Introdução à Inteligência Artificial

PVP 5C – Procura Adversa
Monte Carlo

José Coelho,
2023

Índice

Procura Monte Carlo

- Recebe um estado e retorna uma ação e utilidade

$$MonteCarlo: S \rightarrow A \times (P \rightarrow \mathbb{Z})$$

  - $MonteCarlo(s)$:
    1. $teste(s) = 1$, retornar $(\_, util(s))$
    2. Caso contrário, enquanto puder: $\forall a \in A$
       1. $(\_, f) \leftarrow SimularJogo(exe(s, a))$
       2. $tf_a \leftarrow tf_a + f$
    3. $ma \leftarrow argmax_{a \in A}\, tf_{a,jog(s)}$
    4. Retornar (ma, tf_{ma})
  - $SimularJogo(s)$:
    1. $teste(s) = 1$, retornar $(\_, util(s))$
    2. Caso contrário, retornar $SimularJogo\left(exe(s, PolíticaJogo(s))\right)$

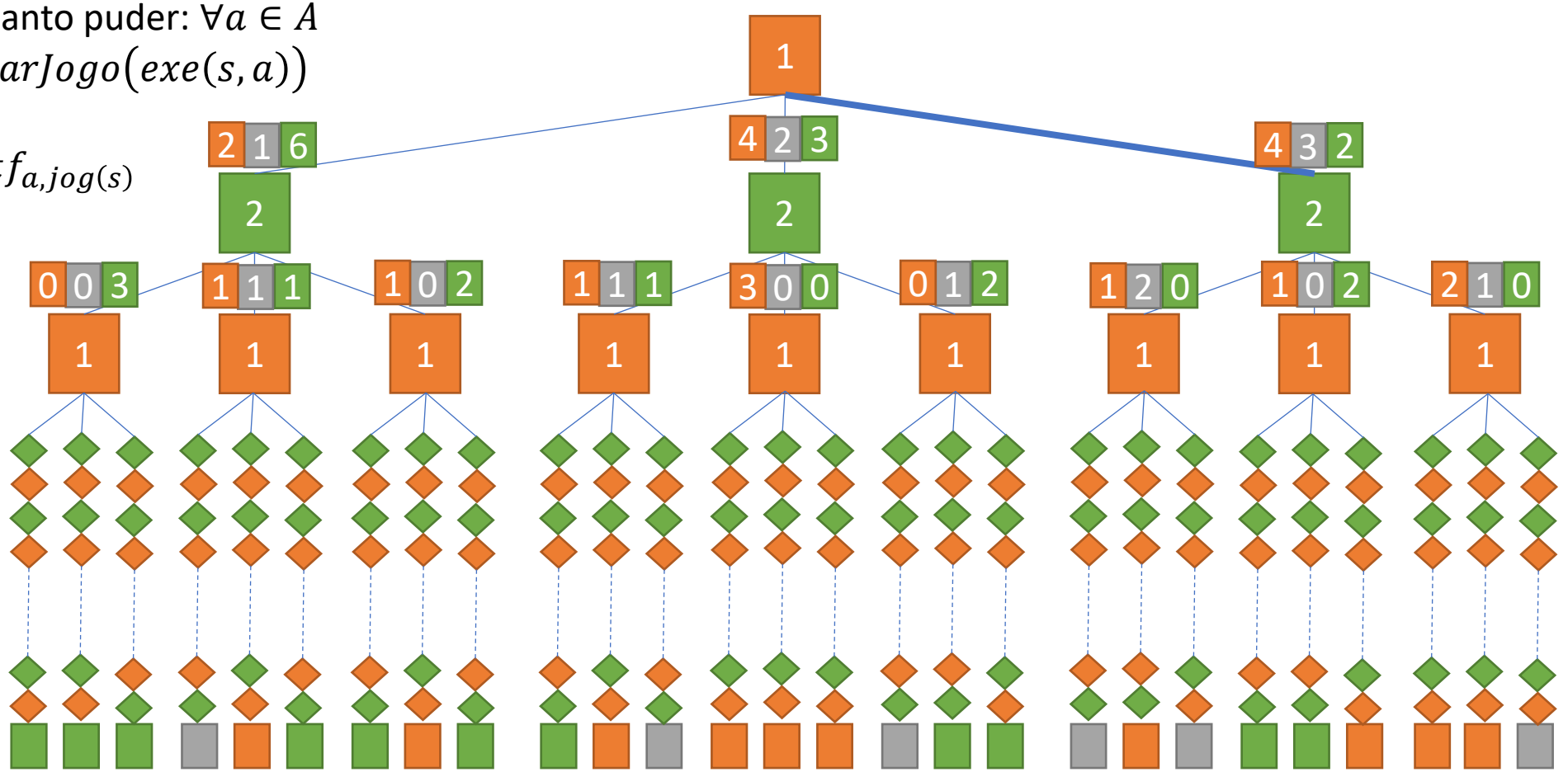
Procura Monte Carlo

- Ideia:
  - Simular muitos jogos completos
  - Jogar o lance com maior valor esperado
- Alternativa à procura na árvore de jogo
  - Realizar uma árvore de jogo limita, para contabilização de resultados
- Grande vantagem quando a ramificação é elevada
- Política de jogo:
  - Aleatória
  - Enviesada para lances óbvios
  - Utilização da avaliação heurística, mantendo aleatoriedade

Exemplo Monte Carlo, 2 jogadores

$MonteCarlo(s)$:

1. $teste(s) = 1$, retornar $(\_, util(s))$
2. Caso contrário, enquanto puder: $\forall a \in A$
   1. $(\_, f) \leftarrow SimularJogo(exe(s, a))$
   2. $tf_a \leftarrow tf_a + f$
3. $ma \leftarrow argmax_{a \in A}\, tf_{a,jog(s)}$
4. Retornar (ma, tf_{ma})



Vitória 1

Vitória 2

Empate

Exemplo Monte Carlo, 3 jogadores

$MonteCarlo(s)$:

1. $teste(s) = 1$, retornar $(\_, util(s))$
2. Caso contrário, enquanto puder: $\forall a \in A$
   1. $(\_, f) \leftarrow SimularJogo(exe(s, a))$
   2. $tf_a \leftarrow tf_a + f$
3. $ma \leftarrow argmax_{a \in A}\, tf_{a,jog(s)}$
4. Retornar (ma, tf_{ma})

Vitória/Empate/Derrota:

123

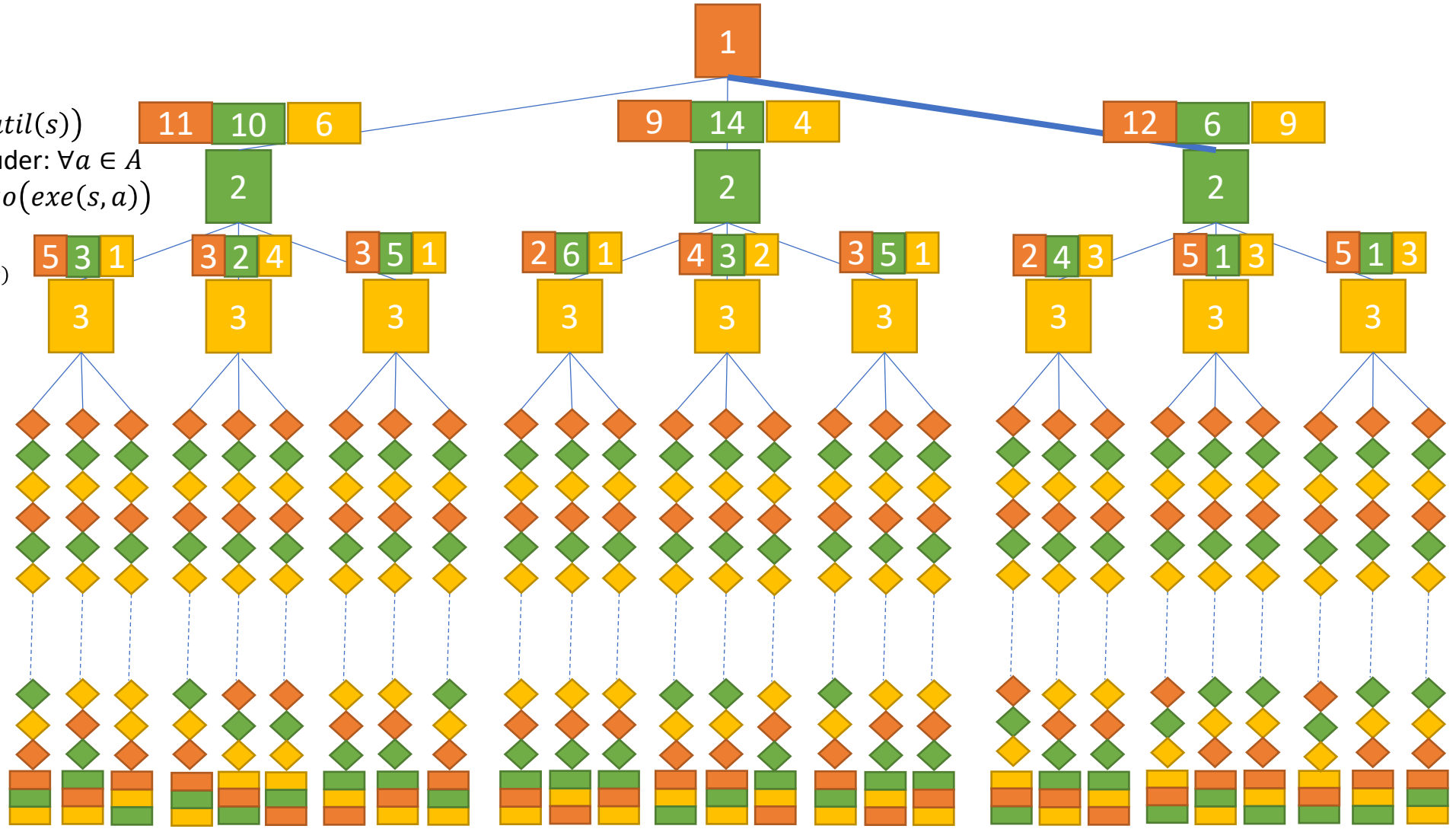
132

231

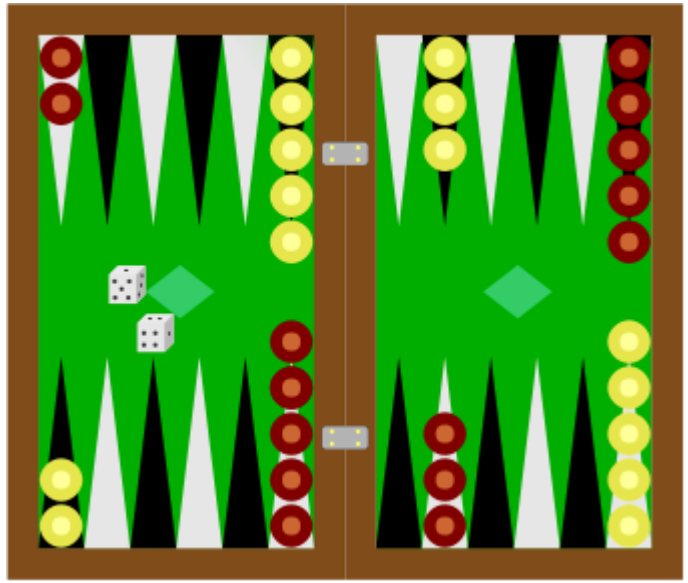
213

312

321



Jogos em ambiente aleatório

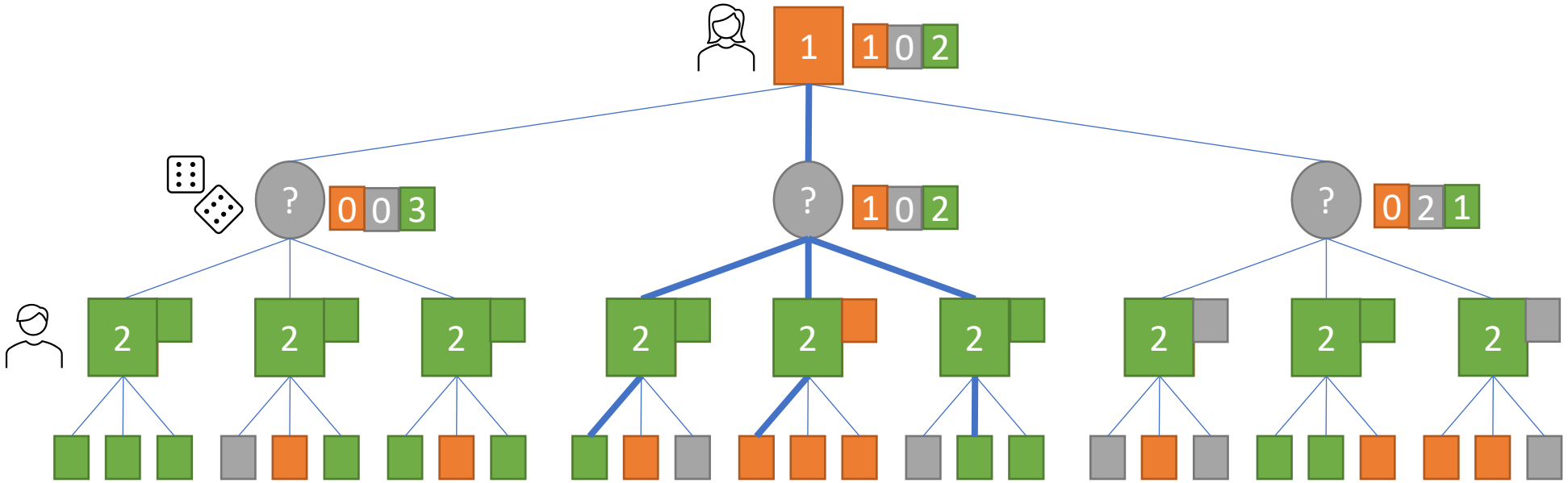


- Idêntico a existir um jogador aleatório
- Alterações:
  - MiniMax:
    - Considerar o valor médio
  - MiniMaxAB:
    - Valores de alfa/beta não alterados nestes nós
  - Monte Carlo:
    - Considerar o nó aleatório como sendo um jogador aleatório com política de jogo aleatória
- Performance:
  - MiniMax / MiniMaxAB: degrada-se, sendo idêntico a um aumento da ramificação
  - Monte Carlo: sem alterações
- Exemplo: Gamão

Exemplo MiniMax, 2 jogadores, aleatório

$MiniMax(s)$:

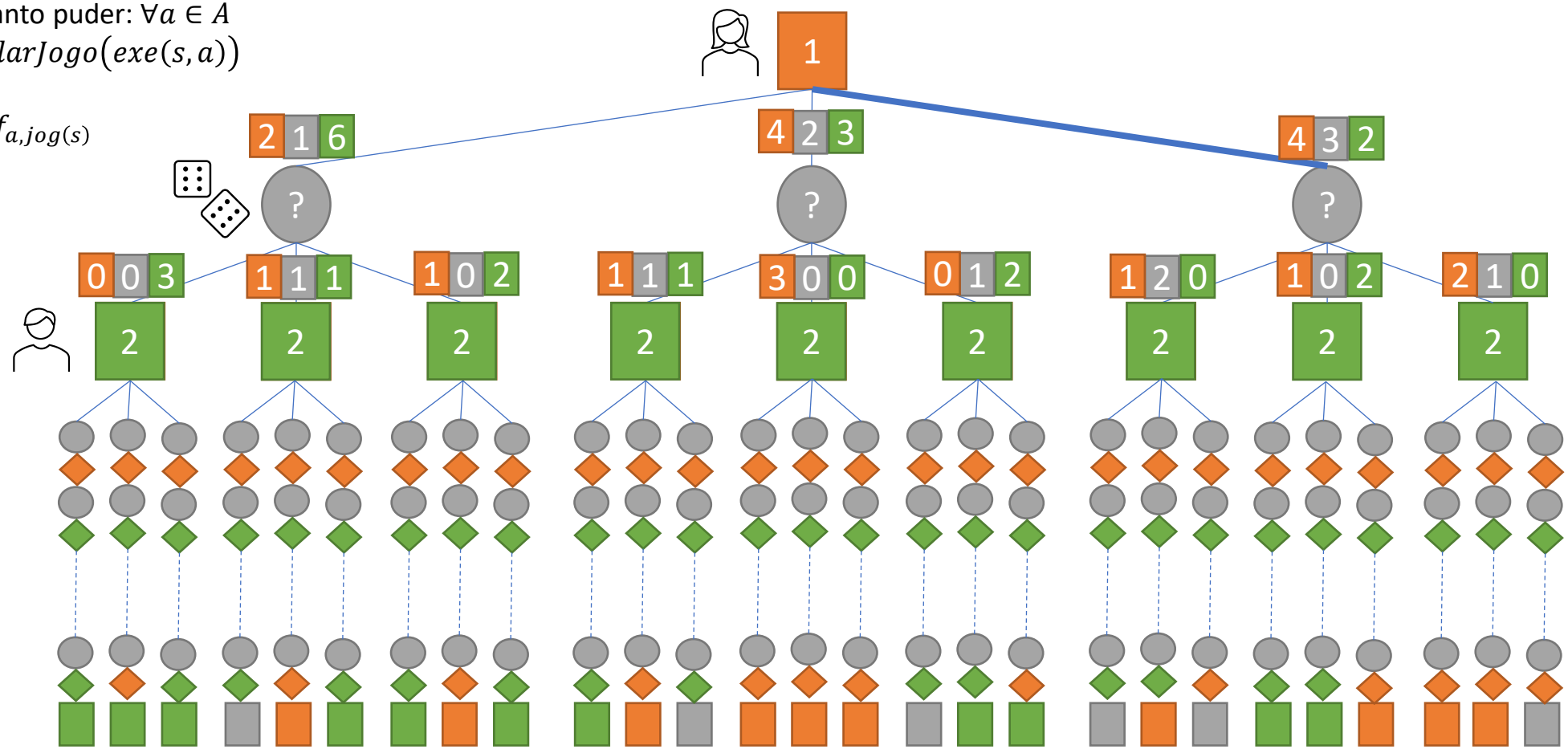
1. $teste(s) = 1$, retornar $(\_, util(s))$
2. Caso contrário: $\forall a \in A$
   1. $(\_, f) \leftarrow MinMax(exe(s, a))$
   2. Se $f_{jog(s)} > mf_{jog(s)}$ então $mf \leftarrow f, ma \leftarrow a$
3. Retornar (ma, mf)



Exemplo Monte Carlo, 2 jogadores, aleatório

$MonteCarlo(s)$:

1. $teste(s) = 1$, retornar $(\_, util(s))$
2. Caso contrário, enquanto puder: $\forall a \in A$
   1. $(\_, f) \leftarrow SimularJogo(exe(s, a))$
   2. $tf_a \leftarrow tf_a + f$
3. $ma \leftarrow argmax_{a \in A}\, tf_{a,jog(s)}$
4. Retornar (ma, tf_{ma})



Vitória 1

Vitória 2

Empate

Jogos parcialmente observáveis

- Parte do estado do jogo não é observável a alguns agentes
- O estado atual é um subconjunto dos estados possíveis S
  - Os algoritmos seriam válidos considerando um nó da árvore de jogo um subconjunto de S

$s \subset S$

- Ações sobre um subconjunto, origina outro subconjunto:
  - Muito pesada a geração de sucessores caso S seja numeroso
  - Impraticável MiniMax e Monte Carlo

$exe: 2^S \times A \rightarrow 2^S \cup \{Imp.\}$

- Estado de crença:
  - Identificar um potencial estado, de entre o subconjunto
  - Executar os algoritmos como se estivéssemos nesse estado
  - Trocar o estado de crença assim que existam evidências para tal
- Exemplo: jogo de cartas

Atividades formativas

AF4a - Procuras Adversas (PnP)

Tentativas a realizar: 1

Exercícios de problemas do manual, para resolver com a técnica de papel-e-lápis (PnP)

AF4c - Procuras Adversas (implementar)

Tentativas a realizar: 1

Exercícios em que se solicita a implementação um problema parecido com os do manual, podendo reutilizar código existente.

AF4b - Procuras Adversas (executar)

Tentativas a realizar: 1

Exercícios para executar código com a implementação de problemas do manual, e observar as diferentes características dos algoritmos na prática.

Recursos utilizados

- Microsoft Power Point
- Clipchamp, voz de síntese Duarte
- Vimeo
- Russell, S. J. & Norvig, P. (2010). Artificial intelligence: A modern approach (3rd ed). Prentice Hall.