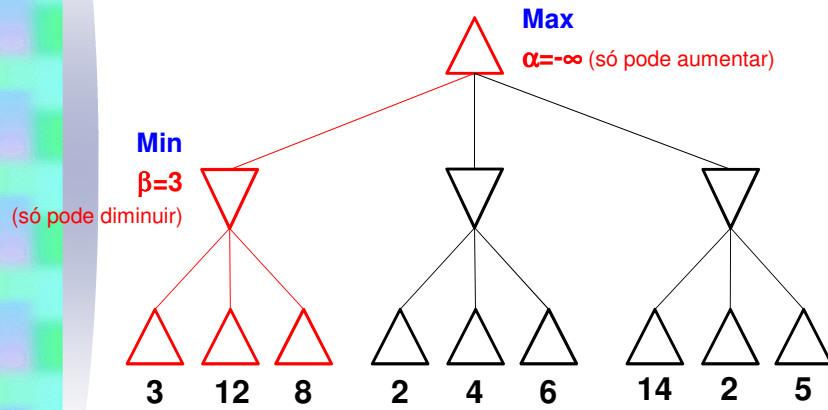
## Exemplo: 2 turnos



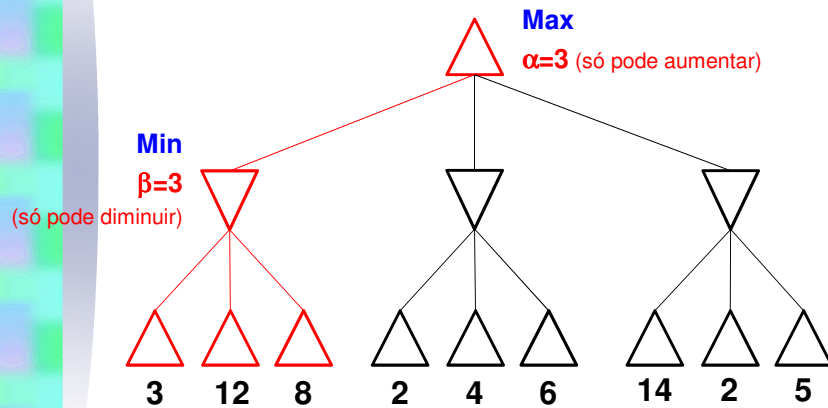
## Exemplo: 2 turnos



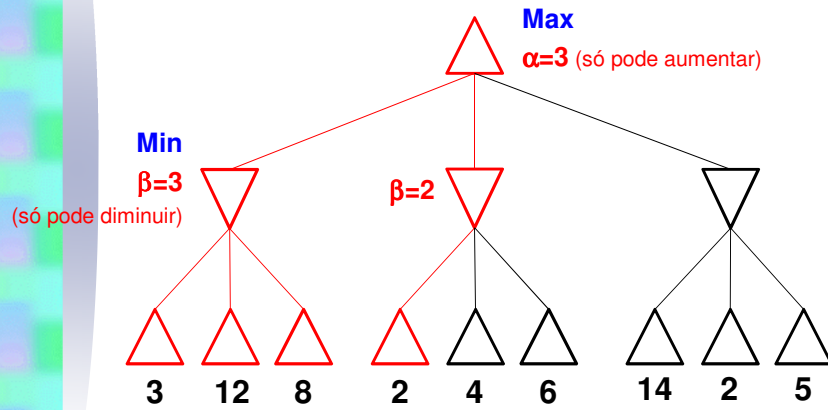
## Exemplo: 2 turnos



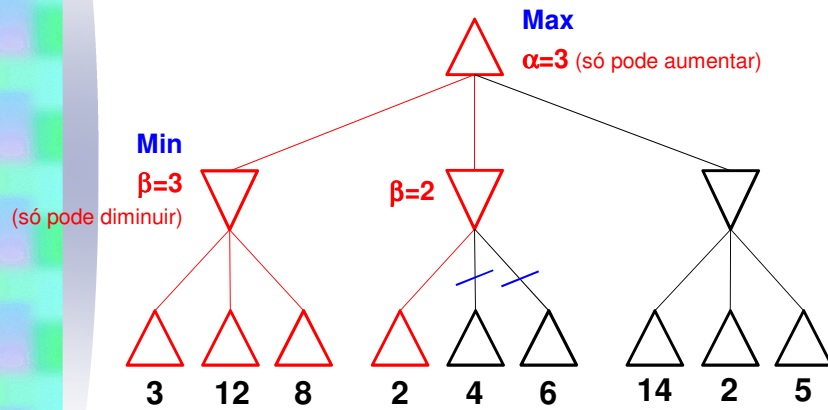
## Exemplo: 2 turnos



## Exemplo: 2 turnos

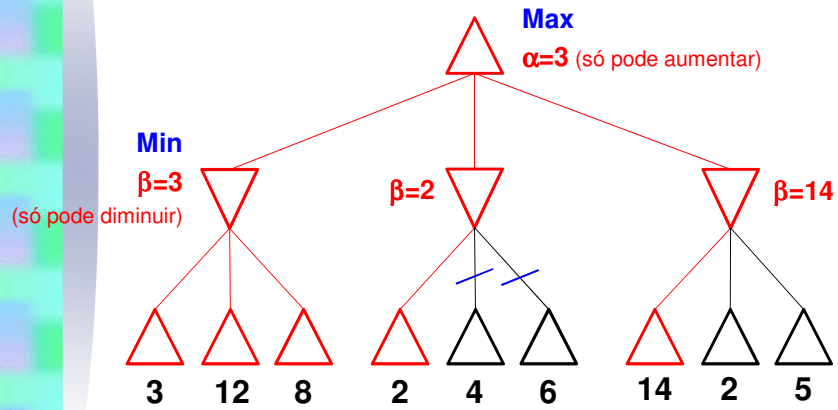


## Exemplo: 2 turnos

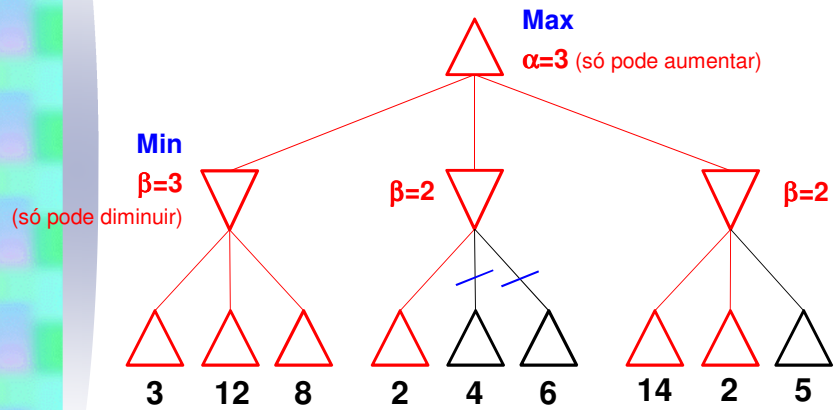


Não importa o que vier depois,  $\alpha$  não será afetado

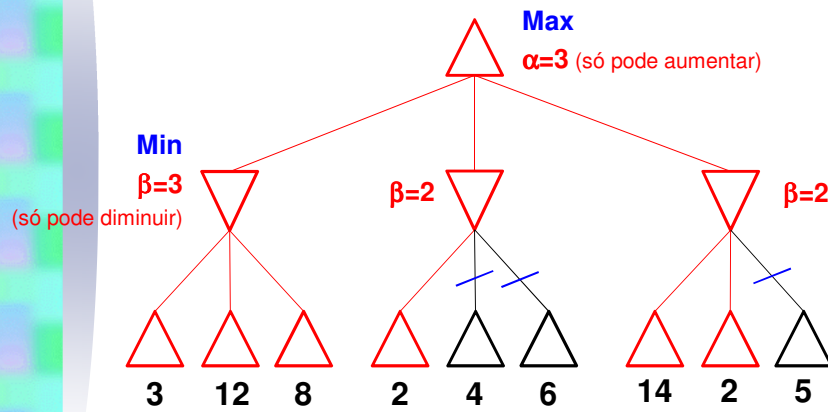
## Exemplo: 2 turnos



## Exemplo: 2 turnos



## Exemplo: 2 turnos



Não importa o que vier depois,  $\alpha$  não será afetado

## Poda alfa-beta

### ◆ Regra geral

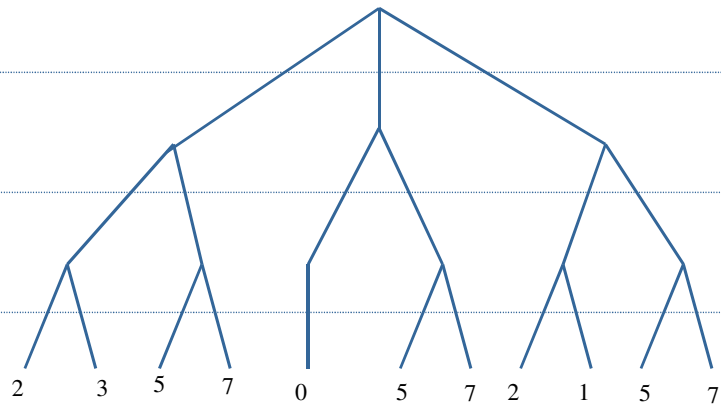
Se beta é menor que alfa, faça a poda dos ramos restantes na subárvore em que está

**Exercício: que caminhos podem ser podados?**

MAX

MIN

MAX

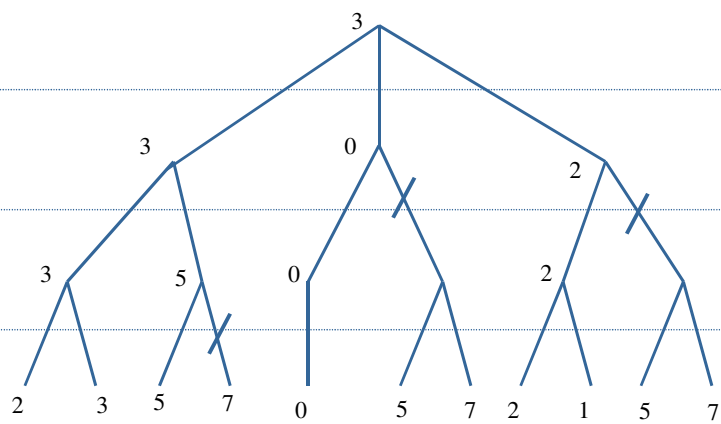


Supondo que menor valor de função de avaliação é 0 (zero)

MAX

MIN

MAX





## Jogos determinísticos na prática

- ◆ **Damas:** **Chinook** acabou com o reinado de 40 anos do campeão humano mundial Marion Tinsley em 1994. Utilizou-se de uma base de dados sobre finais de jogo que definia jogadas perfeitas para todas as posições envolvendo 8 ou menos peças no tabuleiro, um total de **443.748.401.247 posições**.
- ◆ **Xadrez:** **Deep Blue** derrotou o campeão humano mundial Gary Kasparov numa partida de seis jogos em 1997. Deep Blue procura em 200 milhões de posições por segundo, usando avaliação muito sofisticada, e alguns métodos não divulgados estendendo alguns caminhos de busca até **40 jogadas a frente**