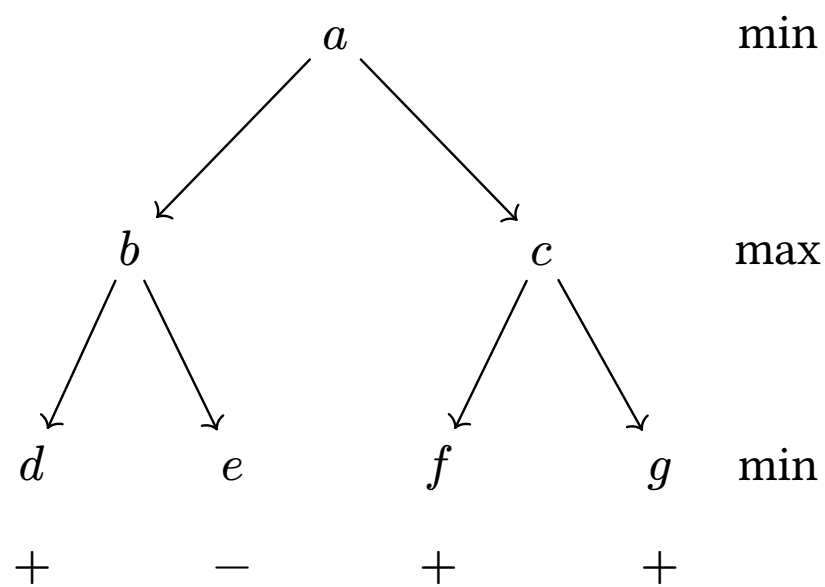
Monte Carlo Tree Search

Noah Syrkis

September 8, 2025

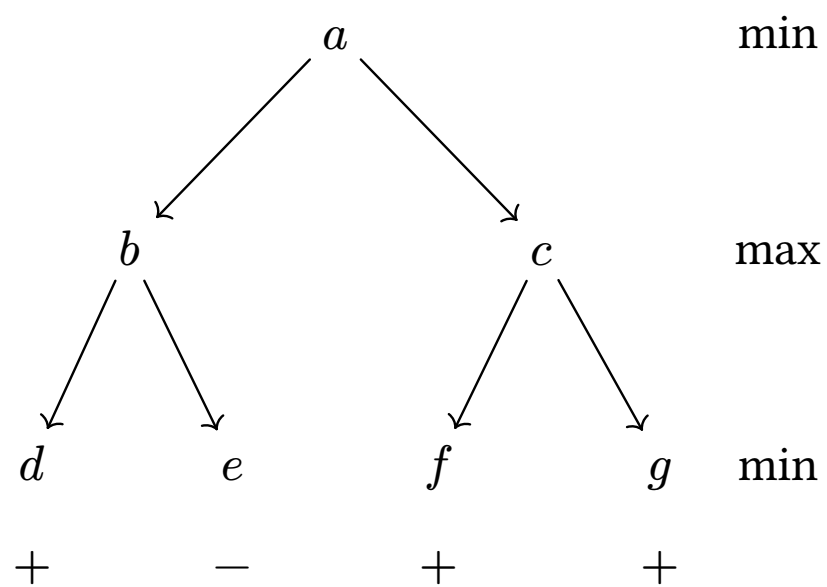
1 | Minimax

- Suppose we have a function that:
- given a state and an action returns a new state,
- and another that given a state returns who won
- What can we do?



1 | Minimax

- Suppose we have a function that:
- given a state and an action returns a new state,
- and another that given a state returns who won
- What can we do? Play perfectly and never loose



1 | Minimax

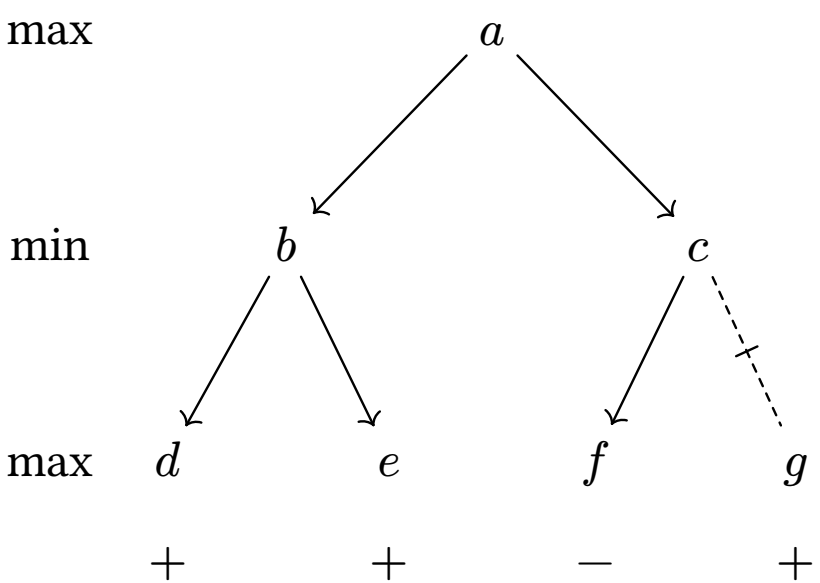
- We can win (or at least not loose) any game[1] by:
  1. Calling the minimax function for all actions
  2. Storing the values of each action in a list
  3. Taking the action with the highest value
- How can we do better? What are the issues?

Algorithm 1: minimax(node, maxim)

```
1 if node is terminal
2     return the value of node
3 bestValue = −∞ if maxim else ∞
4 condition = max if maxim else min
5 for each child of node
6     value = minimax(child, not maxim)
7     bestValue = condition(bestValue, value)
8 return bestValue
```

[1] that is two player, winnable, deterministic, etc.

2 | $\alpha - \beta$ pruning



- Skip branches worse than current floor
- α and β refer to those precisely floors

2 | $\alpha - \beta$ pruning

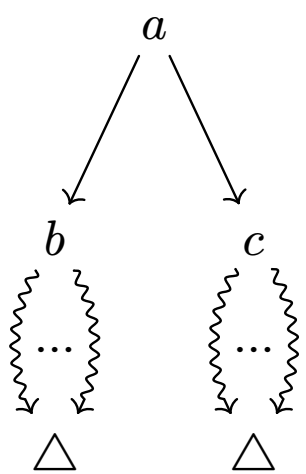
- Algorithm 2 looks daunting but the idea is:
- Stop exploring paths you already know are bad

Algorithm 2: $\alpha - \beta$ pruning(node, maxim, α, β)

```
1  if node is terminal
2      return the value of node
3  bestValue = −∞ if maxim else ∞
4  condition = max if maxim else min
5  for each child of node
6      value = minimax(child, not maxim, α, β)
7      bestValue = condition(bestValue, value)
8      α = (condition(α, value) if maxim else α)
9      β = (condition(β, value) if not maxim else β)
10     if α >= β; break
11 return bestValue
```

3 | MCTS

- Monte Carlo (random) tree search
- Core idea: sample from bottom of each branchj
- How much to sample from each branch?
- How should we reach the bottom?



3 | MCTS

- The intuitive idea of