



ÁRVORE DE JOGOS

Uma introdução à estruturas decisórias utilizando o jogo da velha

PALESTRANTE

Allan Kássio Beckman Soares da Cruz

Doutorando em Ciência da Computação (UFMA)

Mestre em Administração Geral (Steinbeis-SIBE / Alemanha)

Mestre em Design (UFMA)

Especialista em Gestão da Qualidade (UNIASSELVI)

Especialista em Informática na Educação (IFMA)

Bacharel em Ciência da Computação (UFMA)

Python | PHP | Java | JavaScript | HTML | CSS | UX/UI | OOP | PostgreSQL |
Usability | Wordpress | Scrum | Agile | Pokémon Professor



Senior Software Engineer



allankassiocruz



allankassio@gmail.com

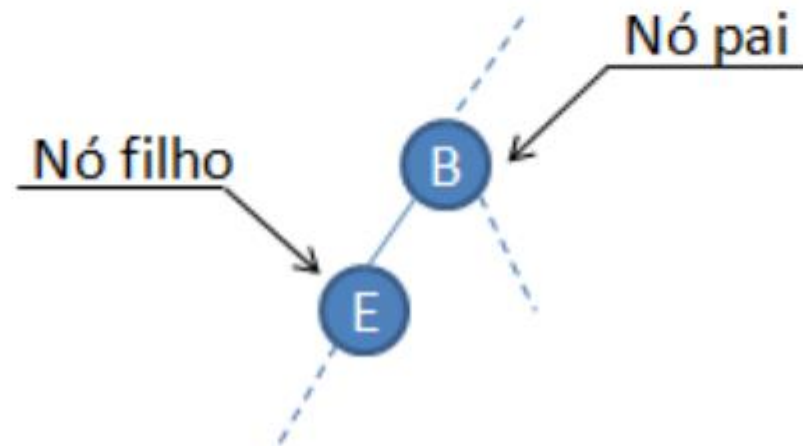
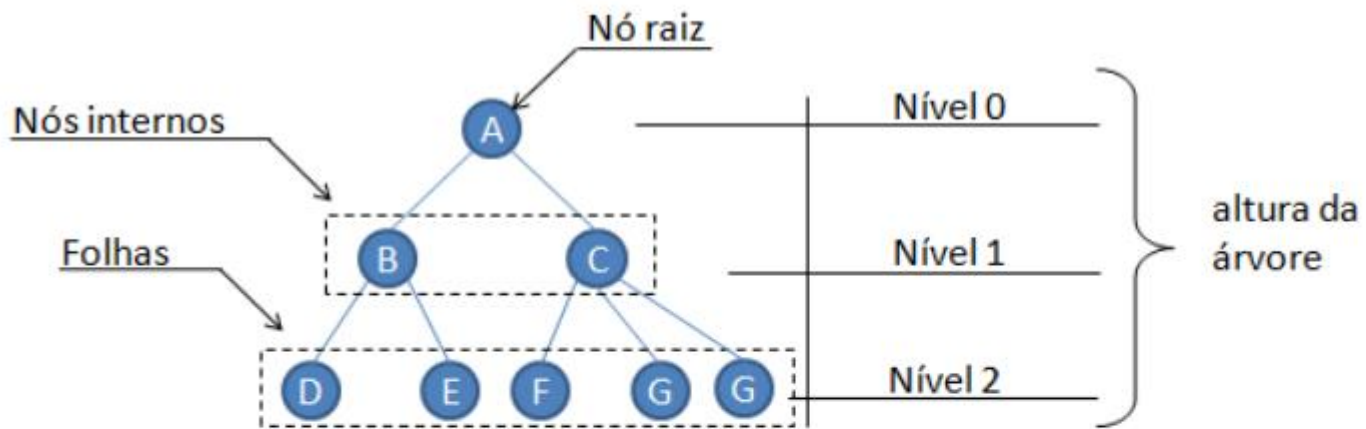


allankassio

ROTEIRO

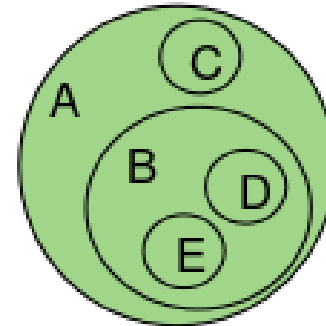
- ❑ Árvores
- ❑ Teoria dos Jogos
- ❑ Árvore de Jogos
- ❑ Algoritmo MINIMAX
- ❑ Construindo um Jogo da Velha inteligente

ÁRVORES

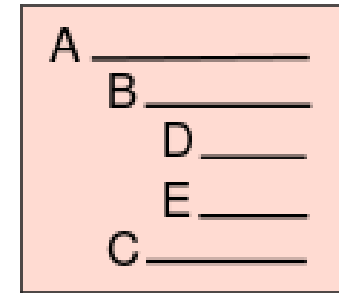


ÁRVORES

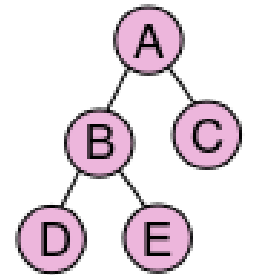
Representações



c)



b)



a)

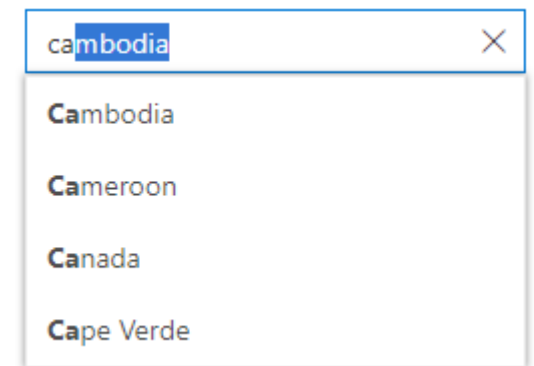
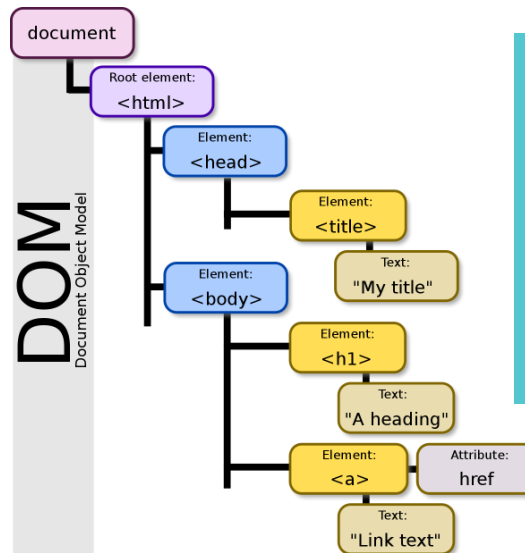
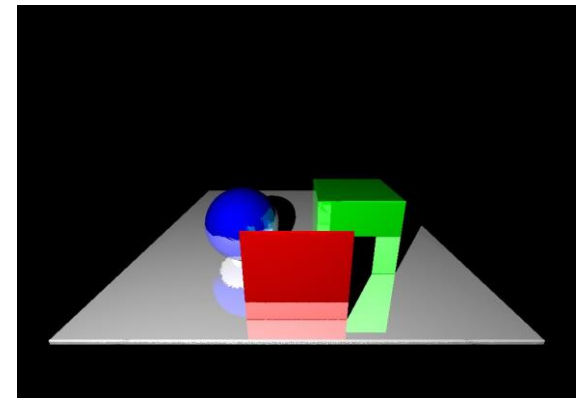
1A; 1.1B; 1.1.1D; 1.1.2E; 1.2C

e)

(A(B(D)(E))(C))

d)

APLICAÇÕES DE ÁRVORES?



APLICAÇÕES DE ÁRVORES?



TEORIA DOS JOGOS

A Teoria dos Jogos é uma análise matemática de situações que abrangem interesses em conflito, com o objetivo de apontar as melhores opções para se alcançar um objetivo.

TEORIA DOS JOGOS

Só restou à Cielo a pior alternativa

Esquema de jogo sequencial em forma estendida



Fonte: Fiani, Ronaldo; Teoria dos Jogos. Adaptado pelo autor

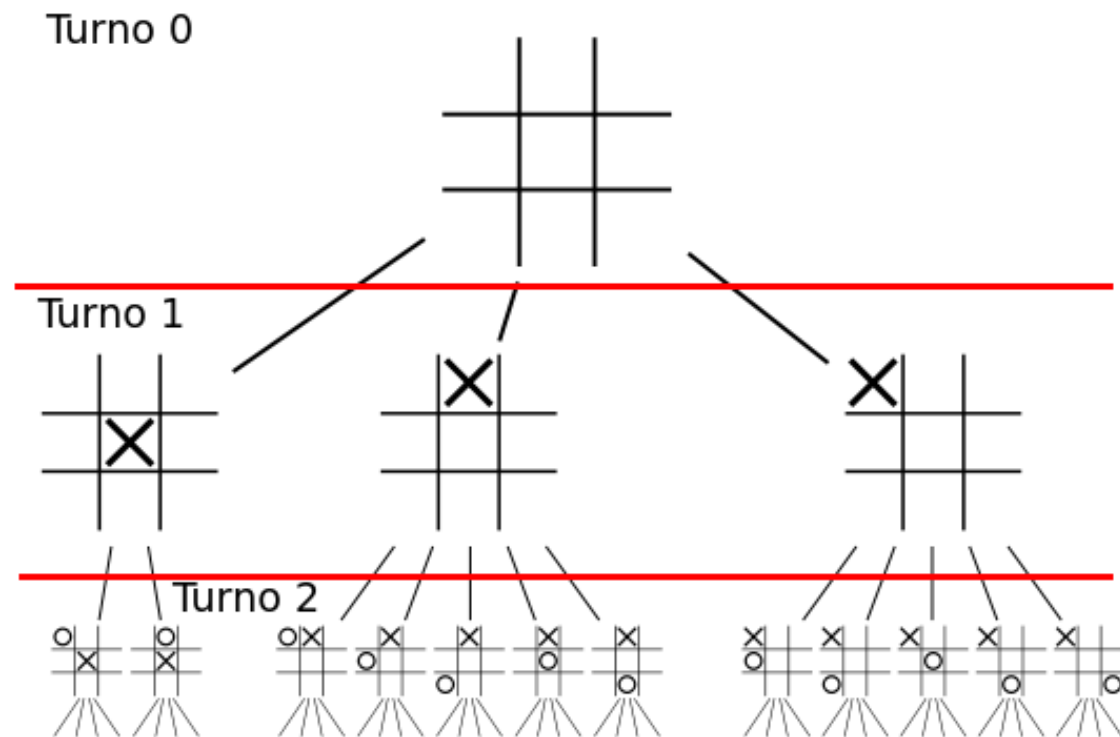
ÁRVORE DE JOGOS

Para jogos como o **Xadrez**, **Damas** e **Jogo da Velha**, podemos representar todas as possibilidades de jogada em uma **Árvore de Jogos**.

Jogo da Velha = 255.168 nós

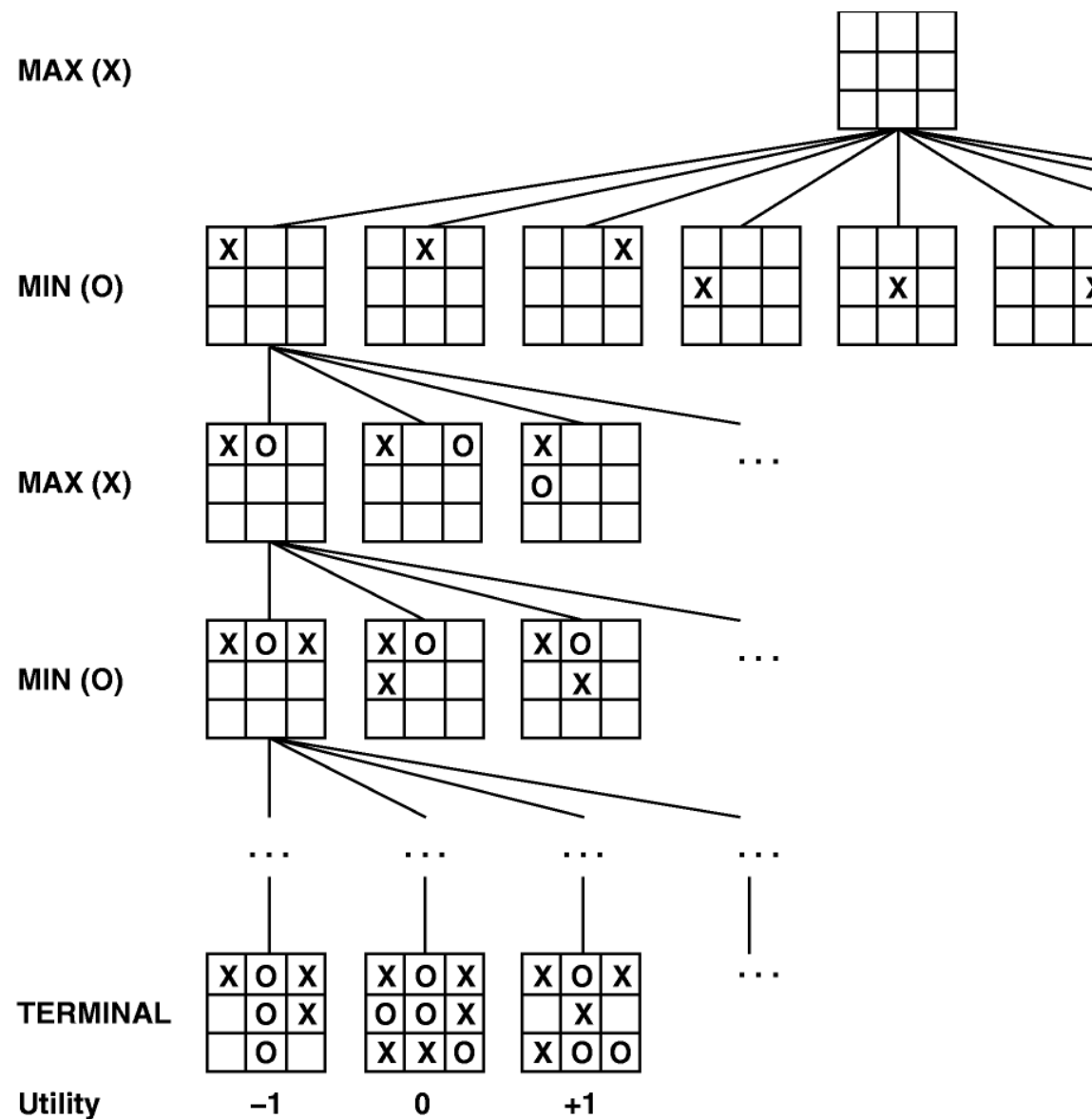
Xadrez > qtd átomos no universo observável

Magic The Gathering > Xadrez



ALGORITMO MINIMAX

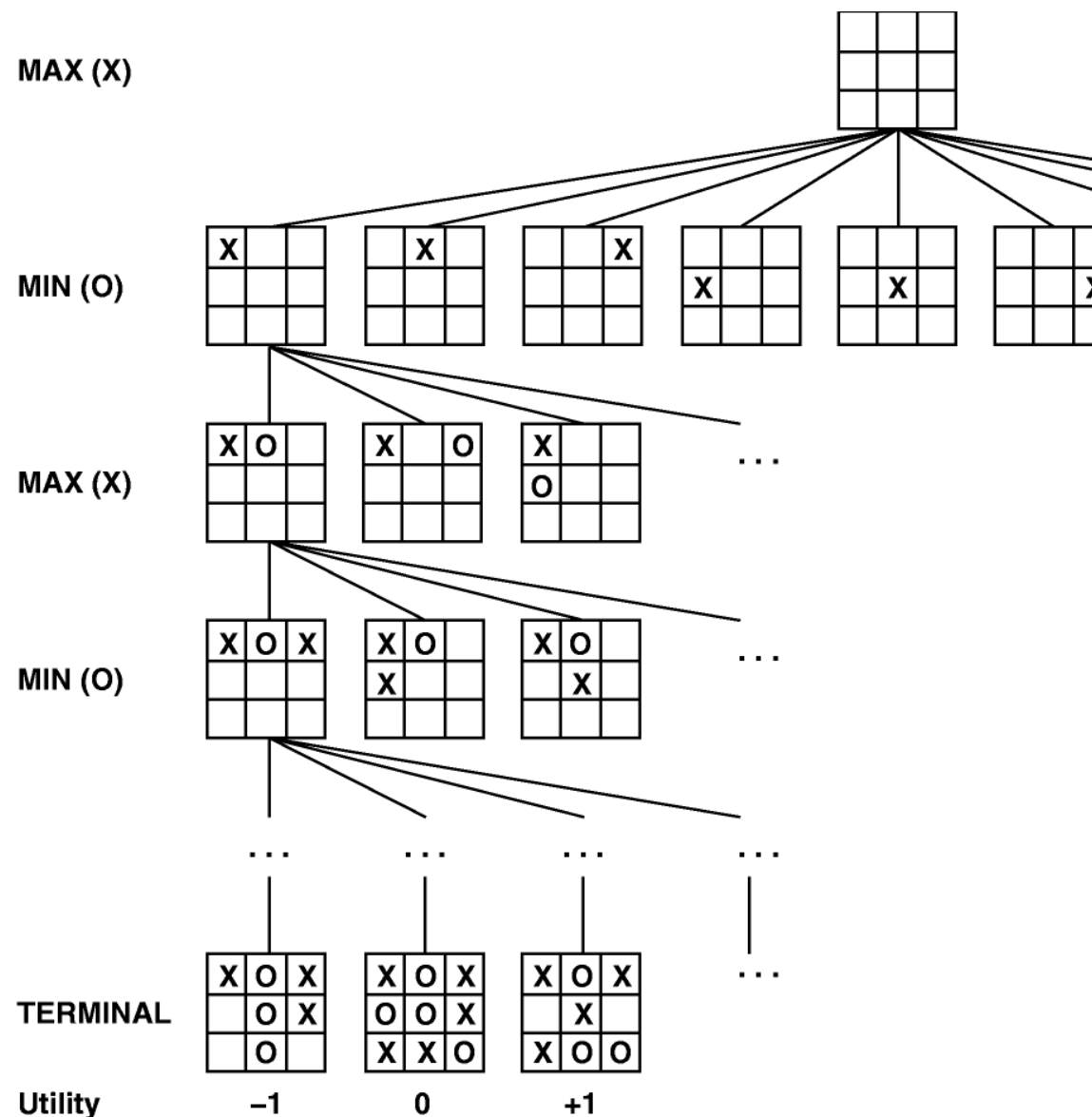
Com a Árvore de Jogos é possível identificar qual caminho seguir para ganhar o jogo de uma maneira que o inimigo não consiga interferir.



ALGORITMO MINIMAX

O resultado (Terminal) pode ser **-1 (Perder)**, **0 (Empatar)** ou **1 (Ganhar)**.

Basta a inteligência artificial seguir caminhos que o **levam para um +1** que terá uma vitória certa.



ALGORITMO MINIMAX

O resultado (Terminal)
pode ser -1 (Perder), 0
(Empatar) ou 1
(Ganhar).

Basta a inteligência
artificial seguir os
caminhos que levam
para um -1 que terá
uma vitória certa.

MAX (X)

MIN (O)

(X)

MIN (O)

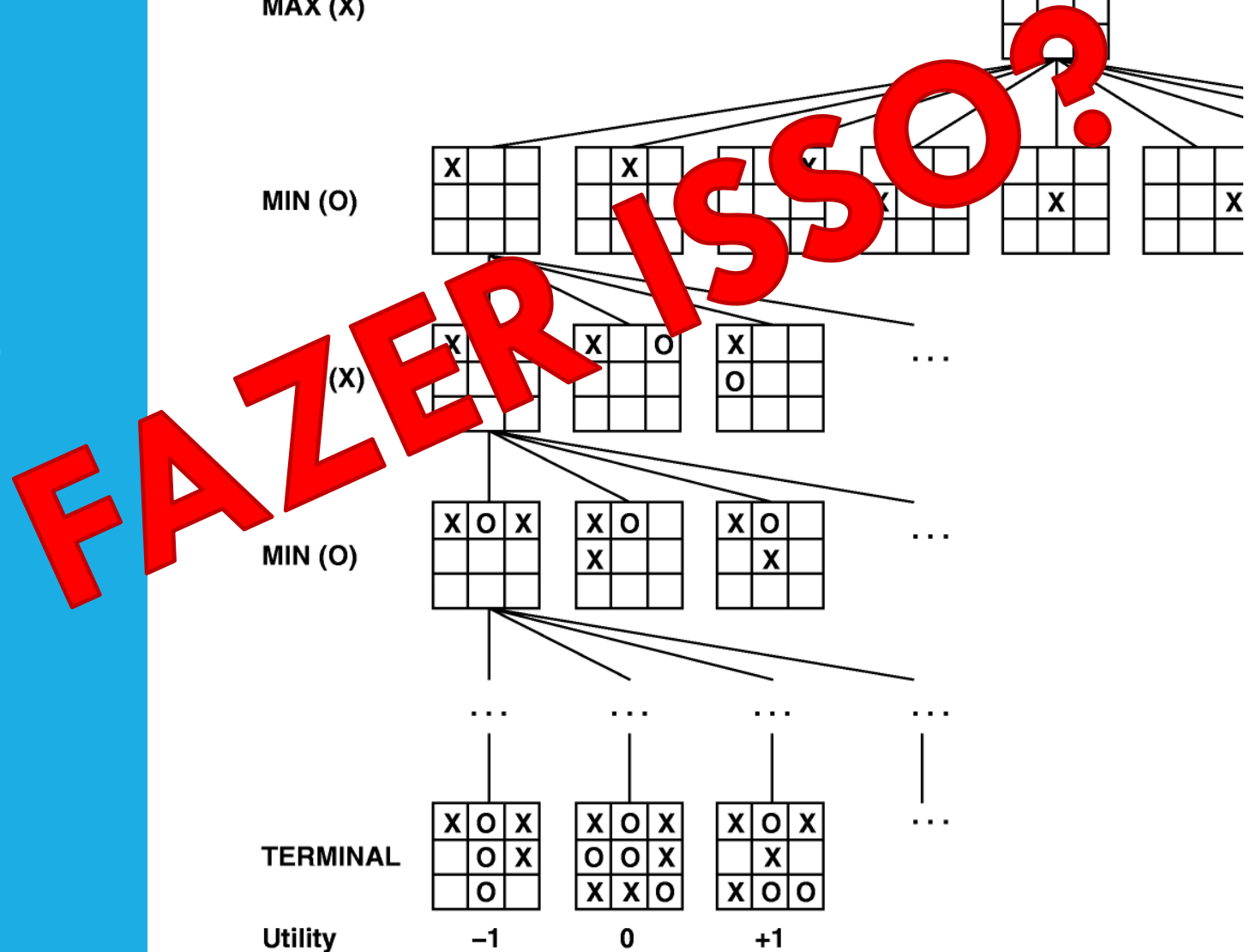
TERMINAL

Utility

-1

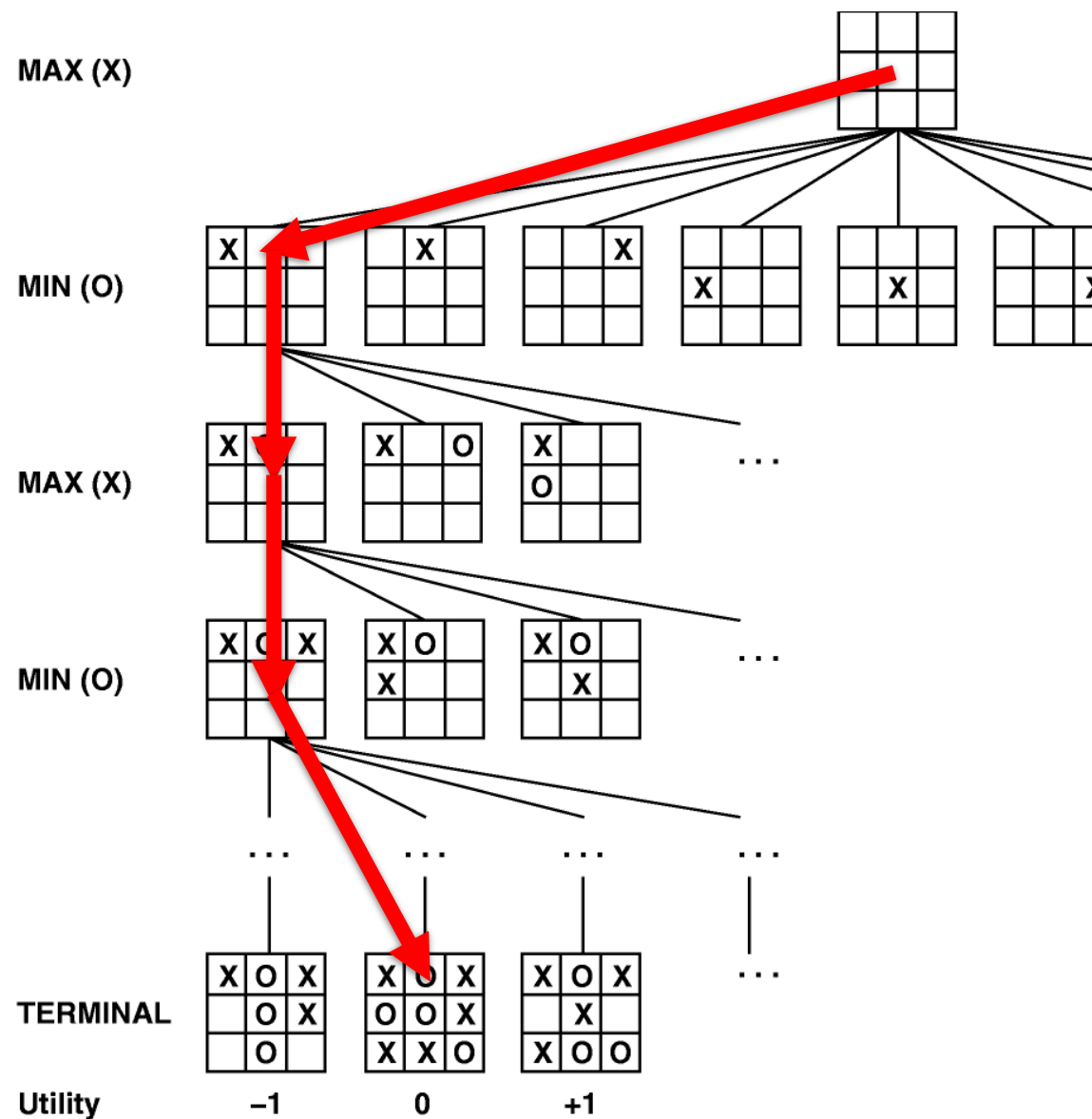
0

+1



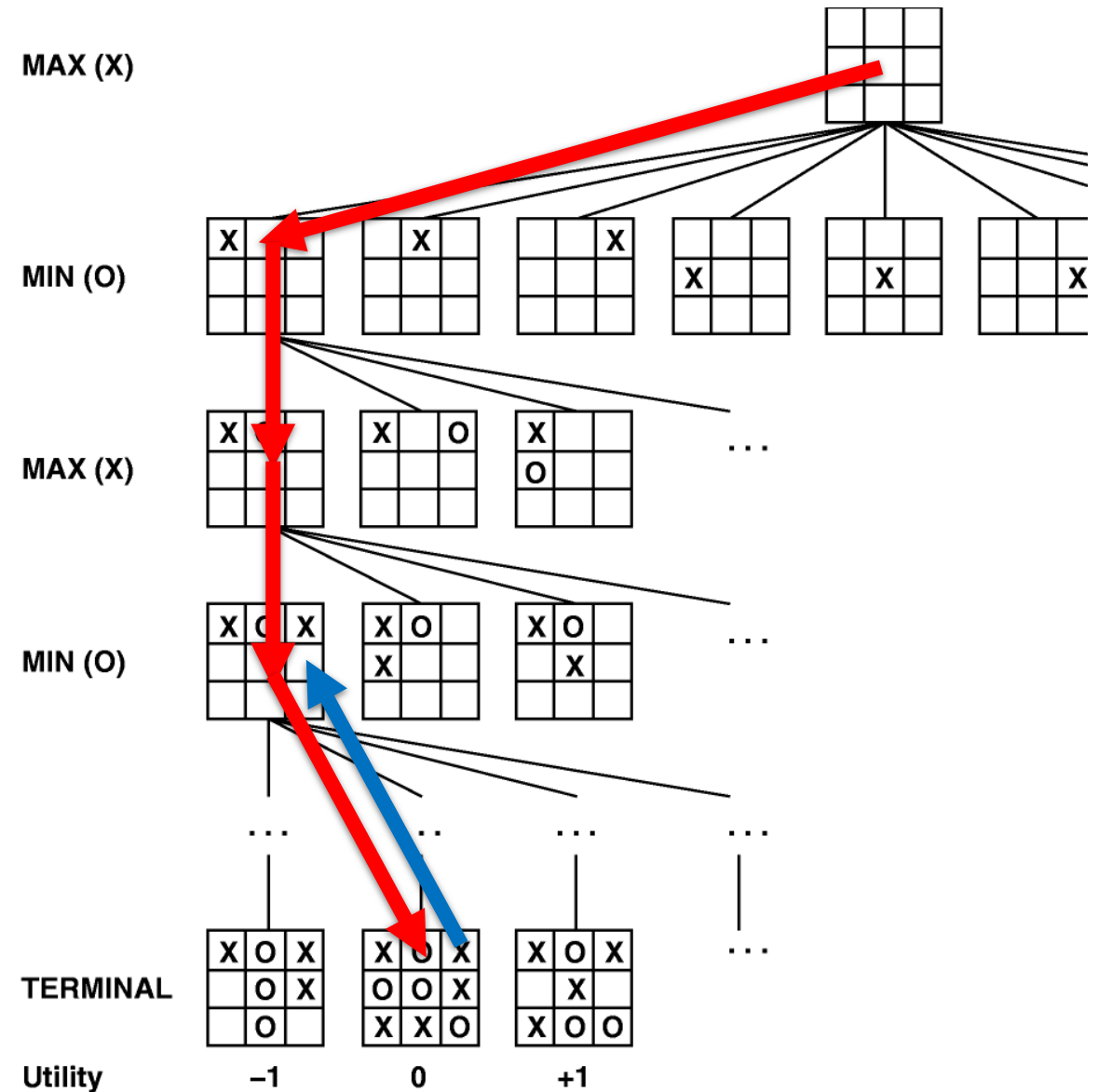
ALGORITMO MINIMAX

O princípio do Minimax é descer os 'nós' da Game Tree até chegar no término do jogo, identificando se o jogador perdeu, empatou ou ganhou.



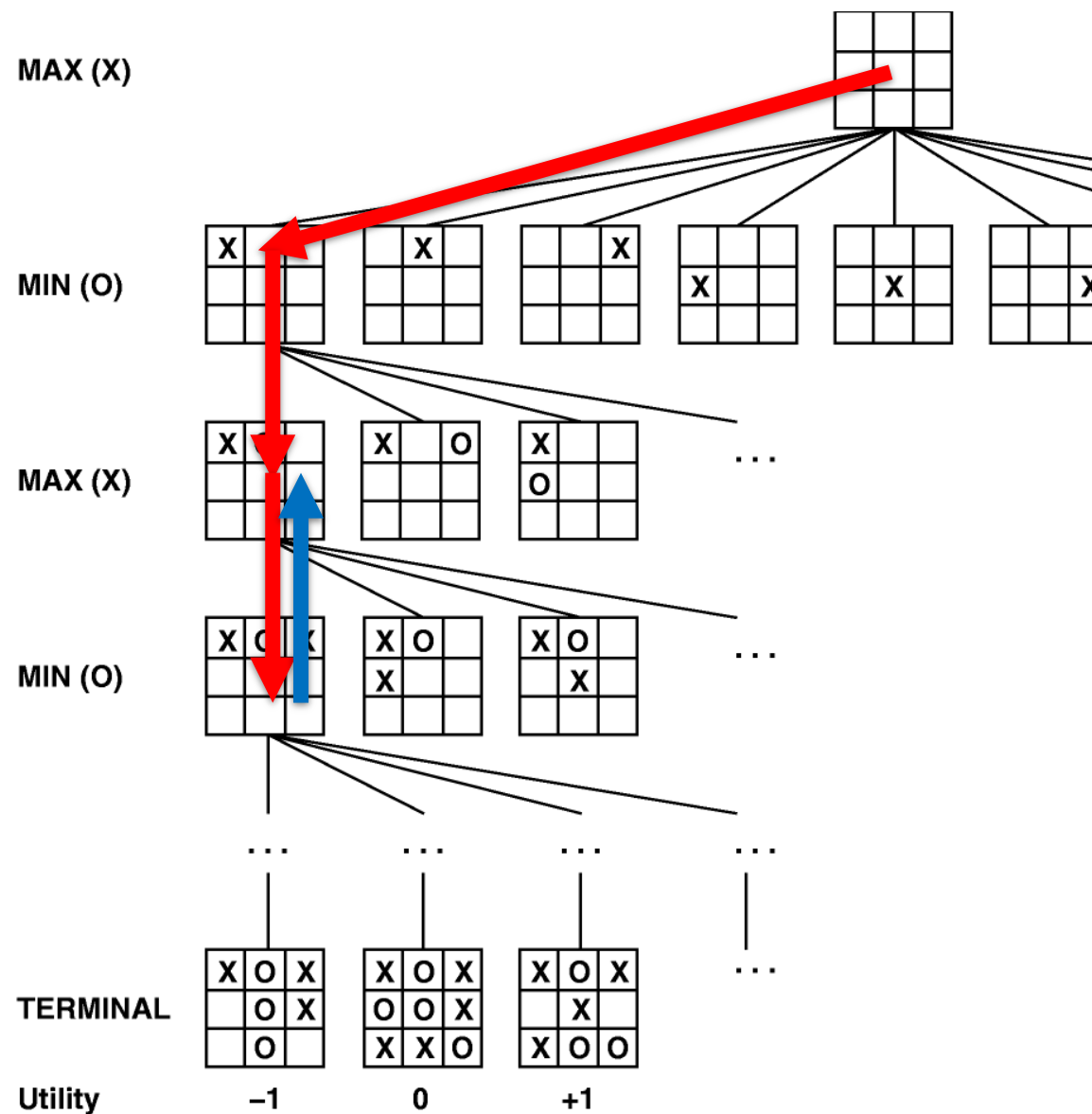
ALGORITMO MINIMAX

Após isso, o algoritmo sobe um 'nó' e identifica de quem é o turno (jogador ou oponente). Caso for o turno do oponente, o algoritmo guarda o menor (Min) resultado de suas respectivas ramificações.



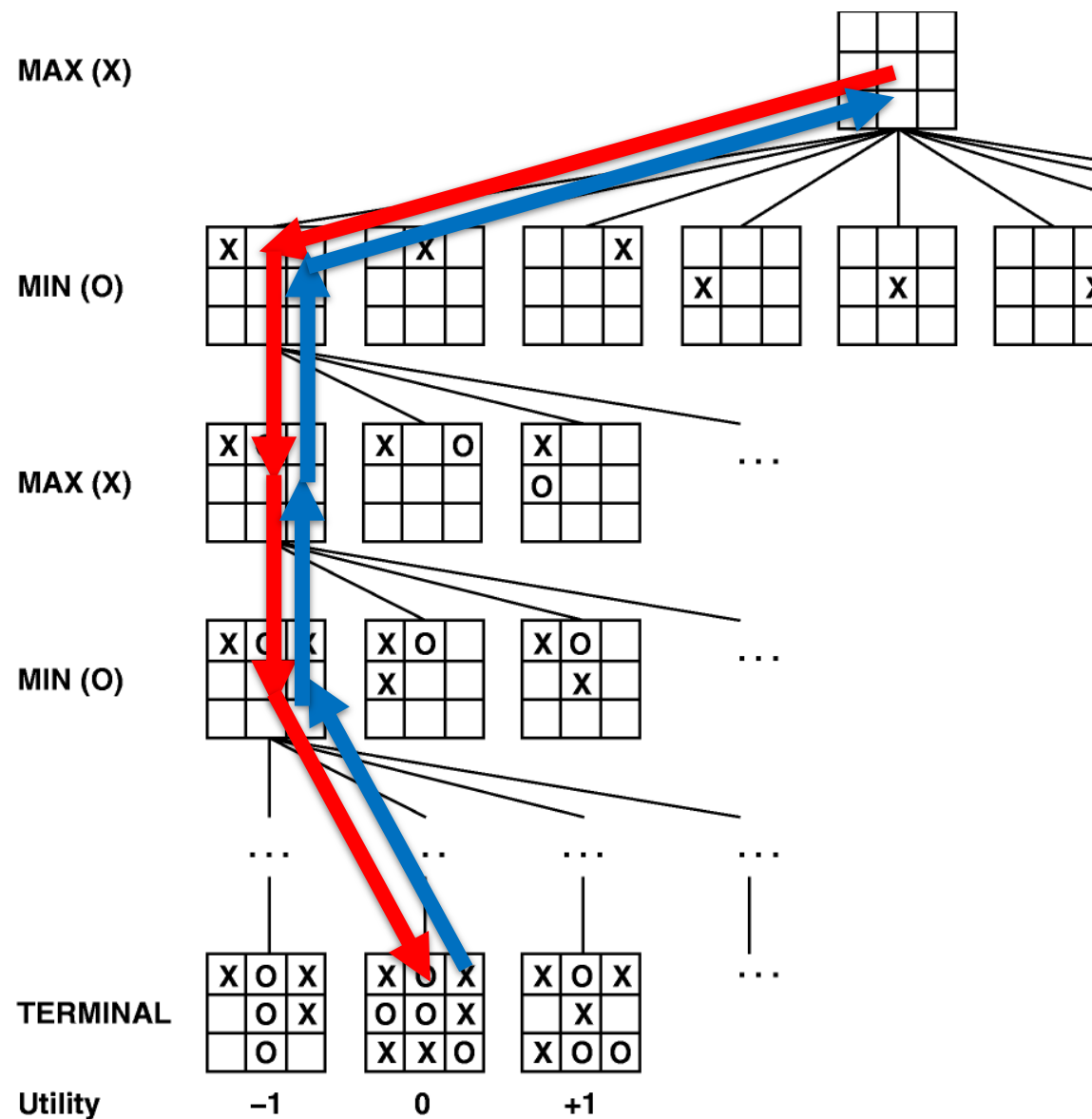
ALGORITMO MINIMAX

Caso for o turno do jogador, o algoritmo guarda o maior (Max) resultado de suas respectivas ramificações.



ALGORITMO MINIMAX

Este processo se repete até chegar no primeiro 'nó' da Árvore.



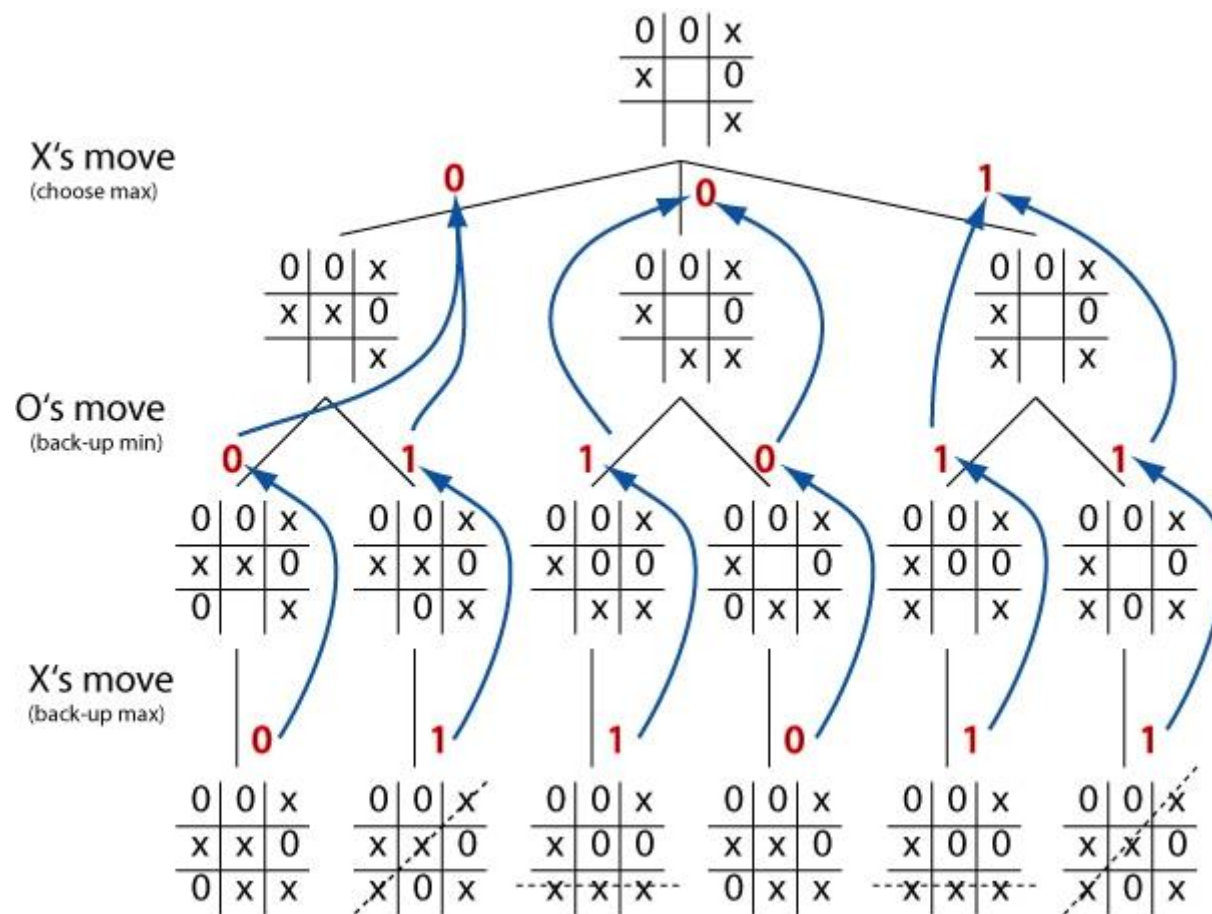
ALGORITMO MINIMAX

Isso significa que:

caso a decisão do jogador ganhar ou perder esteja nas mãos do oponente, o Minimax atribui como não sendo seguro entrar por este caminho,

indicando somente como '1' os caminhos em que o jogador possa vencer a partida

mesmo que o oponente jogue da melhor forma possível.



PSEUDOCÓDIGO DO ALGORITMO MINIMAX

Fonte: Artificial Intelligence a
Modern Approach, 3rd edition

Duas funções Mutuamente
Recursivas que percorrem os
'nós' Min e Max para achar o
respectivo valor da ramificação,
e possui uma função decisiva
que escolhe o movimento
otimizado.

```
function MINIMAX-DECISION(state) returns an action  
  return arg max  $a \in \text{ACTIONS}(s)$  MIN-VALUE(RESULT(state, a))
```

```
function MAX-VALUE(state) returns a utility value  
  if TERMINAL-TEST(state) then return UTILITY(state)  
   $v \leftarrow -\infty$   
  for each a in ACTIONS(state) do  
     $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(\text{RESULT}(\text{state}, a)))$   
  return v
```

```
function MIN-VALUE(state) returns a utility value  
  if TERMINAL-TEST(state) then return UTILITY(state)  
   $v \leftarrow \infty$   
  for each a in ACTIONS(state) do  
     $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(\text{RESULT}(\text{state}, a)))$   
  return v
```

PSEUDOCÓDIGO DO ALGORITMO MINIMAX

Também é comum ver o minimax representado como uma simples função recursiva:

```
ROTINA minimax(nó, profundidade)
  SE nó é um nó terminal OU profundidade = 0 ENTÃO
    RETORNE o valor da heurística do nó
  SENÃO SE o nó representa a jogada de algum adversário ENTÃO
     $\alpha \leftarrow +\infty$ 
    PARA CADA filho DE nó
       $\alpha \leftarrow \min(\alpha, \text{minimax}(\text{filho}, \text{profundidade}-1))$ 
    FIM PARA
    RETORNE  $\alpha$ 
  SENÃO
     $\alpha \leftarrow -\infty$ 
    PARA CADA filho DE nó
       $\alpha \leftarrow \max(\alpha, \text{minimax}(\text{filho}, \text{profundidade}-1))$ 
    FIM PARA
    RETORNE  $\alpha$ 
  FIM SE
FIM ROTINA
```

CRIANDO UM JOGO DA VELHA

Agora vamos criar um simples Jogo da Velha, com as seguintes características:

- Humano x Máquina
- Máquina “burra” (joga aleatoriamente)

```
Jogador O, escolha um espaço.  
5
```

```
  X | 2 | 3  
---|---|---  
  4 | 0 | 6  
---|---|---  
  7 | 8 | 9
```

```
Jogador X, escolha um espaço.  
9
```

```
  X | 2 | 3  
---|---|---  
  4 | 0 | 6  
---|---|---  
  7 | 8 | X
```

```
Jogador O, escolha um espaço.  
█
```

main

1 branch

0 tags

Go to file

Add file

Code

About

allankassio

Update README.md

bdb5cd2 1 minute ago 3 commits

.gitignore	Initial commit	9 minutes ago
LICENSE	Initial commit	9 minutes ago
README.md	Update README.md	1 minute ago
dummy.py	Add files via upload	8 minutes ago
minimax.py	Add files via upload	8 minutes ago

README.md

tictactoe

This project was created with the aim of presenting the concept of a game tree in Data Structure tutoring for students of UNDB technology courses.

The project has two files. One implements tic-tac-toe with a dumb, random AI:

No description, website, or topics provided.

Readme

GPL-3.0 license

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

CRIANDO UM JOGO DA VELHA

<https://github.com/allankassio/tictactoe>



DÚVIDAS?