Section 5: Game Theory and AI Game Playing

Lecturer: Ariel Procaccia *Author: Eric Helmold and Sanjana Singh*

1 Game Theory

1.1 Nash Equilibrium

Definition 1 (Nash Equilibrium) *A vector of strategies* $\boldsymbol{s} = (s_1, \ldots, s_n) \in S^n$ *such that for all* $i \in N$ *and for all* $s'_i \in S$ *the following is true:* $u_i(\boldsymbol{s}) \geq u_i(s_1, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_n)$

In other words, a Nash equilibrium exists when no player in the game is incentivized by their utility function u_i to unilaterally deviate from the strategy they are currently playing s_i to another s'_i, given the strategies of all other players denoted s_j or all $j \neq i$. This means that each of the N players can say that from their perspective, their strategy is optimal (utility maximizing) given the strategy being played by all other players.

A Nash equilibrium is not necessarily an efficient outcome meaning that it may not maximize the total utility of players in the game (e.g. prisoner's dilemma), but it is set of strategies where all players have no regrets about their strategy choices i.e. they are playing the best strategy given the strategies of all the other players.

1.2 Strategy Dominance

1. **Strictly Dominant** - A strategy is said to be strictly dominant if it is the player's best choice regardless of what other players choose i.e. it has higher payoffs in all settings than any other possible strategy the player might adopt
2. **Weakly Dominant** - A strategy s is said to weakly dominant another strategy s' if s has payoffs that are at least as high as s' regardless of what other players choose, and in at least one case higher

Theorem 2 (Nash's Theorem) *In any finite game there exists at least one (possibly mixed strategy) Nash equilibrium. A finite game is a game with a finite number of players each having a finite number of pure strategies.*

1.3 Pure and Mixed Strategies

1. **Pure Strategy**: A strategy profile where a player always plays a specific action e.g. always playing paper in rock-paper-scissors
2. **Mixed Strategy**: A strategy profile where a player plays actions according to a randomized distribution over the possible actions available e.g. play R, P and S with equal (1/3) probability of rock-paper-scissors

It can be shown that a best response to any strategy is a pure strategy. Therefore, if a mixed strategy Nash equilibrium exists where both players play mixed strategies, it must be a strategy that player 1 plays, which makes player 2 indifferent between their various pure strategy options. In such a case, if player 1 plays a mixed strategy, then player 2 might also choose a mixed strategy since it can have the same expected utility as a best response from selecting a pure strategy.

For an intuitive understanding of this condition and approach, consider the following. Given player 1's strategy, let the payoffs of player 2's pure strategy options be $x_1, ..., x_n$. Player 2's best response therefore is to select the argmax of this payoffs vector as their best response. Selecting as a strategy any other option cannot result in a NE since player 2 would have an incentive to change their strategy and do better by selecting the argmax strategy. Therefore, the only way for a NE to occur and for player 2 to play a mixed strategy is if there was more than one pure strategy option that had equal expected value, in which case, selecting some weighted combination of them (i.e a mixed strategy) will result in the same expected utility as selecting anyone of them which will not create an incentive to deviate. The same logic can be applied the other way around for player 1 to want to play a mixed strategy. Satisfying both conditions simultaneously i.e. that the strategy played by the one player makes the expected value of the other's pure strategy options equal, will allow us to recover a mixed strategy NE.

Problem 1 *Consider the following payoff matrix. Identify any pure strategy Nash equilibria if any exist. Identify the mixed strategies that player 1 and player 2 could play that would lead to a mixed strategy Nash equilibrium.*

		Player 2			
		A		B	
Player 1	A	2	0	0	2
	B	-1	2	1	0

1.4 Supplemental Examples and Problems

Example 1 *Alice and Bob are both CS concentrators and also both always eat dinner at the dining hall at 6pm. The CS department is hosting a social event this Friday at 6pm. They are good friends and therefore enjoy each other's company. Going to the CS social event together will be more fun than going to the dining hall together, but either is preferable to going to separate locations. What should Alice and Bob do on Friday night? Identify the Nash equilibrium(s) in the following payoff matrix that describes this setting:*

		Alice			
		CS Event		Dining Hall	
Bob	CS Event	10	10	3	3
	Dining Hall	3	3	5	5

Here there are 2 pure strategy NEs: Both going to the CS department social event or both going to the dining hall. Notice, that if Alice goes to the CS event, but Bob does not, she would have been better off going to the dining hall instead and if Bob goes to the dining hall, but Alice does not, then he would have been better off going to the CS event instead. The only arrangements where neither would have been happier making a different choice is when they both end up choosing the same Friday night destination. Both choosing the dining hall is a NE, but not an efficient outcome because both would be happier having both gone to the CS event instead.

Example 2 *The Ultimatum Game - Yasmine and Zach have $100 to split between them. Yasmine writes down on a piece of paper the amount to give Zach and the rest she keeps. Zach writes down on a piece of paper simultaneously whether he will accept the offer without knowing what Yasmine wrote. Zach can either choose to accept the proposal and the $100 will be split accordingly, or Zach can refuse the offer and neither get anything. What is the pure strategy Nash equilibrium?*

In this case, the only pure strategy Nash equilibrium is for Yasmine to offer Zach $0 and to keep $100 and for Zach to accept the offer. Zach is no worse off accepting than rejecting with a proposal of $0 for him to keep and Yasmine would be better off always offering Zach less given that he accepts the proposal. Both strategies are weakly dominant.

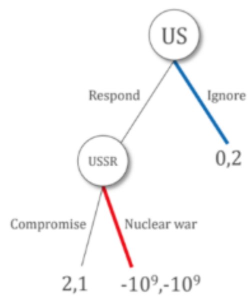
Problem 2 *Suppose there is a mature market where there are only 2 major sellers (called a duopoly). If they both advertise they both incur high costs with no extra sales. If they both do not advertise, they will both have higher profits. If one advertises but the other does not, then one will take significant market share from the other. Identify the pure strategy Nash equilibrium for this game:*

		Seller B	
		Advertise	No Ads
Seller A	Advertise	-1, -1	9, -5
	No Ads	-5, 9	5, 5

2 AI Game Playing

2.1 Extensive form games

Extensive-form is a game-tree used to represent sequential games, games involving a sequence of turns. Consider the below example from lecture – the extensive form is on the left. It is sequential, rather than normal form which is simultaneous.



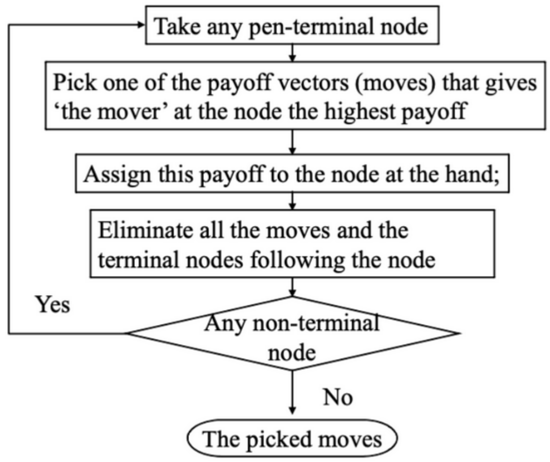
	Compromise	Nuclear war
Respond	2,1	-10^9,-10^9
Ignore	0,2	0,2

(Ignore, Nuclear War) is a Nash Equilibrium in a simultaneous, normal-form game where both players decide at once. But if this game is sequential, and players take turns making decisions, the USSR only continues playing if the US "Responds". Then, the USSR chooses between Compromise and Nuclear War, and Compromise clearly dominates Nuclear War. Thus, the threat of Nuclear War is not a credible threat, because if the USSR is actually called upon to make a decision, she will choose Compromise over Nuclear War. **So does it make sense for (Ignore, Nuclear War) to be a Nash Equilibria?**

2.2 Sub-game perfect Nash Equilibrium

The concept of **subgame perfect Nash equilibrium** is a refinement of Nash Equilibrium that deals with this problem. To define it, we need the idea of a subgame: every decision state in a game tree (including the initial state) defines a subgame. The above Cold War game has two subgames: one rooted in the US's decision state, and one rooted in the USSR's decision state. A set of strategies is a subgame perfect equilibrium if it is a Nash equilibrium in each subgame.

We can find subgame-perfect equilibria using backward induction, illustrated below.



Concept Check: How does this apply to the Cold War Game?

2.3 Minimax search

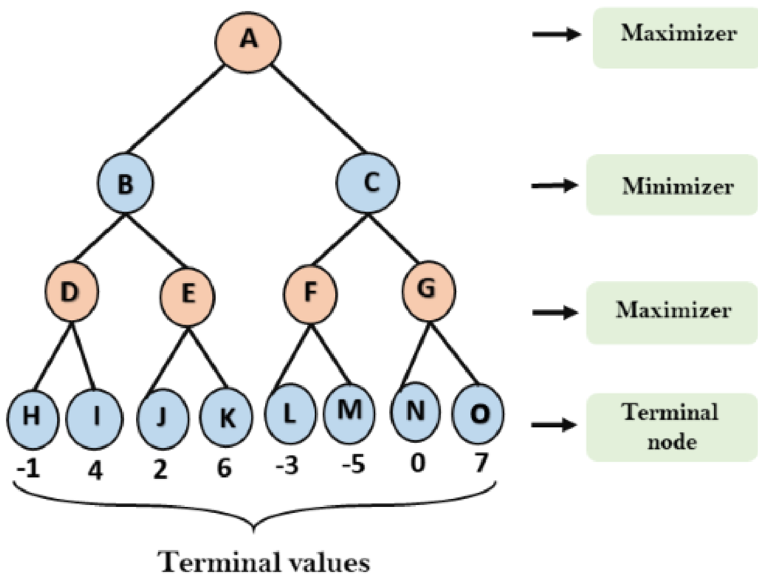
Definition 3 (Minimax search) *Minimax search, intuitively, is backward induction for a zero-sum game- you alternate between the min and max players' decisions. There is a maximum player and a minimum player and the goal of the maximum player is to maximize the final value and the minimum player wants to minimize the final value. The players alternate moves and this algorithm is better understood with pseudocode and examples.*

```
function MaxEval (node n)
    if n is a leaf then return PAYOFF(n)
    v ← −∞
    for all children n' of n
        v ← MAX(v, MINEVAL(n'))
    return v

function MinEval (node n)
    if n is a leaf then return PAYOFF(n)
    v ← ∞
    for all children n' of n
        v ← MIN(v, MAXEVAL(n'))
    return v
```

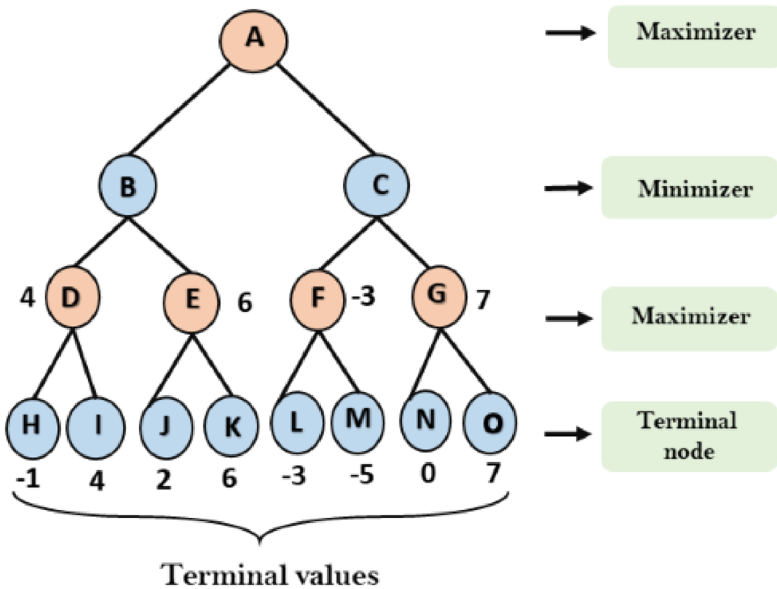
Example 3 *For an example, let us consider this diagram:*

A is the initial state of the tree. Suppose maximizer takes first turn which has worst-case initial value =- infinity, and minimizer will take next turn which has worst-case initial value = +infinity.



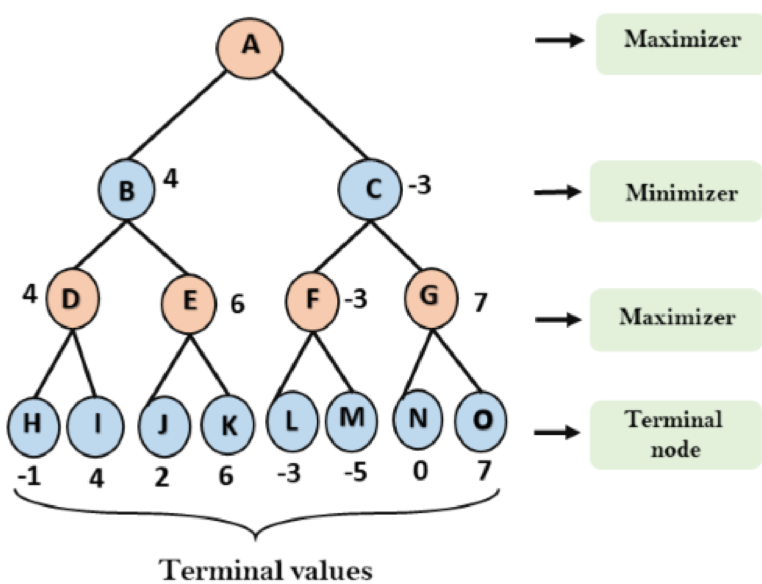
Now, we have the turn of the max player- its initial value is -infinity so we chooose the maximum between each node and $-\infty$.

- *For node D,* $max(-1, -\infty) => max(-1, 4) = 4$
- *For node E,* $max(2, -\infty) => max(2, 6) = 6$
- *For Node F,* $max(-3, -\infty) => max(-3, -5) = -3$
- *For node G,* $max(0, -\infty) = max(0, 7) = 7$



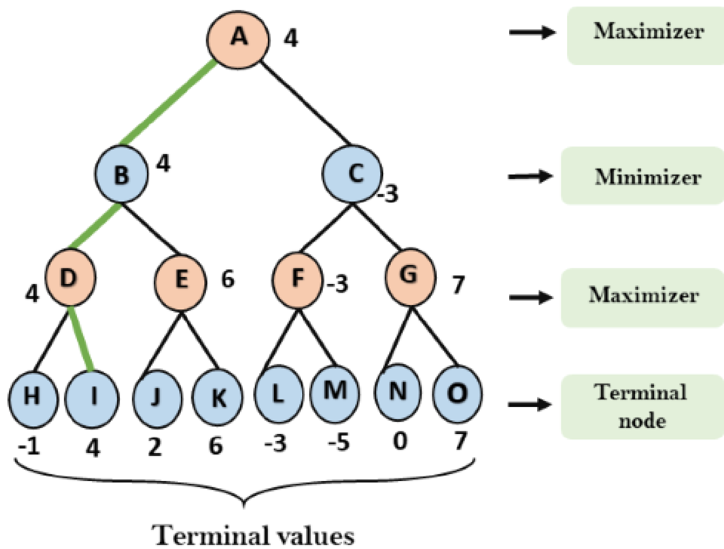
Now, the minimizer will play- they compare each node value and pick the minimum and will provide the third layer node values.

- *For node* $B = min(4, 6) = 4$
- *For node* $C = min(-3, 7) = -3$



Finally, Maximizer will play, and it will again choose the maximum of all nodes value and find the maximum value for the root node.

For node $A = max(4, -3) = 4$.



From https://www.javatpoint.com/mini-max-algorithm-in-ai.

2.4 Alpha-Beta Pruning

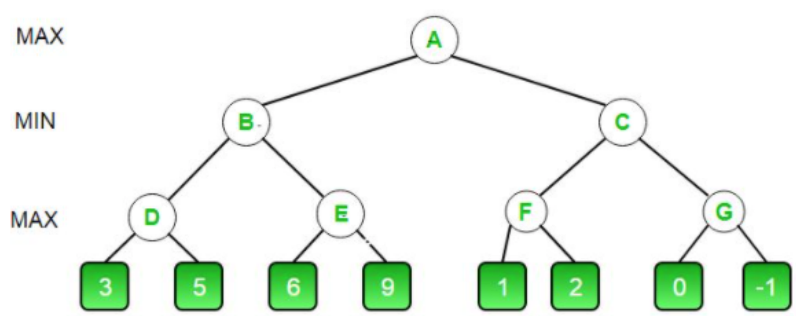
Definition 4 (Alpha-Beta Pruning) *Alpha-Beta pruning builds on minimax search. There are parameters* α, β *that we update and these guarantee ranges- the maximum player guarantees that the final value will be* $\geq \alpha$ *and the minimum player guarantees that the final value will be* $\leq \beta$ *so if a value node does not satisfy these conditions, we will prune that node and its descendants (will not search those spaces).*

A branch will be pruned if $\beta \leq \alpha$ *as these nodes will not be reached in actual play. Like minimax search, it is better explained with pseudocode and examples.*

```
function MaxEval (node n, numbers α,β)
        // max can guarantee ≥ α
        // min can guarantee ≤ β
        if n is a leaf then return PAYOFF(n)
        v ← α
        for all children n′ of n
                v ← MAX(v,MINEVAL(n′,v,β))
                if v ≥ β then return v
        return v

function MinEval (node n, numbers α,β)
        if n is a leaf then return PAYOFF(n)
        v ← β
        for all children n′ of n
                v ← MIN(v,MAXEVAL(n′,α,v))
                if v ≤ α then return v
        return v
```

Example 4 *Alpha-Beta Pruning example*



1. *The initial call starts from A. The value of alpha here is -INFINITY and the value of beta is +INFINITY. These values are passed down to subsequent nodes in the tree. At A the maximizer must choose max of B and C, so A calls B first*
2. *At B it the minimizer must choose min of D and E and hence calls D first.*
3. *At D, it looks at its left child which is a leaf node. This node returns a value of 3. Now the value of alpha at D is max(-INF, 3) which is 3.*
4. *To decide whether its worth looking at its right node or not, it checks the condition*

beta¡=alpha. This is false since beta = +INF and alpha = 3. So it continues the search.

5. *D now looks at its right child which returns a value of 5.At D, alpha = max(3, 5) which is 5. Now the value of node D is 5*

6. *D now looks at its right child which returns a value of 5.At D, alpha = max(3, 5) which is 5. Now the value of node D is 5*

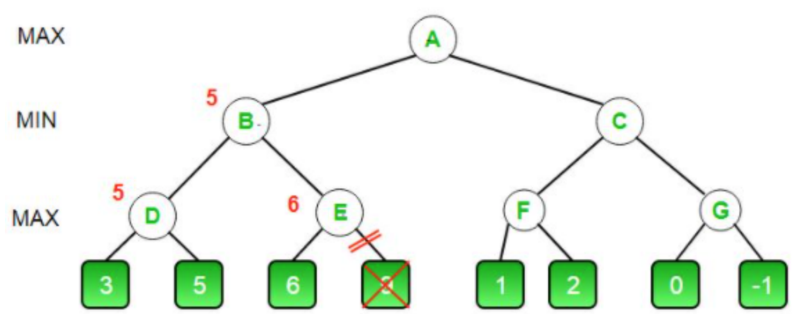
7. *D returns a value of 5 to B. At B, beta = min(+INF, 5) which is 5. The minimizer is now guaranteed a value of 5 or lesser. B now calls E to see if he can get a lower value than 5.*

8. *At E the values of alpha and beta is not -INF and +INF but instead -INF and 5 respectively, because the value of beta was changed at B and that is what B passed down to E*

9. *Now E looks at its left child which is 6. At E, alpha = max(-INF, 6) which is 6. Here the condition becomes true. beta is 5 and alpha is 6. So beta¡=alpha is true. Hence it breaks and E returns 6 to B*

10. *Note how it did not matter what the value of E's right child is. It could have been +INF or -INF, it still wouldn't matter, We never even had to look at it because the minimizer was guaranteed a value of 5 or lesser. So as soon as the maximizer saw the 6 he knew the minimizer would never come this way because he can get a 5 on the left side of B. This way we didn't have to look at that 9 and hence saved computation time.*

11. *E returns a value of 6 to B. At B, beta = min(5, 6) which is 5.The value of node B is also 5*



12. *B returns 5 to A. At A, alpha = max(-INF, 5) which is 5. Now the maximizer is guaranteed a value of 5 or greater. A now calls C to see if it can get a higher value than 5.*

13. *At C, alpha = 5 and beta = +INF. C calls F*

14. *At F, alpha = 5 and beta = +INF. F looks at its left child which is a 1. alpha = max(5, 1) which is still 5.*

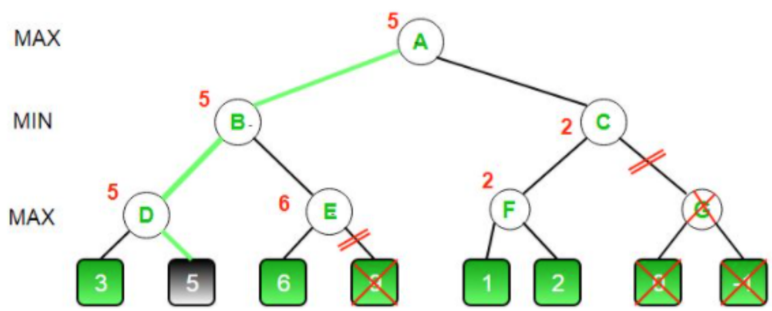
15. *F looks at its right child which is a 2. Hence the best value of this node is 2. Alpha still remains 5*

16. *F returns a value of 2 to C. At C, beta = min(+INF, 2). The condition beta ¡= alpha becomes true as beta = 2 and alpha = 5. So it breaks and it does not even have to compute the entire sub-tree of G.*

17. *The intuition behind this break off is that, at C the minimizer was guaranteed a value of 2 or lesser. But the maximizer was already guaranteed a value of 5 if he choose B. So why would the maximizer ever choose C and get a value less than 2 ? Again you can see that it did not matter what those last 2 values were. We also saved a lot of computation by skipping a whole sub tree.*

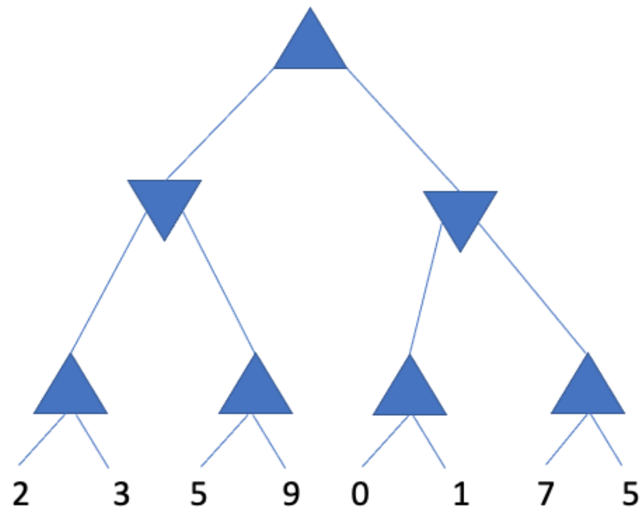
18. *C now returns a value of 2 to A. Therefore the best value at A is max(5, 2) which is a 5.*

19. *Thus, the optimal value that the maximizer can get is 5*



From https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/.

Problem 3 *Carry out alpha-beta pruning on this tree.*



2.5 Supplemental Examples and Problems

Problem 4 *Consider the below extensive game involving splitting 10 dollars. What are the sub- game perfect equilibria of this game?*