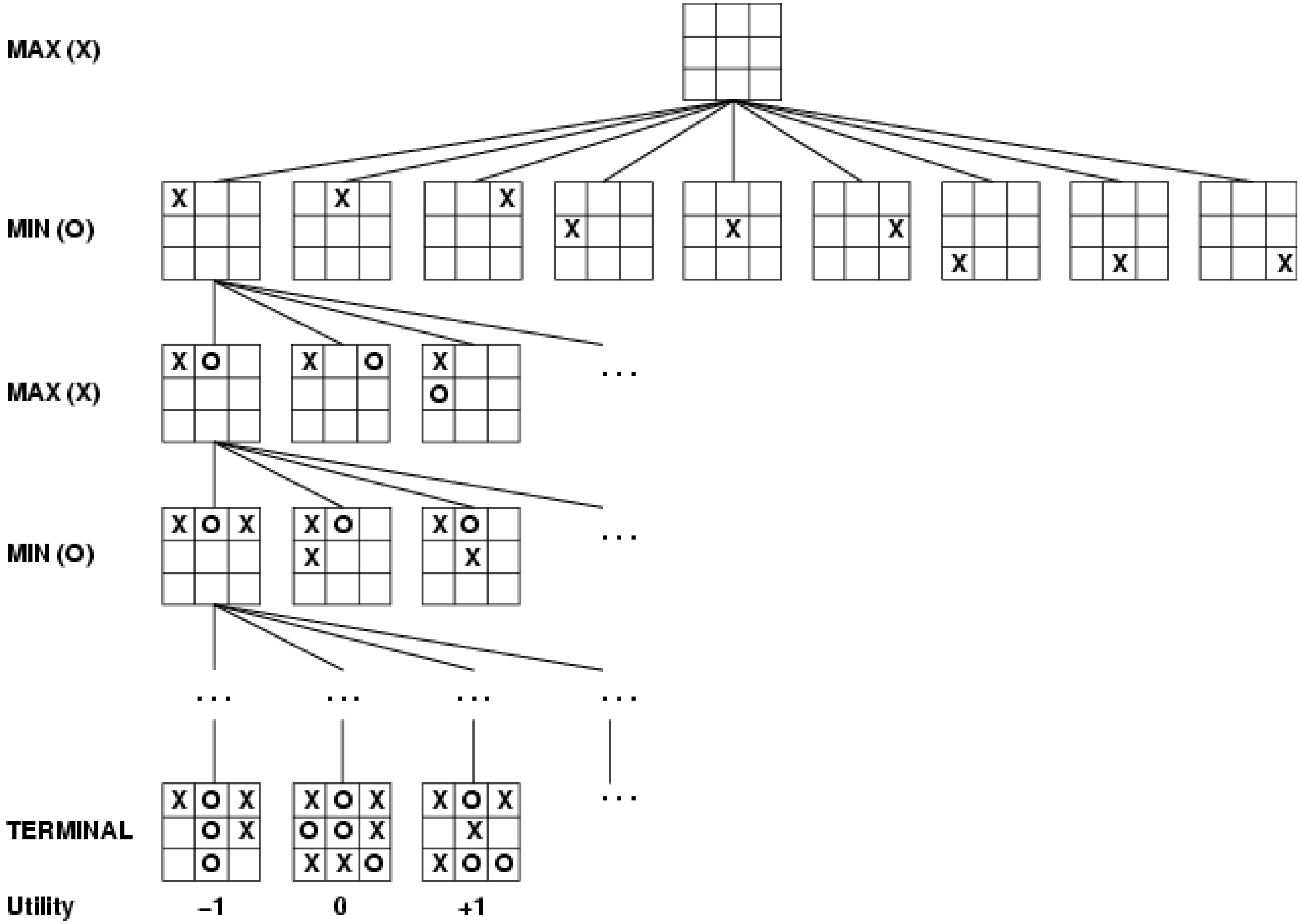
Adversarial Search

Games vs. search problems

- "Unpredictable" opponent → specifying a move for every possible opponent reply
-
- Time limits → unlikely to find goal, must approximate
-

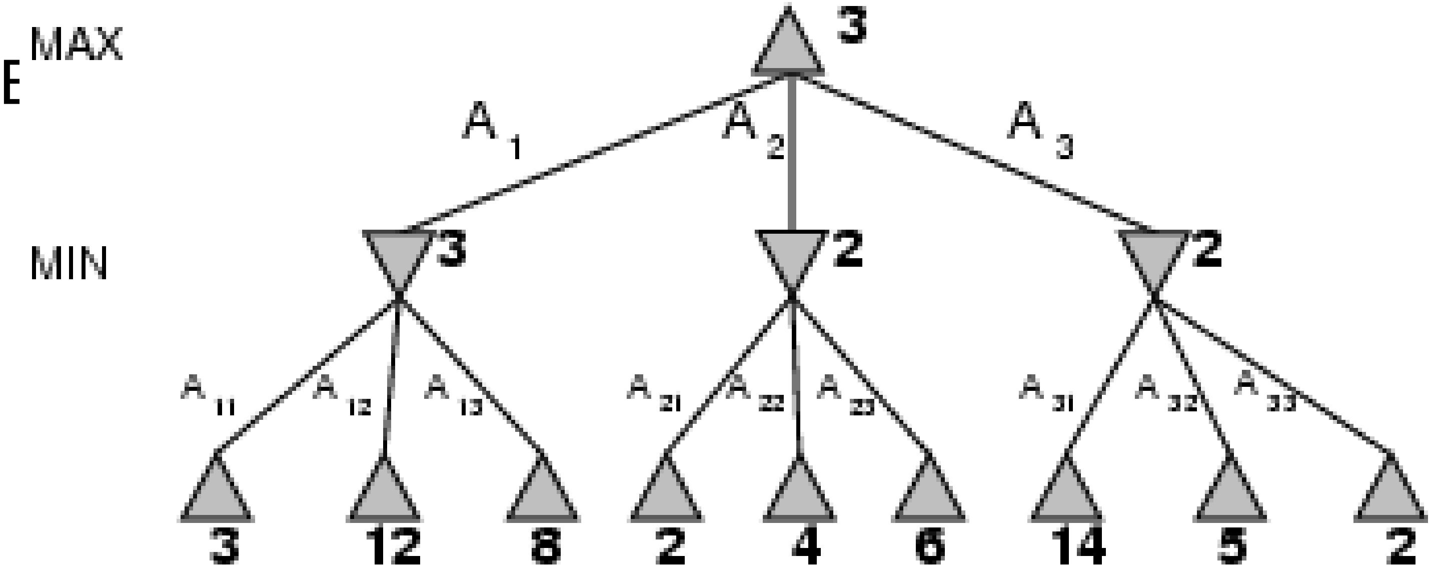
Minimax

Game tree (2-player, deterministic, turns)



Minimax

- Perfect play for deterministic games
-
- Idea: choose move to position with highest minimax value = best achievable payoff against best play
-
- E
-



Minimax algorithm

```
function MINIMAX-DECISION(state) returns an action
    v ← MAX-VALUE(state)
    return the action in SUCCESSORS(state) with value v

function MAX-VALUE(state) returns a utility value
    if TERMINAL-TEST(state) then return UTILITY(state)
    v ← −∞
    for a, s in SUCCESSORS(state) do
        v ← MAX(v, MIN-VALUE(s))
    return v

function MIN-VALUE(state) returns a utility value
    if TERMINAL-TEST(state) then return UTILITY(state)
    v ← ∞
    for a, s in SUCCESSORS(state) do
        v ← MIN(v, MAX-VALUE(s))
    return v
```

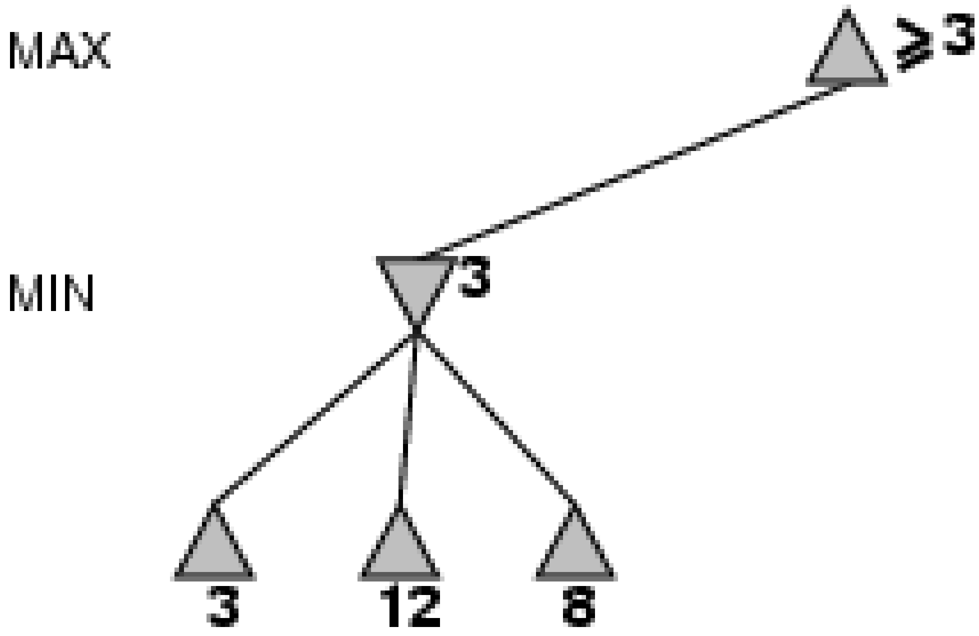
Properties of minimax

- Complete? Yes (if tree is finite)
-
- Optimal? Yes (against an optimal opponent)
-
- Time complexity? $O(b^m)$
-
- Space complexity? $O(bm)$ (depth-first exploration)
-

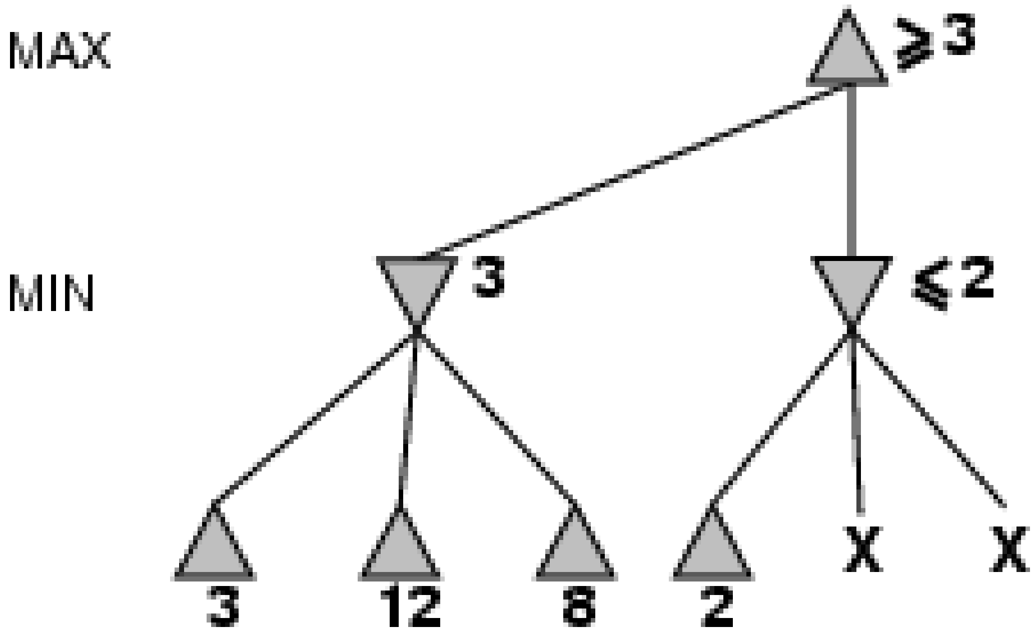
- For chess, $b \approx 35$, $m \approx 100$ for "reasonable" games → exact solution completely infeasible
-

α-β algorithm

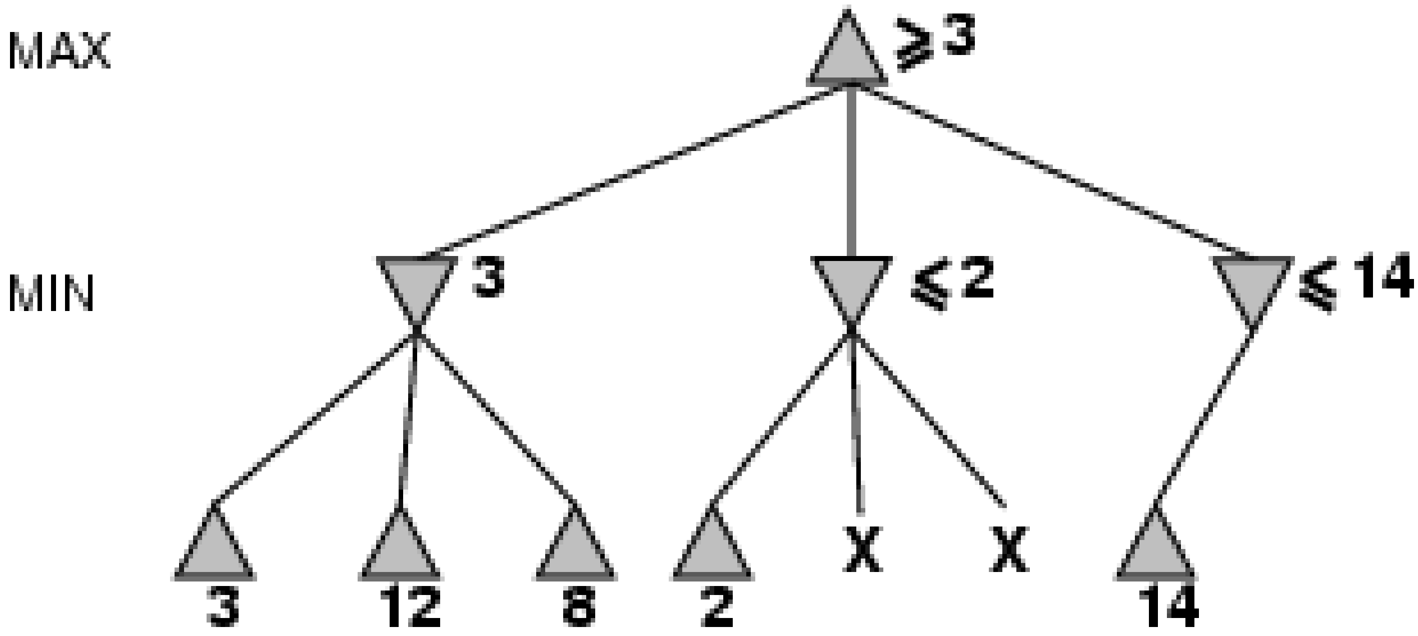
α-β pruning example



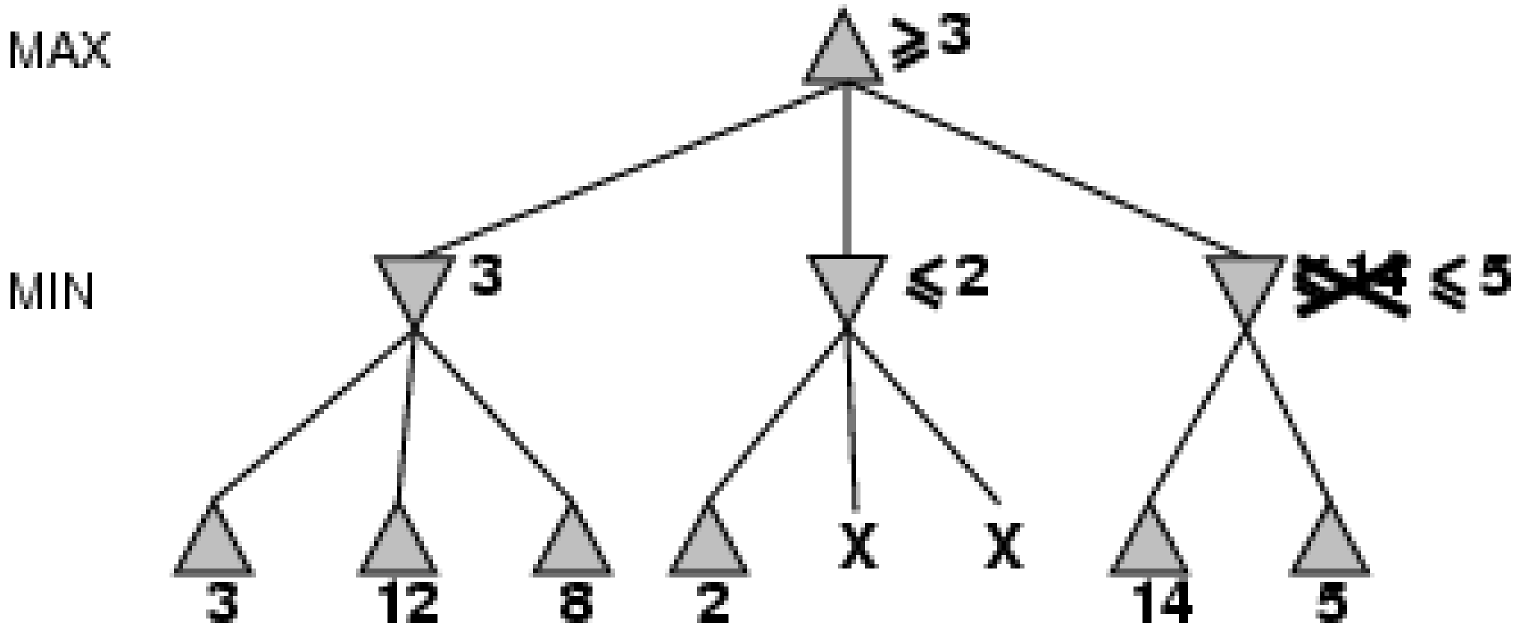
α-β pruning example



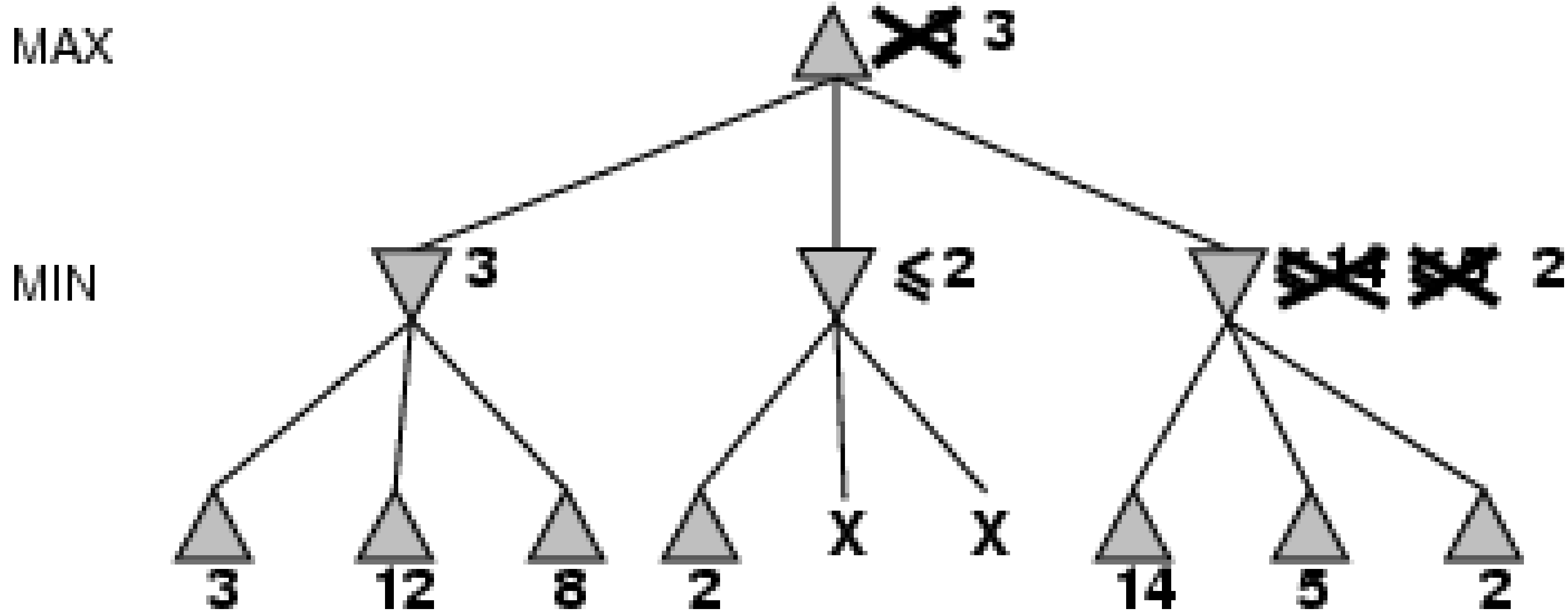
α-β pruning example



α-β pruning example



α-β pruning example



Properties of α-β

- Pruning does not affect final result
-
- Good move ordering improves effectiveness of pruning
-
- With "perfect ordering," time complexity = $O(b^{m/2})$
  → doubles depth of search
- A simple example of the value of reasoning about which computations are relevant (a form of metareasoning)
-

Why is it called α-β?

- α is the value of the best (i.e., highest-value) choice found so far at any choice point along the path for *max*
-
- If *v* is worse than α, *max* will avoid it
-

  → prune that branch