
Jogos de Tabuleiro e Busca Competitiva

Fabrício Barth

Insper

Abril de 2022

Disclaimer

Este não é o material mais atualizado sobre o assunto.

Para acessar o material mais atualizado. Por favor, acesse o site <http://fbarth.net.br/Connect4-Python/>

Sumário

- Características e Exemplos
- Histórico
- Árvore de busca
- Avaliação Estática
- Algoritmo **Min Max**
- Poda **Alpha Beta**
- Questões práticas
- AlphaGo, ..., MuZero.

Características e Exemplos

- São jogados por duas pessoas (agentes).
- Trata-se de uma competição.
- Não tem variável aleatória.



Histórico

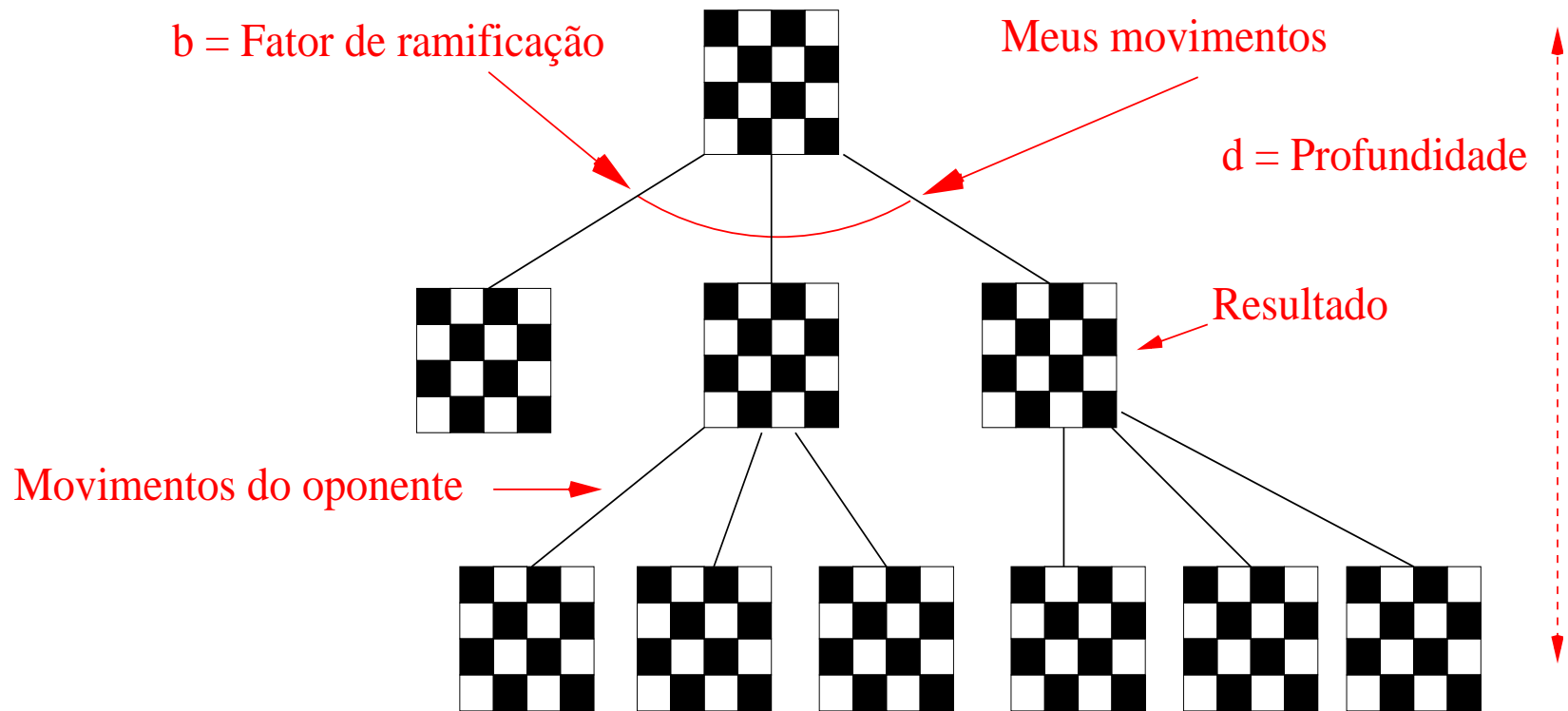
- **Shannon**. *Programming a Computer for Playing Chess*. 1950 : O uso do algoritmo **Min Max** no jogo de Xadrez.
- **Deep Blue** Wins, 1996 e 1997.
- **AlphaGo** versus Lee Sedol, 2016.
- **MuZero** plays Go, Chess, Shogi and Atari, 2020.

Árvore de Busca para Jogos

- **Estado inicial**: tabuleiro na posição inicial e jogador.
- **Operadores**: movimentos permitidos.
- **Estados Objetivo**: posições vencedoras para o meu jogador no tabuleiro.
- **Função de Utilidade**: determina um valor para cada estado.
- **Árvore de Busca**: mostra todas as possibilidades de jogo.

- Não estamos procurando por um caminho. Apenas pelo próximo movimento (espera-se que este movimento me leve à vitória).
- Meus melhores movimentos dependem dos movimentos do meu adversário.

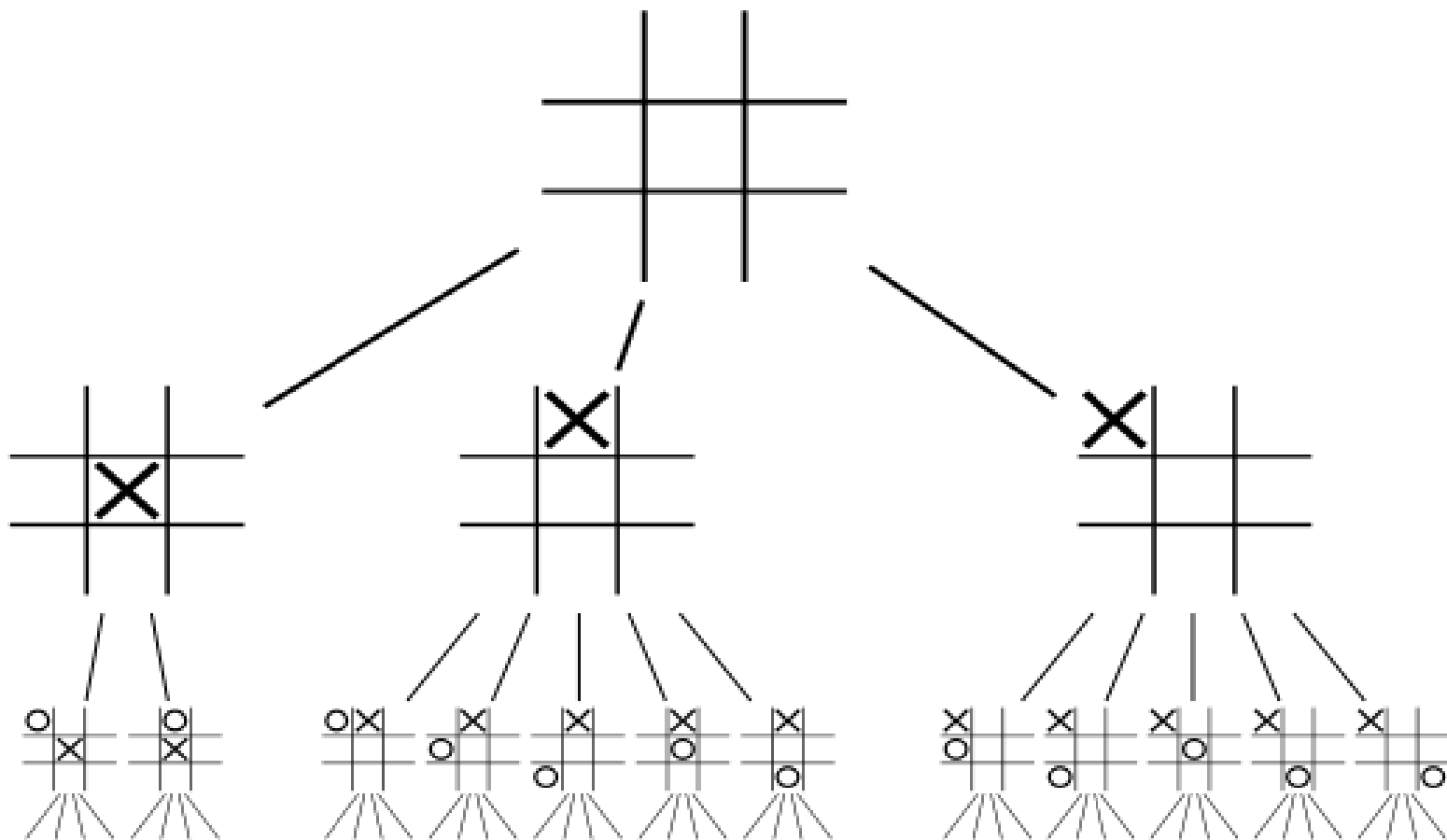
Árvore de Busca



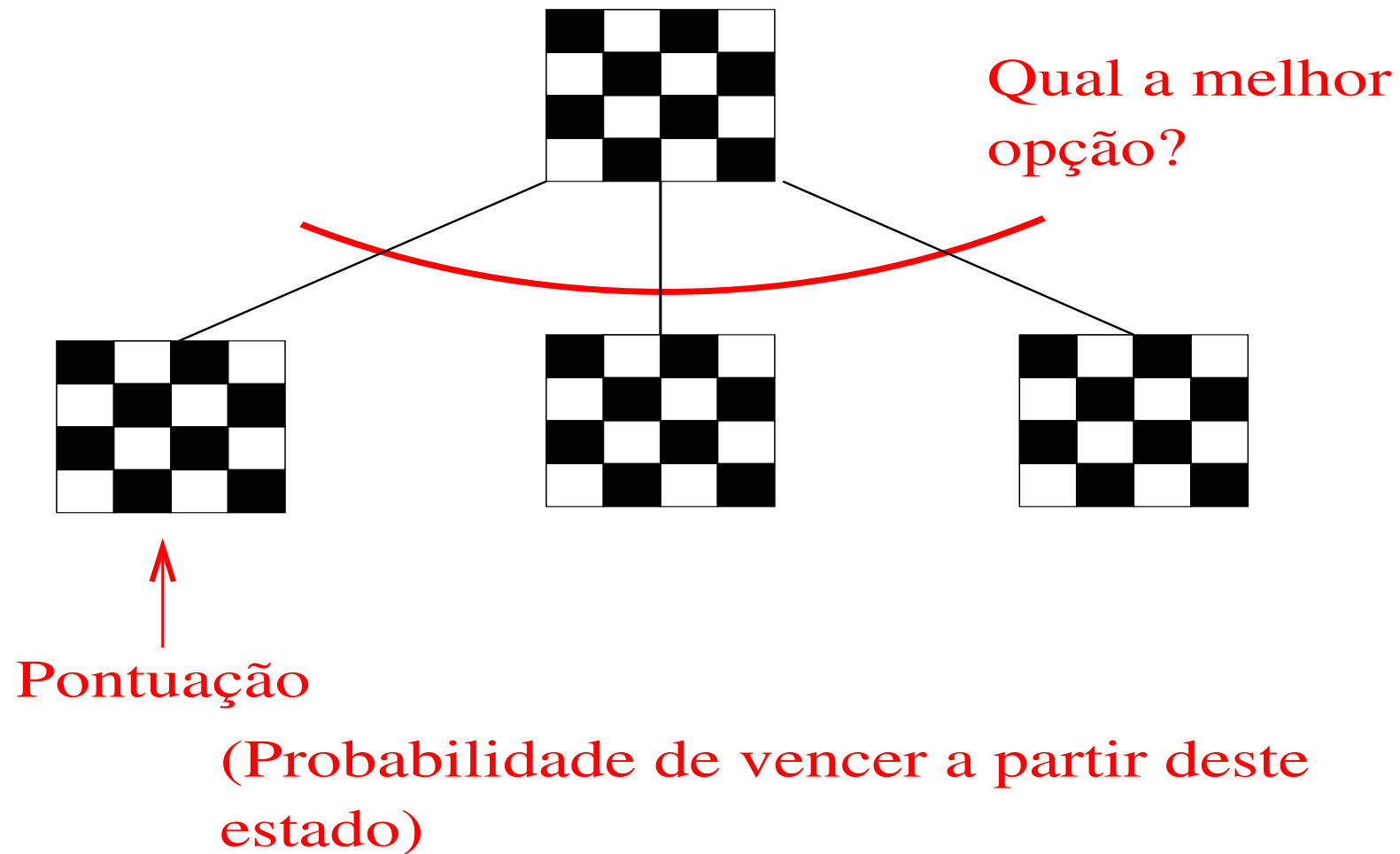
Xadrez

$b = 36$
 $d > 40$
 grande!

Árvore de Busca Parcial para o Jogo da Velha



Função de Utilidade



Definição da função de utilidade para o xadrez

$$material = numeroPeao \times 1 + \dots + numeroDama \times 9 \quad (1)$$

$$v_1 = c_1 \times material \quad (2)$$

$$v_2 = c_2 \times mobilidade \quad (3)$$

$$v_3 = c_3 \times segurancaRei \quad (4)$$

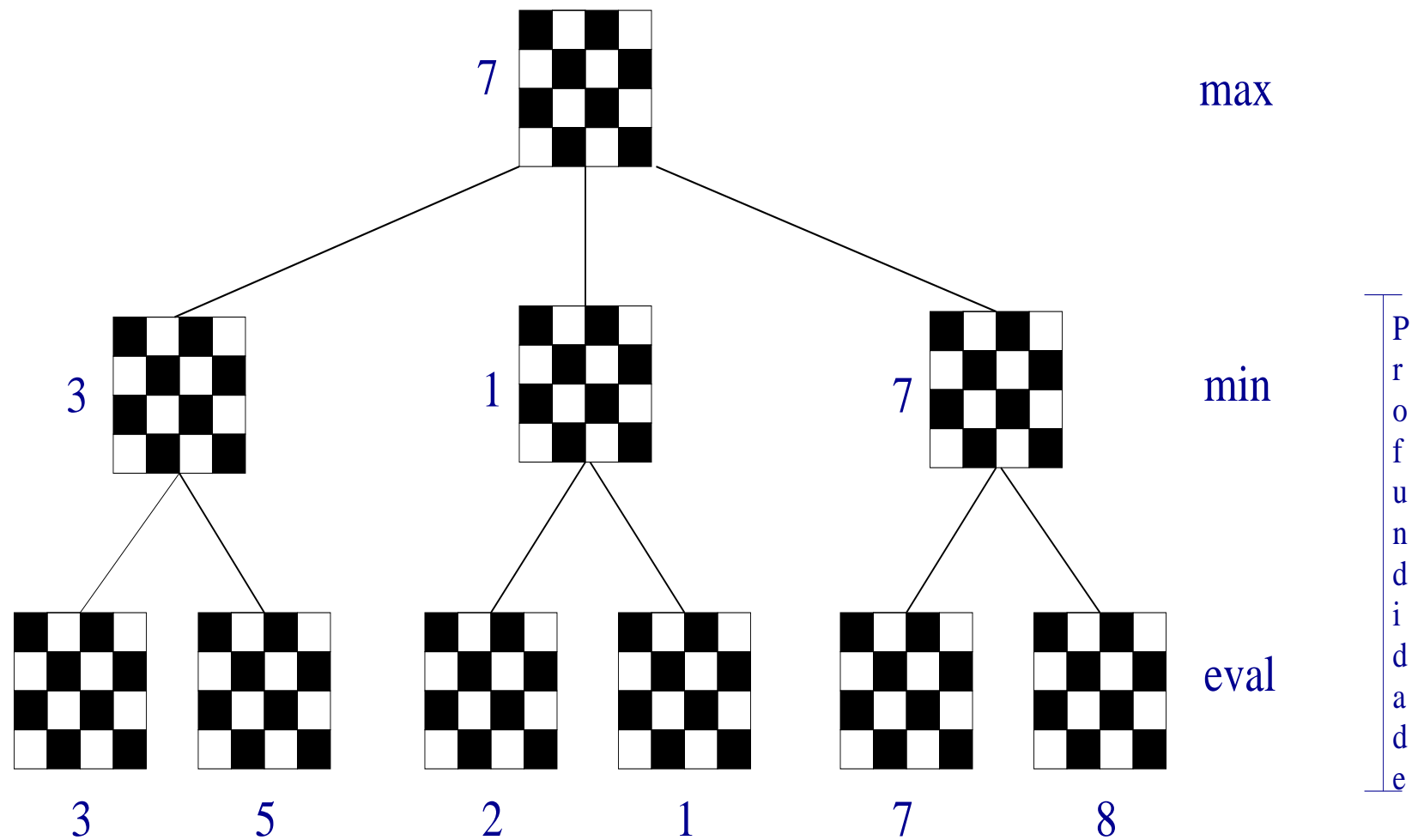
$$v_4 = c_4 \times controleCentro \quad (5)$$

$$v_5 = \dots \quad (6)$$

$$Utilidade = \sum_{i=0}^n v_i \quad (7)$$

- Muito fraco para prever o sucesso final do jogo!

Olhar adiante + função de utilidade (**MinMax**)



Min-Max

chamada inicial MAX-VALUE(estado,max-p)

function MAX-VALUE(Estado *estado*, int *p*)

if $p == 0$ **then**

 return EVAL(*estado*)

end if

$v = -\infty$

for $s \in \text{SUCESSORES}(\textit{estado})$ **do**

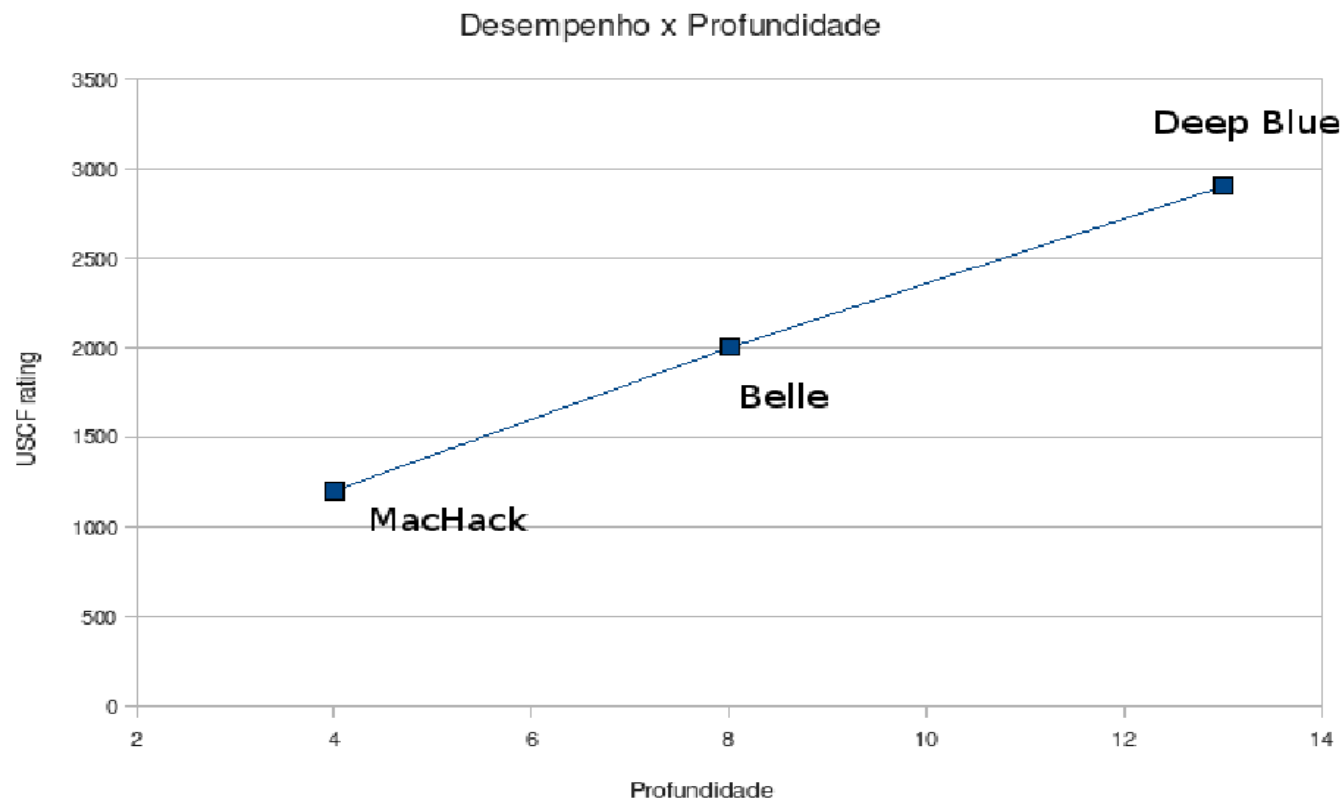
$v = \text{MAX}(v, \text{MIN-VALUE}(s, p - 1))$

end for

return v

```
function MIN-VALUE(Estado estado, int p)  
if p==0 then  
    return EVAL(estado)  
end if  
  
v =  $\infty$   
for s  $\in$  SUCESSORES(estado) do  
    v = MIN(v, MAX-VALUE(s, p - 1))  
end for  
return v
```

Desempenho x Profundidade



Deep Blue \simeq Força Bruta

- 256 processadores dedicados.
- Examina em torno de 30 bilhões de movimentos por minuto.
- A profundidade geralmente é 13. No entanto, em determinadas situações, pode chegar até 30.

Min-Max $\alpha - \beta$

chamada MAX-VALUE(estado, $-\infty$, ∞ , max-p)

function MAX-VALUE(Estado *estado*, α , β , int *p*)

if $p == 0$ **then**

 return EVAL(*estado*)

end if

for $s \in$ SUCESSORES(*estado*) **do**

$\alpha = \text{MAX}(\alpha, \text{MIN-VALUE}(s, \alpha, \beta, p - 1))$

if $\alpha \geq \beta$ **then**

 return α //cutoff

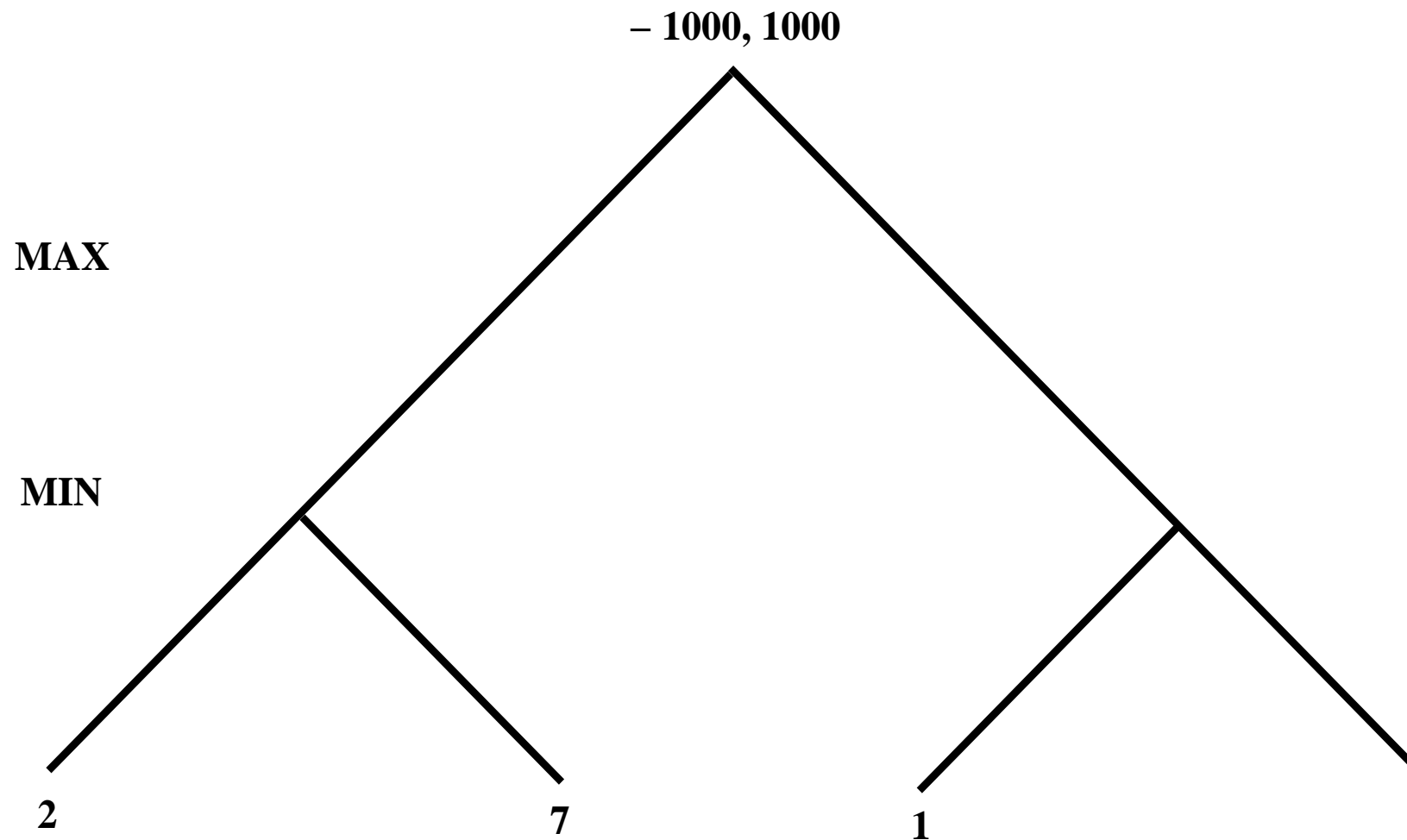
end if

end for

return α

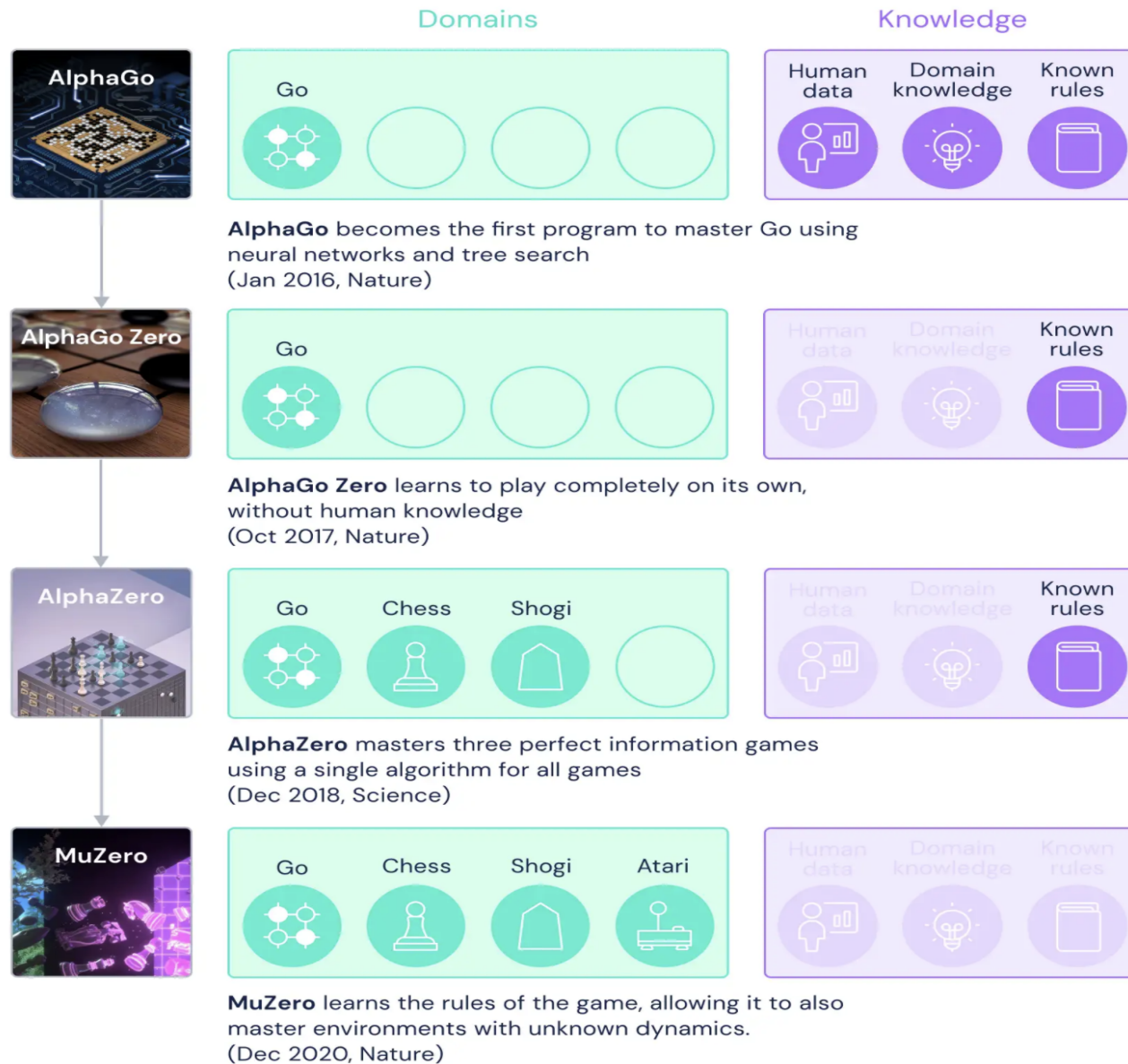
```
function MIN-VALUE(Estado estado,  $\alpha$ ,  $\beta$ , int p)  
if  $p == 0$  then  
    return EVAL(estado)  
end if  
for  $s \in \text{SUCESSORES}(\textit{estado})$  do  
     $\beta = \text{MIN}(\beta, \text{MAX-VALUE}(s, \alpha, \beta, p - 1))$   
    if  $\beta \leq \alpha$  then  
        return  $\beta$  //cutoff  
    end if  
end for  
return  $\beta$ 
```

Exemplo $\alpha - \beta$



AlphaGo, MuZero

Artigos da Deepmind



Material de **consulta**

- Capítulo 6 do livro do Russell & Norvig.
- <http://en.wikipedia.org/wiki/Minimax>.
- <https://deepmind.com/blog/article/muzero-mastering-go-chess-shogi-and-atari-without-rules>