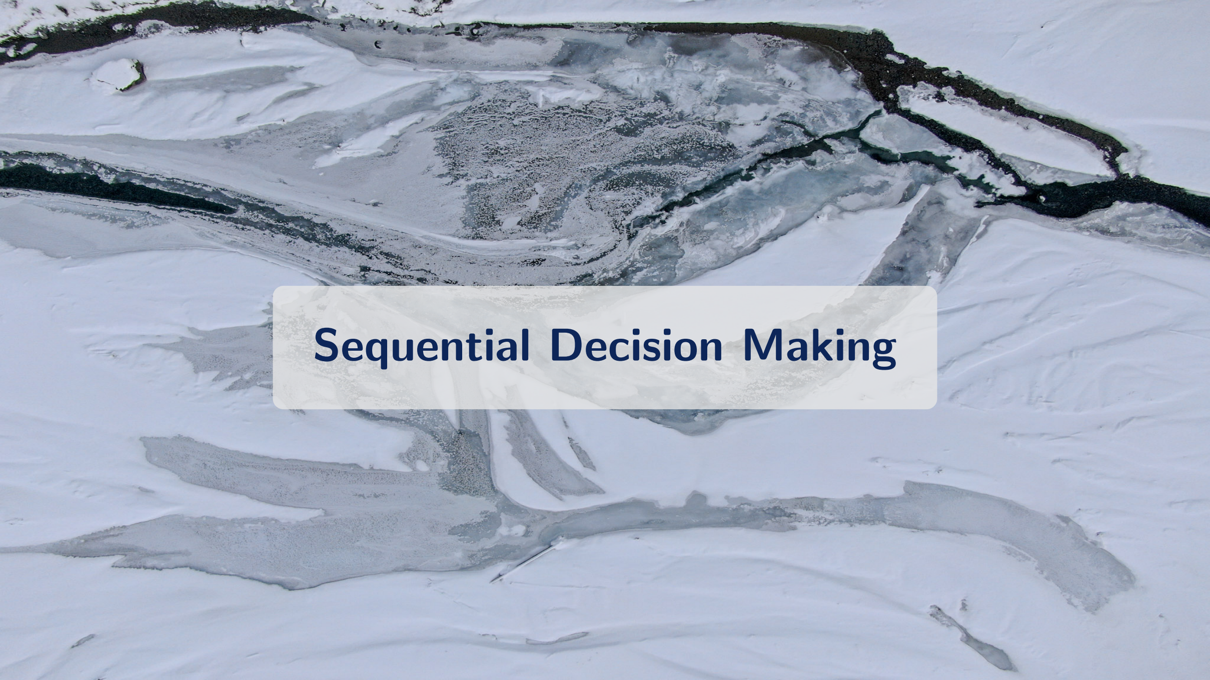
Modern Algorithmic Game Theory

Martin Schmid
Department of Applied Mathematics
Charles University in Prague

November 19, 2025



Sequential Decision Making

Extensive-Form Games

- We will model sequential decision making as an **extensive-form game**
- EFGs use a tree-like structure similar to perfect information game trees
- However, we need to deal with imperfect information, e.g. a player cannot see their opponent's hand in Poker
- Consider these two situations in Heads Up Texas Hold'em Poker:
  - Player 1 has $(A\spadesuit, 8\diamondsuit)$ and Player 2 has $(K\heartsuit, K\diamondsuit)$
  - Player 1 has $(A\spadesuit, 8\diamondsuit)$ and Player 2 has $(2\heartsuit, 7\heartsuit)$
- Even though these two situations are different, Player 1 cannot distinguish them
- We will model these situations by combining groups of indistinguishable states from each player's point of view
- This forces each player to use the same strategy in each state they cannot tell apart

Extensive-Form Games Formalization

Extensive-form game is a tuple consisting of:

- a finite set of players $\mathcal{N} = \{1, 2, \ldots, n\}$
- a finite set of action sequences $\mathcal{H}$. Each sequence $h \in \mathcal{H}$ is called a **history**. The empty sequence $\emptyset$ is in $\mathcal{H}$ and every prefix of a history is also a history, i.e. $(h, a) \in \mathcal{H} \implies h \in \mathcal{H}$. $h'a \sqsubseteq h$ denotes that h' is a prefix of h
- $\mathcal{Z} \subseteq \mathcal{H}$ is the set of terminal histories
- a finite set of actions available in every non-terminal history $\mathcal{A}(h) = \{a : (h, a) \in \mathcal{H}\}$
- a function $p \colon \mathcal{H} \setminus \mathcal{Z} \to \mathcal{N} \cup \{c\}$ that assigns each non-terminal history $h \in \mathcal{H} \setminus \mathcal{Z}$ an **acting player** (either $i \in \mathcal{N}$ or the chance player c)
- a function f_c that assigns every history $h \colon p(h) = c$, a probability measure over $\mathcal{A}(h)$
- a partition $\mathcal{S}_i$ of histories $h \in \mathcal{H} \colon p(h) = i$ where player i is to act. A set $s_i \in \mathcal{S}_i$ is an **information state** of player i and $\mathcal{H}(s_i)$ is a set of histories belonging to s_i
- a **utility function** $u_i \colon \mathcal{Z} \to \mathbb{R}$ for every player $i \in \mathcal{N}$

Extensive-Form Tree Example

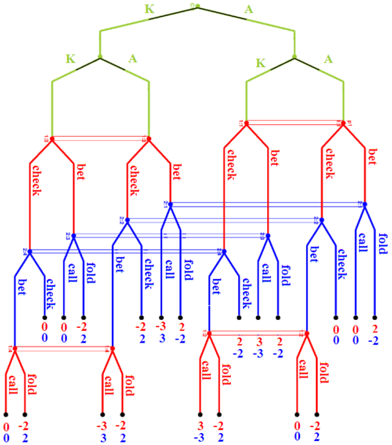
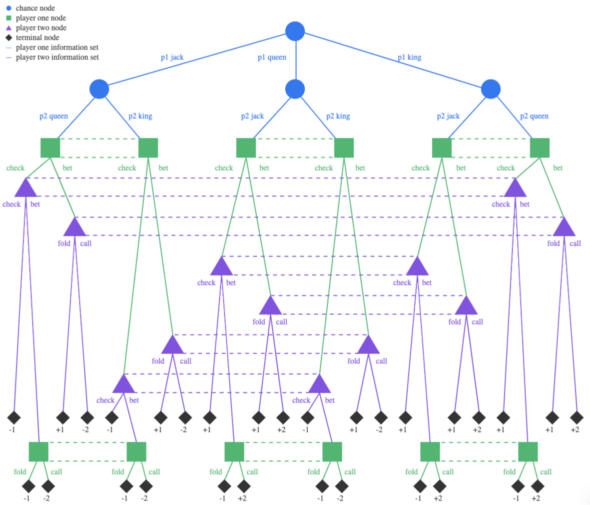


Figure: An game tree of a simple Poker-like game

Kuhn Poker Game Tree



Kuhn Poker Formalization Example

- $\mathcal{N} = \{1, 2\}$, plus the chance player denoted by c
- $\mathcal{H} = \{(\emptyset), (\mathtt{J}), (\mathtt{J}, \mathtt{Q}), (\mathtt{J}, \mathtt{Q}, \mathtt{bet}), (\mathtt{J}, \mathtt{Q}, \mathtt{bet}, \mathtt{call}), \dots\}$, $|\mathcal{H}| = 58$
- $\mathcal{Z} \subseteq \mathcal{H}$, $\mathcal{Z} = \{(\mathtt{J}, \mathtt{Q}, \mathtt{bet}, \mathtt{call}), (\mathtt{Q}, \mathtt{J}, \mathtt{bet}, \mathtt{fold}), \dots\}$, $|\mathcal{Z}| = 30$
- $\mathcal{A}(\mathtt{K}, \mathtt{J}) = \{\mathtt{bet}, \mathtt{fold}\}$, $\mathcal{A}(\mathtt{K}, \mathtt{J}, \mathtt{bet}) = \{\mathtt{call}, \mathtt{fold}\}$
- $p(\emptyset) = c$, $p(\mathtt{K}) = c$, $p(\mathtt{K}, \mathtt{J}) = 1$, $p(\mathtt{K}, \mathtt{J}, \mathtt{check}) = 2$
- $f_c(\emptyset) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, $f_c(\mathtt{J}) = (0, \frac{1}{2}, \frac{1}{2})$
- $\mathcal{S}_1 = \{\{(\mathtt{K}, \mathtt{J}), (\mathtt{K}, \mathtt{Q})\}, \{(\mathtt{K}, \mathtt{J}, \mathtt{check}), (\mathtt{K}, \mathtt{Q}, \mathtt{check})\}, \dots\}$, $|\mathcal{S}_1| = 6$
  $\mathcal{S}_2 = \{\{(\mathtt{J}, \mathtt{K}), (\mathtt{Q}, \mathtt{K})\}, \{(\mathtt{J}, \mathtt{K}, \mathtt{check}), (\mathtt{Q}, \mathtt{K}, \mathtt{check})\}, \dots\}$, $|\mathcal{S}_2| = 6$
- $u_1(\mathtt{K}, \mathtt{J}, \mathtt{bet}, \mathtt{call}) = 2$, $u_2(\mathtt{K}, \mathtt{Q}, \mathtt{bet}, \mathtt{fold}) = -1$

Perfect Recall

- Extensive-form games is a powerful model of sequential decision making
- Unfortunately, it allows us to capture some non-realistic properties when we allow grouping arbitrary histories into information states, e.g. a history and its prefix both being in the same information state
- For example, rational players do not forget information they already know, e.g. what actions they played in previous rounds
- **Perfect recall** formalizes this property; we say that an extensive-form game satisfies perfect recall, if all players can recall their previous actions and the corresponding information states
- This is a critical property as in games with imperfect recall even basic operations (e.g. best response calculation) become hard

Strategies

- In extensive-form games players do not choose a row/column, but instead an edge in the game tree
- Since we need to ensure the players cannot distinguish states merged into information sets, we allow the players to choose an action only in information states in contrast to histories/states
- This way, the players must play the same strategy in all histories grouped in an information state
- We distinguish among three types of strategies in extensive-form games – **pure**, **mixed** and **behavioral**

Strategies

- A **pure** strategy is a complete plan of actions, one for each possible information state in a game, chosen **once** at the start of a game
- A **mixed** strategy is a probability distribution over pure strategies
- A **behavioral** strategy is a mapping from an information state to a distribution over the available actions $s_i \to \Delta(\mathcal{A}(s_i))$
- In other words, a behavioral strategy is a collection of independent probability distributions $\pi_i(s_i)$ over actions $\mathcal{A}(s_i)$, one for each information state s_i
- In games with perfect recall, it can be shown that mixed and behavioral strategies are **realization equivalent**, i.e. any outcome that is achievable using one strategy is also achievable using the other and vice versa
- This equivalence is known as **Kuhn's theorem**

Reach Probabilities

- Given a strategy profile π, we define the **reach probability of a history** $h \in \mathcal{H}$ as

$$P^{\pi}(h) = \prod_{h'a \sqsubseteq h} \pi(h', a)$$

- $P^{\pi}(h)$ can also be rewritten as a product of individual player's contribution $P_i^{\pi}(h)$
- Naturally, the **reach probability of an information state** $s_i \in \mathcal{S}_i$ is defined as

$$P^{\pi}(s) = \sum_{h \in \mathcal{H}(s_i)} P^{\pi}(h)$$

- Note that, for brevity, we overload the notation $P^{\pi}(\cdot)$ for both the histories and information states. The same applies to $\pi(\cdot, a)$.

Strategy Evaluation

- Expected utility $u_i(\pi)$ of player i given a strategy profile π is a weighted sum over the terminal histories $h \in \mathcal{Z}$
- Mathematically, this is defined as

$$u_i(\pi) = \sum_{h \in \mathcal{Z}} P^\pi(h) u_i(h)$$

- Similarly to reinforcement learning, we define the **history value function** $v_i^\pi(h)$ as the expected utility of player i when starting in $h \in \mathcal{H}$ and following strategy π
- The **history-action value function** $q_i^\pi(h, a)$ is defined analogously, except that player i first takes action $a \in \mathcal{A}(h)$ and then follows their policy π

State-Value & Action-Value Functions

- Similarly to the reach probability of an information state, the value and action value functions of an information state are defined as sums over its histories
- However, compared to reinforcement learning, these values are not conditional on reaching a particular state, but instead depend on the reach probabilities
- The **state** value $v_i^\pi(s_i)$ of an information state $s_i \in \mathcal{S}_i$ is defined as

$$v_i^\pi(s) = \sum_{h \in \mathcal{H}(s)} P^\pi(h) v_i^\pi(h)$$

- Similarly, the **state-action** value $q_i^\pi(s_i, a)$ of an information state $s_i \in \mathcal{S}_i$ and an action $a \in \mathcal{A}(s_i)$ is defined as

$$q_i^\pi(s, a) = \sum_{h \in \mathcal{H}(s)} P^\pi(h) q_i^\pi(h, a)$$

Normal-Form vs. Extensive-Form

Normal-Form to Extensive-Form Games

- We can convert any normal-form game to an equivalent extensive-form game
- We simply let one player act first and group all subsequent histories into a single information state so that the opponent doesn't know what move the first player made

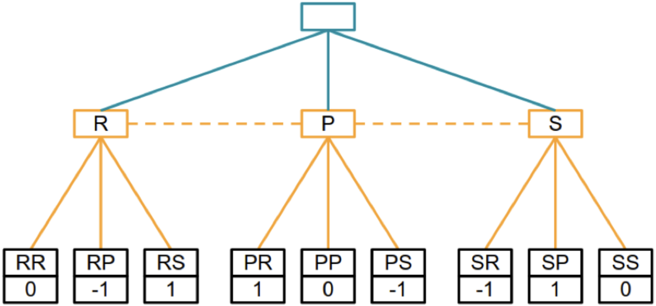
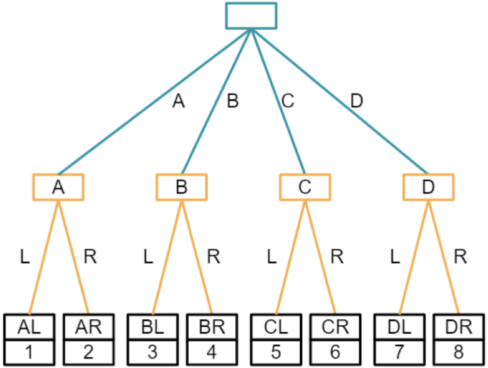


Figure: Sequential Rock-Paper-Scissors

Extensive-Form to Normal-Form Games

- Any extensive-form game that satisfies perfect recall, can be converted to an equivalent normal-form game
- Unfortunately, the resulting normal-form game can be exponentially larger than the original extensive-form game
- The set of each player's actions in the resulting normal-form game consists of all **pure strategies** in the original game
- The utility matrix consists of **expected payoffs** of the corresponding pure strategies in the original game

Extensive-Form to Normal-Form Games



	A	B	C	D
AL-BL-CL-DL	1	3	5	7
AL-BL-CL-DR	1	3	5	8
AL-BL-CR-DL	1	3	6	7
AL-BL-CR-DR	1	3	6	8
AL-BR-CL-DL	1	4	5	7
AL-BR-CL-DR	1	4	5	8
AL-BR-CR-DL	1	4	6	7
AL-BR-CR-DR	1	4	6	8
AR-BL-CL-DL	2	3	5	7
AR-BL-CL-DR	2	3	5	8
AR-BL-CR-DL	2	3	6	7
AR-BL-CR-DR	2	3	6	8
AR-BR-CL-DL	2	4	5	7
AR-BR-CL-DR	2	4	5	8
AR-BR-CR-DL	2	4	6	7
AR-BR-CR-DR	2	4	6	8

Extensive-Form to Normal-Form Games

- As any extensive-form game with perfect recall can be converted to an equivalent normal-form game, all theorems and results that hold in normal-form games also hold in extensive-form games
- Namely,
  - the existence of Nash equilibria,
  - the existence of a pure best response,
  - the collapse of Nash equilibria and Maximin strategies in zero-sum games,
  - not-so-nice properties of other games, such as the multiple equilibria problem
- For small enough games (think Kuhn Poker) we can even use the algorithms that we discussed in the previous lectures to find (converge to) Nash equilibria in these games

Extensive-Form Fictitious Play

Extensive-Form Fictitious Play

- A generalization of normal-form Fictitious Play to extensive-form games
- First introduced by Johannes Heinrich, Marc Lanctot and David Silver in a paper named *Fictitious Self-Play in Extensive-Form Games*
- It follows the same structure as normal-form Fictitious Play where in each iteration:
  - We compute a best response to the average strategy of our opponent
  - We add the computed best response to the running average strategy
- However, each of these two steps is more involved than its normal-form counterpart

Best Response Computation

- Just like in perfect information games, we traverse the tree bottom-up and greedily select the action with the highest value, while propagating the values up the tree.
- However, in imperfect information games, action values in an information state depend on action values of each history in that particular information state
- Furthermore, these histories need to be weighted by their reach probabilities $P^{\pi}(h)$ which require knowing strategies of **all** players even in the preceding states
- How can we know our best-response strategy up the tree if we are computing it bottom-up?
- Perfect recall guarantees that our contribution to $P^{\pi}(h)$ is the same for all $h \in \mathcal{H}(s)$! Mathematically, $P_i^{\pi}(h) = P_i^{\pi}(h') \; \forall h, h' \in \mathcal{H}(s)$
- Thus, when computing a best response, we can weigh the history-action values only using opponents' policies π_{-i}

Counterfactual Reach Probability

- The notion of excluding one player's policy when computing a reach probability leads to the notion of a **counterfactual reach probability**
- Given a strategy profile π, we define the counterfactual reach probability of history $h \in \mathcal{H}$ as

$$P^{\pi}_{-i}(h) = \prod_{j \in \mathcal{N} \setminus \{i\}} P^{\pi}_{j}(h) = \prod_{h'a \sqsubseteq h \colon p(h') \neq i} \pi(h', a)$$

- It is the probability of reaching history h when player i **attempts** to reach that particular history and all other players stick to their strategies.
- In other words, at every history h' that is a prefix of history h, where player i has to act, they place all of its probability mass on the particular action a that is on the path to h, i.e. $\pi_i(h', a) = 1$, where $h'a \sqsubseteq h$ and $p(h') = i$.

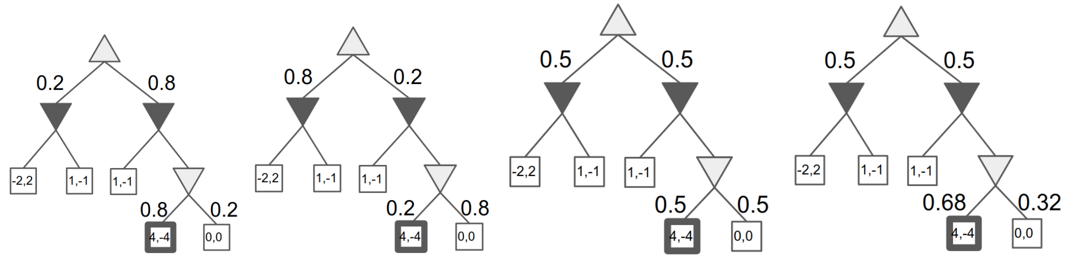
Best Response Computation

- We have now seen that we only need to consider our opponent's strategies when computing a best response against them
- Thus, starting at the leaf nodes and moving upwards until we reach the root node, we will compute the counterfactually-weighted action values in each information state, choose the action maximizing the action values and propagating its value up the tree
- $\sum_{h \in \mathcal{H}(s)} P^{\pi}_{-i}(h) q^{\pi}_{i}(h, a)$

Policy Averaging

- Averaging in the space of behavioral strategies is not as straightforward and we need to exercise caution
- A common mistake is to simply average the strategies locally, in each individual information state
- This is incorrect as for $\pi_1^c = 0.5\pi_1^a + 0.5\pi_1^b$, we expect $u_1(\pi_1^c, \pi_2) = 0.5u_1(\pi_1^a, \pi_2) + 0.5u_1(\pi_1^b, \pi_2)$
- The issue is due to the dynamics being inherently sequential — the distribution over the terminal states $P^\pi(z)$ depends on the full sequence of actions leading to z
- Proper averaging in the behavioral strategy space must take into account reach probabilities of each information state

Policy Averaging Example



(a) π_1^a (b) π_1^b (c) π_1^c - naive averaging (d) π_1^c - proper averaging

Figure: Consider the highlighted state s playing against an opponent who plays to reach that state, and its reach probability $P^{\pi}(s)$. a) $P^{\pi_1^a}(s) = 0.64$ b) $P^{\pi_1^b}(s) = 0.04$ c) Naive strategy averaging simply averages the strategies in isolation: $P^{\pi_1^c}(s) = 0.25 \neq 0.5 \cdot 0.64 + 0.5 \cdot 0.04$ d) Proper strategy averaging takes reach probabilities into consideration

Week 7 Homework

You can find more detailed descriptions of homework tasks in the GitHub repository.

1. Strategy profile evaluation
2. Best response calculation
3. Strategy averaging
4. Extensive-form Fictitious play