Lecture 6

Multi-agent Decision Making
(Original: adversarial search in game playing)

Ning Xiong

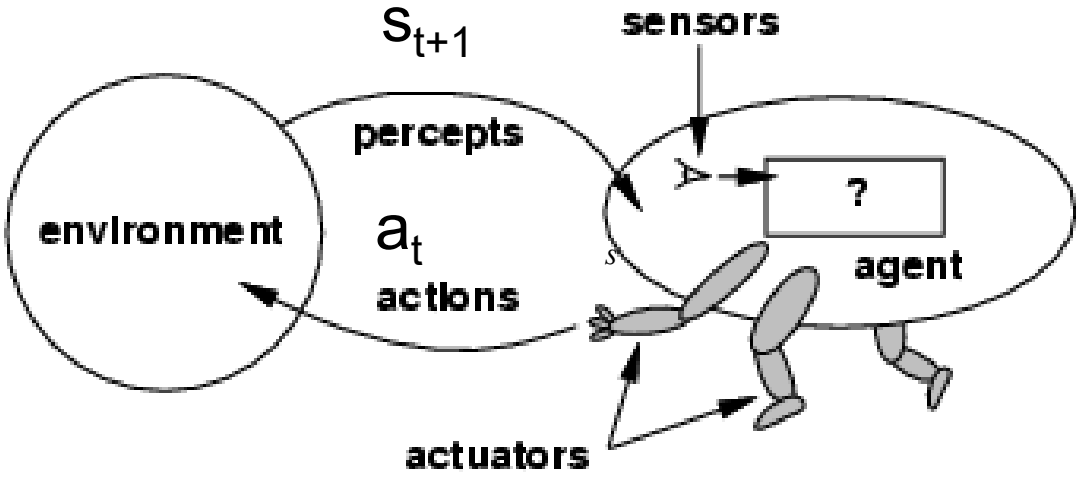
Mälardalen University

Content

The original aim is to teach the **adversarial search** in game playing. But this lecture will attempt to present knowledge in a broader perspective.

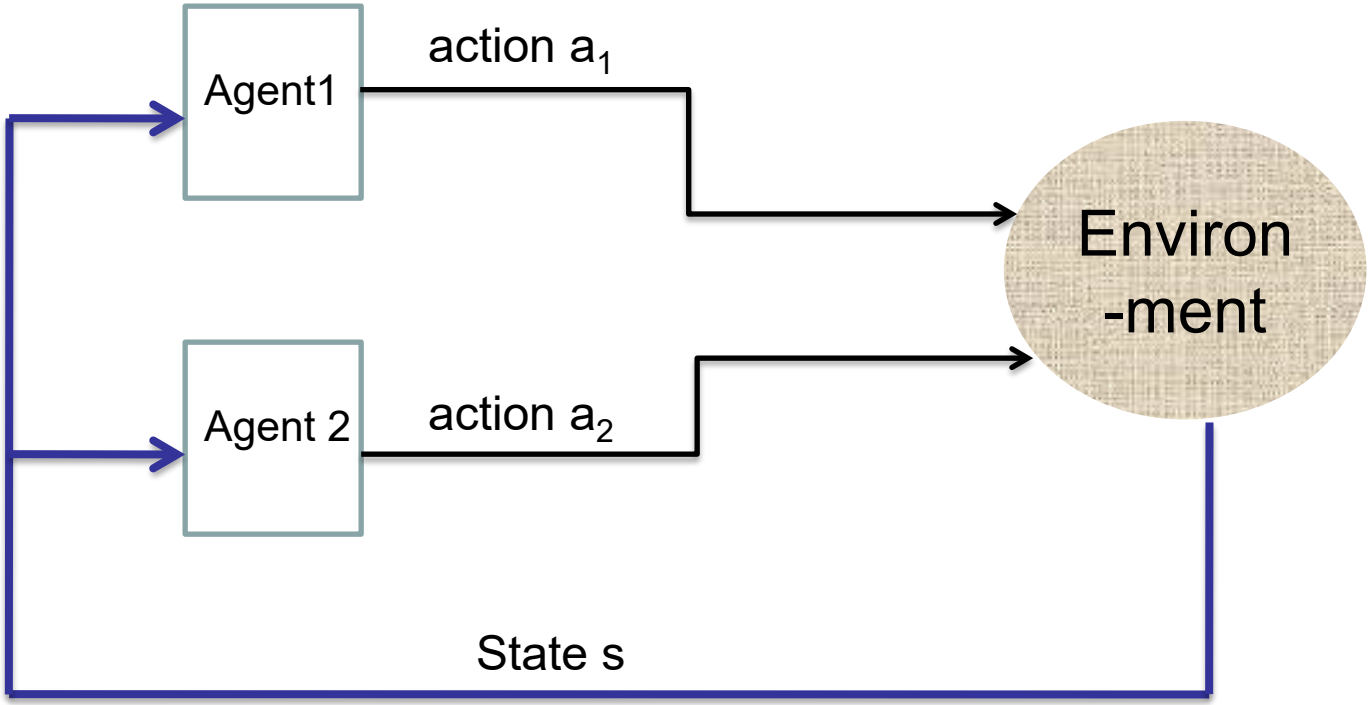
- Characteristic of multi-agent decision making
- Game theory at a shallow level
- Minimax search
- α-β pruning
- Heuristic in minimax search

Single Agent



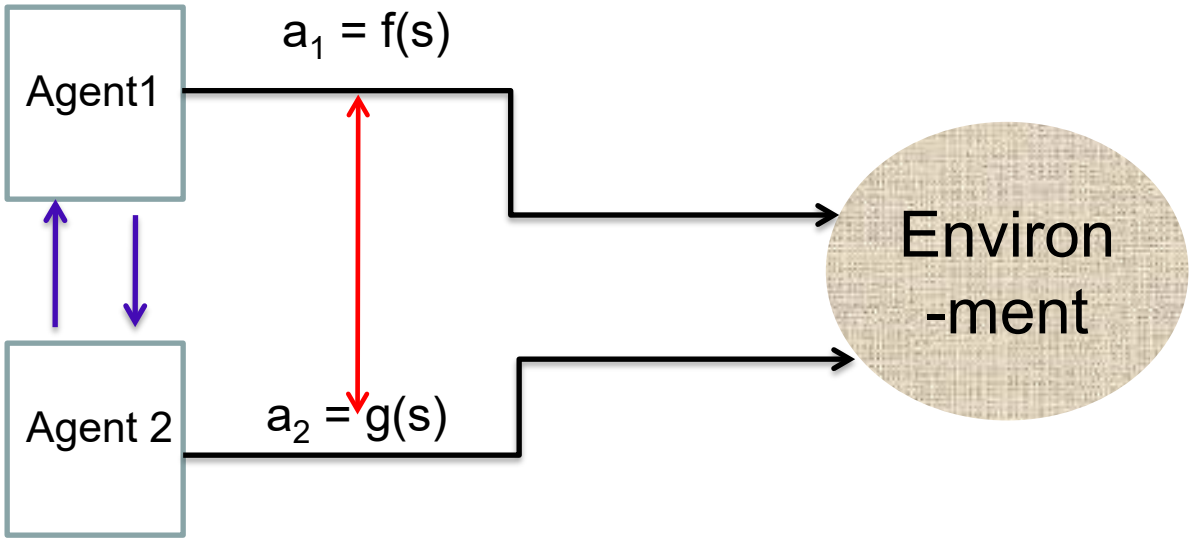
- The agent can take actions to change the environment to meet its goal
- The next state s_{t+1} is only affected by the action a_t

Multi-Agent System



- The environment is affected by multiple agents.
- Agents are self-interested (they don't have the common goal)
- The agents take decisions and actions to maximize their own benefit.

Interactive Decision Making



There is interplay between two agents in decision making. Agent 1 will consider what strategy other agent will use when deciding his own strategy f. So will agent 2.

About Game Theory

- The basic concepts arouse from studying the games such as chess and checkers
- The results and techniques can be applied to all interplay in multi-agent systems where agents don’t have a common goal.
- Game theory aims to make the most rational choice under the expectation of other agents to be also rational.

Strategic Game

A strategic game is a model of interactive decision making in which each agent makes a choice once and for all, and these choices are made at the same time.

Example: prisoner's dilemma

Nash Equilibrium in Strategic Games

Game theory suggests all agents take their actions in terms of **Nash equilibrium**

- **A Nash equilibrium** of a strategic game is a profile (a_i^*) of actions such that **for every agent *i*** we have

$$u_i(a_{-i}^*, a_i^*) \geq u_i(a_{-i}^*, a_i)$$

where u_i denotes the payoff of agent i

- Nash equilibrium denotes a steady profile of actions that no one can be better of by deviation if others keep the choices in the equilibrium

Explanation of Equilibrium

Equilibrium: (confess, confess)

Person 1 \ Person 2	Don't confess	Confess
Don't confess	-1, -1	-4, 0
Confess	0, -4	-3, -3

Deviation by person 1:

U1(confess, confess)=-3, U1(not confess, confess)=-4

U1(confess, confess) > U1(not confess, confess)

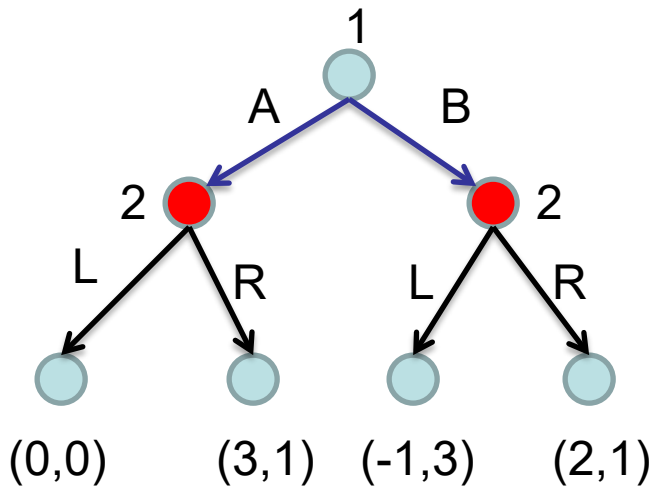
Deviation by person 2:

U2(confess, confess)=-3, U2(confess, not confess)=-4

U2(confess, confess) > U2(confess, not confess)

Extensive Game

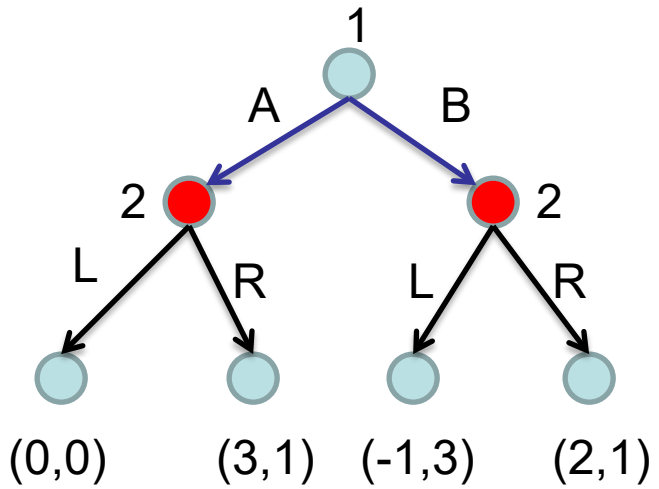
• An extensive game is a description of the sequential structure of decision making, in which agents make moves one by one and no two agents make the decisions at the same time.



• An extensive game can be represented by a tree or graph . A link between two nodes represent an action.

Strategy in an Extensive Game

• A strategy of an agent in an extensive game is the specification of the actions at all those nodes under which it is the turn of the agent to play



- The strategies for agent 1: A, B
- The strategies for agent 2: LL, LR, RL, RR

Equilibrium in Extensive Games

Game theory suggests all agents take their strategies following **Nash equilibrium**

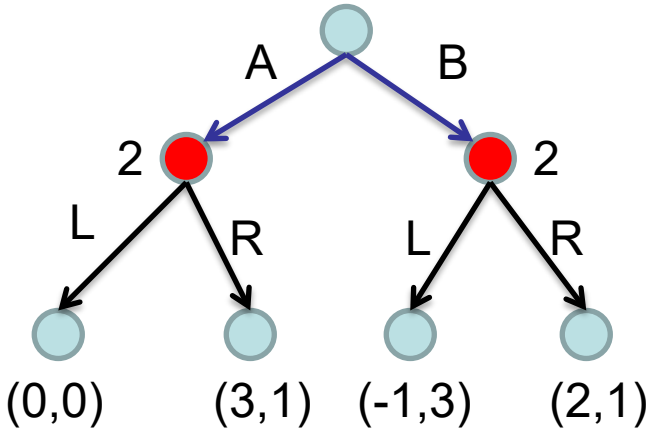
- **A Nash equilibrium** of an extensive game is a profile (P_i^*) of strategies (policies) such that **for every agent i** we have

$$u_i(P_{-i}^*, P_i^*) \geq u_i(P_{-i}^*, P_i)$$

where u_i denotes the payoff of agent i

- Nash equilibrium denotes a steady profile of strategies that no one can be better of by changing its strategy if others honor the strategies specified in the equilibrium.

Explanation for Equilibrium



Equilibrium: (A, RR)

Deviation by agent 1:
U1(A, RR)=3, U1(B, RR)=2,
U1(B, RR)<U1(A, RR)

Deviation by agent 2:
U2(A, RR)=1,
U2(A, RL)=1, U2(A,RR)=U2(A,RL)
U2(A, LR)=0, U2(A,RR)>U2(A,LR)
U2(A, LL)=0, U2(A,RR)>U2(A,LL)

Applying Game Theory to Constant-Sum Extensive Games

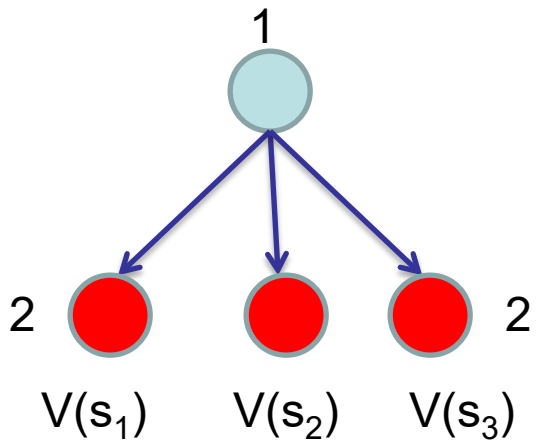
- In a constant-sum game the sum of final payoffs for agents is always constant, e.g, checker, chess, resource allocation …..

- We study what are the rational strategies in terms of Nash equilibrium.

- We define the value of a state under the condition of Nash equilibrium. That is, **the value of a state is defined as the final payoff that one agent will achieve from that state if both agents follow the strategies of the Nash equilibrium.**

Rational Strategy for Agent 1

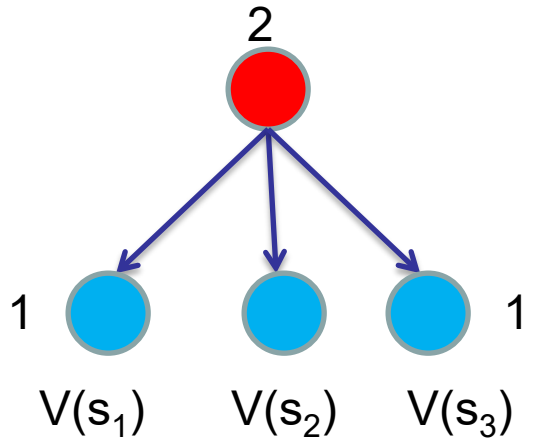
- Suppose the value function is defined for agent 1



- According to Nash equilibrium, at any node for agent 1 to play, it should take the act to move to the successor state which has the highest value.
  - The value max[$V(s_i)$] will be its final payoff if both agents follow the policies of Nash equilibrium. In case agent 1 chooses another node, its final payoff will be less
- Agent 1 is called MAX agent. It always goes to the successor with maximum value

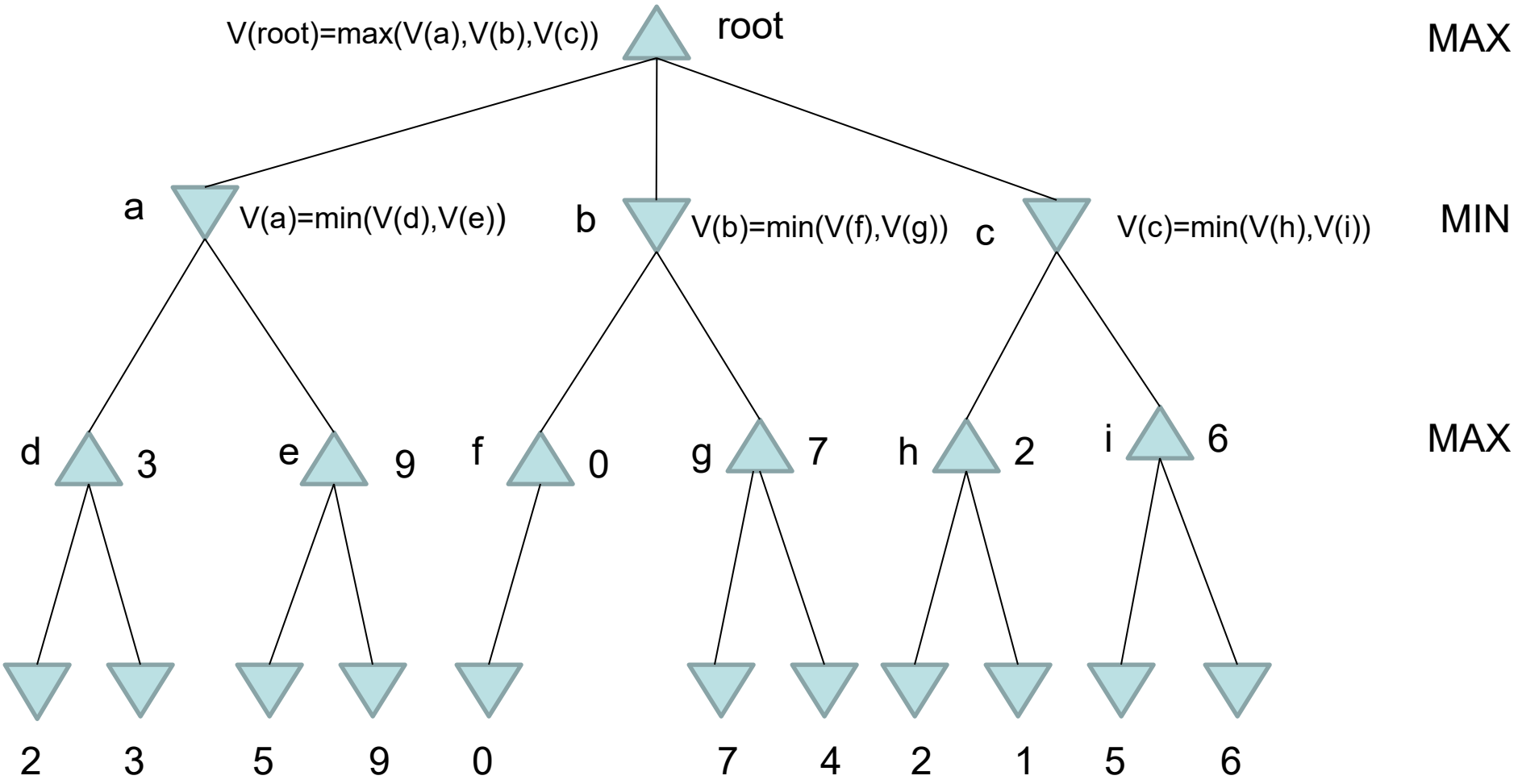
Rational Strategy for Agent 2

- The goal of agent 2 is to minimize the payoff of its opponent



- According to Nash equilibrium, at any node for agent 2 to play, it should take the act to move to the successor state which has the lowest value.
  - The value $\min[V(s_i)]$ will be the final payoff of its opponent if both agents follow the policies of Nash equilibrium. In case agent 2 chooses another node, the final payoff of its opponent will be more
- Agent 2 is called MIN agent. It always goes to the successor with minimum value

How to Get the Values of States



MINIMAX-Value

The values of states generated when agents follow Nash equilibrium strategies are called **MINIMAX-values**.

$$MINIMAX-VALUE(n) = \begin{cases} Payoff(n) & if \ \ n \ \ is \ \ a \ \ terminal \ \ state \\ \max_{s \in Successors(n)} MINIMAX-VALUE(s) & if \ \ n \ \ is \ \ a \ \ MAX \ \ node \\ \min_{s \in Successors(n)} MINIMAX-VALUE(s) & if \ \ n \ \ is \ \ a \ \ MIN \ \ node \end{cases}$$

This recursive definition implies that the derivation of MINIMAX values should start from bottom of the tree, with payoffs of terminal states being backed up level by level to other nodes.

Minimax Principle

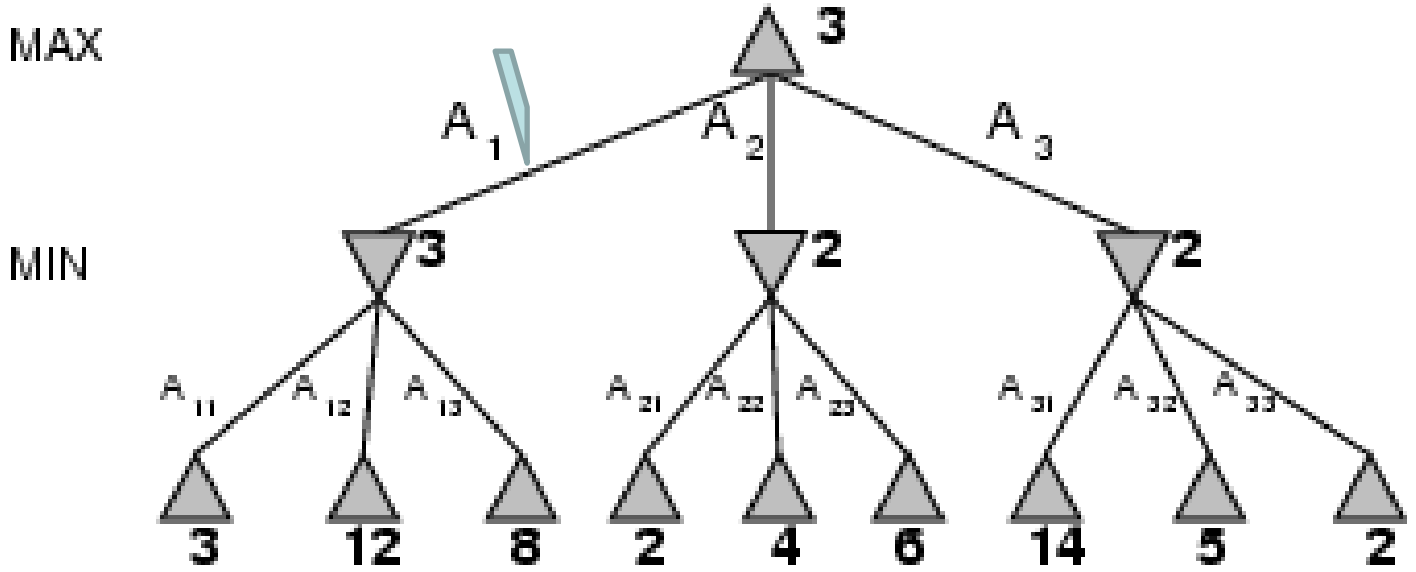
The strategies derived from Nash equilibrium are called minimax principle. The minimax principle claims that

1. The optimal strategy for MAX agent is to choose the move leading to the successor with the maximum of the minimax values

2. The optimal strategy for MIN agent is to choose the move leading to the successor with the minimum of the minimax values

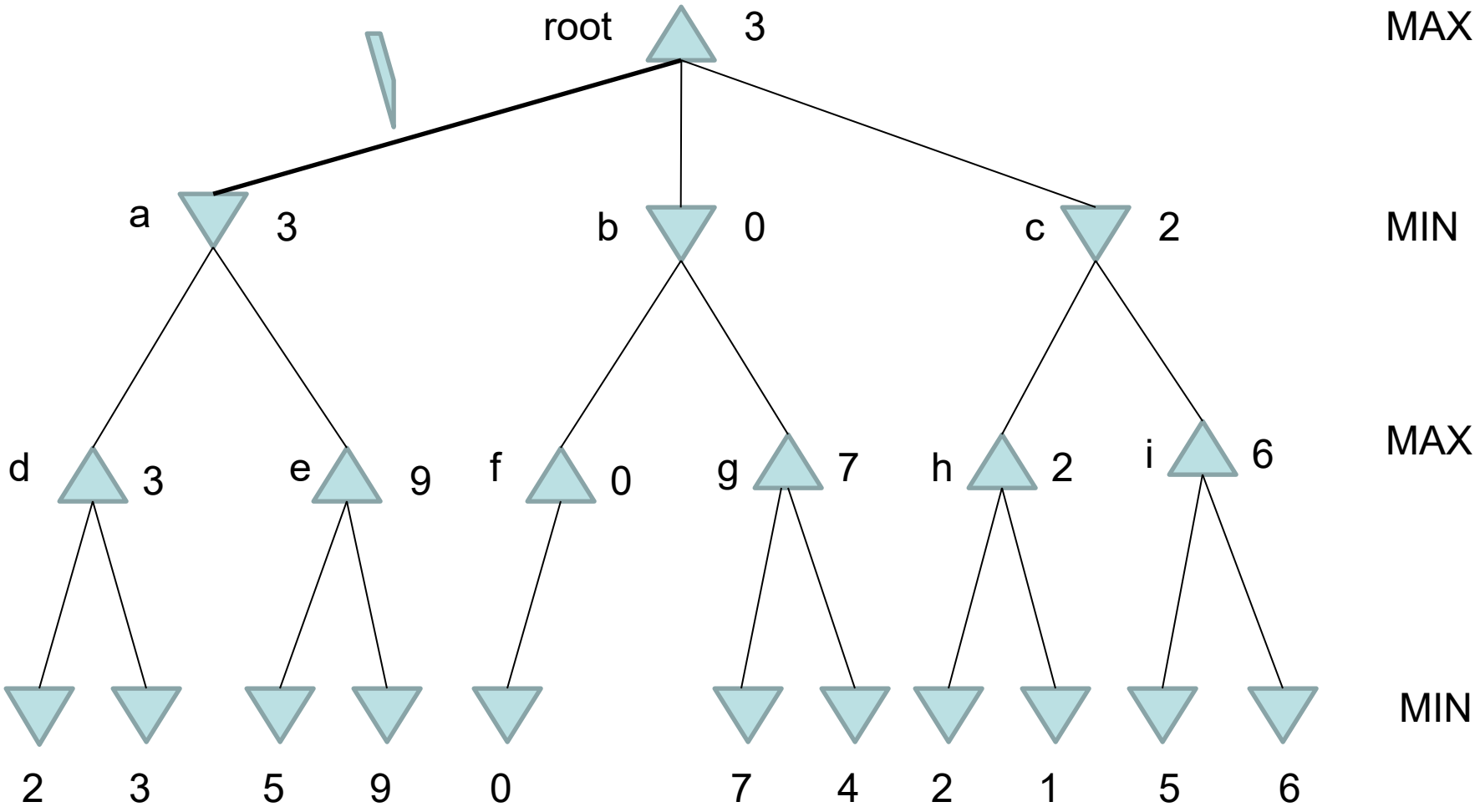
Example of Minimax

- Strategy: choose move to position with highest minimax value



This example from Russel's book

Minimax on Previous Example



Properties of minimax

- Complete? Yes (if state space is finite)
- Optimal? Yes (against a rational opponent following the the same strategy)
- Exhaustive search: construct the complete tree
- Time complexity? $O(b^m)$, where m is the step number and b is branching factor (number of choices)
- For chess, b ≈ 35, m ≈100 → exact solution completely infeasible

Reducing Complexity for Minimax Search

1. Alpha-Beta pruning: ignoring some nodes while not affecting Minimax decision.

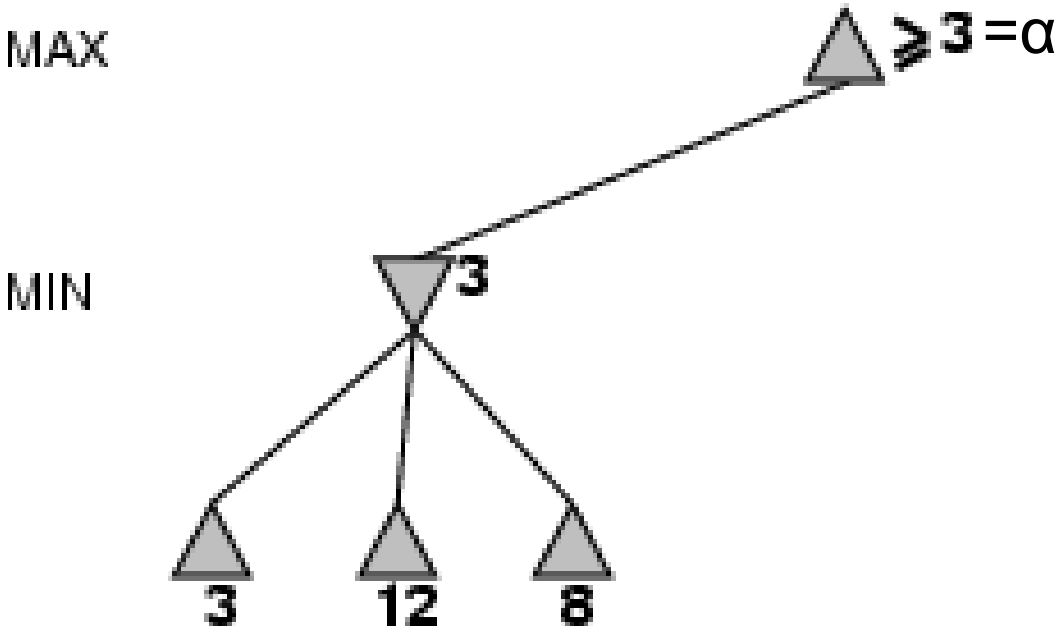
2. Heuristic approach: terminate before the end of the game, heuristic evaluation of leave nodes

Alpha-Beta Pruning

In practice quite a few nodes in the tree will never be reached if agents follow the Minimax strategy, hence they can be ignored during the search without changing the final decisions.

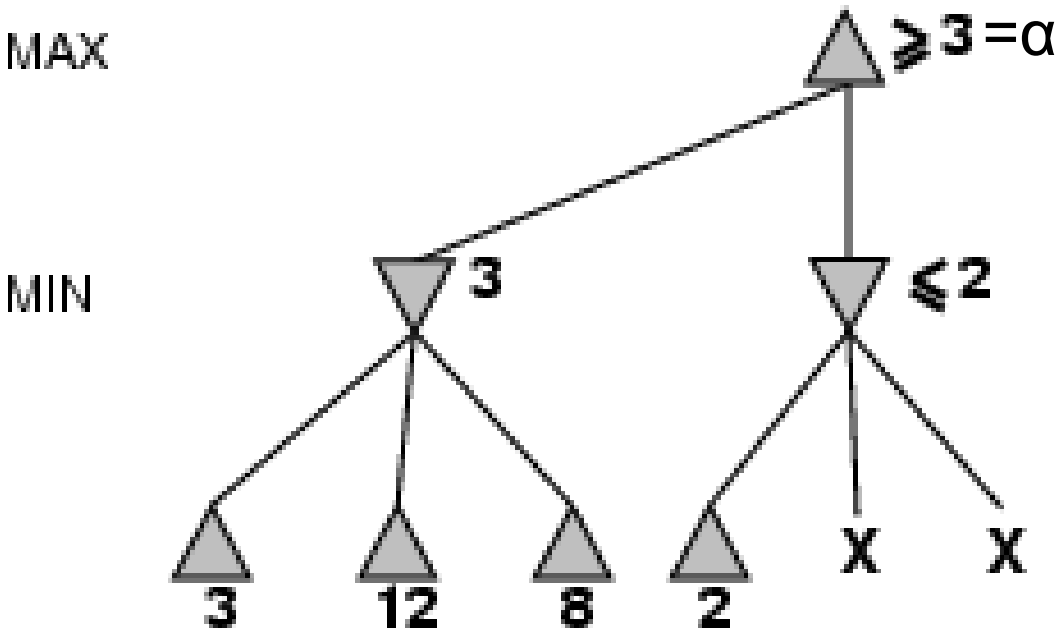
This technique to rule out such parts of the tree is called alpha-beta pruning.

α pruning Example

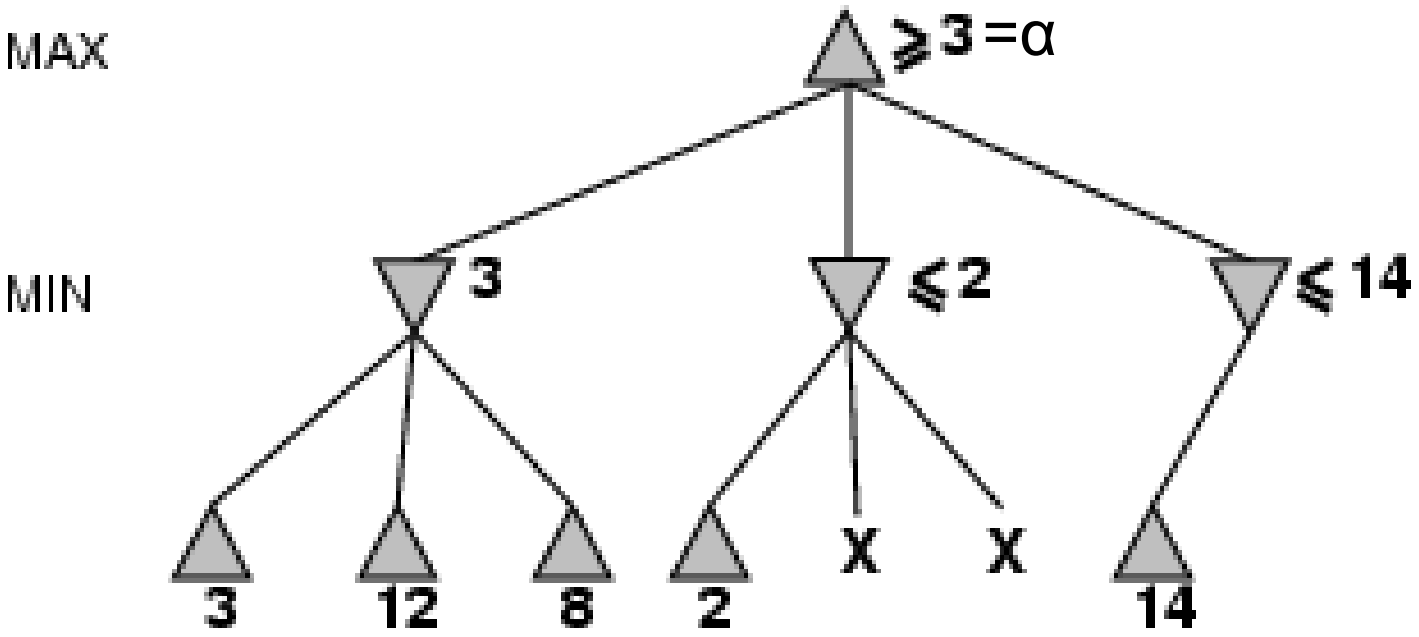


This example from Russel's book

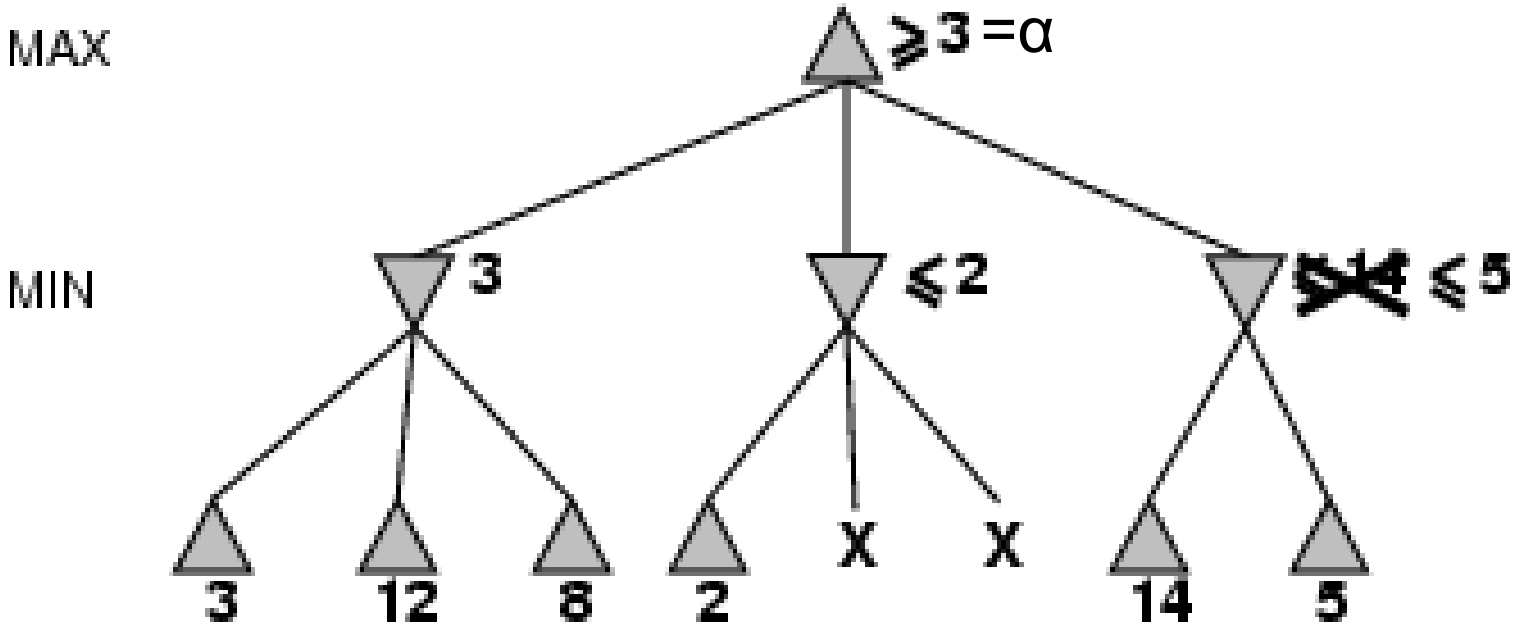
α -pruning Example



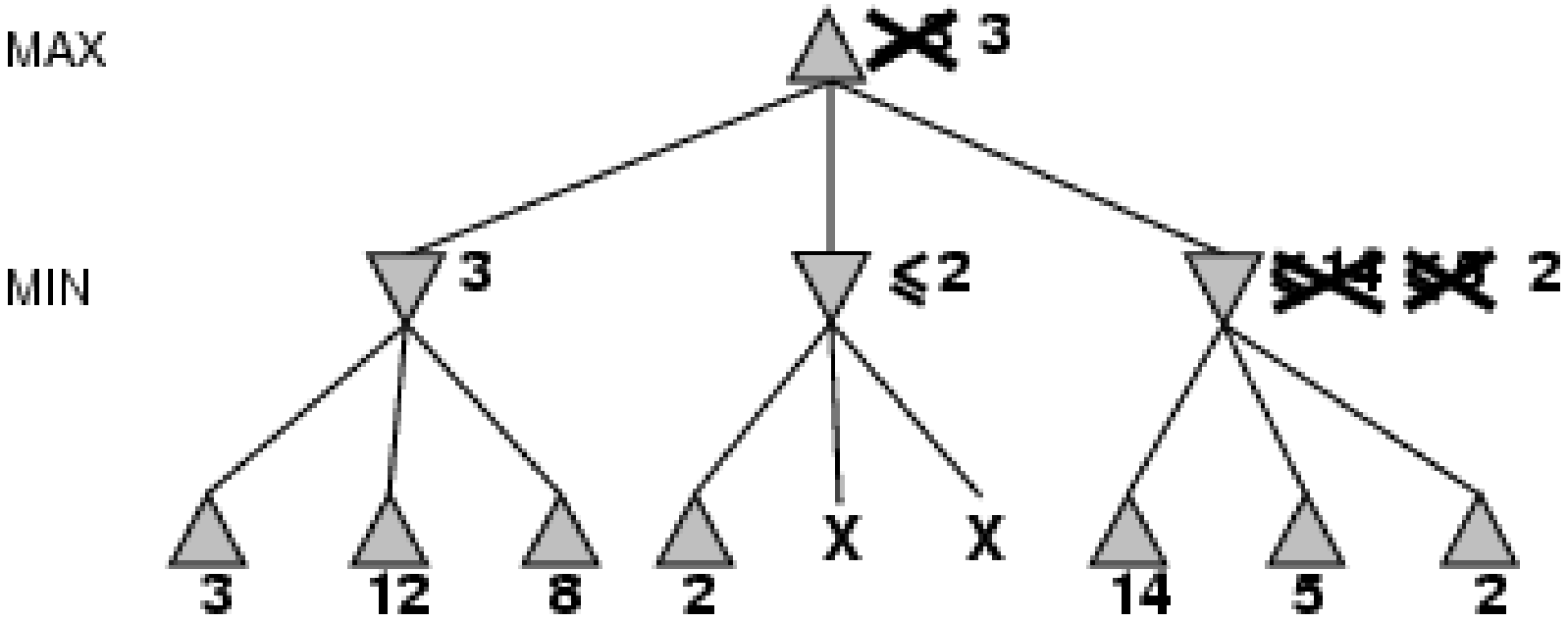
α pruning Example



α pruning Example



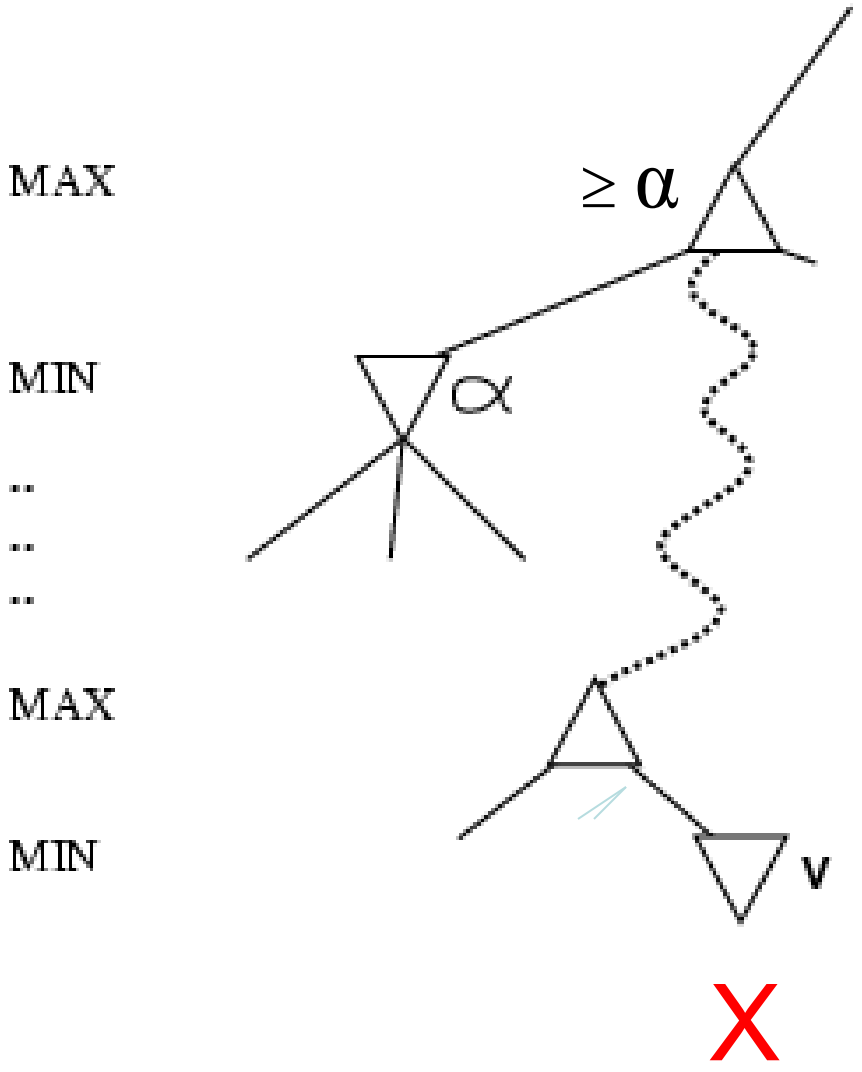
α pruning Example



Alpha Pruning (MAX agent)

- α is the value of the best choice (i.e. highest value) found so far for an MAX point along the path
- If *v* for a later MIN node is worse than α, *max* will avoid it
-

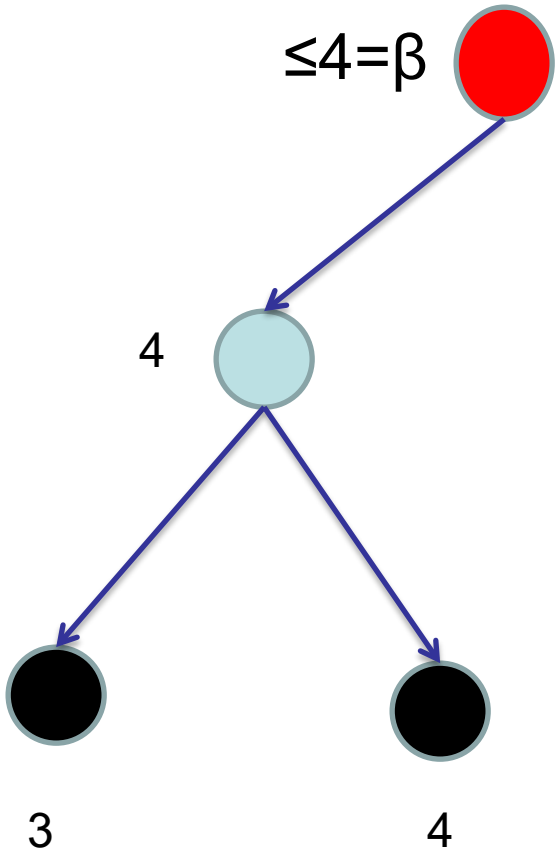
→prune the branches of the MIN node



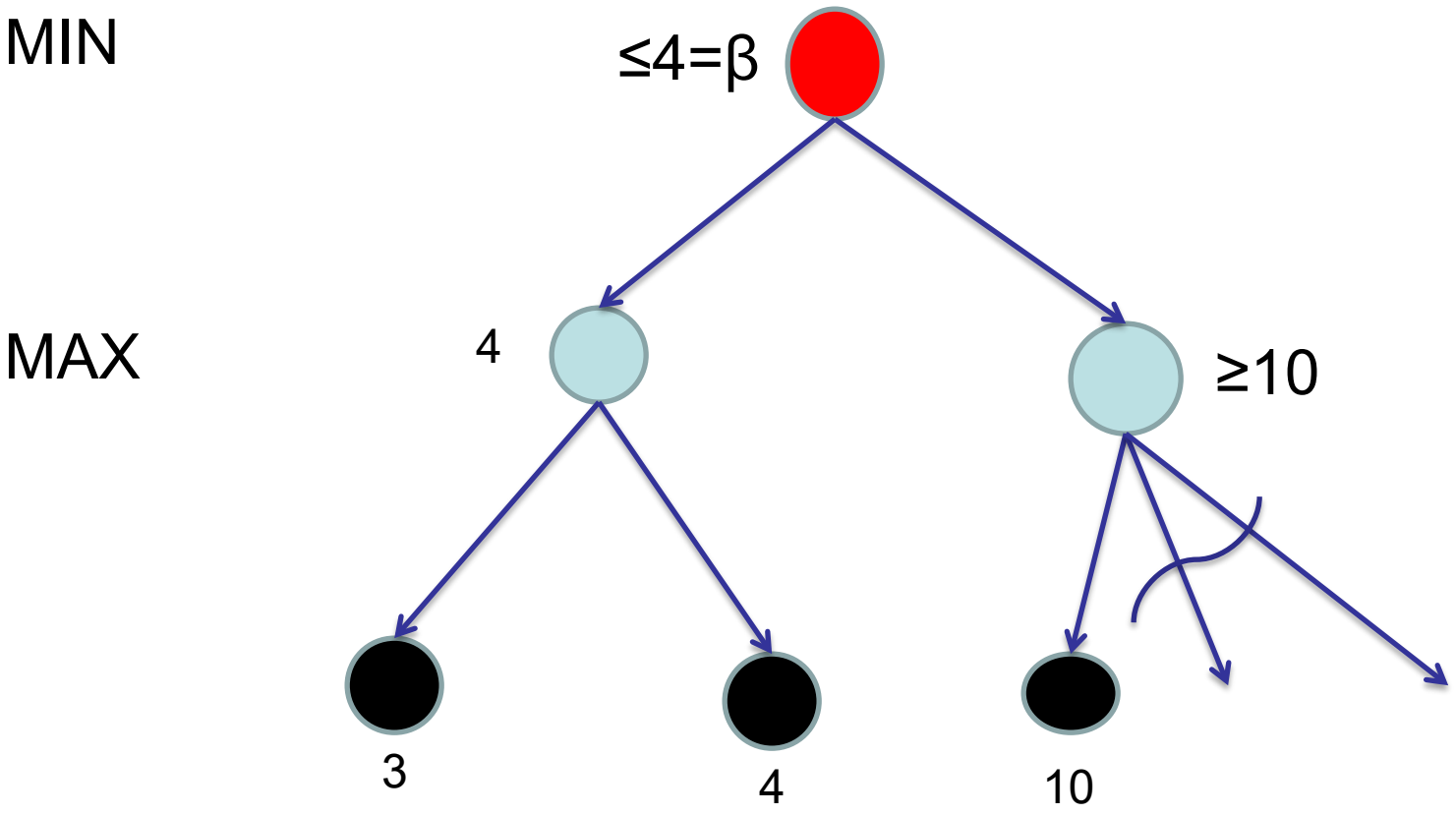
Beta Pruning

MIN

MAX



Beta Pruning

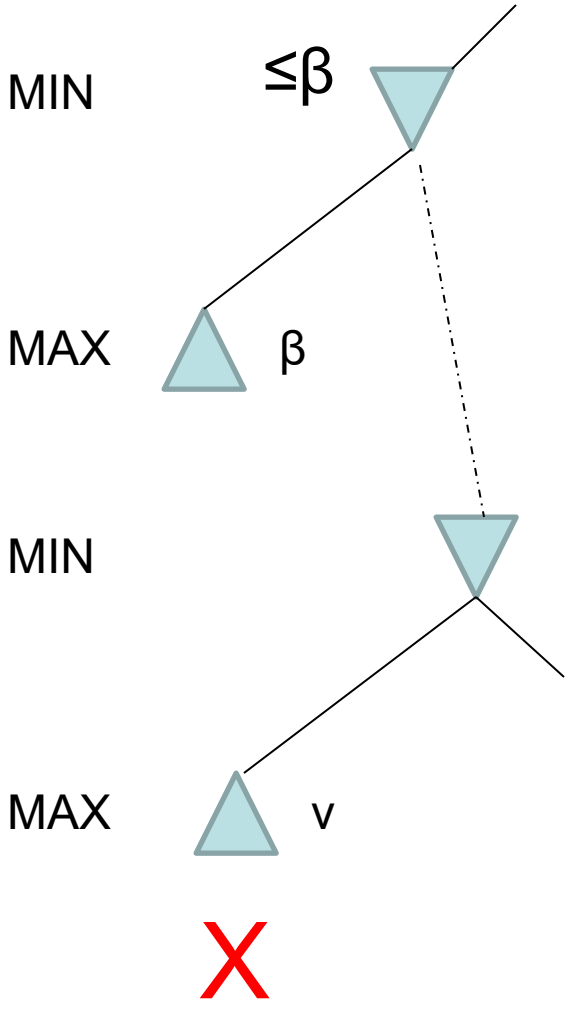


Beta Pruning (MIN agent)

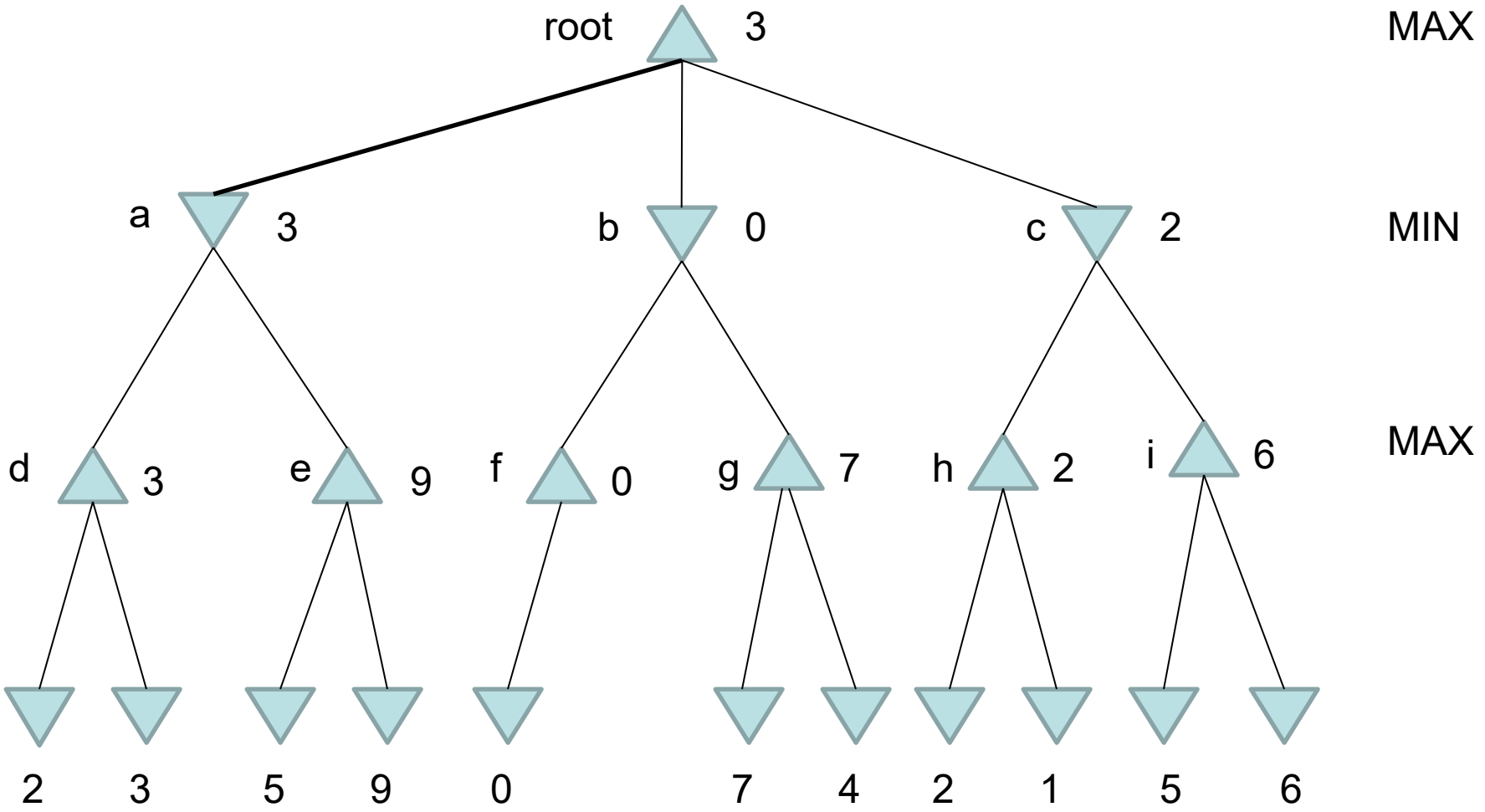
- β is the value of the best choice (i.e., lowest-value) found so far for a MIN point along the path

- If *v* for a later MAX node is bigger than β, MIN agent will avoid it

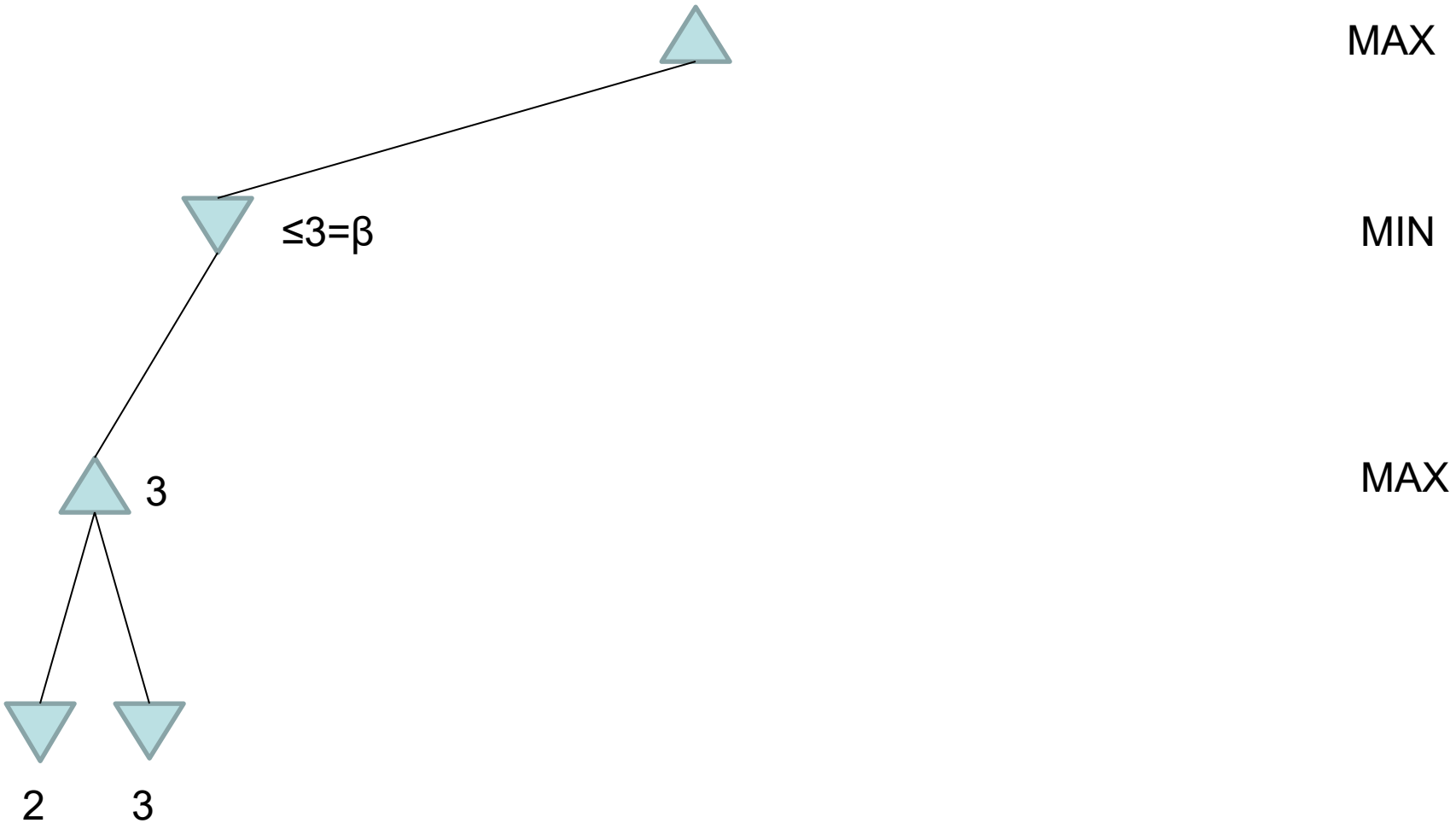
- →prune the branches of the MAX node



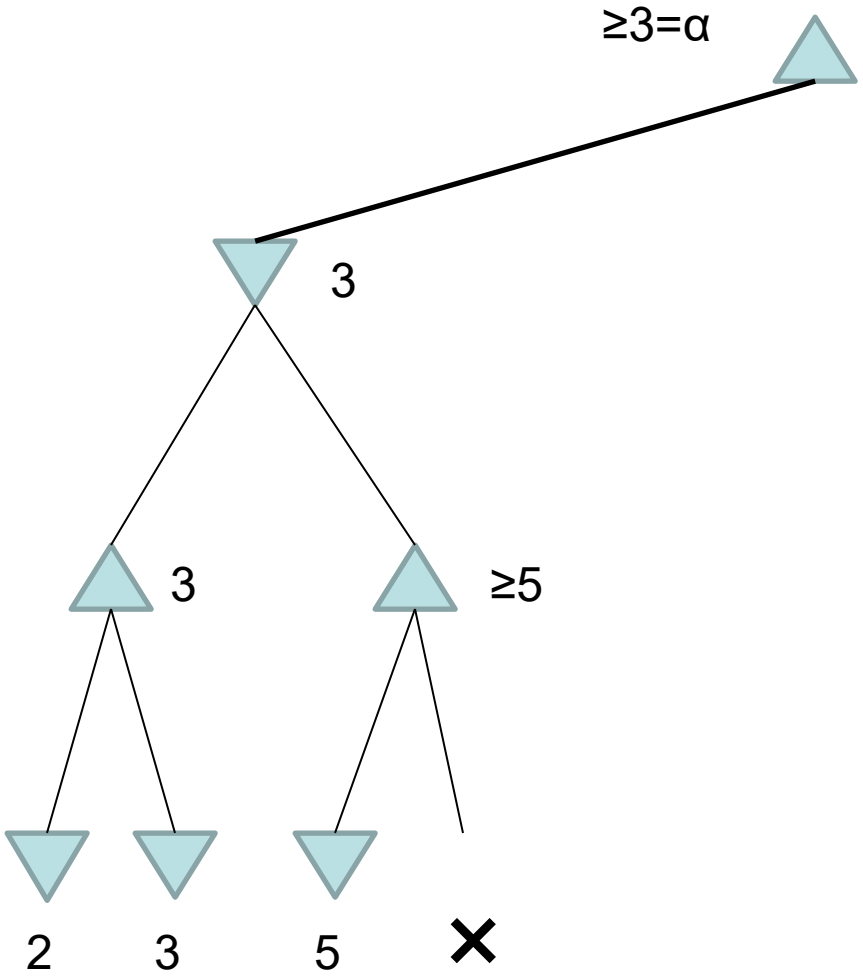
α-β Pruning Example



α-β Pruning Example



α-β Pruning Example

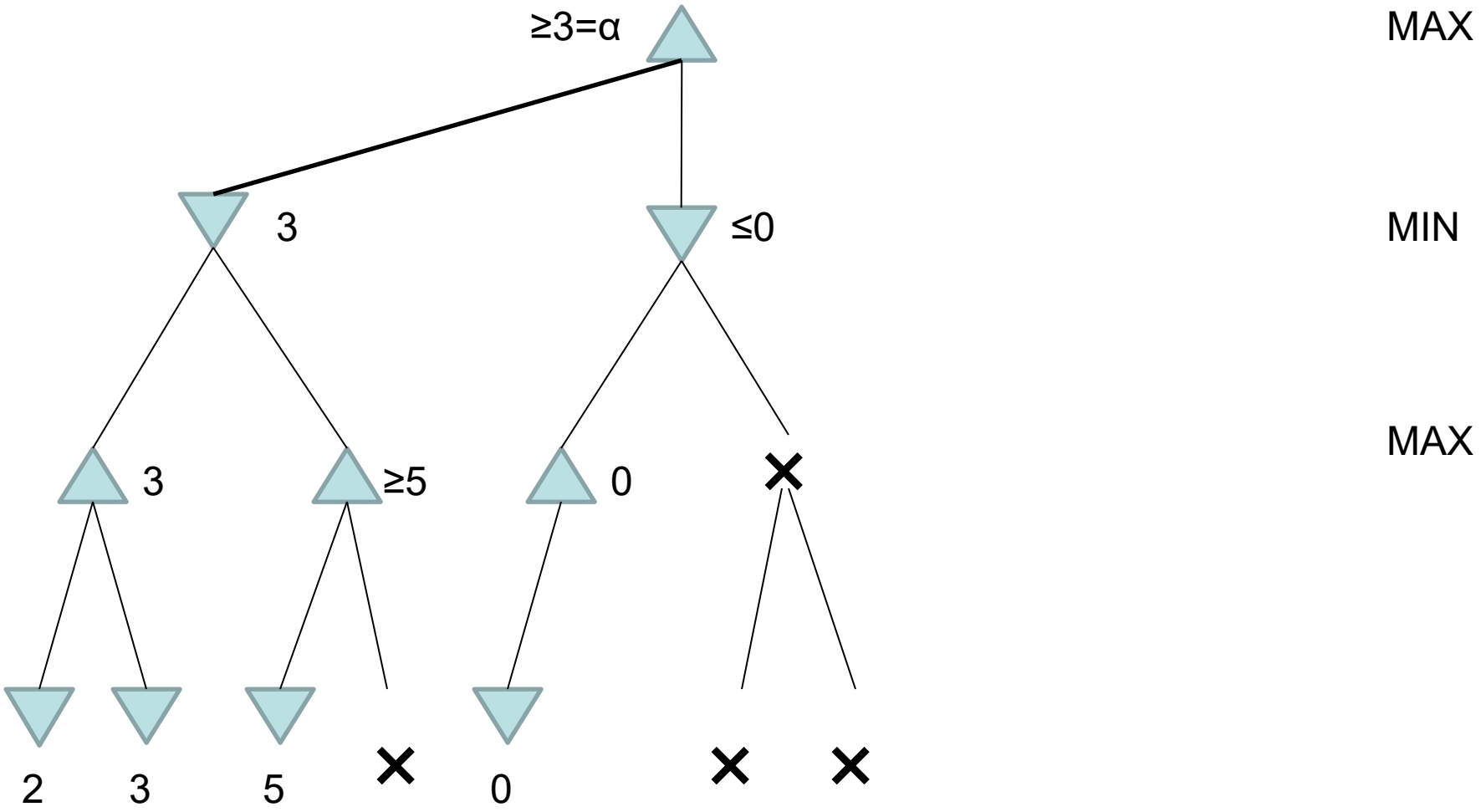


MAX

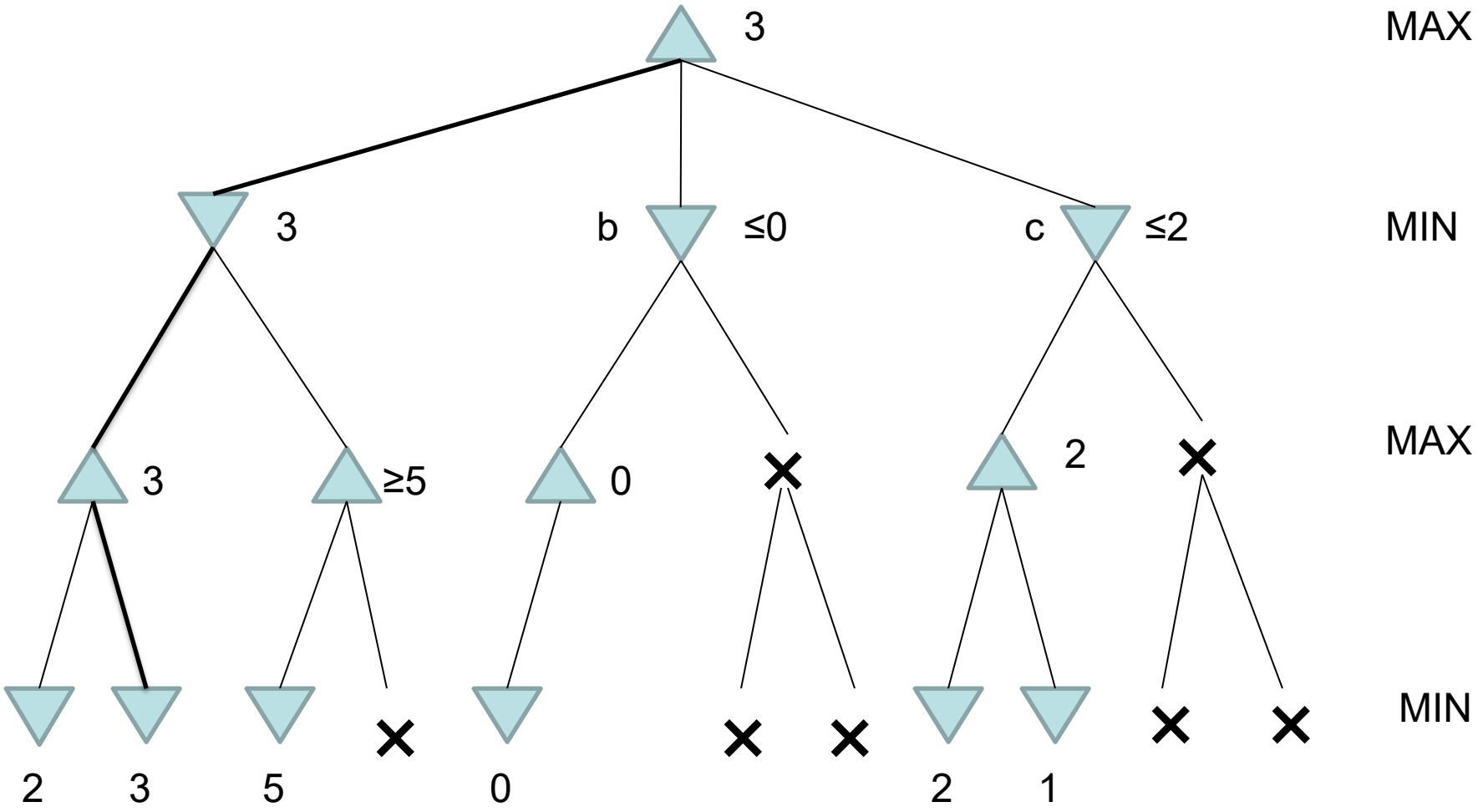
MIN

MAX

α-β Pruning Example



α-β Pruning Example



Reducing the Depth of Search

In practice it is not feasible to search many steps till end of the game (terminal states) due to time limits.

Approximate solution

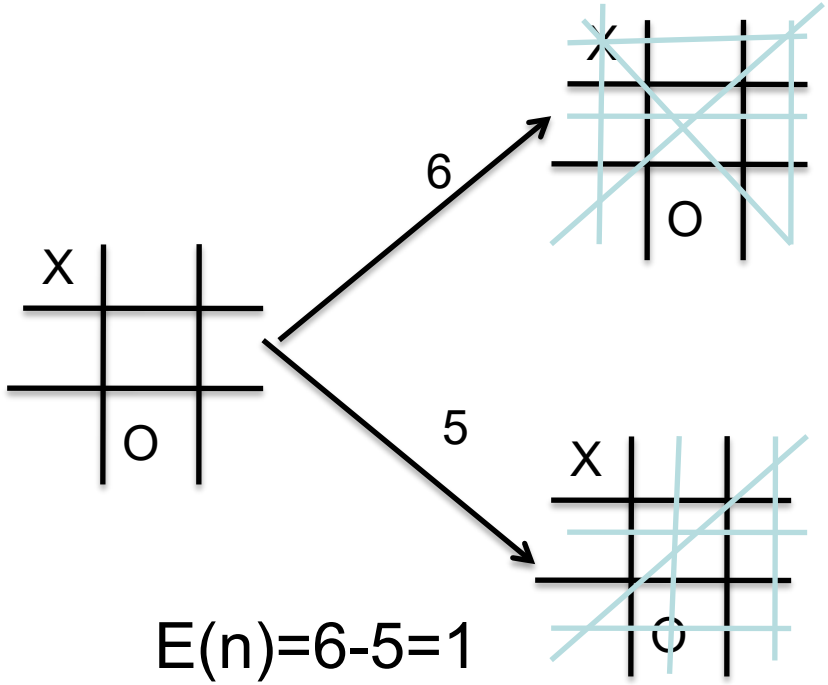
1. Terminate early before the game is over
2. Heuristic evaluation function to estimate the values of terminal leaves

Minimax strategy as well as alpha-beta pruning can be applied to game tree with terminal leaves assigned with heuristic values.

Heuristic Evaluation Function

A heuristic evaluation function returns the estimated value from a given position. It turns non-terminal states of the game into terminal leaves of the tree.

Evaluation Function for Tic-Tac-Toe



Heuristic: E(n)=M(n)-O(n)

where
M(n) is the total of my possible winning lines;
O(n) is the total of opponent's Winning lines

Cutting off Search

• The most straightforward method is to set a fixed depth limit for search, i.e. n-ply look-ahead, where **n** is the number of levels to explore. *n* is decided by the amount of time the agent is given for a single decision as well as computer speed.

Two-Ply Ahead Tree