Week One

MDST F24

Game Theory

- **Definition:** the study of strategic interactions between rational decision-makers. It provides mathematical frameworks to model situations where the outcome for each participant depends not only on their own actions but also on the actions of others.

- **Importance:**
  - **Strategic Decision-Making**: Game theory is crucial in economics, political science, biology, and computer science, helping predict and understand the behavior of individuals or groups in competitive situations.
  - **Applications**: From business strategies, auctions, and voting systems to military tactics and evolutionary biology, game theory offers insights into optimizing outcomes in various scenarios.

Type of Games

- **Cooperative vs. Non-Cooperative Games**:
  - **Cooperative Games:** Players can form coalitions and make binding agreements. The focus is on how to distribute collective gains.
  - **Non-Cooperative Games:** Each player acts independently, aiming to maximize their own payoff without binding agreements. The focus is on strategy and individual optimization.

- **Zero-Sum vs. Non-Zero-Sum Games**:
  - **Zero-Sum Games:** One player's gain is exactly equal to another player's loss. The total benefit available to all players remains constant.
  - **Non-Zero-Sum Games:** The total payoff to all players can vary; cooperation can lead to mutually beneficial outcomes.

Key Ideas

- **Players**: The decision-makers in the game. Each player aims to maximize their own payoff.
- **Strategies**: The set of possible actions a player can take. A strategy profile lists the strategies chosen by each player.
- **Payoffs**: The reward or outcome each player receives based on the strategy profile chosen. Payoffs can be represented in a matrix or as utility functions.

- **Games of Perfect vs. Imperfect Information**:
  - **Perfect Information**: All players are fully aware of all previous actions and the current state of the game (e.g., chess).
  - **Imperfect Information**: Some information is hidden from players (e.g., poker, where players cannot see each other's cards).
- For our project we will be working with non-cooperative, two-player zero-sum, imperfect information games.

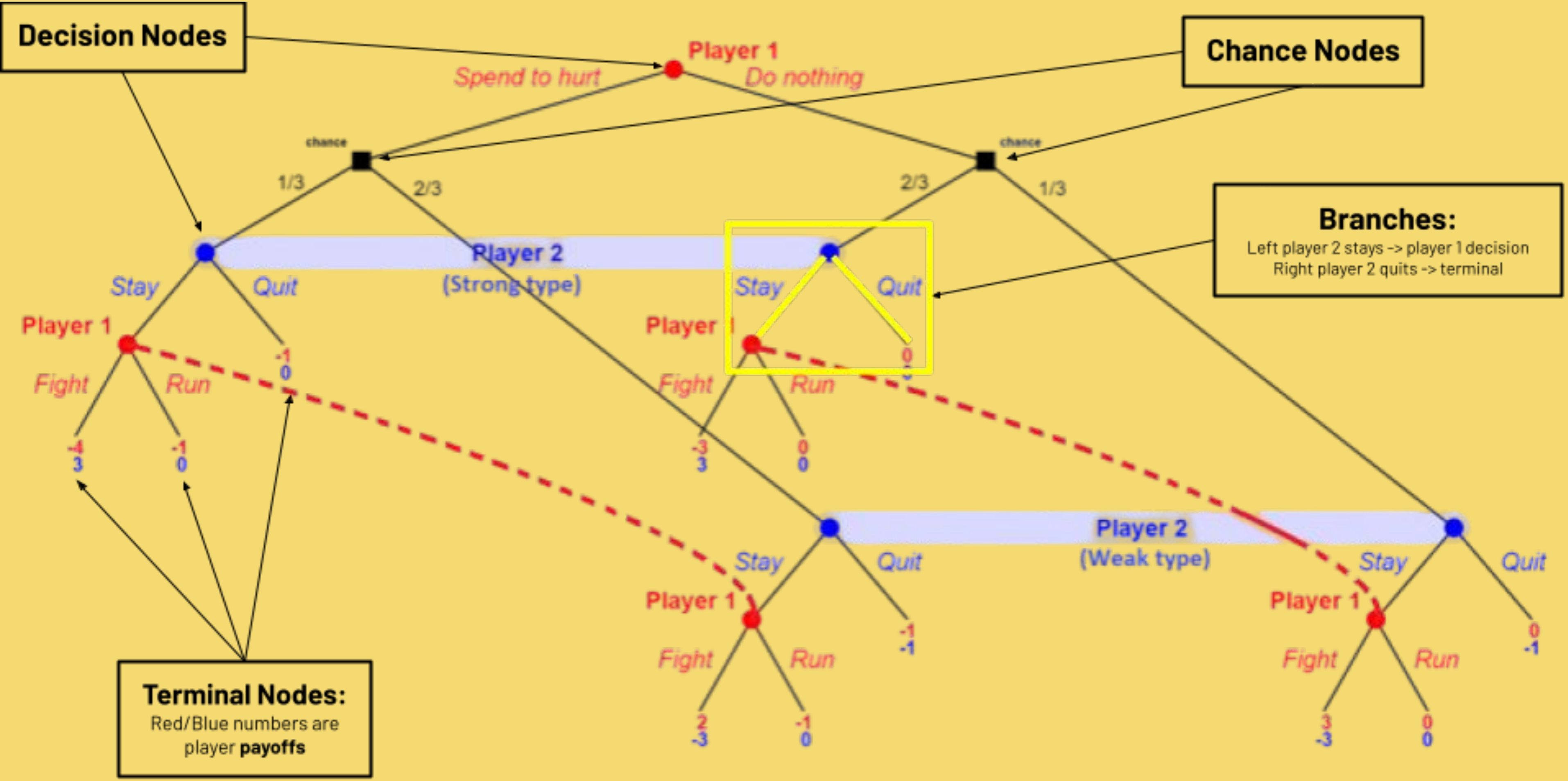
Extensive Form Representation

- **Definition**: a way to model games where players make decisions at different points in time. It provides a comprehensive view of all possible actions and their consequences, including the timing of moves, the available information at each decision point, and the possible outcomes.

- Useful for games where the <u>order of moves is important</u> and where players might have <u>different information</u> at different stages of the game.

Representation Components

- **Nodes (Vertices)**:
  - Represent decision points where a player (or chance) chooses an action.
  - **Decision nodes** are controlled by players, while **chance nodes** are controlled by a randomizing device (like dice or cards).
- **Branches (Edges)**:
  - Represent the actions that can be taken at each decision point.
  - Each branch leads to a new node or an outcome, showing the consequences of a decision.
- **Terminal Nodes**:
  - Represent the end of the game.
  - Each terminal node corresponds to a possible outcome of the game, often associated with a payoff for each player.
- **Payoffs**:
  - Each terminal node is associated with a payoff vector, indicating the reward each player receives if the game ends at that node.



Decision Nodes
Player 1
Spend to hurt
Do nothing
Chance Nodes
chance
chance
1/3
2/3
2/3
1/3
Branches:
Left player 2 stays -> player 1 decision
Right player 2 quits -> terminal
Player 2
(Strong type)
Stay
Quit
Stay
Quit
Player 1
-1
0
Player 1
0
0
Fight
Run
Fight
Run
-4
3
-1
0
-3
3
0
0
Player 2
(Weak type)
Stay
Quit
Stay
Quit
Player 1
-1
-1
Player 1
0
-1
Fight
Run
Fight
Run
2
-3
-1
0
3
-3
0
0
Terminal Nodes:
Red/Blue numbers are player payoffs

Abstraction

- **Definition:** Abstraction in game theory involves simplifying a game by focusing on essential elements like players, actions, and payoffs, while omitting non-essential details. This process helps in analyzing the game more efficiently.

- **Five Steps:**
  a. Identify the players (Player 1 and Player 2 in our case, in general we define the player number as N)
  b. Determine sequence of moves (i.e. the order of play)
  c. Identify possible actions at each decision node (example: pass or bet)
  d. Represent the game as a tree, with the root being the first action taken, and the leafs being the terminal nodes. (example: one terminal node could be a player folding)
  e. Assign payoffs to the terminal nodes representing the winner and loser in that sequence. (Example: can use (+1,-1) to represent player one winning and player two losing)

- Following these guidelines allows us to simplify the game, and helps us eventually find the optimal strategy through backtracking and regret minimization

Kuhn Poker

- A simplified poker variant introduced by Harold W. Kuhn in 1950. It's often used to illustrate fundamental concepts in game theory due to its simplicity and strategic depth.
- **Game Setup:**
  - There are two players: Player 1 (P1) and Player 2 (P2).
  - The deck contains three cards: King (K), Queen (Q), and Jack (J).
  - Each player is dealt one card, and the remaining card is hidden.

Kuhn Poker Rules

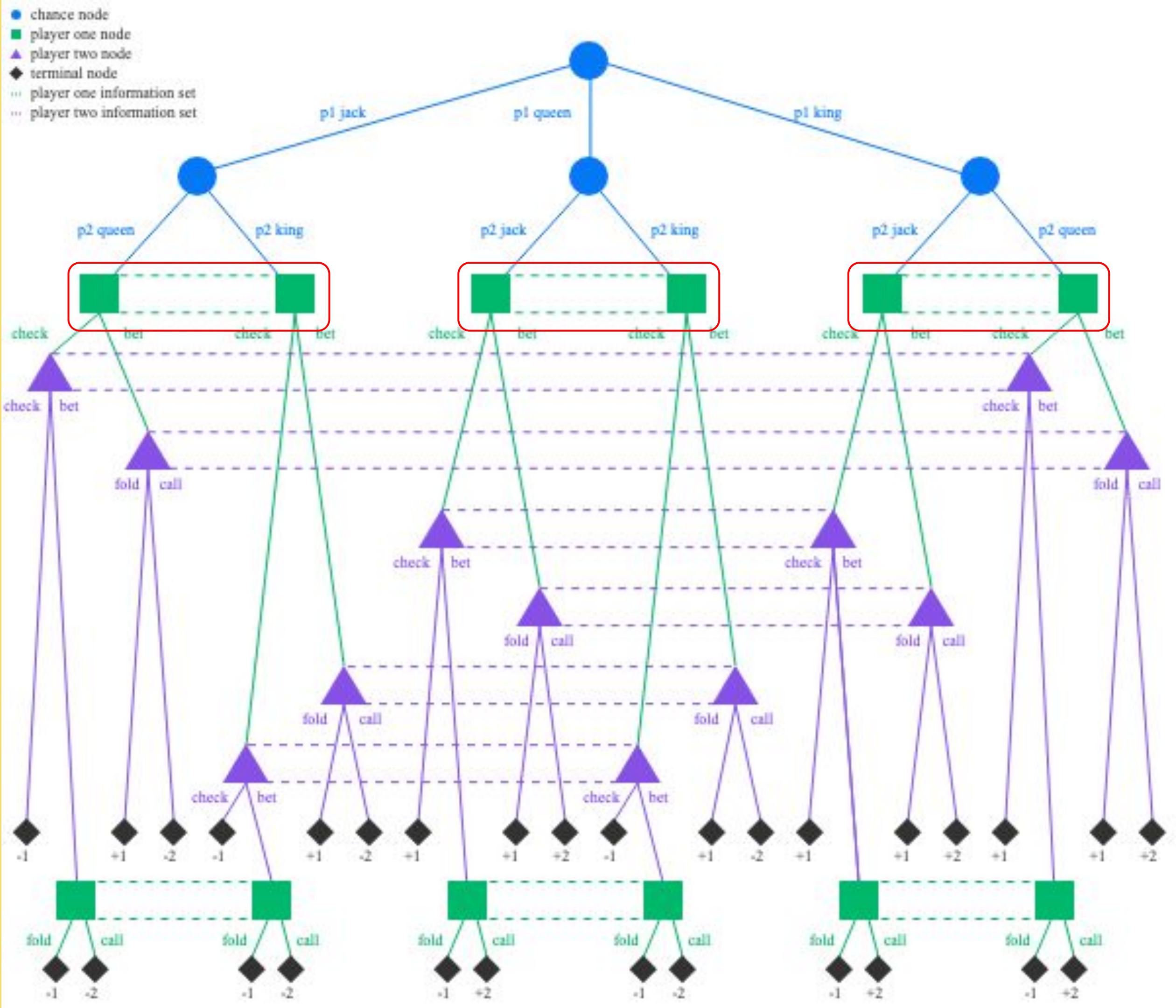
- **Starting the Game**:
  - Each player antes 1 chip into the pot, so the initial pot is 2 chips.
  - Player 1 (P1) is dealt a card, followed by Player 2 (P2).
  - The third card is not revealed and remains hidden.
- **Actions**:
  - The game proceeds with one betting round.
  - **Player 1's Turn**: P1 can either **bet** (raise) or **pass** (check).
  - **Player 2's Response**:
  - If P1 passes, P2 can either bet or pass.
  - If P1 bets, P2 can either **call** (match the bet) or **fold** (give up).
- **Outcomes**:
  - If both players pass, the player with the higher card wins the pot.
  - If a bet is made and the other player folds, the bettor wins the pot.
  - If P2 calls a bet, the cards are revealed, and the player with the higher card wins the pot.
  - The pot is split if both players end up with the same card, but this scenario is impossible in Kuhn Poker due to the three-card deck.

Kuhn Poker Abstraction

- **Initial Node:**
  - Represents the start of the game where P1 is dealt a card.
- **Branches for P1:**
  - P1 can either bet or pass.
- **Subsequent Nodes:**
  - If P1 passes, a new node represents P2's decision (bet or pass).
  - If P1 bets, another node represents P2's decision (call or fold).
- **Terminal Nodes:**
  - Represent the end of the game, with each terminal node indicating the outcome (who wins the pot and how much).
- **Information Set:**
  - Definition: An information set is a collection of decision points (nodes) in a game tree where a player, when making a decision, cannot distinguish between them.
  - P2 does not know what card P1 holds. Thus, P2's decision nodes after P1 bets are in the same information set.
  - This means P2 must choose the same strategy regardless of the specific node they are at within their information set.

Extensive Form for Kuhn Poker

Notice how the information sets are structured, for example, P1 at his first decision node, has the same information set regardless of the card P2 is assigned since he only knows his own card. The dotted lines attaching the different decision nodes to one another represent situations in which a P1 or P2 are not sure of which game state they are in. For example, if P1 has the King he is oblivious to whether P2 has a Jack or Queen.



Optimal Play

- **Definition:** Optimal play refers to a strategy that maximizes a player's payoff, given the strategies of other players. In the context of game theory, this often means finding a strategy that no player can improve upon, assuming the other players' strategies remain fixed.
- **Rationality:**
  - In game theory, players are assumed to be rational, meaning they will choose strategies that maximize their expected utility.
  - Optimal play is derived from the assumption that each player will choose the best possible action, considering the potential actions of the other players.
- **Dominant Strategies:**
  - A dominant strategy is one that provides a higher payoff regardless of what the other players do.
  - If a player has a dominant strategy, they will always play it, as it represents their optimal play, however not all games have dominant strategies.
- **Mixed Strategies:**
  - In some games, no dominant strategy is optimal. Instead, players may use mixed strategies, where they randomize over possible actions.
  - Mixed strategies can often be optimal in games with no clear dominant strategy, ensuring that a player is unpredictable and cannot be exploited by opponents.

Nash Equilibrium

- **Definition:** A Nash equilibrium occurs when each player's strategy is optimal given the strategies of the other players. No player can unilaterally change their strategy to achieve a better outcome.

- In equilibrium, every player is playing optimally, considering the strategies of others.

- An important thing to note is that in some games, regardless if you are playing optimally, you can still lose hands based on sheer luck. Nash equilibrium guarantees us that in the long run our strategy is unbeatable, and we will be net positive over time.

- This is the concept behind why casinos are willing to lose significantly on some nights since in the long run the odds are in their favor.

Introduction to CFR

- **Definition:** Counterfactual Regret Minimization is an iterative algorithm used to compute approximate Nash equilibria in large games. It is particularly effective in extensive form games like poker, where decisions are made sequentially, and players have imperfect information.
- **Purpose:** CFR aims to minimize regret over time, ensuring that the strategies converge towards an equilibrium where no player has an incentive to deviate.
- **Regret:** In the context of game theory, regret is the difference between the payoff a player could have received by playing a different strategy and the payoff they actually received. If a player regrets not choosing a different action, this regret informs how they might adjust their strategy in future iterations. We will look to minimize regret by optimizing our strategy.
- **Counterfactual Regret:** This specific type of regret considers what the outcome would have been if a player had taken a different action, given the sequence of events that actually occurred.

Counterfactual value

$$v_i(\sigma, h) = \sum_{z \in Z, h \sqsubset z} \pi^{\sigma}_{-i}(h)\pi^{\sigma}(h, z)u_i(z).$$

- $\pi^{\sigma}_{-i}(h)$ The probability of reaching either history h, assuming that player i takes whatever action needed to reach that state
- $\pi^{\sigma}(h, z)$ is the probability of going from non-terminal game history h to terminal game history z
- Z is the set of terminal game histories, $h \sqsubset z$ is the set of nonterminal game histories that lead to the terminal game histories in Z.

This formula gives us the utility gained from player *i* at node *h*, weighted by the chance that they actually reach that node and the terminal state

Computing Counterfactual Regret

The counter-factual regret for not selecting action a given history h is given by

$$r(h, a) = v_i(\sigma_{I \to a}, h) - v_i(\sigma, h).$$

Which means that the counter factual regret of player i for not selecting action a given *information state* I is given by

$$r_i(I, a) = \sum_{h \in I} r_i(h, a).$$

And the cumulative regret is:

$$R_i^T(I, a) = \sum_{t=1}^{T} r_i^t(I, a).$$

Updating Strategy

- Using the 'regret matching' algorithm, we can determine the strategy profile to be used in the next iteration of CFR:
- Average of all strategy profiles from Sigma^1 to Sigma^T converges to equilibrium

$$\sigma_i^{T+1}(I,a) = \begin{cases} \frac{R_i^{T,+}(I,a)}{\sum_{a \in A(I)} R_i^{T,+}(I,a)} & \text{if } \sum_{a \in A(I)} R_i^{T,+}(I,a) > 0 \\ \frac{1}{|A(I)|} & \text{otherwise.} \end{cases}$$

R^T+ signifies that we only include **positive regrets** into summation

Useful Resources

- CFR is the most conceptually difficult component of this project. Much of this may be new to you including the mathematics, backtracking, tree representation, and game theory background so familiarizing yourself with the material will be very important
- Some good resources to check out on this algorithm include:
  - https://aipokertutorial.com/the-cfr-algorithm/
  - https://youtu.be/ygDt_AumPr0?si=xcilQ4Y__LwCc45h
  - http://modelai.gettysburg.edu/2013/cfr/cfr.pdf* (Algorithm you will be implementing)
  - https://poker.cs.ualberta.ca/publications/NIPS07-cfr.pdf
- Don't feel discouraged if you were unable to follow along with the entire algorithm, feel free to ask any questions and spend time trying to understand how it works. In the next section we will have you look at pseudocode for the CFR algorithm, then you will have the opportunity to implement it for Kuhn Poker as our last activity.

CFR Pseudocode

Walk Tree

This function walks the game tree.

- `h` is the current history h
- `i` is the player i that we are computing regrets of
- `pi_i` is $\pi_i^{\sigma^t}(h)$
- `pi_neg_i` is $\pi_{-i}^{\sigma^t}(h)$

It returns the expected utility, for the history h

$$\sum_{z \in Z_h} \pi^\sigma(h, z) u_i(z)$$

where Z_h is the set of terminal histories with prefix h

While walking the tee it updates the total regrets $R_i^T(I, a)$.

Activity

Hint Available on next slide

Algorithm 1 Counterfactual Regret Minimization (with chance sampling)

Initialize cumulative regret tables: $\forall I, r_I[a] \leftarrow 0$.
Initialize cumulative strategy tables: $\forall I, s_I[a] \leftarrow 0$.
Initialize initial profile: $\sigma^1(I, a) \leftarrow 1/|A(I)|$

function CFR(h, i, t, π_1, π_2):
if h is terminal **then**
    return $u_i(h)$
else if h is a chance node **then**
    Sample a single outcome $a \sim \sigma_c(h, a)$
    return CFR(ha, i, t, π_1, π_2)
end if
Let I be the information set containing h.
$v_\sigma \leftarrow 0$
$v_{\sigma_{I \rightarrow a}}[a] \leftarrow 0$ for all $a \in A(I)$
for $a \in A(I)$ **do**
    if $P(h) = 1$ **then**
        $v_{\sigma_{I \rightarrow a}}[a] \leftarrow$ CFR(ha, i, t, $\sigma^t(I, a) \cdot \pi_1$, π_2)
    else if $P(h) = 2$ **then**
        $v_{\sigma_{I \rightarrow a}}[a] \leftarrow$ CFR(ha, i, t, π_1, $\sigma^t(I, a) \cdot \pi_2$)
    end if
    $v_\sigma \leftarrow v_\sigma + \sigma^t(I, a) \cdot v_{\sigma_{I \rightarrow a}}[a]$
end for
if $P(h) = i$ **then**
    for $a \in A(I)$ **do**
        $r_I[a] \leftarrow r_I[a] + \pi_{-i} \cdot (v_{\sigma_{I \rightarrow a}}[a] - v_\sigma)$
        $s_I[a] \leftarrow s_I[a] + \pi_i \cdot \sigma^t(I, a)$
    end for
    $\sigma^{t+1}(I) \leftarrow$ regret-matching values computed using Equation 5 and regret table r_I
end if
return v_σ

Each of these numbers corresponds to a snippet of code you should write to complete this function. Good Luck!

1 If it's a terminal history $h \in Z$ return the terminal utility $u_i(h)$.

2 If it's a chance event $P(h) = c$ sample a and go to next step.

3 Get current player's information set for h

4 To store $\sum_{z \in Z_h} \pi^{\sigma}(h, z) u_i(z)$

5 To store

$$\sum_{z \in Z_h} \pi^{\sigma^t|_{I \to a}}(h, z) u_i(z)$$

for each action $a \in A(h)$

6 Iterate through all actions

7 If the current player is i,

8

$$\pi_i^{\sigma^t}(h + a) = \pi_i^{\sigma^t}(h) {\sigma^t}_i(I)(a)$$
$$\pi_{-i}^{\sigma^t}(h + a) = \pi_{-i}^{\sigma^t}(h)$$

9 Otherwise,

10

$$\pi_i^{\sigma^t}(h + a) = \pi_i^{\sigma^t}(h)$$
$$\pi_{-i}^{\sigma^t}(h + a) = \pi_{-i}^{\sigma^t}(h) * {\sigma^t}_i(I)(a)$$

11

$$\sum_{z \in Z_h} \pi^{\sigma}(h, z) u_i(z) = \sum_{a \in A(I)} \left[{\sigma^t}_i(I)(a) \sum_{z \in Z_h} \pi^{\sigma^t|_{I \to a}}(h, z) u_i(z) \right]$$

12 If the current player is i, update the cumulative strategies and total regrets

13 Update cumulative strategies

$$\sum_{t=1}^{T} \pi_i^{\sigma^t}(I) \sigma^t(I)(a) = \sum_{t=1}^{T} \left[\sum_{h \in I} \pi_i^{\sigma^t}(h) \sigma^t(I)(a) \right]$$

14

$$\tilde{r}_i^t(I, a) = \tilde{v}_i(\sigma^t|_{I \to a}, I) - \tilde{v}_i(\sigma^t, I)$$
$$= \pi_{-i}^{\sigma^t}(h) \Big(\sum_{z \in Z_h} \pi^{\sigma^t|_{I \to a}}(h, z) u_i(z) - \sum_{z \in Z_h} \pi^{\sigma}(h, z) u_i(z) \Big)$$

$$TR_i^T(I, a) = \sum_{t=1}^{T} \tilde{r}_i^t(I, a)$$

15 Update the strategy $\sigma^t(I)(a)$

16 Return the expected utility for player i,

$$\sum_{z \in Z_h} \pi^{\sigma}(h, z) u_i(z)$$

Nash Equilibrium in Kuhn Poker

- **Player 1:**
  - **Optimal Strategy:** Player 1 should bet with the King and bluff with the Jack about 1/3 of the time. This bluff is a form of mixed strategy.
  - **Reasoning:** Betting with the King is a dominant strategy, as it is the highest card. Bluffing with the Jack introduces uncertainty for Player 2, making it harder for them to play optimally.
- **Player 2:**
  - **Optimal Strategy:** If Player 1 bets, Player 2 should call if they hold a King or Queen and fold if they hold a Jack. This balances the expectation of Player 1's bluffing frequency.
  - **Reasoning:** Betting with the King is a dominant strategy, as it is the highest card. Bluffing with the Jack introduces uncertainty for Player 2, making it harder for them to play optimally.
- **Results:** The Nash equilibrium in Kuhn Poker is reached when both players adopt these strategies, where Player 1 occasionally bluffs with the Jack, and Player 2 responds optimally based on the probability of a bluff.
- **Takeaways:**
  - This equilibrium is **not purely deterministic**; it involves mixed strategies, highlighting the strategic depth even in simplified games.