INSTITUT TEKNOLOGI BANDUNG

1920

Modul 4: Adversarial Search

Adversarial Search

Masayu Leylia Khodra
(masayu@informatika.org)

KK IF – Teknik Informatika – STEI ITB

Inteligensi Buatan
(*Artificial Intelligence*)

Adversarial Search

Non-Adversarial Search

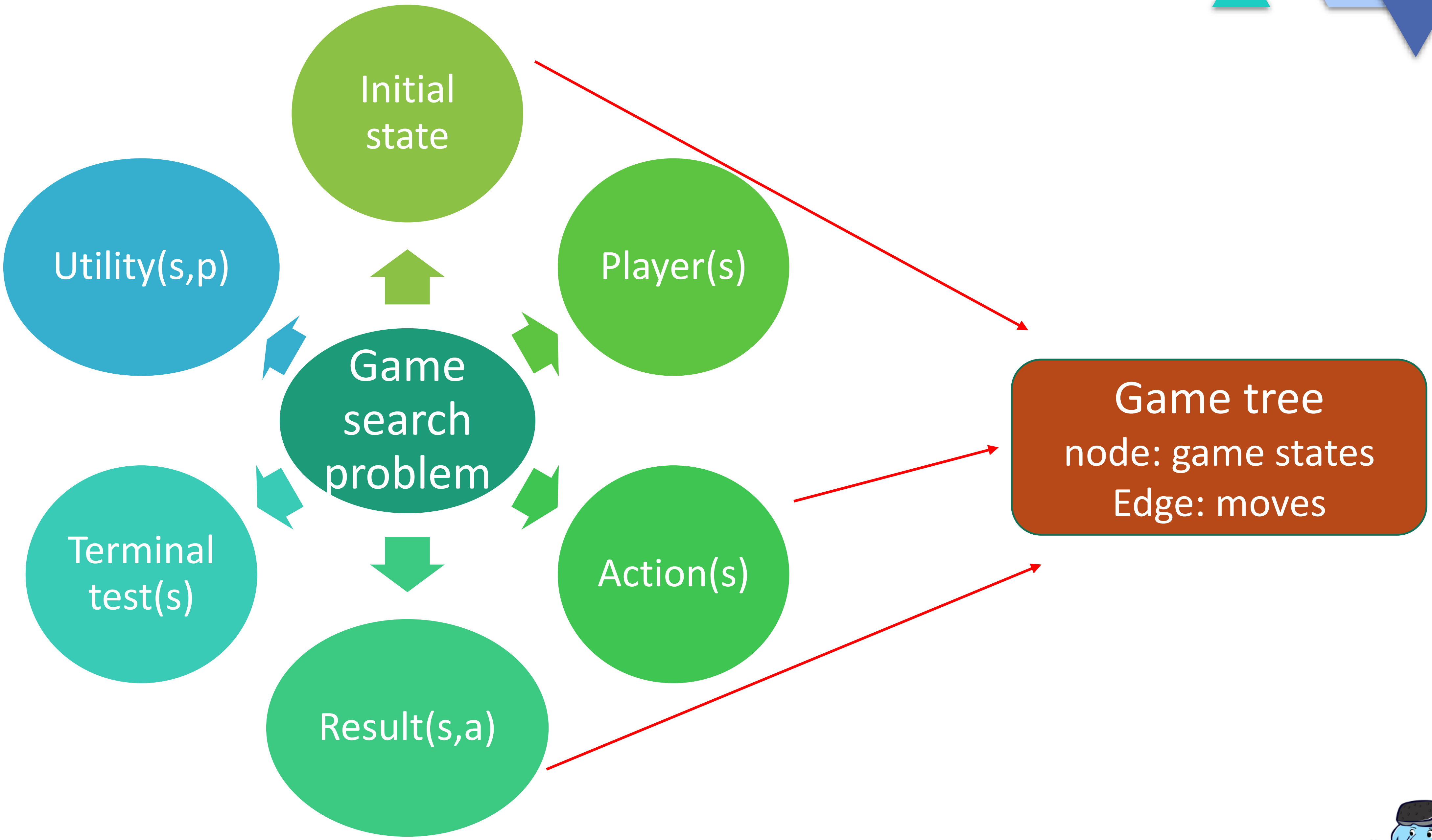
Approximate solution: strategy

Multiagent: competitive environment (games)

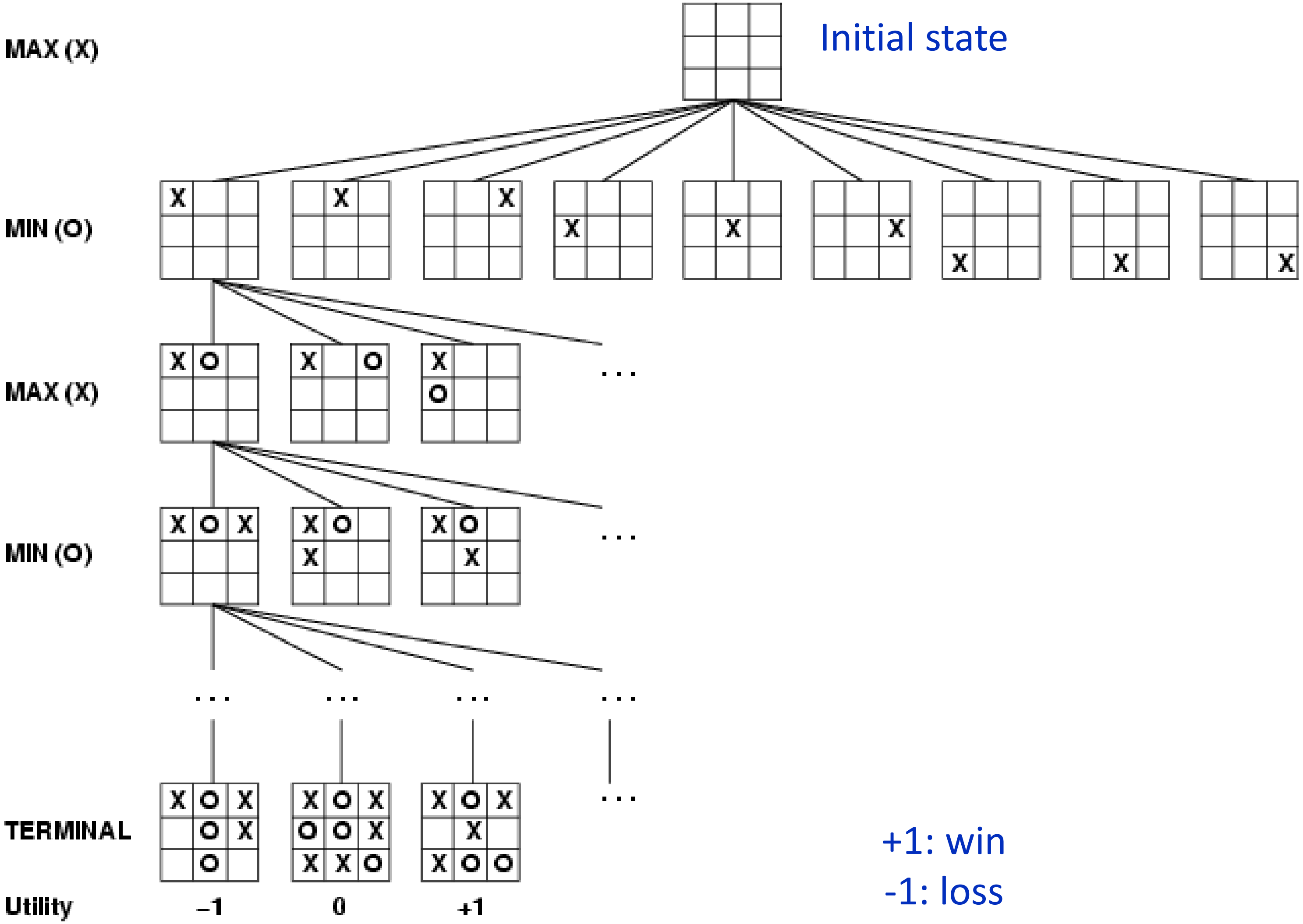
Optimize solution: path to goal or solution state

Deterministic and fully observable

Turn-taking


Initial state
Player(s)
Action(s)
Result(s,a)
Terminal test(s)
Utility(s,p)
Game search problem
Game tree
node: game states
Edge: moves

Game tree (2-players, tic-tac-toe)



Initial state

node: game states
Edge: moves

+1: win
-1: loss
0: draw

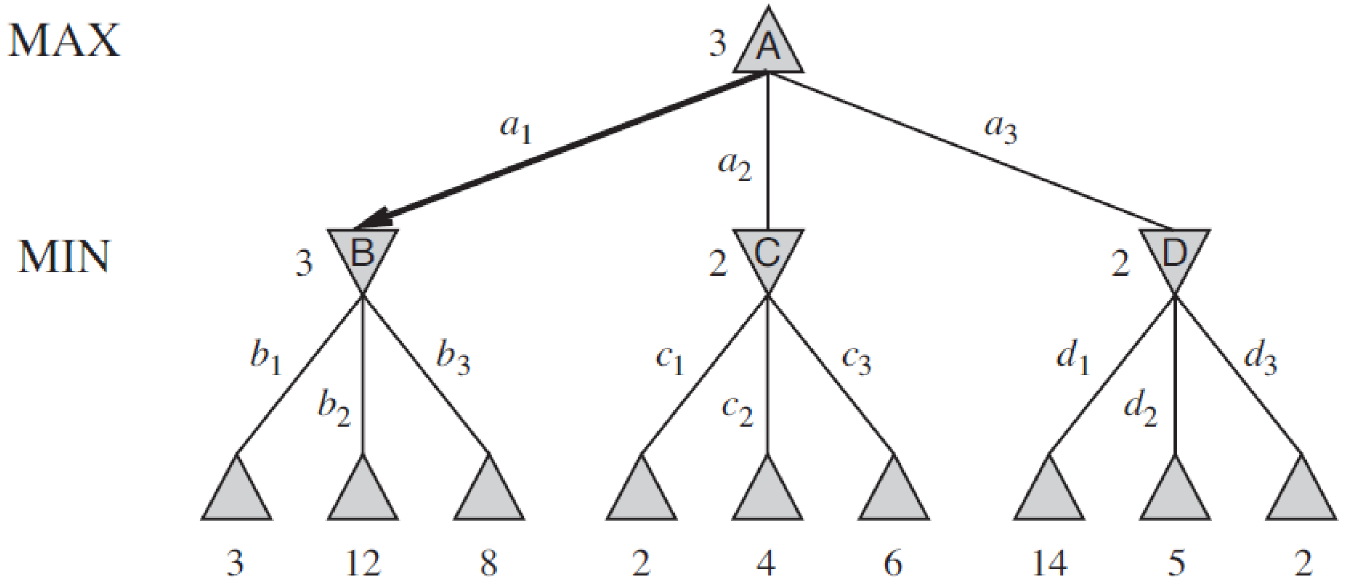
Optimal Decision with Minimax Algorithm

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

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**
    $v \leftarrow$ MIN(v, MAX-VALUE(RESULT(s, a)))
  return v

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**
    $v \leftarrow$ MAX(v, MIN-VALUE(RESULT(s, a)))
  return v

Minimax: 2-ply Game Tree

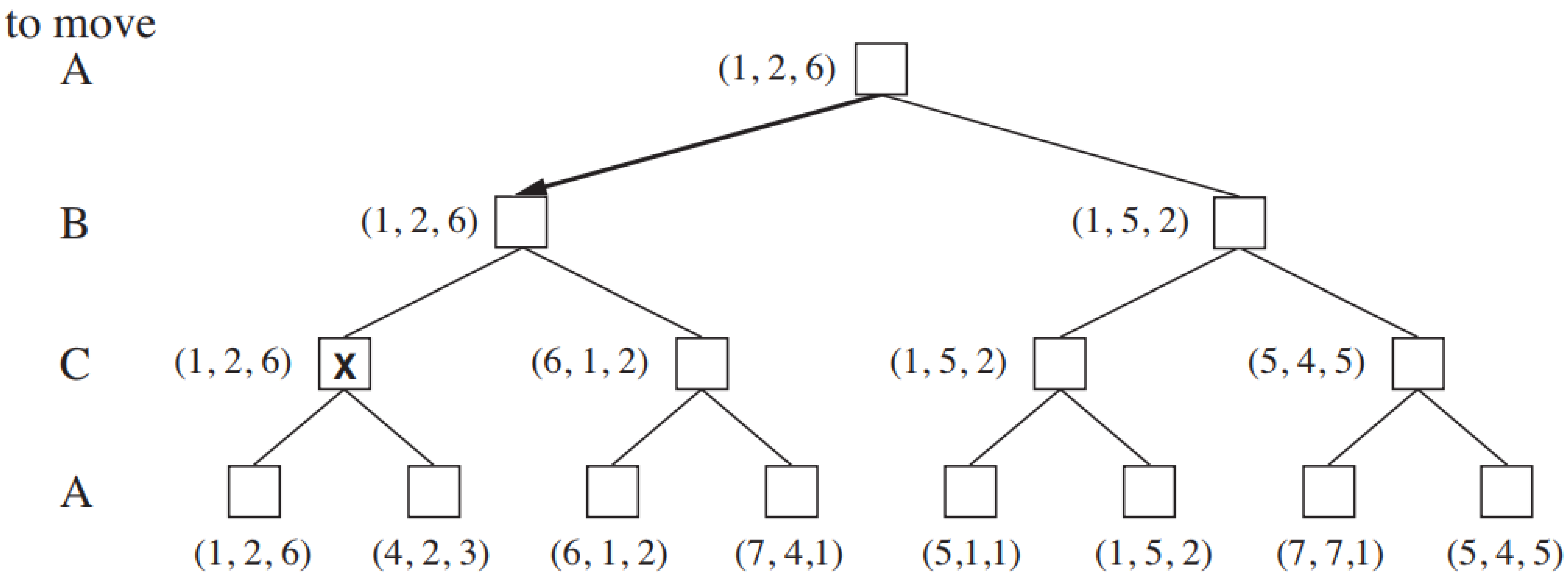


Minimax(B)=min(
Minimax(result(B,b1)),
Minimax(result(B,b2)),
Minimax(result(B,b3)))
=min(3,12,8)=3

Minimax(A)=max(Minimax(B),
Minimax(C),Minimax(D))
=max(3,2,2)=3

$$\text{MINIMAX}(s) = \begin{cases} \text{UTILITY}(s) & \text{if TERMINAL-TEST}(s) \\ \max_{a \in Actions(s)} \text{MINIMAX}(\text{RESULT}(s,a)) & \text{if PLAYER}(s) = \text{MAX} \\ \min_{a \in Actions(s)} \text{MINIMAX}(\text{RESULT}(s,a)) & \text{if PLAYER}(s) = \text{MIN} \end{cases}$$

Minimax 3-ply in Multiplayer Games



Minimax Properties

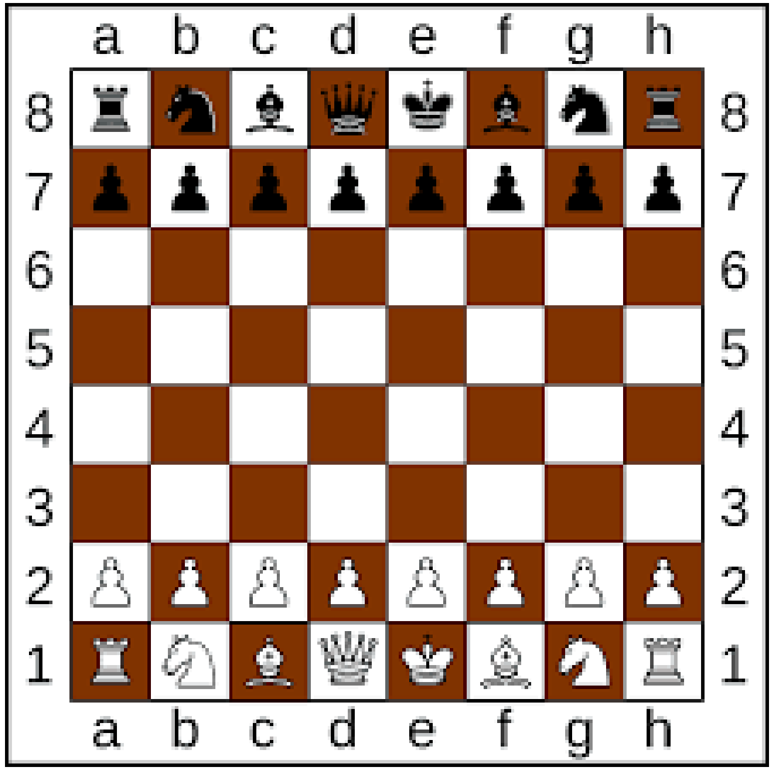
Complete?
Yes (if tree is finite)

Optimal?
Yes (against an optimal opponent)

Time complexity?
$O(b^m)$

Space complexity?
O(bm) (depth-first exploration)

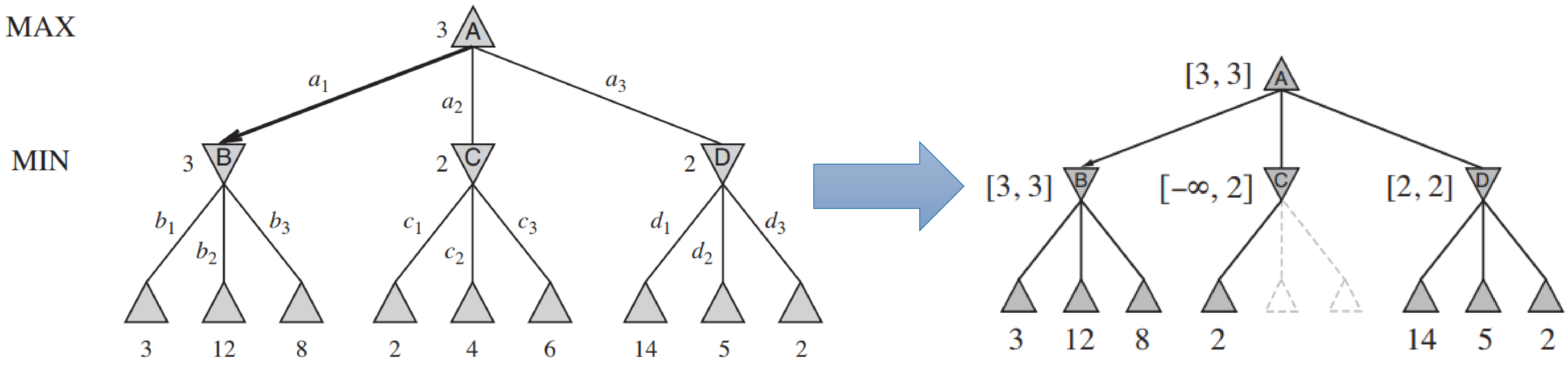
Minimax for Chess



https://commons.wikimedia.org/wiki/File:AAA_SVG_Chessboard_and_chess_pieces_02.svg

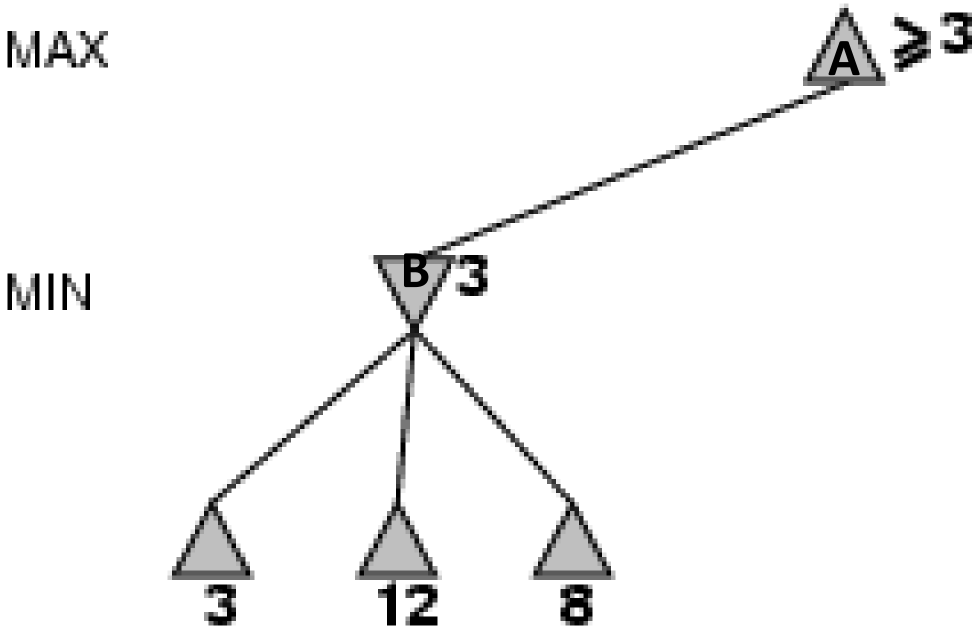
- Branching factor: 35 (avg)
- Games often 50 moves for each player → m=100
- Game states is exponential in the depth of game tree.
- Exact solution is completely infeasible

αβSearch: Minimax with Pruning

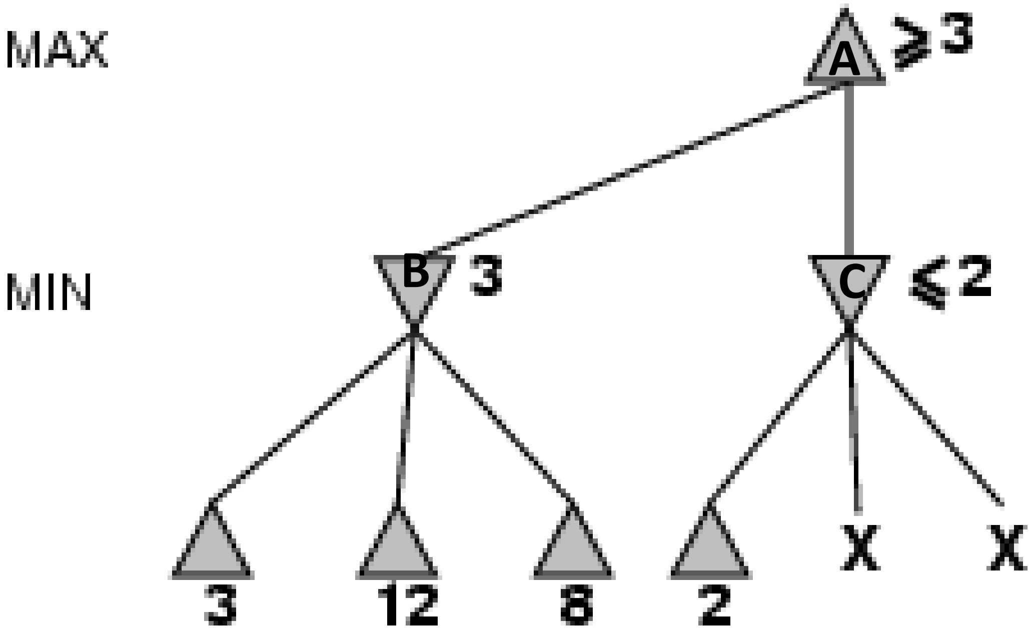


Pruning does not affect final result. It returns the same move as minimax would, but prunes away branches that cannot possibly influence the final decision.

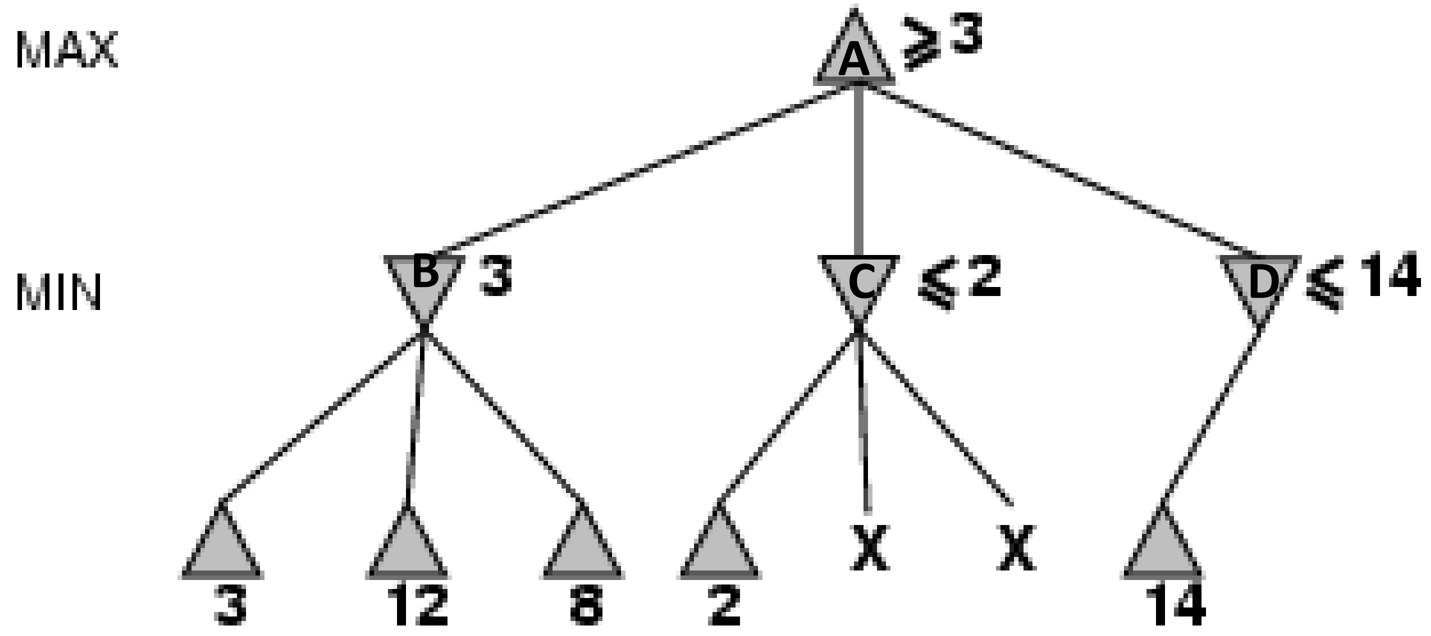
α-β pruning example



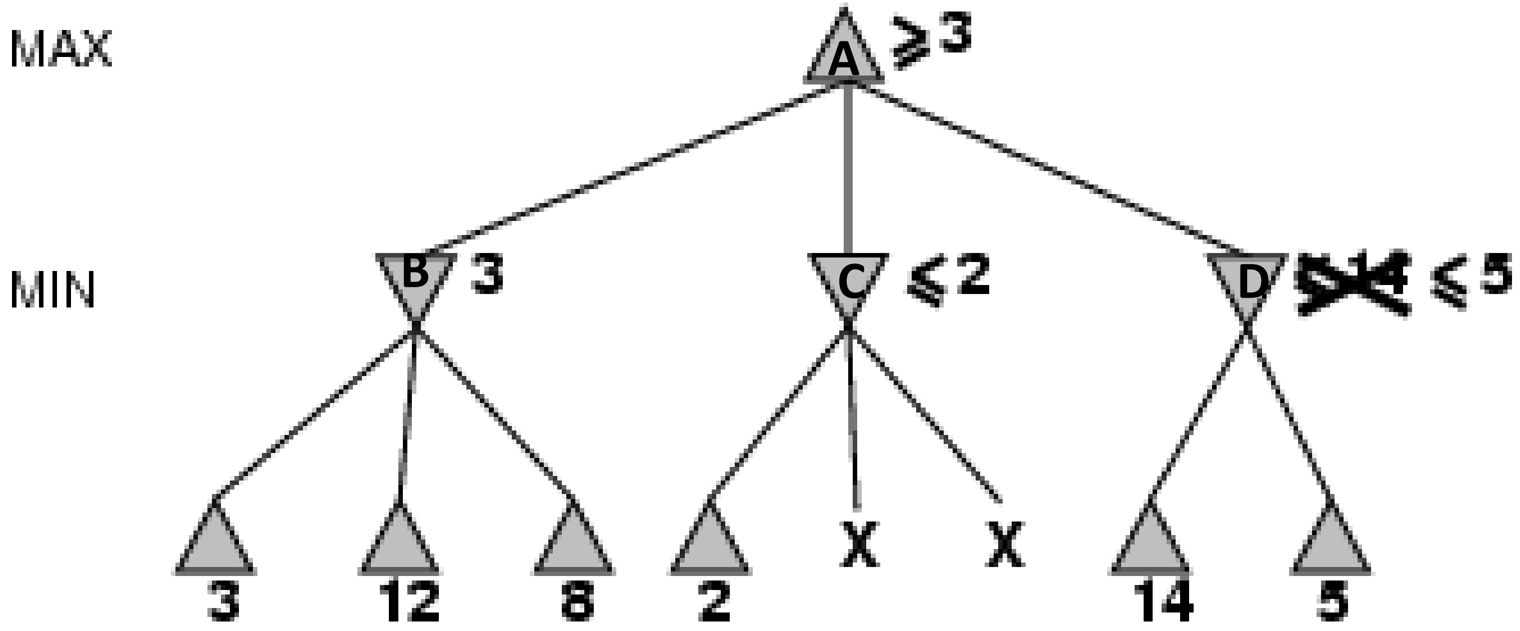
α-β pruning example



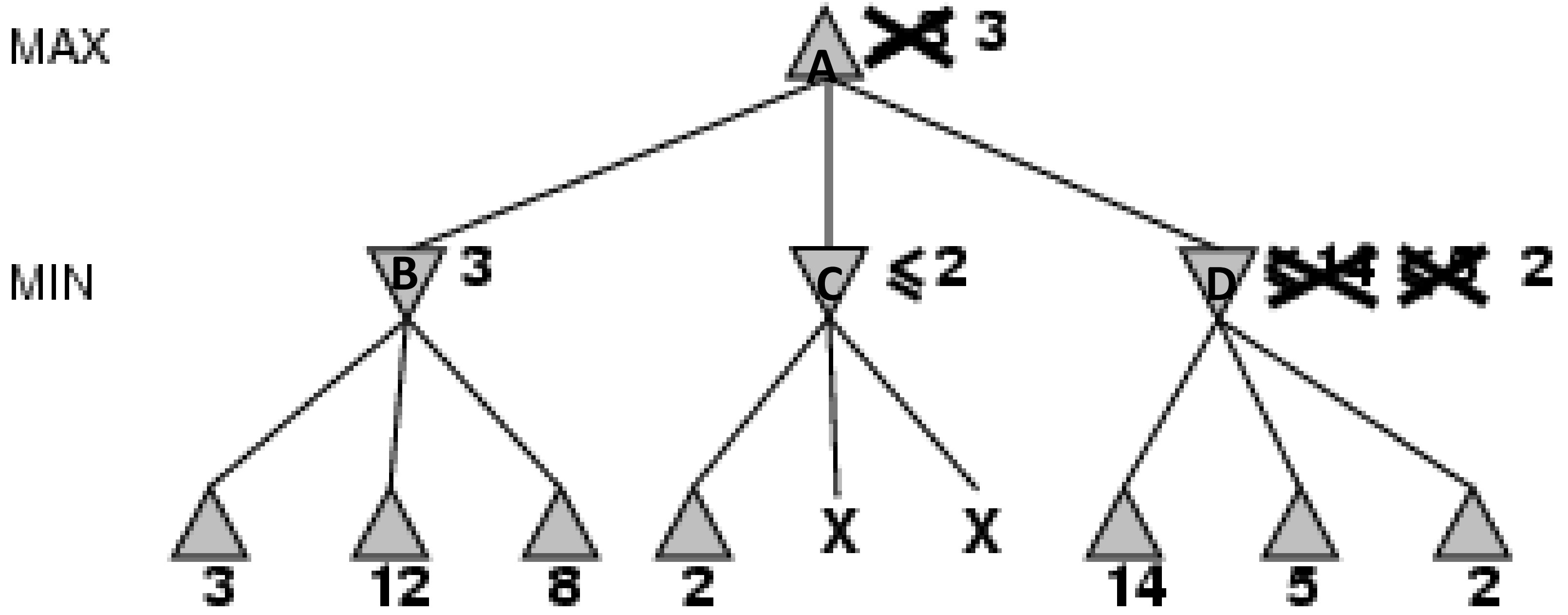
α-β pruning example



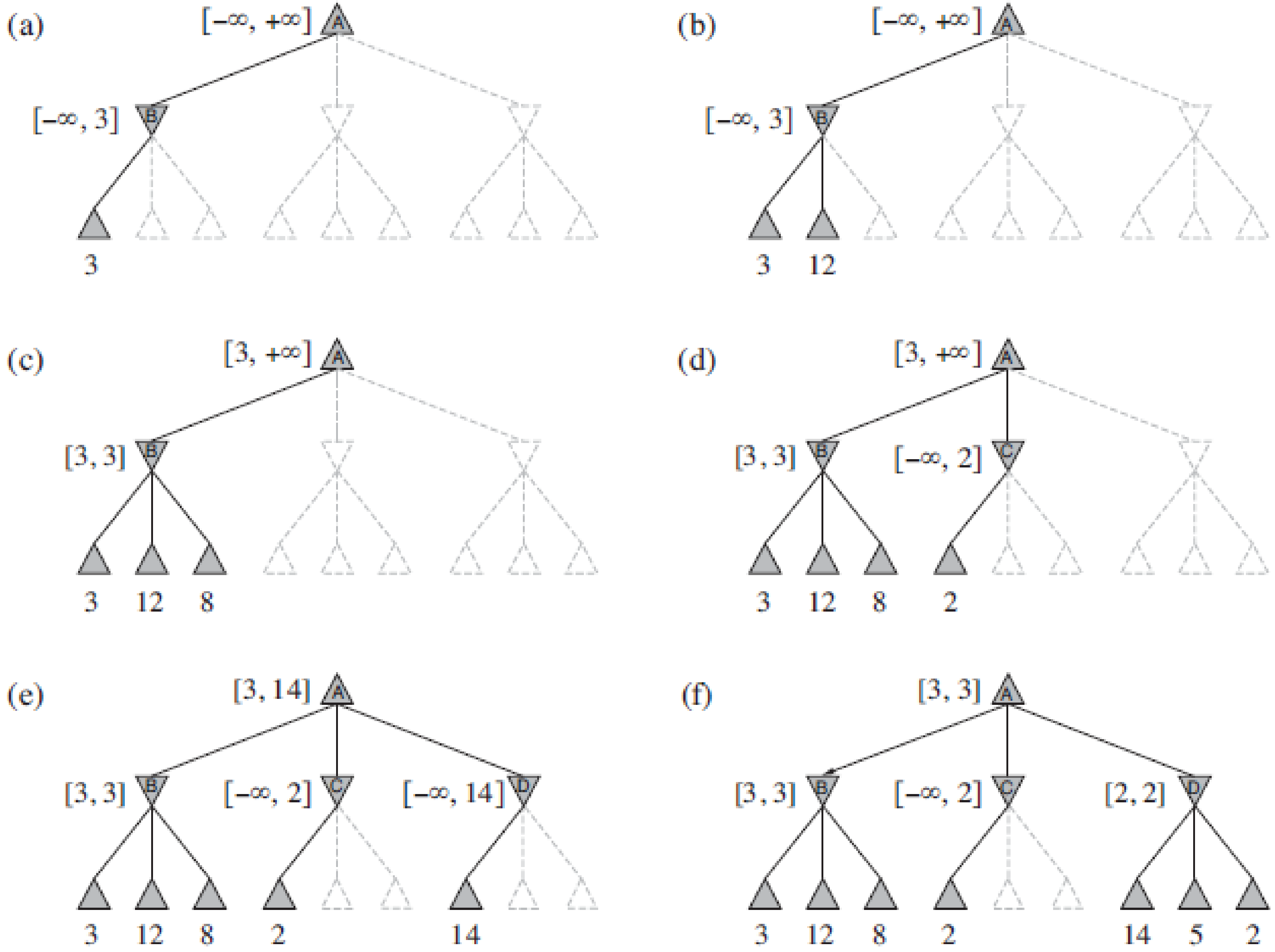
α-β pruning example



α-β pruning example



Algorithm illustration with α-β value



Summary

Adversarial search → Minimax search → αβSearch

INSTITUT TEKNOLOGI BANDUNG

1920