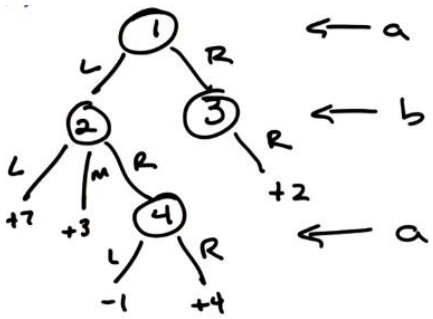
RL03. Game Theory I

What is Game Theory?

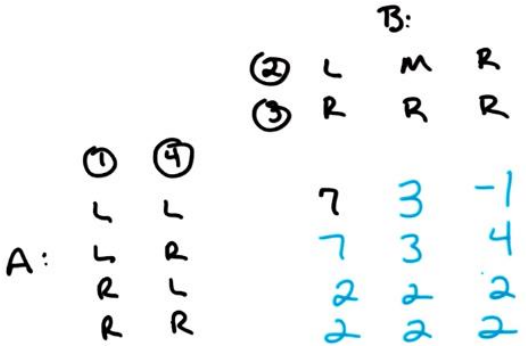
- Game Theory is the mathematics of conflicts of interests when making decisions. This can be related to AI when there're multiple agents acting in the environment.

A Simple Game:

- Imagine a world with two agents, a and b, acting in a simple game where a makes a choice, then b makes a choice, then a might get a chance to make another choice, and so on.
- We can represent the game in this tree, where the leaves represent $rewards$ given to a. Whenever a gets a $reward$, b will get $-reward$.



- This specific game is called "A 2-player zero-sum finite deterministic game of perfect information". It's the simplest possible game.
- In MDP we had the notion of a "Policy", which is mapping from states to actions. In Game Theory we have the notion of a "Strategy".
- A Strategy is mapping of "all possible" states to actions.
- Ultimately, a game can be completely captured in a matrix of all strategies a and b can choose along the axes of a matrix. The outcome of the games per each of these strategies will be represented in the cells of the matrix.
- Minimax: Since a and b are taking decisions against each other's interests, they both have to consider the worst case. So a will always try to choose the "maximum minimum" and b will try to choose the "minimum maximum". This is called the Minimax Algorithm.

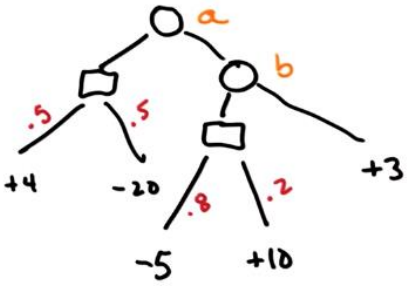


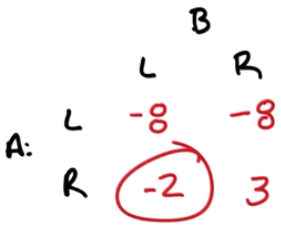
Fundamental Result:

- In a 2-player zero-sum finite deterministic game of perfect information, Minimax is equal to Maximin. There always exists an optimal pure strategy for each player.
- The optimal strategy here means that everyone is trying to maximize his own reward while knowing that everyone else is trying to do the same.

Game Tree:

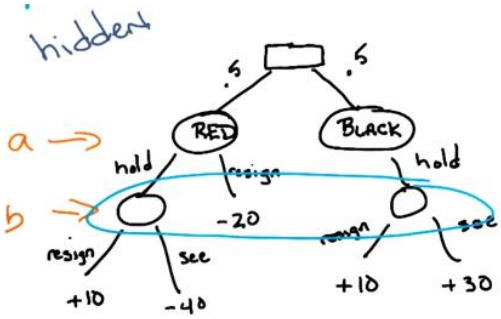
- Now we consider a more complex scenario, a 2-player zero-sum finite **non**-deterministic game of perfect information. The game involves a non-deterministic behavior in certain states.

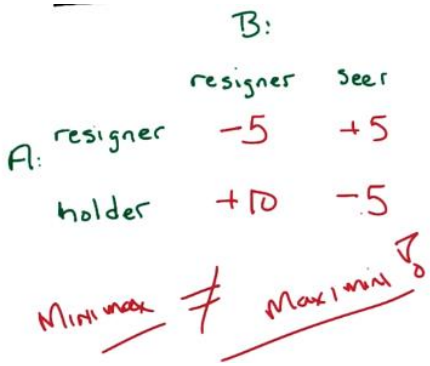




Minipoker:

- Now we consider an even more complex scenario, a 2-player zero-sum finite **non**-deterministic game of ~~perfect~~ **hidden** information.
- Imagine if our players are dealing red/black cards, where red is a bad choice for a, and black is the good choice:
  - a is dealt a card, with a 50% chance of being red or black.
  - a may resign, if red → -20 cents for a
  - Else a may hold:
    1. b resigns → +10 cents.
    2. b sees: if red → -40 cents.
       if black → +30 cents.





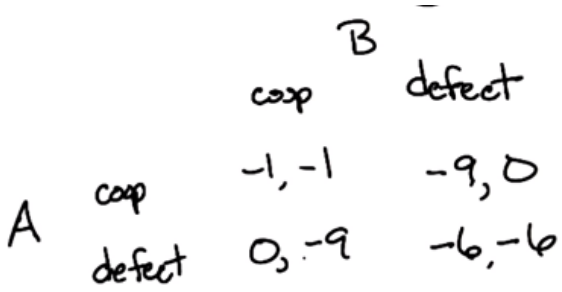
Mixed Strategies:

- In a mixed strategy, instead of making the same decisions given the same circumstances (pure strategy), we assign probabilities to our states (distribution over strategy).
- When using a mixed strategy between 2 options for both players, one can examine one of the players as deterministic and calculate the outcome as a function of the other probabilistic player's

probability of picking a strategy. The maximin of the two functions of the two deterministic possibilities gives the ultimate outcome of the game.

Snitch:

- Now we consider an even more complex scenario, a 2-player **non**-zero-sum finite **non**-deterministic game of ~~perfect~~ **hidden** information.
- Prisoner's dilemma: Imagine 2 criminals caught and put in separate jails. Each person is given the option to snitch on the other guy.
- The "best" outcome on average for both prisoners is to both cooperate, then they spend the least time in prison.
- But, based on the costs, it makes more sense to always defect instead of cooperating. If you're A and B cooperates, defecting will give you no time in jail. If B defects, defecting will give you 6 months instead of 9 months.



- This dominance of defecting always being the better option results in the "worst" outcome.

Nash Equilibrium:

- Given N players with strategies $s_1, s_2, \dots, s_n$,
  Then,

$$s_1{}^* \in s_1, s_2{}^* \in s_2, \dots, s_n{}^* \in s_n \quad iff \quad \bigvee_i s_i{}^* = argmax_{s_i}[utility_i(s_1{}^*, \dots, s_i, \dots, s_n{}^*)]$$

  - A Nash Equilibrium means that we're in a situation where no one has any reason to change his strategy.
- Nash Equilibrium applies to both pure and mixed strategies.
- We have three theorems resulting from Nash Equilibrium:
  - In the n-player pure strategy game, if equilibrium of strictly dominated strategies eliminates all but one combination, that combination is the Nash Equilibrium.
  - Any Nash Equilibrium will survive elimination of strictly dominated strategies.
  - If the number of players is finite and we have finite number of strategies for each player, there exists at least one (possibly mixed) Nash Equilibrium.

The Two Step:

- If we have a two-step prisoner's dilemma, perhaps it's in your interest to cooperate as a signal to the other player that you're the type to cooperate so that the overall cost to you is reduced over several steps.
- However, it really doesn't matter. In the final game there is only one game left and all the previous games are sunk costs. Therefore, it is determined that the players will end up in the (-6, -6) choice.

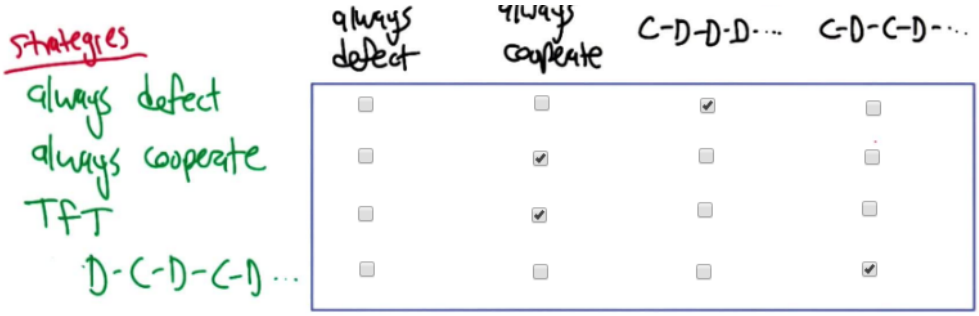
And since we know the outcome of the final (n) game, we can consider the $(n-1)$ game as the "final" game. Therefore, proof by induction dictates that we'll always defect.

- If you have an (n) repeated games, then the (n) repeated Nash equilibrium is the solution.

RL03. Game Theory II

The Iterated Prisoner's Dilemma – IPD:

- We already discussed that in one game, the only rational thing to do is to defect. Even in a finite and known number of games, the ultimate choice would still be defecting.
- The question is, what happens if the number of rounds left unknown?
- The Uncertain End:
  - The probability that the game continues is γ.
  - Each round could be the last one, or not!
  - The expected number of rounds is $\frac{1}{1-\gamma}$
  - γ works similar to the discount factor.
- Tit-for-Tat:
  - This is an IPD strategy where opponents cooperate on the first round, then each one copies the opponent's previous move thereafter.



  - What's the best strategy against TFT?
    1. If we have a low γ, the best strategy is to always defect. The total reward would be:

$$0 + \frac{-6\gamma}{1-\gamma}$$

    2. If we have a high γ, the best strategy is to always cooperate. The total reward would be:

$$\frac{-1}{1-\gamma}$$

    3. The two strategies will result in the same reward if:

$$\frac{-6\gamma}{\cancel{1-\gamma}} = \frac{-1}{\cancel{1-\gamma}}$$

$$-6\gamma = -1$$

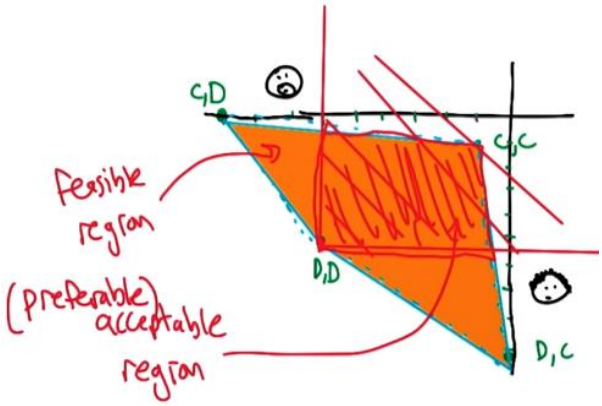
$$\gamma = \frac{1}{6}$$

Best Response to a Finite State Strategy:

- If we were in a one-round game, the matrix would be all what you need to select the best strategy.
- On the other hand, in a multi-round setting, our choice not only impacts our pay off, but also the future decisions of the opponent.
- So, in a multi-round scenario, we need a state machine representation to make sense of the game.
- This state machine actually represents an MDP where we map from the opponent's state to an action for us.

Folk Theorem:

- In a repeated game setting, the possibility of retaliation (defecting) opens the door for cooperation.
- In Game Theory, the Folk Theorem describes the set of payoffs that can result from Nash strategies in repeated games.
- Minmax Profile: A pair of payoffs, one for each player, that represent the payoffs that can be achieved by a player defending itself from a malicious adversary (some other player trying to lower my score).
- Security Level Profile: Minmax of PD is (D, D) because the malicious adversary will defect, and your best option is to defect as well. The region above and to the right of the minmax point is the (preferable) acceptable region, it is better than what you can guarantee against someone acting maliciously.



- Folk Theorem: Any feasible payoff profile that strictly dominates the minmax/security level profile can be realized as a Nash Equilibrium payoff profile, with sufficiently large discount factor.
  Proof: If it strictly dominates the minmax profile, can use it as a threat. Better off doing what you are told.

Grim Trigger:

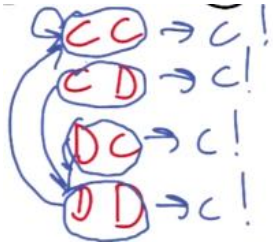
- Cooperation keeps you in the state of mutual benefit, but if defected once, you enter the state of dealing out of vengeance and stay in that state forever. This creates a Nash Equilibrium.
- The only problem with this concept is that the threat is implausible, because it doesn't make since that a player will stop taking the best response (in his interest) only to punish the other player. So, in this setting, we end up with a Subgame Perfect Equilibrium.

Subgame Perfect Equilibrium:

- Subgame Perfect Equilibrium: Each player will always take the best response independent of history.
- For example, if we're playing Grim vs TFT, we have a Nash Equilibrium, but we don't have a Subgame Perfect Equilibrium.
- TFT vs TFT: This is not Subgame Perfect as well. If we initialized with D/C, we'll end up with alternating choices which is worse than cooperating forever.

Pavlov:

- In Pavlov, the player cooperates if the both players agreed on the last round. Will defect otherwise (disagree).
- Pavlov vs Pavlov is a Nash Equilibrium.
- Pavlov vs Pavlov is Subgame Perfect. It will always end up cooperating.
- Pavlov creates plausible threats.



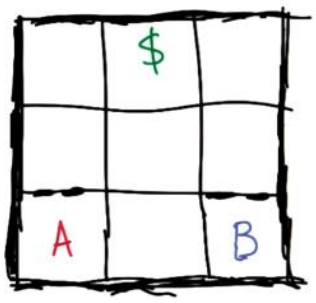
Computational Folk Theorem:

- Given a 2-player bi-matrix game (each player has a separate reward structure), you can build Pavlov-like machines for any game. And using Pavlov you can construct Subgame Perfect Nash Equilibrium for any game in polynomial time.

Stochastic Games and Multiagent RL:

- Stochastic Games is a formal model for multiagent RL (like MDP to RL).
- The Game:
  - 3×3 grid.
  - Two players: A and B.
  - Possible directions: N, S, E, W, X.
  - If both A and B arrived, they both win.
  - Semi-wall: 50% go through.
- Formal definitions:
  - S: States.
  - A_i: Actions of player i (a for player A and b for player B).
  - T: Transactions → $T(s,(a,b),s')$.
  - R_i: Rewards for players i → $R_1\big(s,(a,b)\big), R_2(s,(a,b))$.
  - γ: Discount.
- If we constrained the Stochastic Game model, we end up with:
  - Zero-sum Stochastic Game:

$$R_1 = -R_2$$



- MDP:

$$T(s,(a,b),s') = T(s,(a,b'),s') \bigvee b'$$
$$R_2(s,(a,b)) = 0$$
$$R_1(s,(a,b)) = R_1(s,(a,b')) \bigvee b'$$

- Repeated Game:

$$|s| = 1$$

- Zero-sum Stochastic Games:
  - If we look into the Bellman equation:

$$Q^*{}_i(s,(a,b)) = R_i(s,(a,b)) + \gamma \sum_{s'} T(s,(a,b),s') max_{a',b'}\, Q^*{}_i(s',(a',b'))$$

  - The max in the equation assumes that the joint actions will always benefit you, which is delusional considering a zero-sum game. We will modify the equation to solve this issue:

$$Q^*{}_i(s,(a,b)) = R_i(s,(a,b)) + \gamma \sum_{s'} T(s,(a,b),s') minimax_{a',b'}\, Q^*{}_i(s',(a',b'))$$

  - Now we can solve this using a $Q-$ Learning approach. This is sometimes called "$minimax-Q$":

$$< s,(a,b),(r_1,r_2),s' > :\ Q(s,(a,b)) \leftarrow^{\alpha} r_i + \gamma\, max_{a',b'} Q_i(s',(a',b'))$$

  - We end up with these facts:
    1. Value iteration works.
    2. $minimax-Q$ converges.
    3. Unique solution to Q^*.
    4. Policies can be computed independently: If two players are running Minimax-Q on this own, the policies they will get will converge to $minimax-Q$ optimal policies.
    5. Update is efficient (polynomial) because $minimax$ can be computed using linear programming.
    6. Q functions are sufficient to specify the Policy.
- General-sum Stochastic Games:
  - If we look into the Bellman equation:

$$Q^*{}_i(s,(a,b)) = R_i(s,(a,b)) + \gamma \sum_{s'} T(s,(a,b),s') max_{a',b'}\, Q^*{}_i(s',(a',b'))$$

  - We can't use $minimax$ since it assumes that the other players are trying to minimize my rewards, which is not correct in this setting. We will use Nash Equilibrium instead:

$$Q^*{}_i(s,(a,b)) = R_i(s,(a,b)) + \gamma \sum_{s'} T(s,(a,b),s') Nash_{a',b'}\, Q^*{}_i(s',(a',b'))$$

  - Now we can solve this using a $Q-$ Learning approach. This is sometimes called "$Nash-Q$":

$$< s,(a,b),(r_1,r_2),s' > :\ Q(s,(a,b)) \leftarrow^{\alpha} r_i + \gamma\, Nash_{a',b'} Q_i(s',(a',b'))$$

- We end up with these facts:
  1. Value iteration doesn't work.
  2. $Nash - Q$ doesn't converge.
  3. No unique solution to Q*: We can have Different Nash Equilibria with different values.
  4. Policies cannot be computed independently: Nash Equilibrium is a joint behavior, so we cannot have two different players computing the Q. We will end up with incompatible results.
  5. Update is not efficient.
  6. Q functions are not sufficient to specify the Policy.