6.S890: Topics in Multiagent Learning

Lecture 12 – Prof. Farina

Introduction to Extensive-Form Games

Fall 2023

Extensive-Form Game

- Games played on a game tree (think chess, go, poker, monopoly, Avalon, Liar’s dice, …)
- Stochastic moves are allowed (random draws of cards, random roll of dice, random arrivals, …)

We will be mostly interested in the general case of **imperfect-information** games

(i.e., certain moves or stochastic events are only observed by a subset of players)

Difficulties with Extensive-Form Games

Compared to normal-form games, imperfect-information extensive-form games bring many conceptual challenges

1. The number of (deterministic) strategies grows exponentially in the game tree

2. Imperfect information makes backward induction and local reasoning not viable

   *General principle: you need to think about what the opponents don't know about you and leverage that to your advantage. Sometimes that means **bluffing**, to not reveal private information.*

3. Other players have control over what part of the game tree is visited/explored

- Nonetheless: many positive results 🎉

Imperfect-Information Extensive-Form Games

How it started:

H. W. Kuhn[1]

1950

A fascinating problem for the game theoretician is posed by the common card game, Poker. While generally regarded as partaking of psycho-logical aspects (such as bluffing) which supposedly render it inaccessible to mathematical treatment, it is evident that Poker falls within the general theory of games as elaborated by von Neumann and Morgenstern [1]. Relevant probability problems have been considered by Borel and Ville [2] and several variants are examined by von Neumann [1] and by Bellman and Blackwell [3].

As actually played, Poker is far too complex a game to permit a complete analysis at present; however, this complexity is computational and

How it's going:

Science — Current Issue | First release papers | Archive | About | Submit manus

HOME > SCIENCE > VOL. 359, NO. 6374 > SUPERHUMAN AI FOR HEADS-UP NO-LIMIT POKER: LIBRATUS BEATS TOP PROFESSIONALS

RESEARCH ARTICLE

Superhuman AI for heads-up no-limit poker: Libratus beats top professionals

NOAM BROWN AND TUOMAS SANDHOLM — Authors Info & Affiliations

SCIENCE • 17 Dec 2017 • Vol 359, Issue 6374 • pp. 418-424 • DOI: 10.1126/science.aao1733

4,596 | 149

2017

How Extensive-Form Games Are Drawn

Example (Kuhn poker).

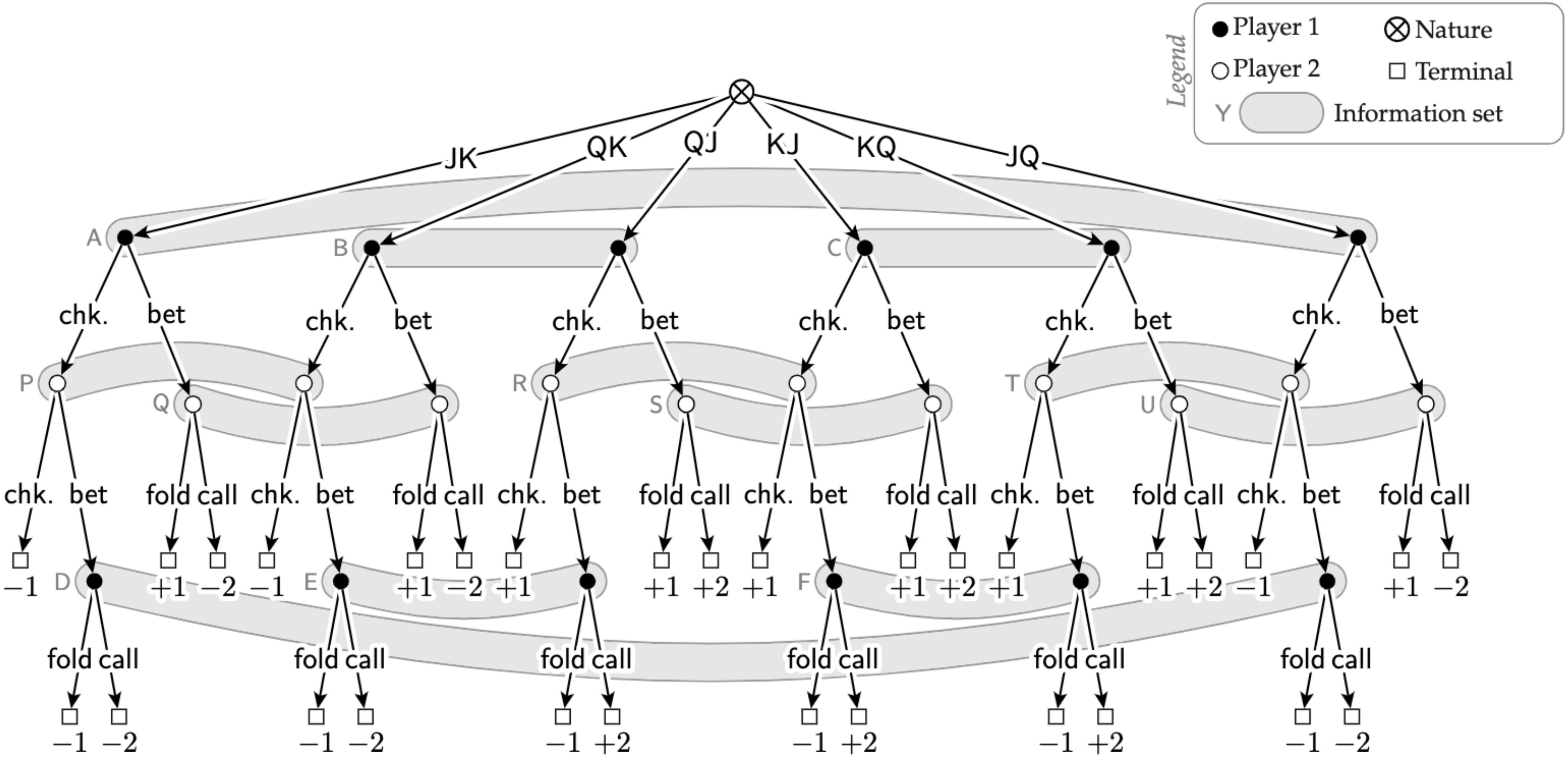
In Kuhn poker, each player puts an ante worth $1 into the pot. Each player is then privately dealt one card from a deck that contains 3 unique cards (Jack, Queen, King). Then, a single round of betting then occurs, with the following dynamics. First, Player 1 decides to either check or bet $1.

Then,

- If Player 1 checks, Player 2 can check or bet another $1 after matching the pot.
    - If Player 2 checks, a showdown occurs; if Player 2 bets, Player 1 can fold or call.
        - If Player 1 folds, Player 2 takes the pot; if Player 1 calls, a showdown occurs.
- If Player 1 bets, Player 2 can fold or call the bet by matching the pot.
    - If Player 2 folds, Player 1 takes the pot; if Player 2 calls, a showdown occurs.

When a showdown occurs, the player with the higher card wins the pot and the game immediately ends

How Extensive-Form Games Are Drawn



As noted by Kuhn himself, even the previous small game already captures central aspects of deceptive behavior

The presence of bluffing and underbidding in these solutions is noteworthy (bluffing means betting with a J ; underbidding means passing on a K). All but the extreme strategies for player I, in terms of the behavior parameters, involve both bluffing and underbidding while player II's single optimal strategy instructs him to bluff with constant probability 1/3 (underbidding is not available to him). These results compare

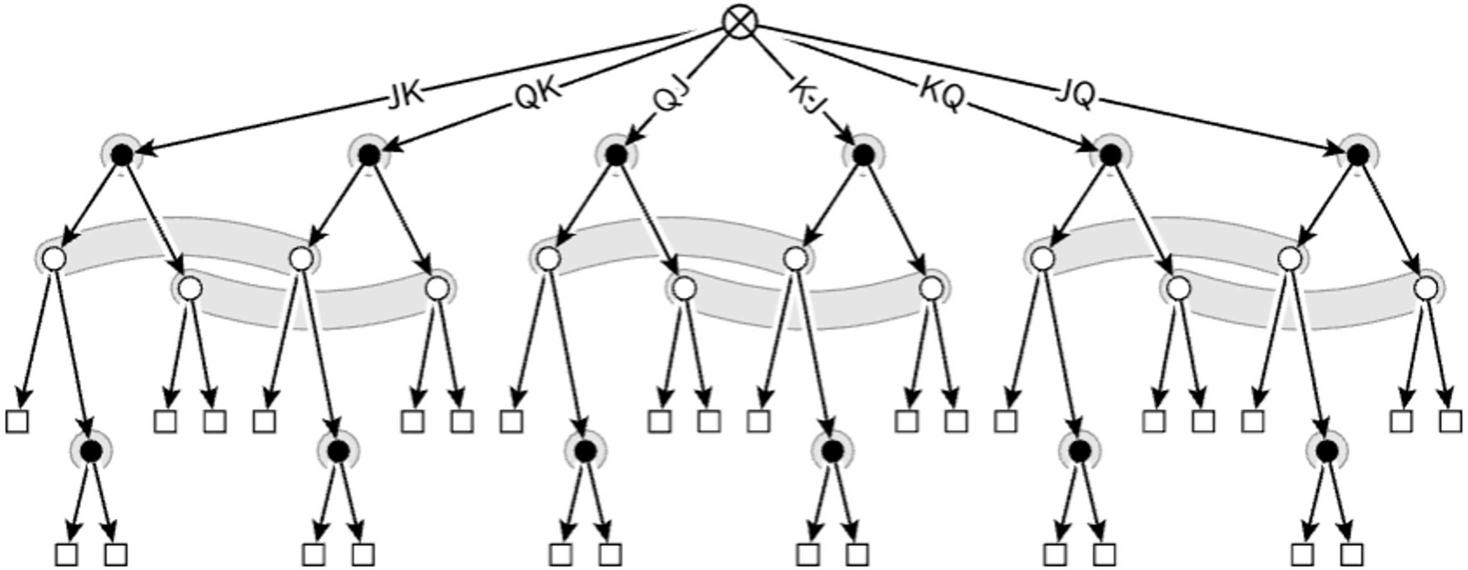
A Bit of Nomenclature

- The nodes of the game tree are often called **histories** (will be denoted with letter h)
- The collection of information sets for a given player is called the **information partition** of the player
- The game has **perfect information** if all information sets are singleton

The structure of Information

■ **First variation**

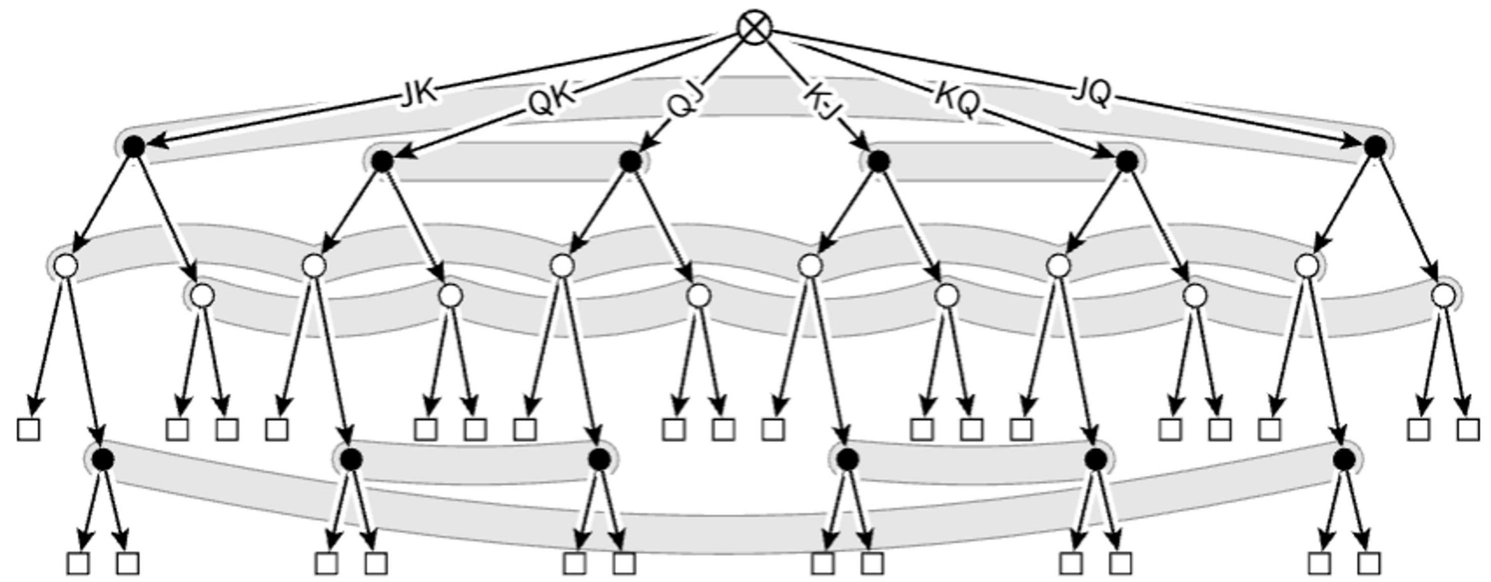
Player 1 is revealed the private card of Player 2 by the dealer.



The structure of Information

■ **Second variation**

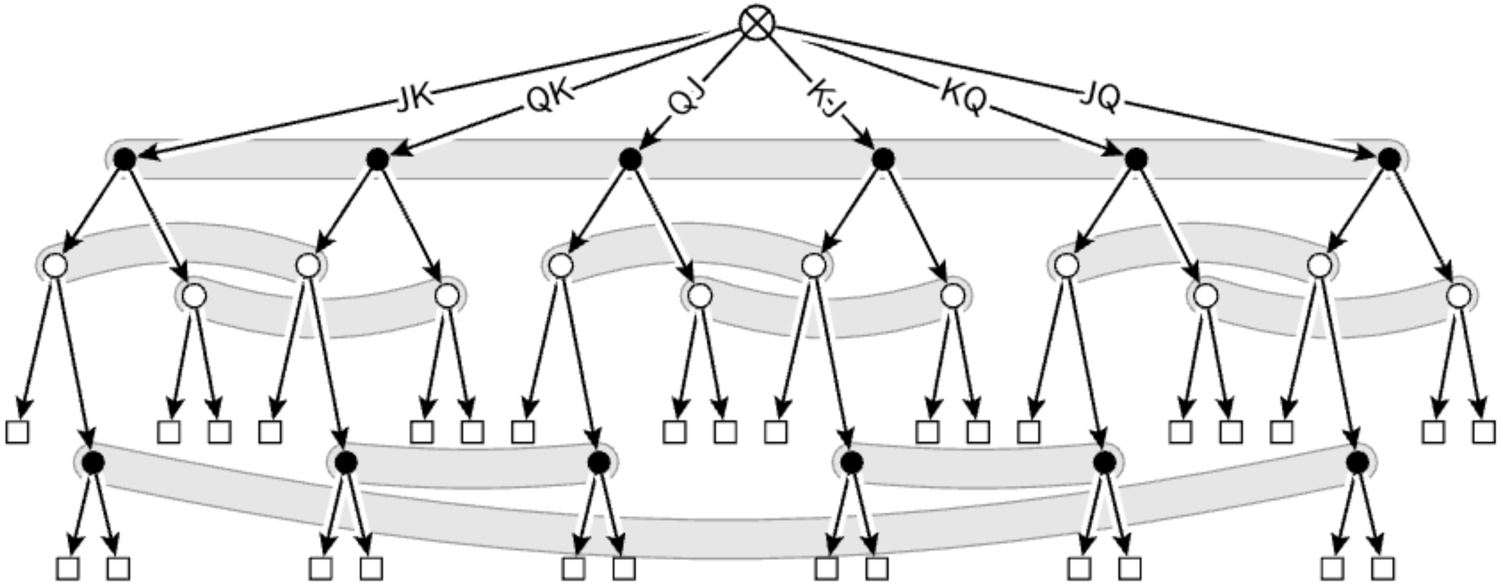
Player 2 does not get to observe her private card.



The structure of Information

■ **Third variation**

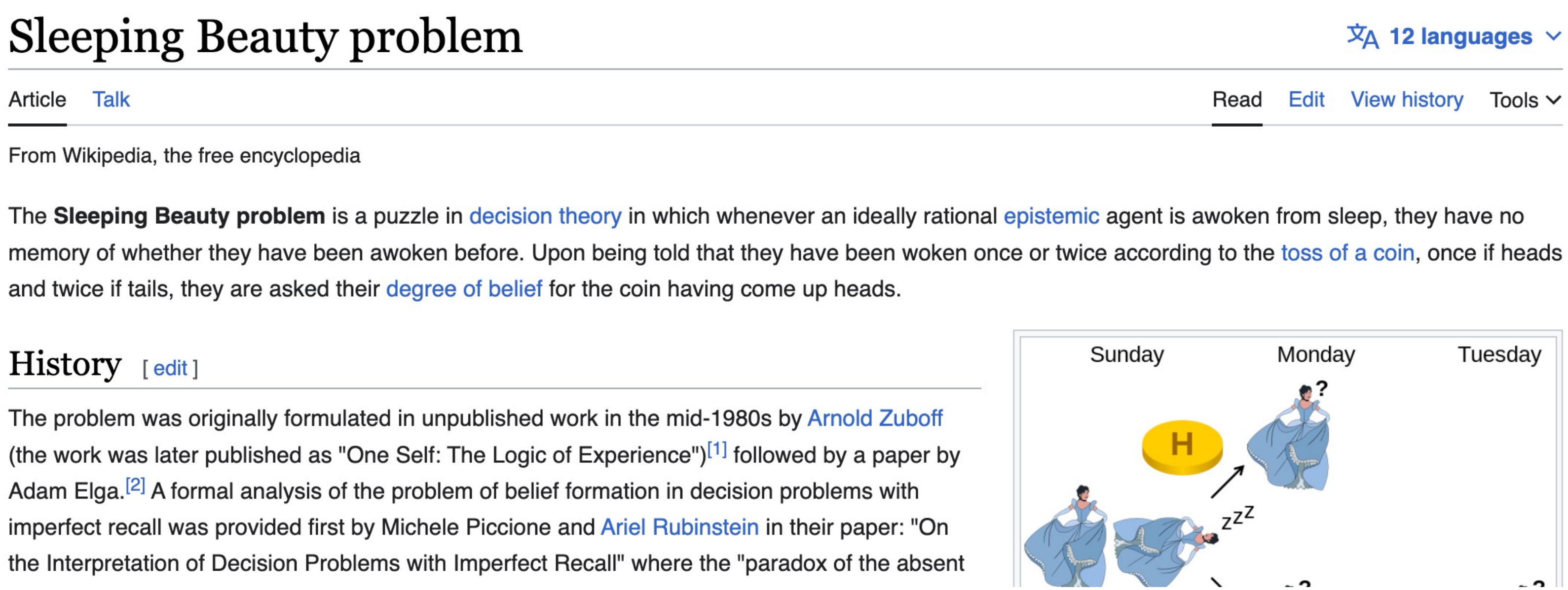
Player 1 is allowed to look at his private card only if he decides to check.



Perfect vs Imperfect Recall

Perfect Recall: information sets satisfy the fact that that no player forgets about their actions, and about information once acquired

🚨 **Danger zone™:** unexpected things happen when trying to formalize optimal strategies in the presence of imperfect recall


Sleeping Beauty problem

文A 12 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

The **Sleeping Beauty problem** is a puzzle in decision theory in which whenever an ideally rational epistemic agent is awoken from sleep, they have no memory of whether they have been awoken before. Upon being told that they have been woken once or twice according to the toss of a coin, once if heads and twice if tails, they are asked their degree of belief for the coin having come up heads.

History [edit]

The problem was originally formulated in unpublished work in the mid-1980s by Arnold Zuboff (the work was later published as "One Self: The Logic of Experience")[1] followed by a paper by Adam Elga.[2] A formal analysis of the problem of belief formation in decision problems with imperfect recall was provided first by Michele Piccione and Ariel Rubinstein in their paper: "On the Interpretation of Decision Problems with Imperfect Recall" where the "paradox of the absent

Perfect vs Imperfect Recall

Perfect Recall: information sets satisfy the fact that that no player forgets about their actions, and about information once acquired

More formally:

> A player $i \in [\![n]\!]$ is said to have *perfect recall* if, for any information set $I \in \mathcal{I}_i$, for any two histories $h, h' \in I$ the sequence of Player i's actions encountered along the path from the root to h and from the root to h' must coincide (or otherwise Player i would be able to distinguish among the histories, since the player remembers all of the actions they played in the past). The game is perfect recall if all players have perfect recall.

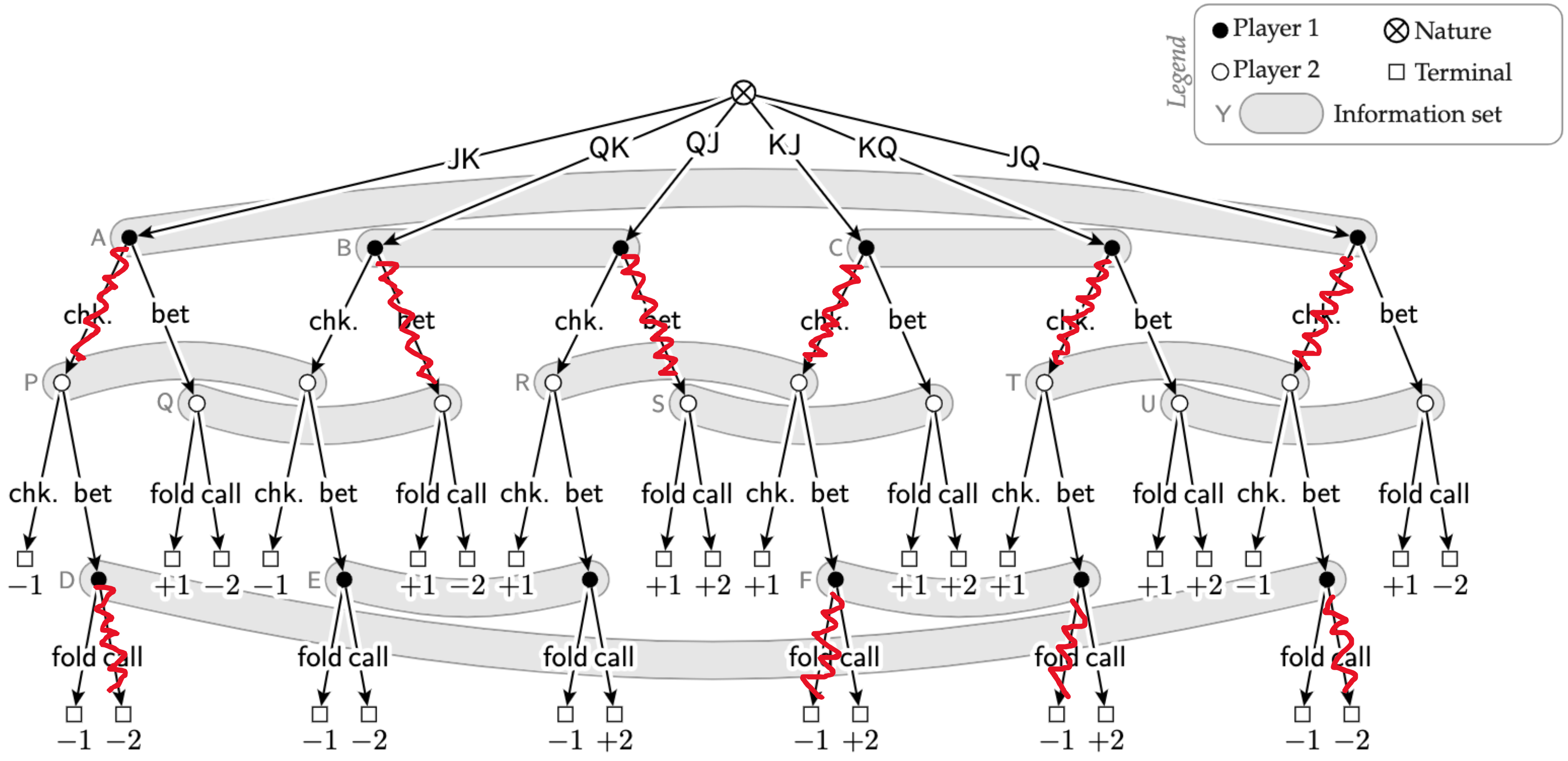
Strategies in Extensive-Form Games

Approach 1: Convert to Normal-Form Game
(aka "reduced normal-form representation")

Approach 2: The RL way: "Behavioral Strategies"

Strategic Form

Idea: Strategy = randomize a deterministic contingency plan



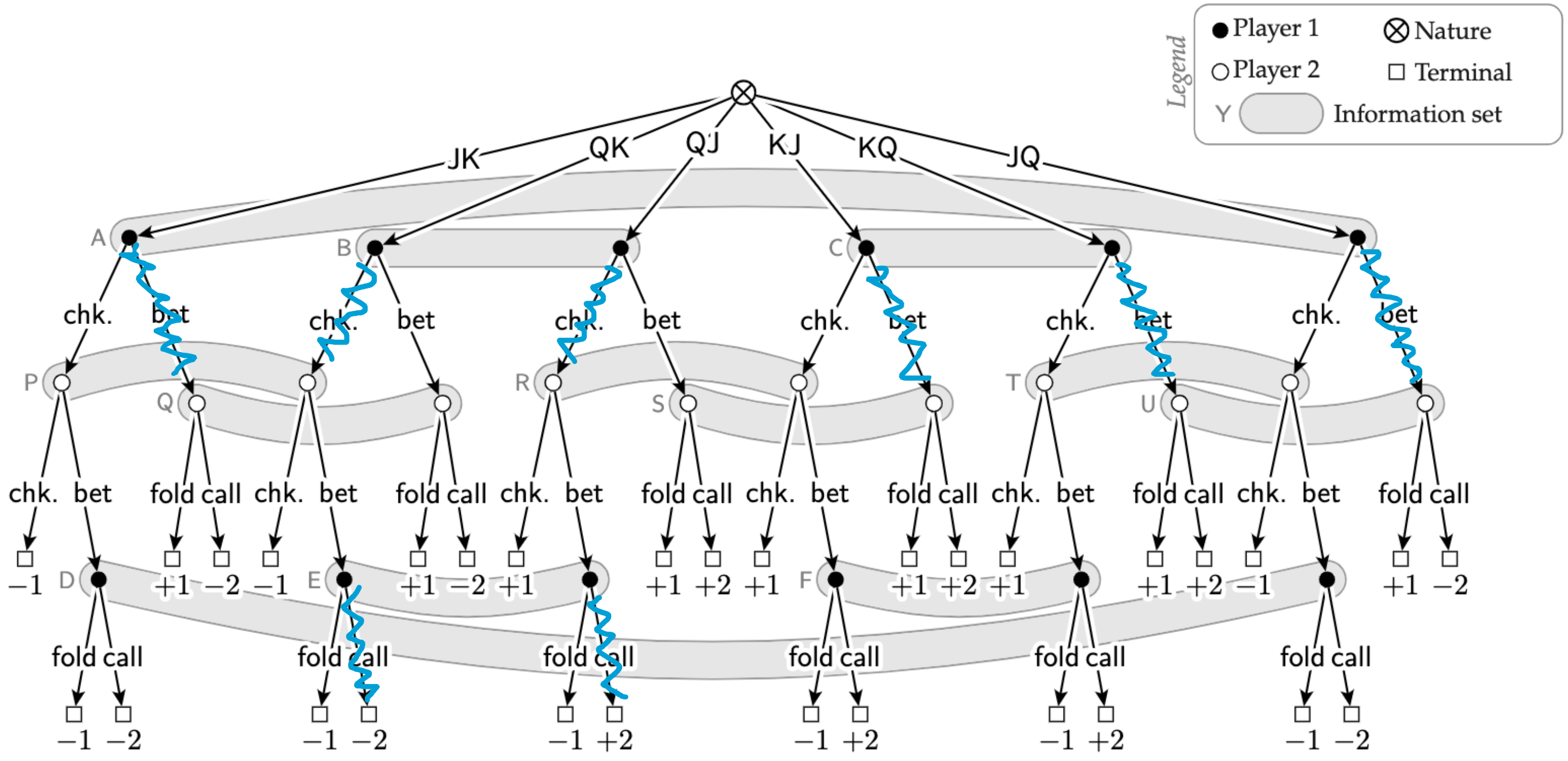
Each player constructs a list of all possible assignments of actions at each information set

Histories in the same information must get assigned the same action

No need to specify actions at histories that are for sure unreachable

Strategic Form

Idea: Strategy = randomize a deterministic contingency plan

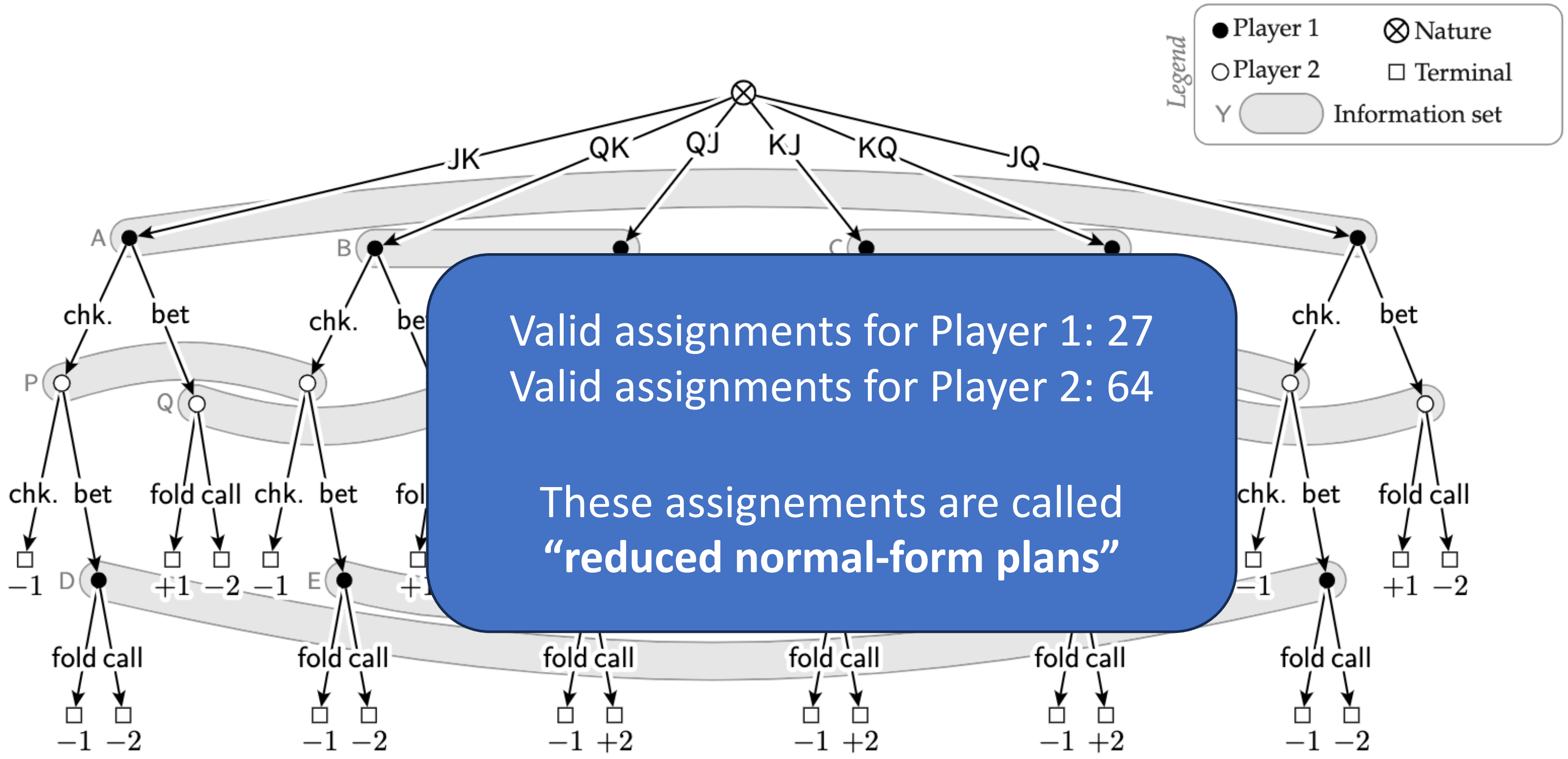


Each player constructs a list of all possible assignments of actions at each information set

(Histories in the same information must get assigned the same action)

Strategic Form

Idea: Strategy = randomize a deterministic contingency plan



Valid assignments for Player 1: 27
Valid assignments for Player 2: 64

These assignements are called **“reduced normal-form plans”**

Each player constructs a list of all possible assignments of actions at each information set

(Histories in the same information must get assigned the same action)

Equivalent Normal-Form Game

Reduced normal-form plans for Player 2

Reduced normal-form plans for Player 1

1/3	0	-1/3	⋯	1/2
0	1/3	0	⋯	0
-1/3	2/3	1/2	⋯	0
⋮	⋮	⋮		⋮
1/2	0	-2/3	⋯	-1/2

(27 x 64 matrix)

Payoff matrix: Each cell contains the expected utility when players use that combination of reduced normal-form plans

Don't forget nature moves

With this, we have reduced the extensive-form game to a normal-form game ("**reduced normal form of the extensive-form game**")

Inherit notions of Nash, correlated equilibrium, coarse correlated equilibrium, …

Equivalent Normal-Form Game

Reduced normal-form plans for Player 2

Reduced normal-form plans for Player 1

1/3	0	-1/3	⋯	1/2
0	1/3	0	⋯	0
-1/3	2/3	1/2	⋯	0
⋮	⋮	⋮		⋮
1/2	0	-2/3	⋯	-1/2

(27 x 64 matrix)

Example: Nash equilibrium in Kuhn poker:

$$\max_{x} \min_{y} x^T A y$$

Distribution over the 27 plans of Player 1

Distribution over the 64 plans of Player 2

Payoff matrix on the left

You can use any technique for normal-form games: learning, linear programming, …

Equivalent Normal-Form Game

Reduced normal-form plans for Player 2

Reduced normal-form plans for Player 1

1/3	0			
0	1/3			
-1/3	2/3			
⋮	⋮			
1/2	0	-2/3	…	-1/2

(27 x 64 matrix)

Big issue: the number of reduced normal-form plans scales exponentially with the game tree size!

This approach is not scalable beyond very small games

We need better techniques

Example: Nash equilibrium in Kuhn ... er:

... $x^T Ay$

Payoff matrix on the left

the 27 plans of Player 1

Distribution over the 64 plans of Player 2

You can use any technique for normal-form games: learning, linear programming, …

Quick Aside

Recent discovery: for certain algorithms, we can actually get around the exponential size and still operate in this exponential representation implicitly via a kernel trick

Specifically, this applies to the multiplicative weights update (MWU) algorithm.

Takeaway

Running MWU on the reduced normal-form representation of an extensive-form game can be done in linear time per iteration in the size of the game tree (as opposed to linear in the number of reduced normal-form plans)

We can use this technique to compute Nash eq. (in two-player zero-sum games) and coarse correlated equilibrium

[Farina et al., 2022] Kernelized Multiplicative Weights for 0/1-Polyhedral Games: Bridging the Gap Between Learning in Extensive-Form and Normal-Form Games

Recap on Normal-Form Strategies

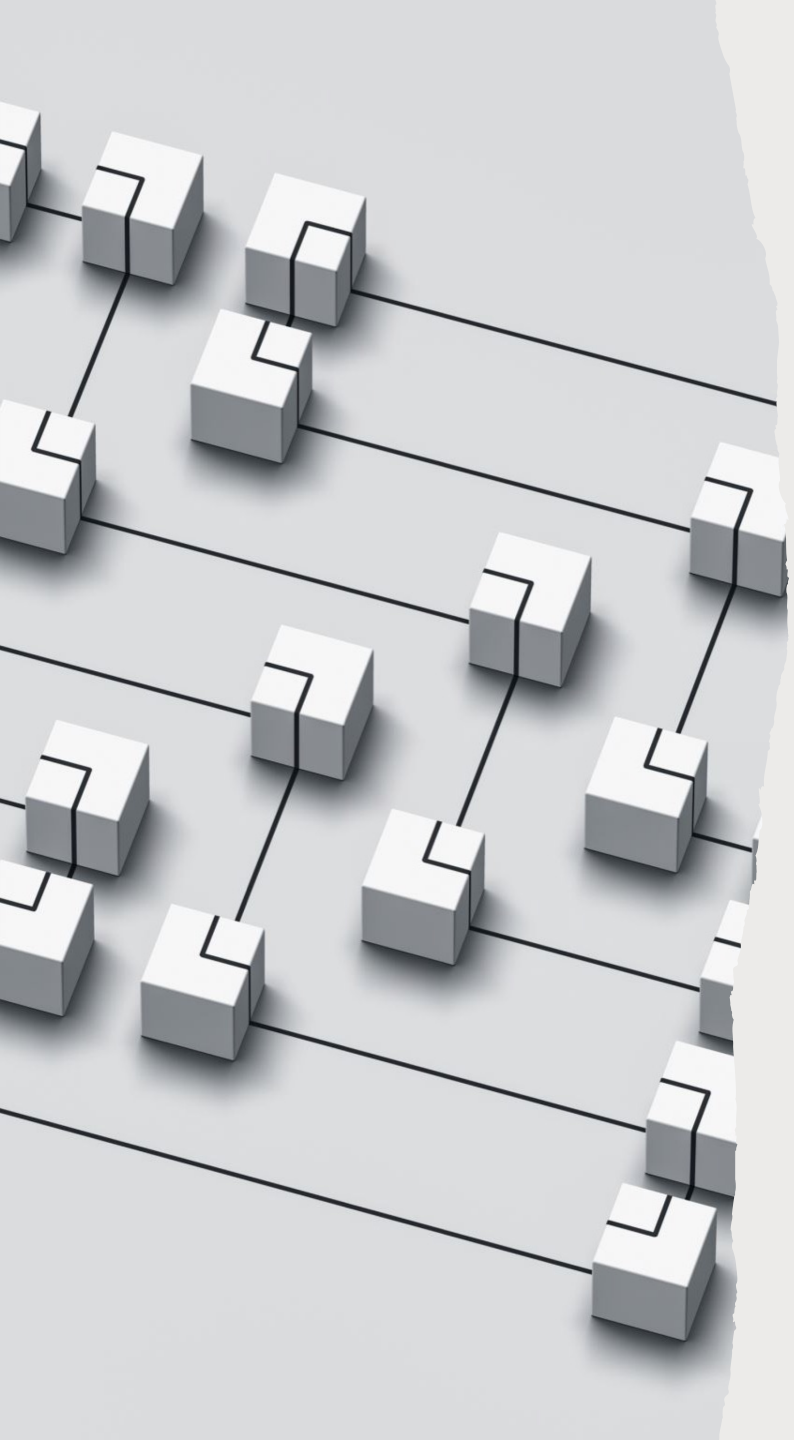
	Idea	Obvious downsides	Good news
(Reduced) Normal-form strategies	Distribution over deterministic strategies $\mu \in \Delta(Plans)$	Exponentially-sized object	In rare cases, it's possible to operate implicitly on the exponential object via a kernel trick

Behavioral Strategies

Idea: Strategy = choice of distribution over available actions at each "decision point"

Information set

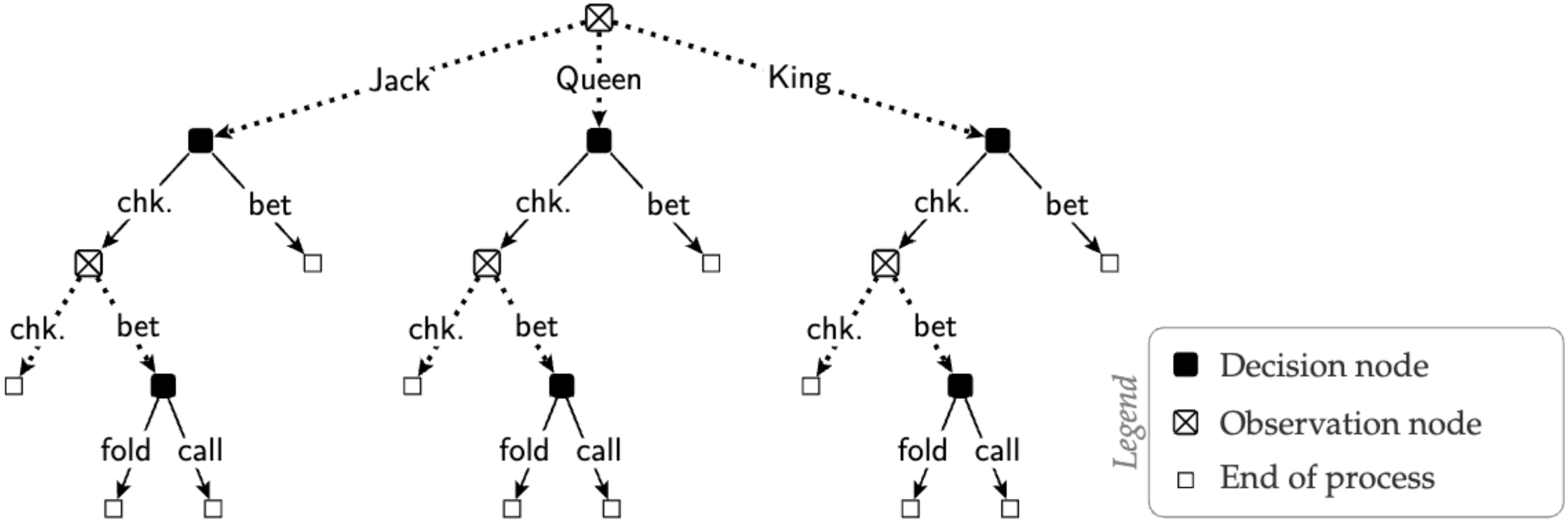
Let's introduce some notation for the tree-form decision process faced by each player…



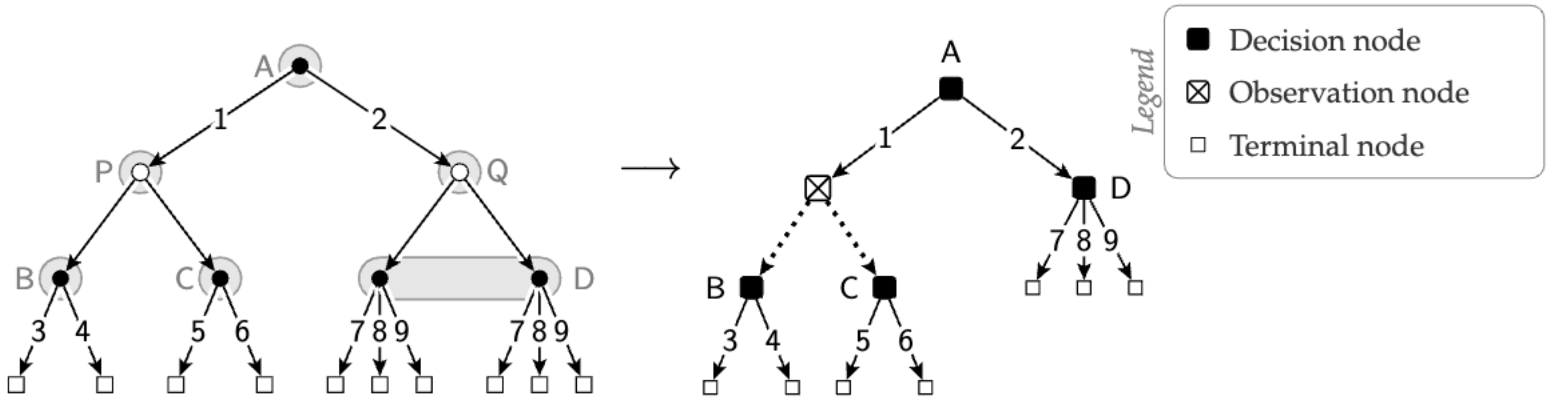
Tree-form Decision Processes

- The **game tree** is a description of the global dynamics of the game, without taking the side of any player in particular
- The problem faced by an individual player is called a tree-form decision process
- TFDP provides a more natural formalism for defining player-specific quantities and procedures, such as strategies and learning algorithms, that inherently refer to the decision space that one player faces while playing the game
- From the point of view of each player, two types of nodes: **decision points** and **observation points**

Example in Kuhn Poker (Player 1)



Another Example



Algorithm for constructing the tree-form decision process of a player:

1. For each information set of the player, construct a corresponding decision node

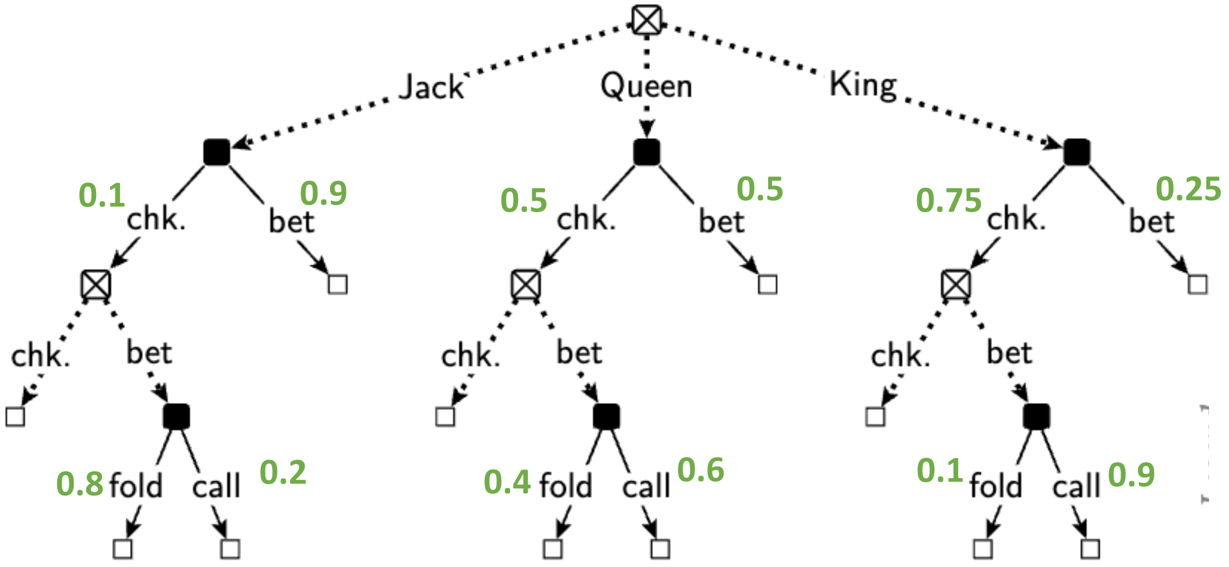
2. The parent of each decision node is the last action of the player on the path from the root of the game tree to any node of the information set

💡 Does not matter which one when the player has perfect recall! (why?)

3. If multiple decision nodes want to have the same parent action, connect with an observation node

Behavioral strategies

Idea: Strategy = choice of distribution over available actions at each **decision point**



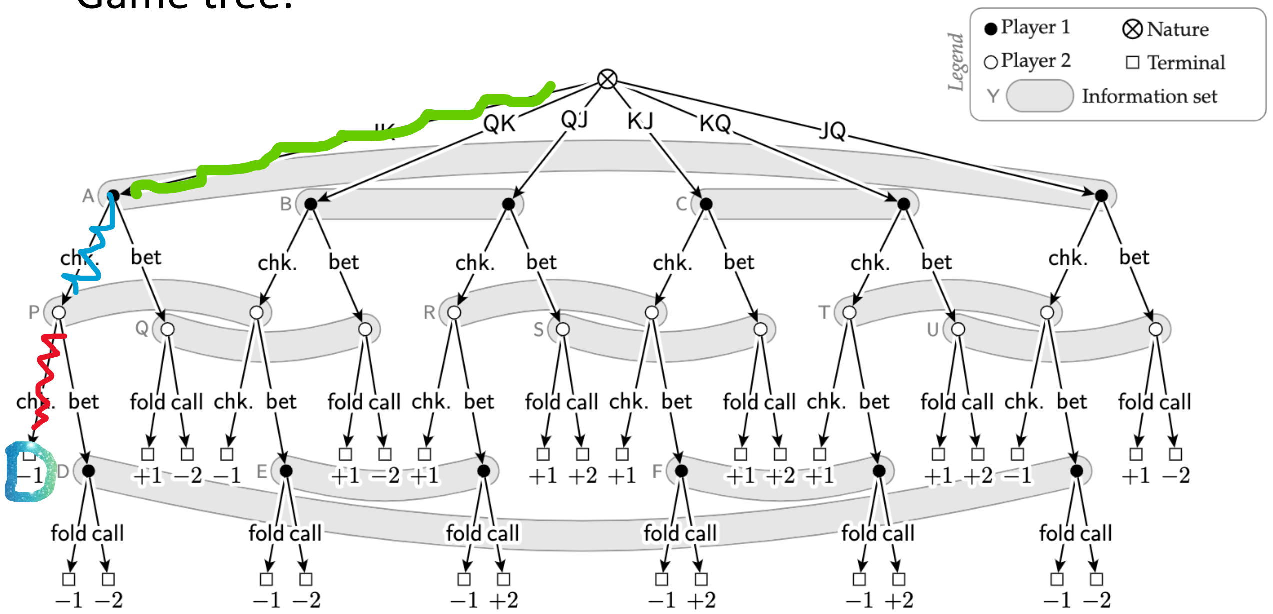
- ✓ Set of strategies is convex
- ✗ Expected utility is **not** linear in this representation

Reason: prob. of reaching a terminal state is **product** of variables

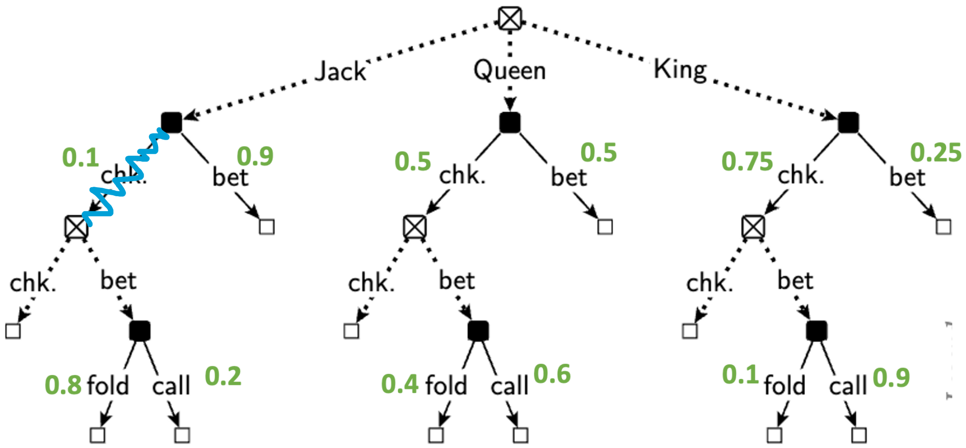
Products = non-convexity 😪

Expected Utility

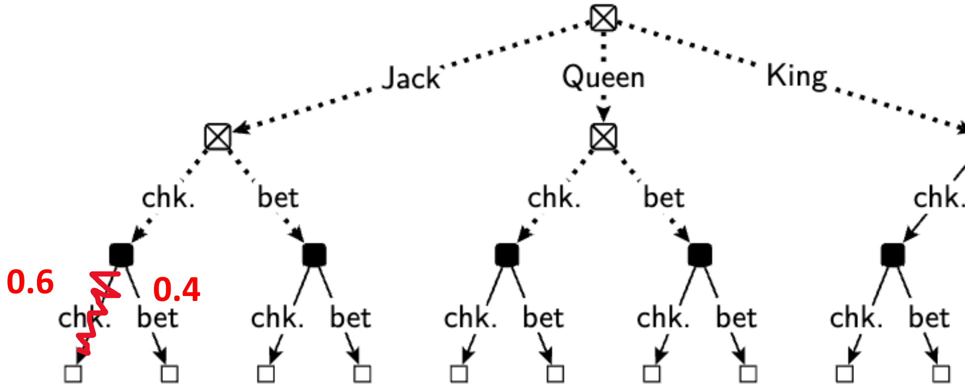
Game tree:



Decision problem and behavioral strategy of Player 1



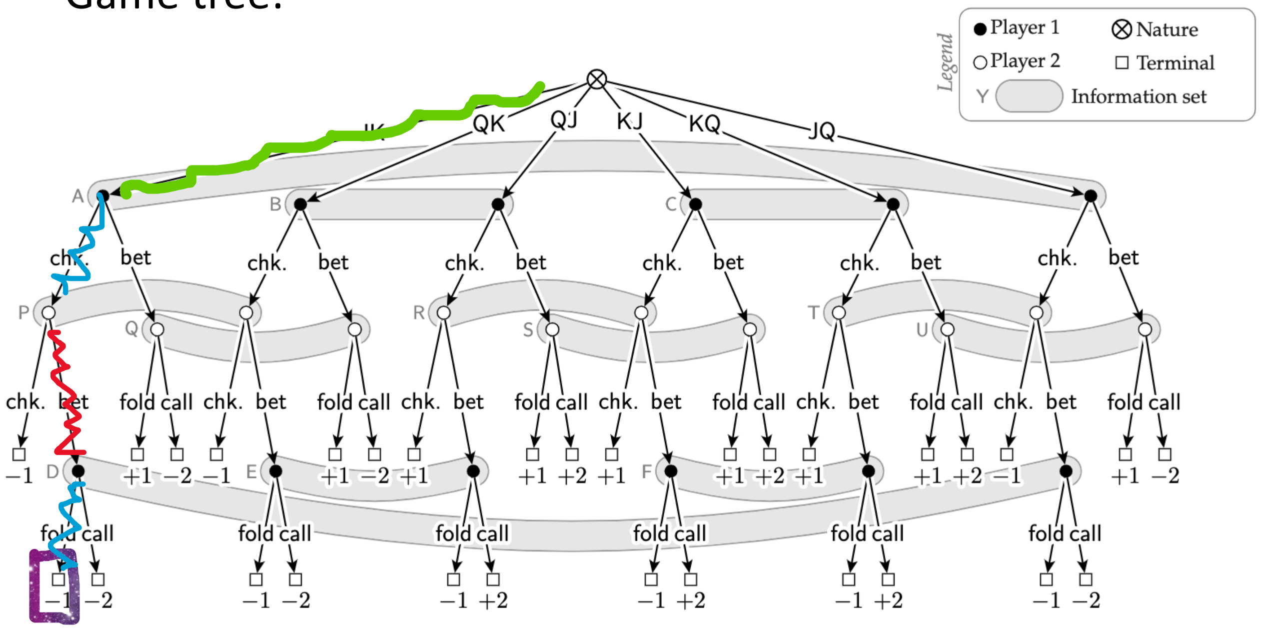
Decision problem and behavioral strategy of Player 2



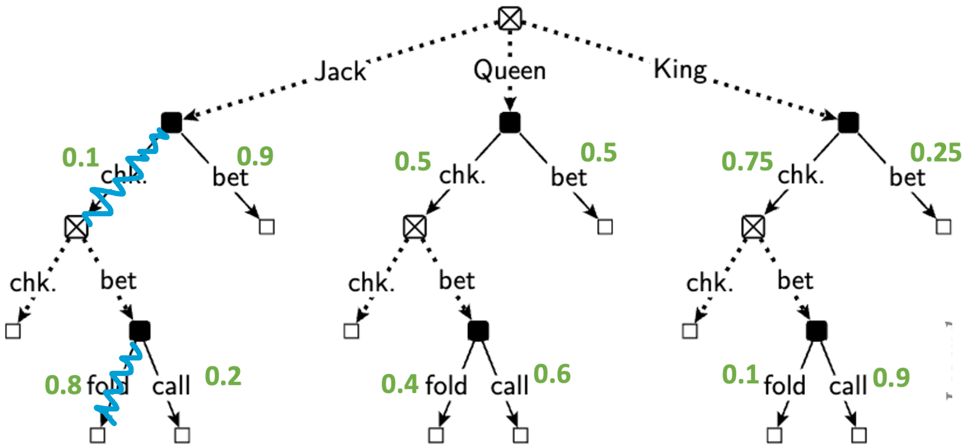
Prob of reaching this terminal state: 1/6 (Nature) x 0.1 (Pl1) x 0.6 (Pl2)

Expected Utility

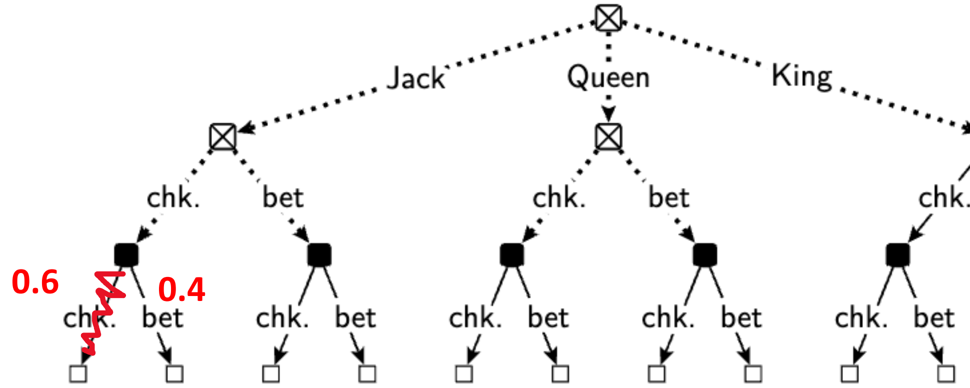
Game tree:



Decision problem and behavioral strategy of Player 1



Decision problem and behavioral strategy of Player 2



Prob of reaching this terminal state: 1/6 (Nature) x 0.1 (Pl1) x 0.4 (Pl2) x 0.8 (Pl1)

When these are variables being optimized, we have a product! Non-convexity in player’s strategy

Kuhn's Theorem

(Under perfect recall assumption)

Normal-form strategies and behavioral strategies are equally powerful

(more formally: they can induce the same distribution over terminal states)

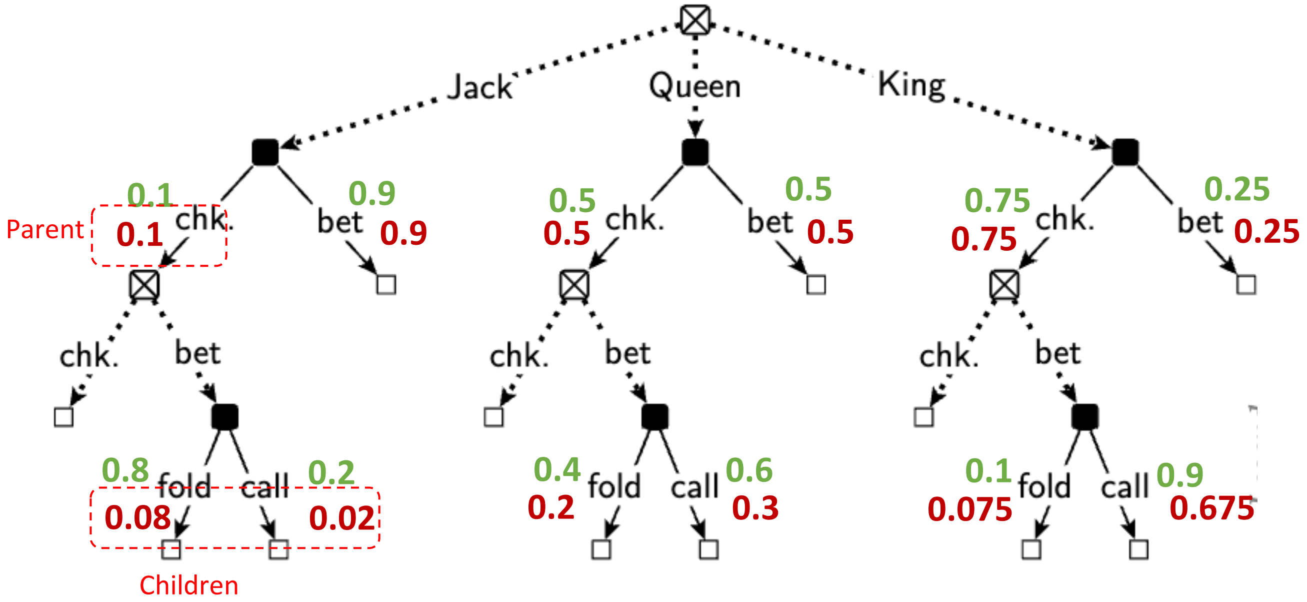
🚨 **Danger zone™:** the theorem is not true anymore if the player does not have perfect recall!

Recap on Behavioral Strategies

	Idea	Obvious downsides	Good news
(Reduced) Normal-form strategies	Distribution over deterministic strategies $\mu \in \Delta(\Pi)$	Exponentially-sized object	In rare cases, it's possible to operate implicitly on the exponential object via a kernel trick
Behavioral strategies	**Local distribution over actions at each decision point** $\boldsymbol{b} \in \times_j \Delta(\boldsymbol{A_j})$	**Expected utility is nonconvex in the the entries of vector $\boldsymbol{b}$**	**Kuhn's theorem: same power as reduced normal-form strategies**

"Fixing" Behavioral Strategies: Sequence-Form Strategies

Idea: Store probability for whole sequences of actions



- ✔ Set of strategies is convex
- ✔ Expected utility is a linear function

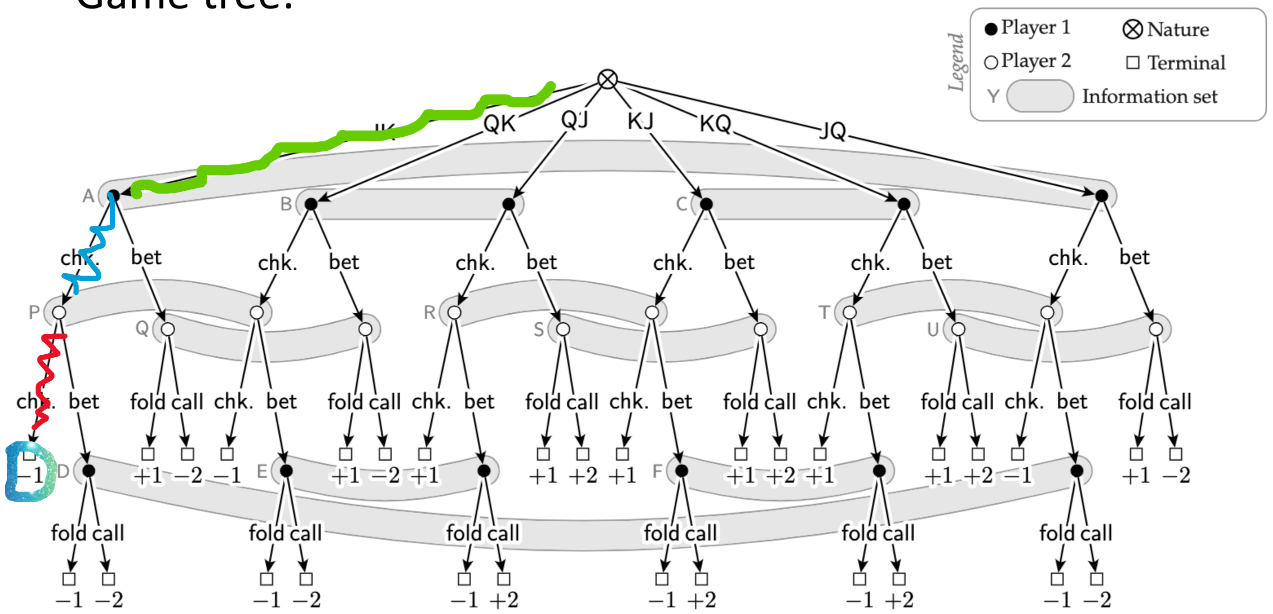
⭐ **Consistency constraints**

1. Entries all non-negative
2. Root sequence has probability 1.0
3. Probability mass conservation

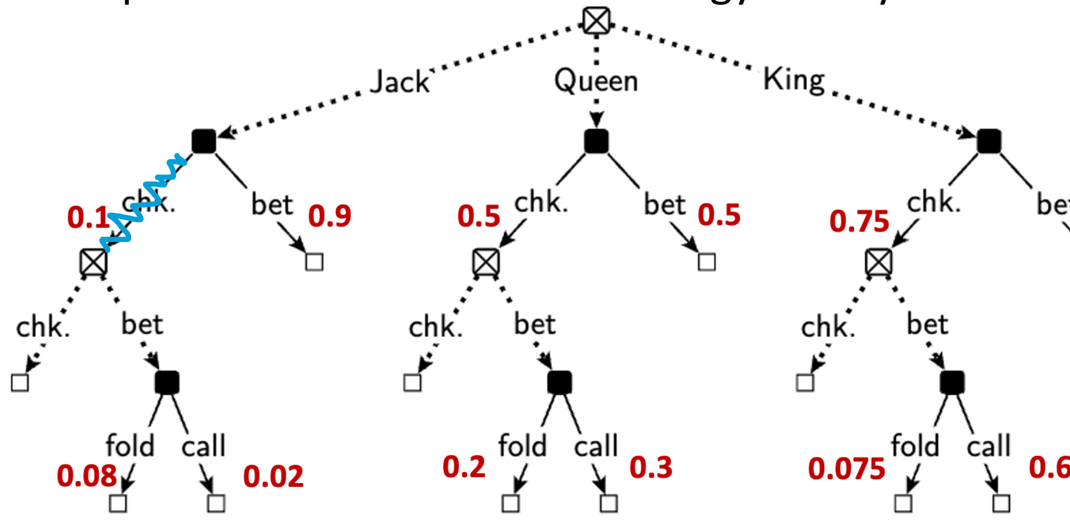
Since sequence-form strategies already automatically encode products of probabilities on paths, expected utility is linear in this strategy representation!

Expected Utility

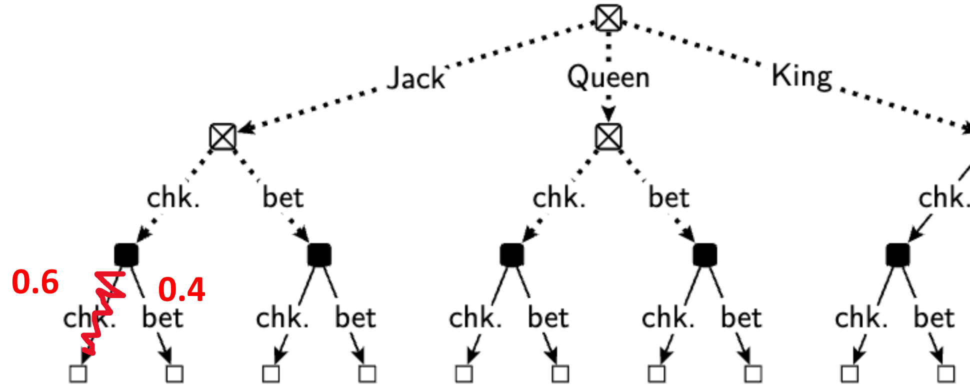
Game tree:



Decision problem and behavioral strategy of Player 1



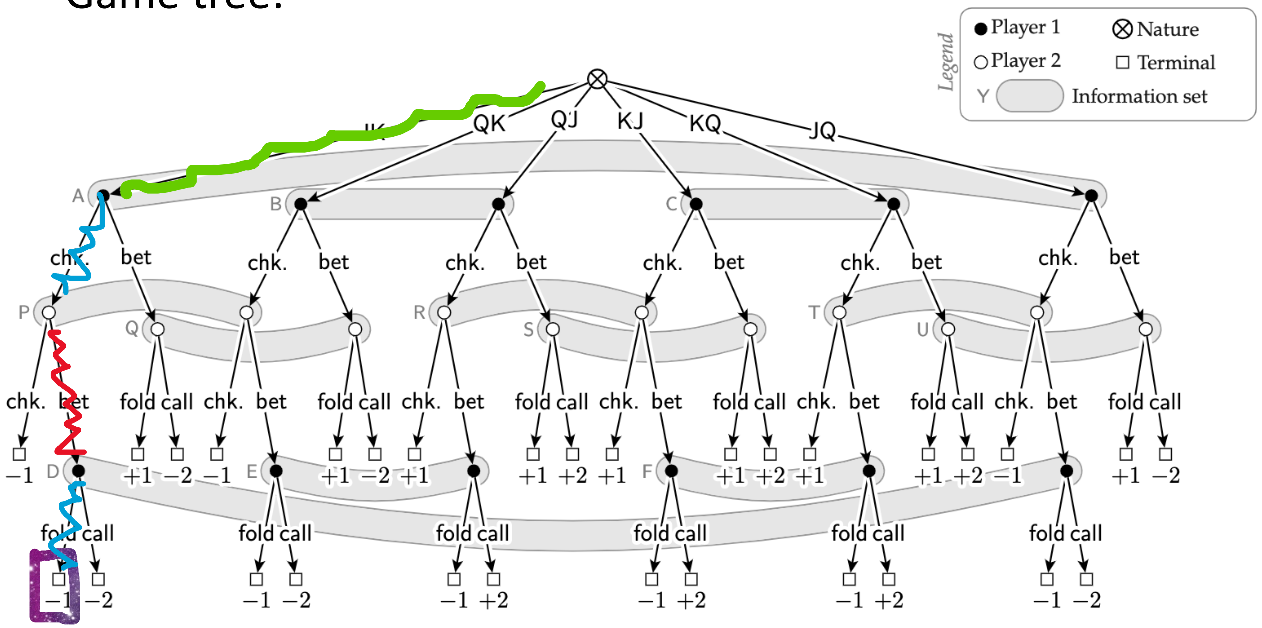
Decision problem and behavioral strategy of Player 2



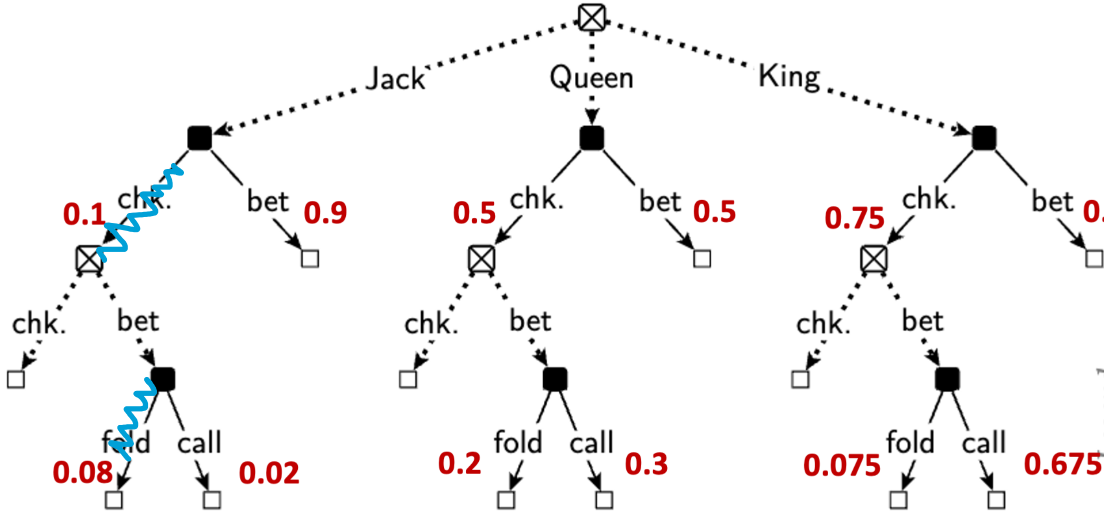
Prob of reaching this terminal state: 1/6 (Nature) x 0.1 (Pl1) x 0.6 (Pl2)

Expected Utility

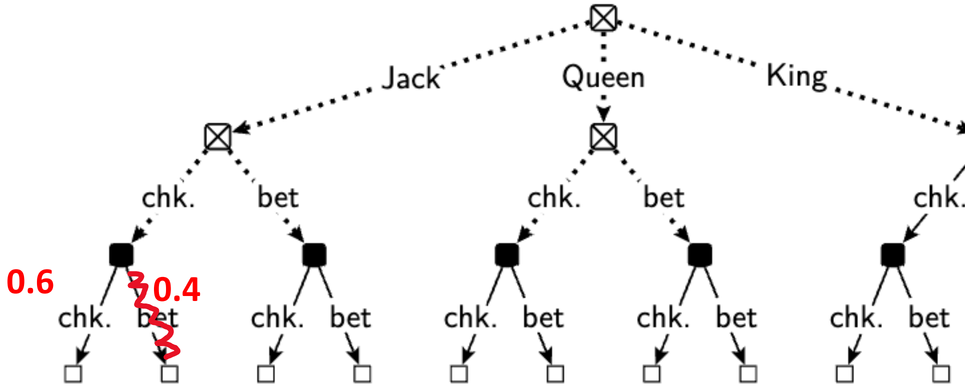
Game tree:



Decision problem and behavioral strategy of Player 1



Decision problem and behavioral strategy of Player 2



Prob of reaching this terminal state: 1/6 (Nature) x 0.08 (Pl1) x 0.4 (Pl2)

Nonlinearity is gone

Sequence-Form Representation

Expected utility is linear in every player's strategy (just like normal-form games)

Where did we pay a price? In normal-form games, strategy set is very simple (simplex). In extensive-form games, we have sequence-form polyoptes

Everything still convex: We can use convex optimization tools

Equilibrium Computation (Extensive-Form)

BEFORE: Reduced–normal form

Nash equilibrium in Kuhn poker:

$$\max_{x} \min_{y} x^T A y$$

Distribution over the 27 plans of Player 1

Distribution over the 64 plans of Player 2

Payoff matrix on the left

You can use any tech... learning, linear program...

Scale exponentially with tree size

NOW: Sequence form

Nash equilibrium in Kuhn poker:

$$\max_{x'} \min_{y'} x'^T B y'$$

Sequence-form polytope of player 1 (dimension 12)

Sequence-form polytope of player 2 (dimension 12)

Sequence-form payoff matrix

You can **still** use learning...

Scale **linearly** with tree size

Recap

	Idea	Obvious downsides	Good news
(Reduced) Normal-form strategies	Distribution over deterministic strategies $\mu \in \Delta(\Pi)$	Exponentially-sized object	In rare cases, it's possible to operate implicitly on the exponential object via a kernel trick
Behavioral strategies	Local distribution over actions at each decision point $b \in \times_j \Delta(A_j)$	Expected utility is nonconvex in the the entries of vector b	Kuhn's theorem: same power as reduced normal-form strategies
Sequence-form strategies	**"Probability flows" on the tree-form decision process** $\boldsymbol{x} \in \boldsymbol{Q}$ **(convex polytope)**	**None**	**Everything is convex!** **Kuhn's theorem applies automatically.**