

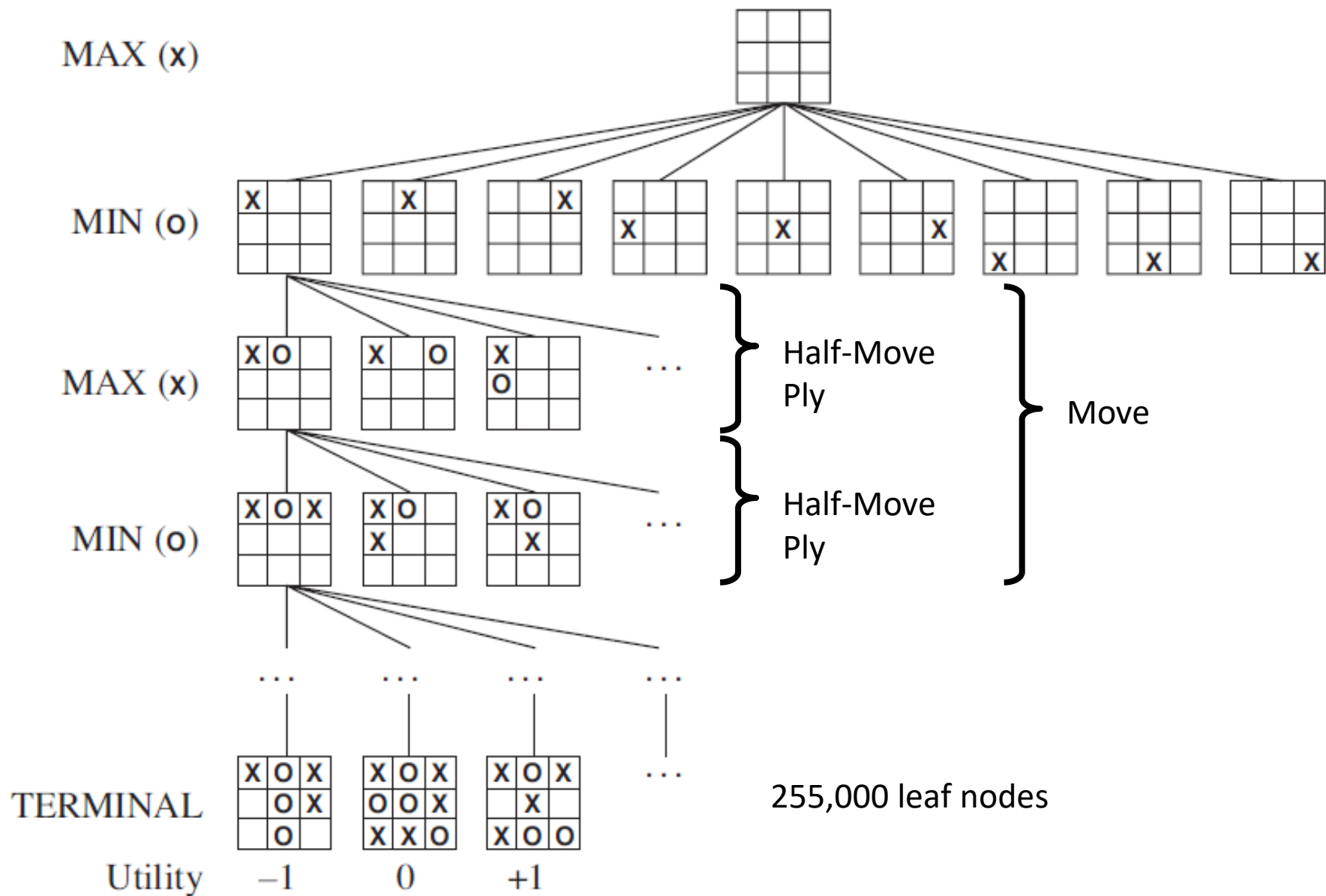
Adversarial Search: Games

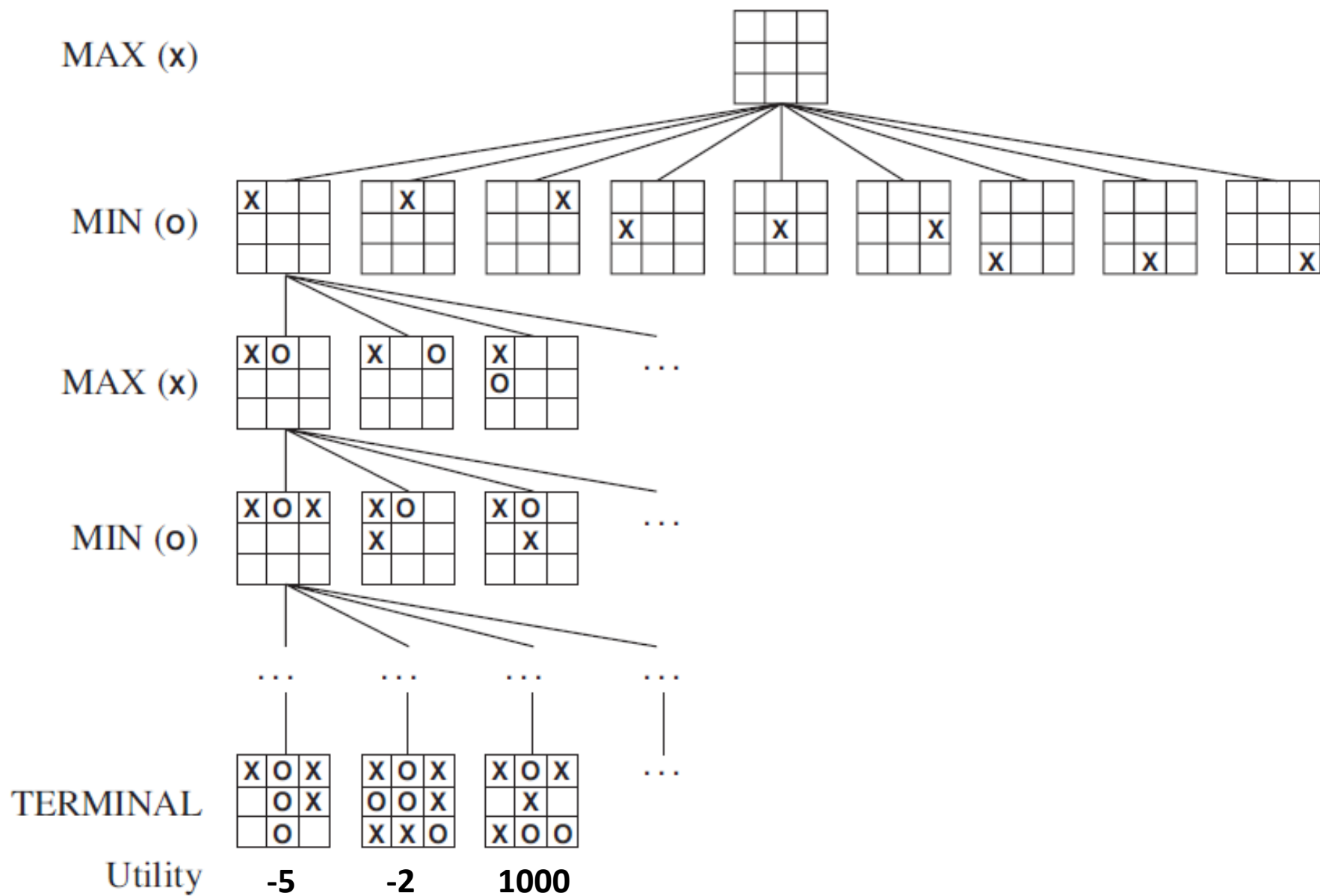
CS161

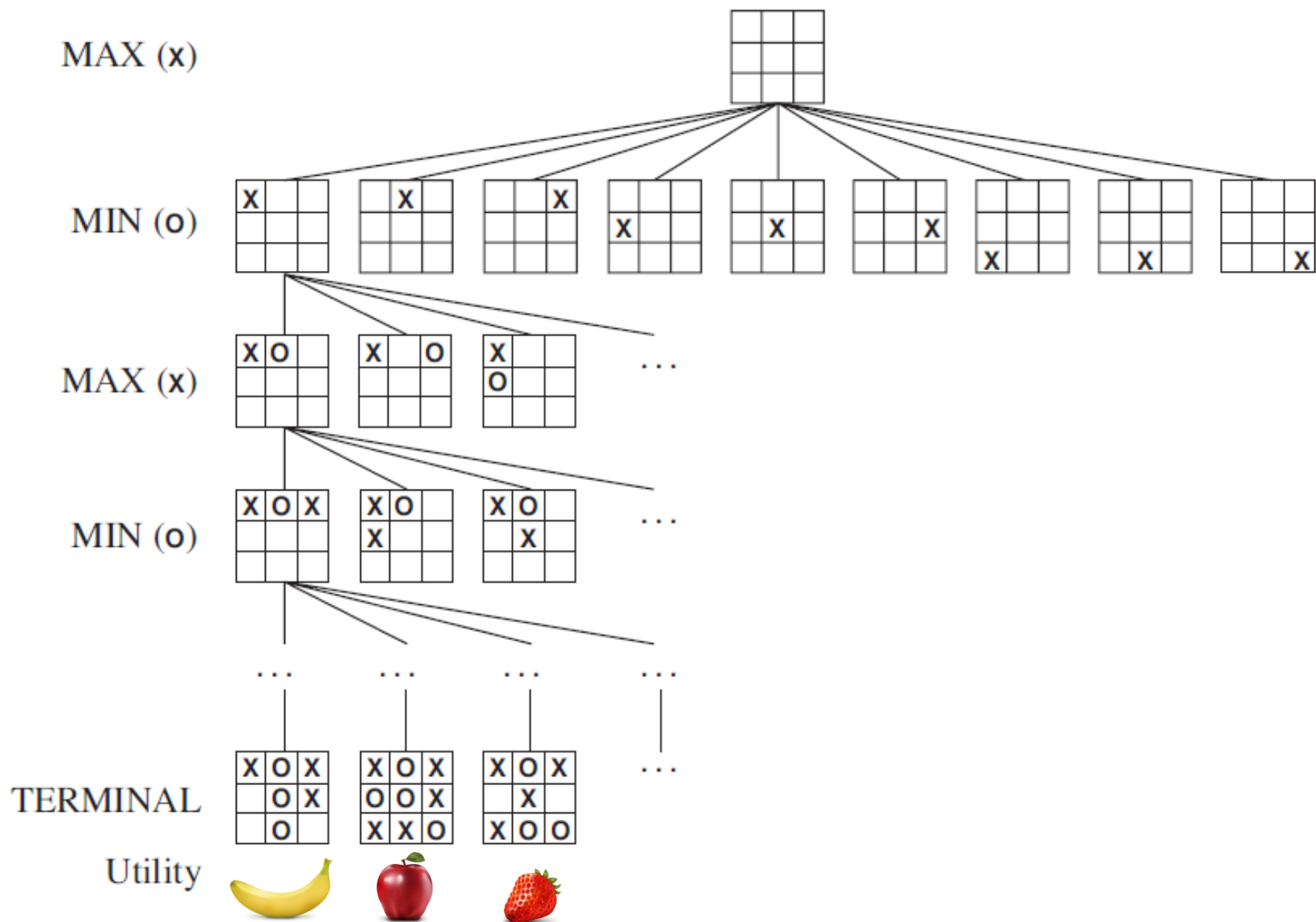
Guy Van den Broeck

Adversarial Search Problems

- Long history in CS
- Like a search problem but
 - Opponent actions (uncertainty)
 - Often large branching factor (35 in chess)
 - Often large depth (around 100 moves in chess)
 - Time limits (sometimes)
 - Utility function (win/loss/draw)
- Different types of games
 - Perfect/imperfect information
 - Deterministic/chance
 - (non-)zero-sum

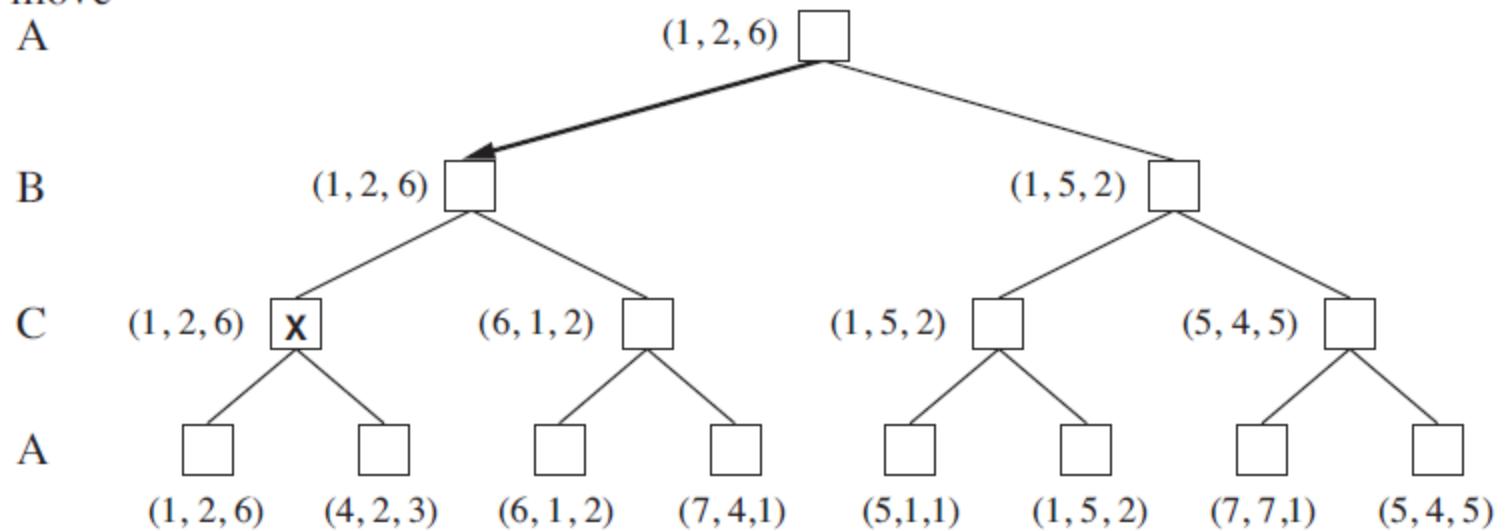






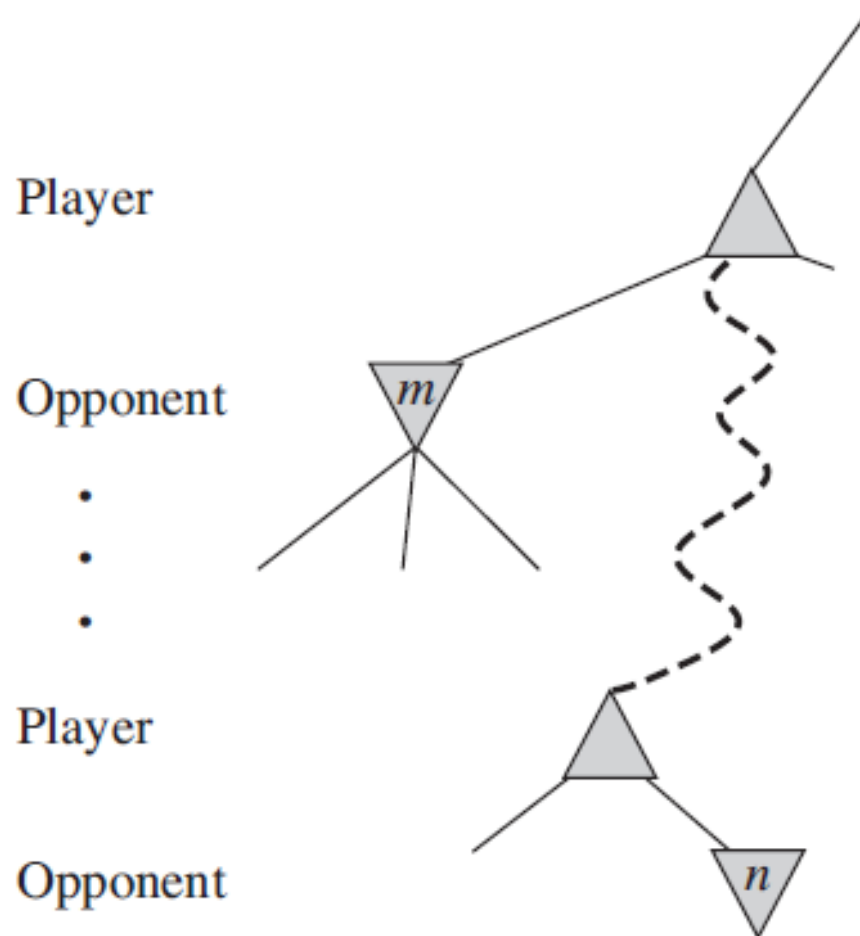
Three Players

to move
A



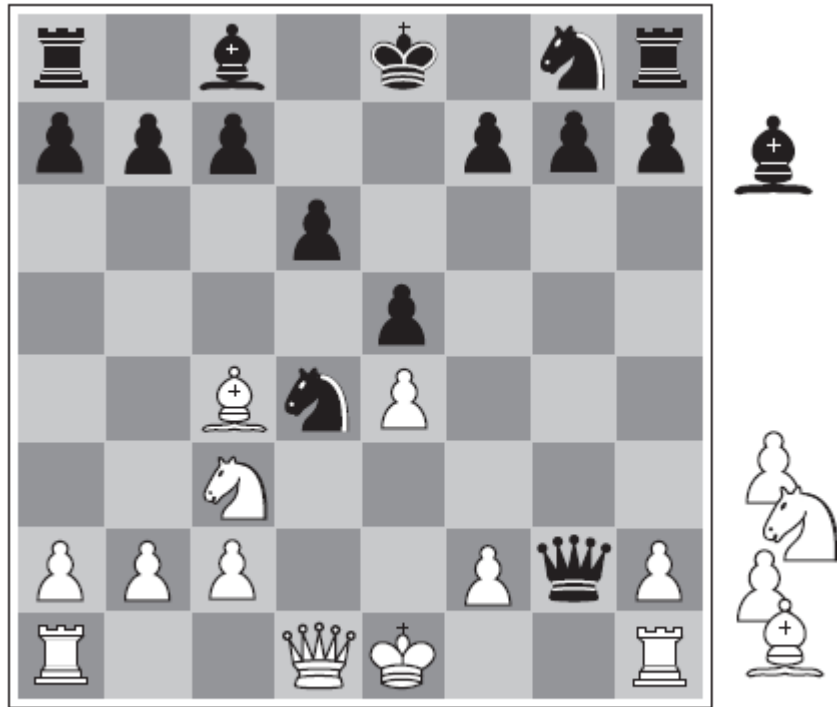
Can we save time?

Pruning

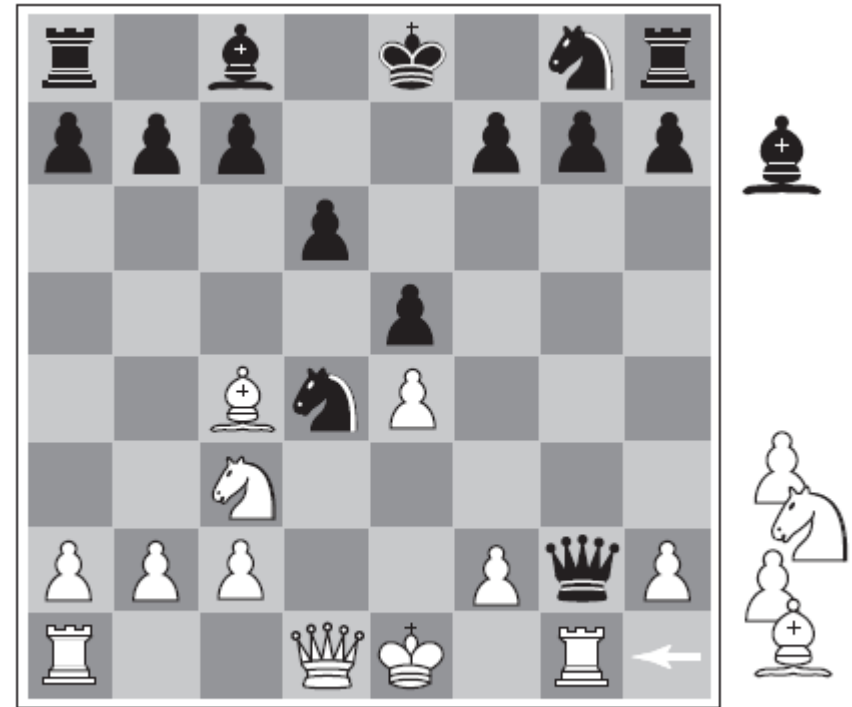


What if game tree still too large?

Evaluation: Tricky



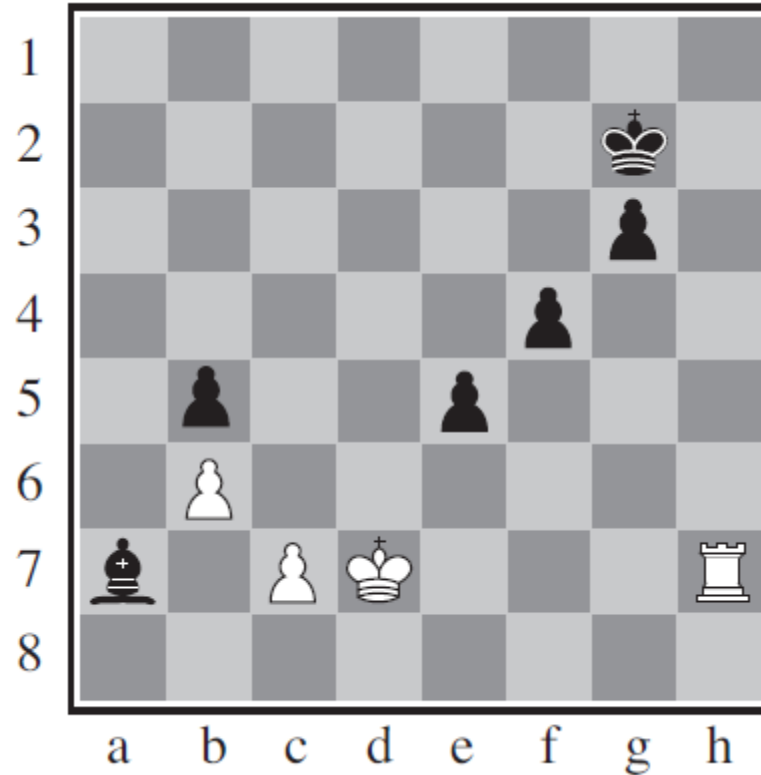
(a) White to move



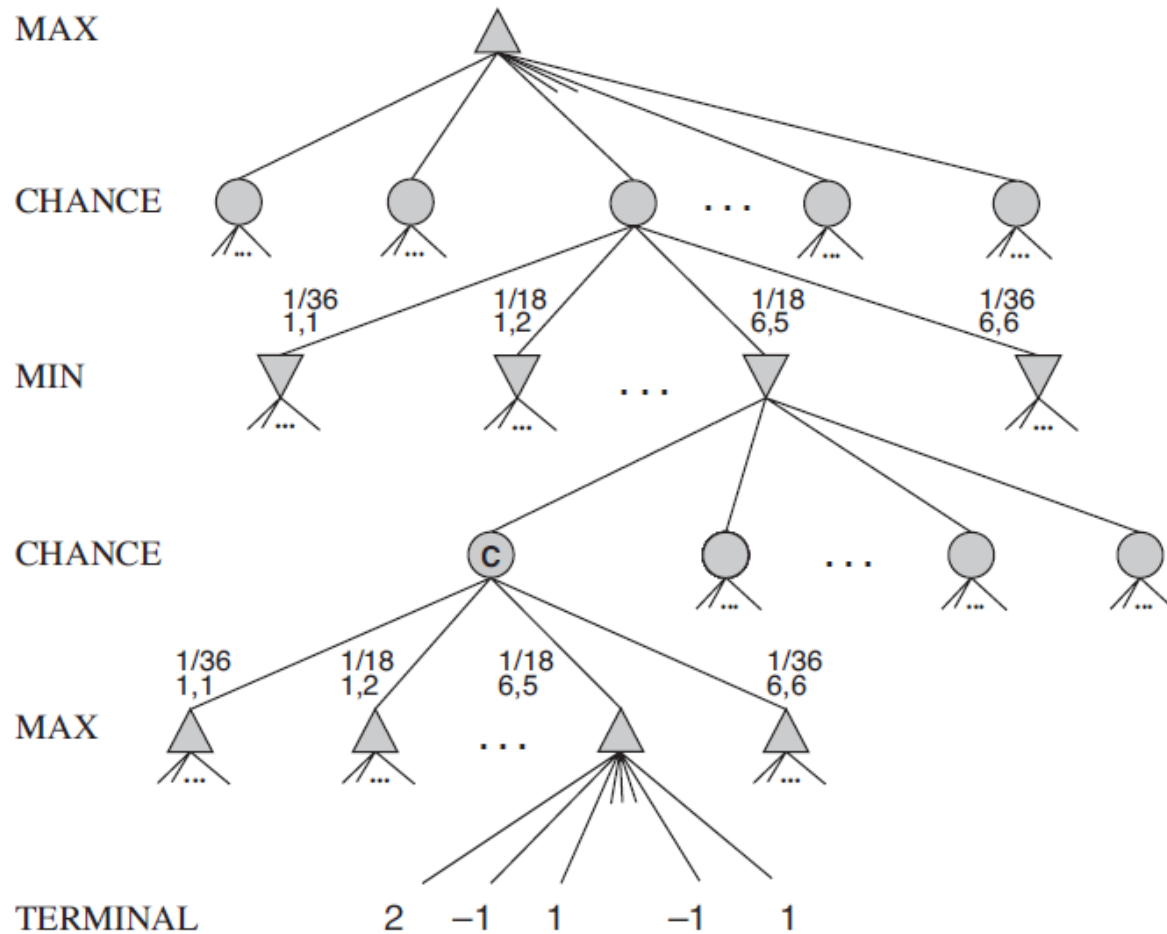
(b) White to move

Two chess positions that differ only in the position of the rook at lower right. In (a), Black has an advantage of a knight and two pawns, which should be enough to win the game. In (b), White will capture the queen, giving it an advantage that should be strong enough to win.

Horizon Effect: Black to Move



Chance: Expecti-Minimax



Changing Rewards

