Module 3: Extensive-Form Games & CFR

CSCE 631 — Algorithmic Game Theory meets LLMs

Week 3: June 9 (Mon) – June 13 (Fri)

Contents

Learning Objectives

By the end of this module, you should be able to:

1. Represent a game in extensive form (game tree with information sets, chance nodes, and payoffs) and convert between extensive-form and normal-form representations.
2. Define and compute subgame perfect equilibria using backward induction, and explain why this refinement matters.
3. Explain the concepts of information sets, perfect recall, and behavioral strategies, and state Kuhn's theorem.
4. Formulate two-player zero-sum extensive-form games as sequence-form LPs and explain the computational advantage over normal-form LPs.
5. Describe the CFR algorithm: counterfactual values, regret accumulation, regret matching per information set, and the decomposition theorem that guarantees convergence.

1 Lecture 10: Extensive-Form Games I (72 min)

The normal-form representation from Weeks 1–2 collapses all strategic reasoning into a single simultaneous choice. Many real-world interactions, however, unfold *sequentially*: players observe some actions before choosing their own, and uncertainty about the state of the world may persist throughout play. The *extensive form* captures this sequential structure explicitly via game trees, and introduces the crucial concept of *information sets* to model what players know—and do not know—at the moment they must act.

1.1 The Extensive-Form Game

Definition: Extensive-Form Game

An **extensive-form game** is specified by:

- A finite **game tree** (V, E) rooted at a distinguished node r. Internal nodes are partitioned into *decision nodes* (owned by a player) and *chance nodes* (owned by Nature). Leaves are *terminal nodes*.
- A **player function** assigning each decision node to a player $i \in N$.
- An **action set** $A(v)$ at each decision node v, labeling the outgoing edges.
- A collection of **information sets** $\mathcal{I}_i$ for each player i, partitioning i's decision nodes into groups of nodes that player i cannot distinguish.
- A probability distribution over actions at each **chance node**.
- A **payoff vector** $u(z) \in \mathbb{R}^n$ at each terminal node z.

In **perfect-information games**, every information set is a singleton: each player knows exactly which node they are at when they act. Chess, Go, and tic-tac-toe are perfect-information games. In **imperfect-information games**, information sets may contain multiple nodes, reflecting genuine uncertainty about the game state. Poker is the canonical example: a player cannot see the opponents' cards, so several dealing histories are lumped into a single information set.

Chance nodes (sometimes labeled "Nature") model exogenous randomness such as card deals, dice rolls, or coin flips. They act according to a fixed, commonly known probability distribution and are not strategic agents.

1.2 Information Sets

Definition: Information Set

An **information set** $I \in \mathcal{I}_i$ for player i is a set of decision nodes belonging to player i such that: (1) i cannot distinguish between nodes in I based on the history of play, and (2) the same set of actions is available at every node in I, i.e., $A(v) = A(v')$ for all $v, v' \in I$.

The information-set structure encodes what a player observes. When player i acts at some node $v \in I$, they know only that they are at *some* node in I—they cannot condition their action on which specific node it is. This is why the same actions must be available: if different actions were available at different nodes in I, the player could infer the node from the action set, contradicting the definition.

1.3 Perfect Recall

Definition: Perfect Recall

A player has **perfect recall** if, at every information set, the player remembers all of their own past actions and all past information sets they have visited. Formally, for every pair of nodes v, v' in the same information set I of player i, the sequence of player i's actions and information sets along the path from the root to v is the same as along the path from the root to v'.

Perfect recall is assumed in virtually all computational game theory. It means that a player may forget what Nature or the opponents did, but never forgets what they themselves did. Without perfect recall, many of the elegant decomposition results that make large games tractable—notably Kuhn's theorem and the CFR decomposition—break down entirely.

1.4 Strategies in Extensive-Form Games

A **pure strategy** for player i in an extensive-form game is a complete contingency plan: it specifies one action for *every* information set of player i, including information sets that may be unreachable given earlier choices. This completeness is essential for Nash equilibrium analysis because opponents' beliefs about off-path behavior determine whether deviations are profitable.

Definition: Behavioral Strategy

A **behavioral strategy** for player i assigns, to each information set $I \in \mathcal{I}_i$, a probability distribution over the actions $A(I)$ available at I. That is, $\beta_i \colon \mathcal{I}_i \to \bigcup_I \Delta(A(I))$ with $\beta_i(I) \in \Delta(A(I))$ for each I.

Behavioral strategies represent *local* randomization: the player independently draws a random action at each information set. This is in contrast to *mixed strategies*, which randomize over *complete plans* (pure strategies). In general, these two forms of randomization are different. However, a foundational result shows they are equivalent in the games we care about.

1.5 Kuhn's Theorem

Theorem: Kuhn's Theorem (1953)

In any finite extensive-form game with **perfect recall**, every mixed strategy of a player is *realization-equivalent* to some behavioral strategy, and vice versa. That is, for any mixed strategy σ_i there exists a behavioral strategy β_i that induces the same probability distribution over terminal nodes (for every fixed strategy of the opponents), and conversely.

Kuhn's theorem is practically important because behavioral strategies are exponentially more compact than mixed strategies. A player with d information sets, each having b actions, has b^d pure strategies, so a mixed strategy lives in a simplex of dimension $b^d - 1$. A behavioral strategy, by contrast, is described by d independent distributions over b actions each—only $d(b-1)$ free parameters. Under perfect recall, this compact representation loses nothing.

Example: Entry Deterrence Game

An entrant decides whether to *enter* a market or *stay out*. If the entrant enters, the incumbent chooses to *accommodate* or *fight*. Payoffs: if the entrant stays out, payoffs are (0, 2); if the entrant enters and the incumbent accommodates, payoffs are (1, 1); if the entrant enters and the incumbent fights, payoffs are $(-1, -1)$.
The game tree has two information sets: the entrant's (singleton) and the incumbent's (singleton, since the incumbent observes entry). The incumbent's threat to fight is *non-credible*: upon entry, fighting yields -1 while accommodating yields 1. Backward induction shows the entrant enters and the incumbent accommodates, giving payoffs (1, 1). While the normal-form game has a Nash equilibrium where the incumbent threatens to fight and the entrant stays out, this equilibrium is not subgame perfect—an observation that motivates the refinements in Lecture 11.

Common Pitfalls

- A pure strategy in extensive form specifies an action at *every* information set, including those unreachable given earlier choices. This completeness is essential for Nash equilibrium analysis.
- Perfect recall is almost always assumed. Without it, Kuhn's theorem fails, and behavioral and mixed strategies are not equivalent.
- The normal-form representation of an extensive-form game can be exponentially larger: a game with d information sets, each with b actions, has b^d pure strategies.

2 Lecture 11: Extensive-Form Games II (72 min)

Nash equilibrium, transplanted into extensive-form games, can support strategies that rely on threats no rational player would actually carry out. This lecture introduces *subgame perfection*—the requirement that strategies form a Nash equilibrium in every subgame—and its extensions to imperfect-information settings via sequential equilibrium and perfect Bayesian equilibrium. These refinements are central to modeling credible commitments, signaling, and strategic communication.

2.1 Subgames

Definition: Subgame

A **subgame** of an extensive-form game is a subtree rooted at a node v that satisfies two conditions: (1) v is a singleton information set (i.e., the player acting at v knows they are at v), and (2) the subtree rooted at v does not "cut" any information set—if a node w is in the subtree and belongs to information set I, then every node in I is also in the subtree. A subgame must be a self-contained game in its own right.

Every extensive-form game has at least one subgame: the entire game itself. In perfect-information games, every decision node roots a subgame. In imperfect-information games, subgames may be scarce because information sets often span across different branches.

2.2 Subgame Perfect Equilibrium

Definition: Subgame Perfect Equilibrium (SPE)

A strategy profile σ is a **subgame perfect equilibrium** if its restriction to every subgame is a Nash equilibrium of that subgame.

Subgame perfection eliminates non-credible threats by requiring rational play even in parts of the game tree that the equilibrium path never reaches. In finite perfect-information games, SPE can be computed by **backward induction**: starting from the terminal nodes, at each decision node the acting player chooses the action maximizing their payoff, assuming all future players also play optimally.

Example: Centipede Game

Two players alternate moves for k rounds. At each turn, the acting player can *take* (ending the game with a payoff split favoring the taker) or *pass* (letting the game continue, increasing the total payoff). Backward induction dictates that the first player takes immediately at the root, yielding a low payoff for both. This is the unique SPE.

The centipede game is famous for the stark contrast between theory and experiment: in laboratory settings, human subjects frequently cooperate (pass) for several rounds, capturing the mutual gains that backward induction forecloses. This discrepancy has fueled extensive debate about whether common knowledge of rationality is a realistic assumption.

2.3 Sequential Equilibrium

In imperfect-information games, subgame perfection may have little bite because there may be few (or no proper) subgames. Sequential equilibrium strengthens the refinement by requiring players to hold beliefs about where they are within an information set and to act optimally given those beliefs.

Definition: Sequential Equilibrium

A **sequential equilibrium** is a pair (σ, μ) consisting of a behavioral strategy profile σ and a **belief system** μ (a probability distribution over nodes at each information set) satisfying:

- **Sequential rationality**: at every information set I, the acting player's strategy is optimal given the beliefs $\mu(I)$ and the continuation strategies of all players.

- **Consistency**: (σ, μ) is the limit of a sequence (σ^k, μ^k) where each σ^k is a completely mixed behavioral strategy (all actions played with positive probability) and μ^k is derived from σ^k by Bayes' rule.

The consistency requirement handles the thorny question of what to believe at information sets reached with zero probability under the equilibrium strategy. By requiring beliefs to be limits of Bayesian posteriors under fully mixed perturbations, sequential equilibrium ensures beliefs are "reasonable" even off the equilibrium path.

2.4 Signaling Games and Perfect Bayesian Equilibrium

Perfect Bayesian equilibrium (PBE) is a simpler variant of sequential equilibrium frequently used in signaling games. In a signaling game, one player (the *sender*) has private information—a *type* drawn by Nature—and chooses a publicly observable *signal*. The other player (the *receiver*) observes the signal, updates beliefs about the sender's type, and then acts. PBE requires specifying beliefs at every information set (including those off the equilibrium path) and sequential rationality given those beliefs.

Signaling games appear throughout economics (job-market signaling, advertising as quality signal), political science, and, increasingly, in multi-agent AI systems where one agent's communication must be interpreted by another.

Common Pitfalls

- SPE is only a strict refinement of NE when there are proper subgames. In simultaneous-move games (or single-round games), SPE coincides with NE.
- Backward induction assumes common knowledge of rationality at every node, even nodes that rational play would never reach. This is philosophically contentious, as the centipede game illustrates.
- Sequential equilibrium requires beliefs even at information sets that are never reached in equilibrium—this is where the concept becomes most subtle.

3 Lecture 12: Extensive-Form Games III — Sequence Form (73 min)

The normal-form representation of an extensive-form game can be computationally catastrophic. This lecture introduces the *sequence form*, an alternative representation that stays compact by exploiting the tree structure. The sequence-form LP is the mathematical foundation of modern game-solving systems including Libratus and Pluribus, which solved heads-up and multiplayer no-limit Texas hold'em.

3.1 The Normal-Form Blowup

Converting an extensive-form game to normal form requires enumerating every pure strategy: every possible mapping from information sets to actions. If a player has d information sets, each with b available actions, then the number of pure strategies is b^d. For a game like poker, where a player may face thousands of information sets, this exponential blowup makes normal-form methods completely infeasible.

3.2 Sequences and Realization Plans

The key insight of the sequence form is to work with *sequences* rather than complete strategies. A sequence captures only the partial plan "play these actions along this path from the root," without specifying what to do at unreachable information sets.

Definition: Sequence

A **sequence** σ for player i is an ordered list of actions taken by player i along some path from the root of the game tree to a node. The *empty sequence* $\emptyset$ corresponds to player i having taken no actions yet. The number of sequences for player i equals the number of player-i actions in the tree plus one (for the empty sequence), and is therefore *linear* in the size of the game tree.

Definition: Realization Plan

A **realization plan** for player i is a function $x_i \colon \Sigma_i \to [0, 1]$ mapping each sequence to its probability of being played. It must satisfy:

- $x_i(\emptyset) = 1$ (the empty sequence is always "played").
- For each information set I of player i with parent sequence σ:

$$\sum_{a \in A(I)} x_i(\sigma \cdot a) = x_i(\sigma) \quad \text{(flow conservation)}.$$

These constraints are linear and ensure x_i is consistent with some behavioral strategy.

The flow-conservation constraints encode the tree structure: the probability of reaching an information set must equal the sum of the probabilities of the actions taken there. Under perfect recall, there is a bijection between valid realization plans and behavioral strategies.

3.3 The Sequence-Form LP

For a two-player zero-sum extensive-form game, the expected payoff can be written as a bilinear form $x_1^\top A\, x_2$ in the two players' realization plans, where A is a payoff matrix indexed by sequence pairs. The Nash equilibrium can then be computed via a linear program.

Theorem: Sequence-Form LP Complexity

The sequence-form LP for a two-player zero-sum extensive-form game has $O(|\Sigma_1| + |\Sigma_2|)$ variables and constraints, where $|\Sigma_i|$ is the number of sequences for player i. Since the number of sequences is linear in the game tree size, this LP is **polynomial** in the size of the game tree, in contrast to the normal-form LP which has $O(|S_1| + |S_2|)$ variables where $|S_i|$ can be exponential.

This result is the mathematical engine behind solving large poker variants. The entire Libratus and Pluribus pipeline—blueprint strategy computation, subgame resolving, real-time endgame search—starts from sequence-form ideas. The practical implication is stark: games with millions of information sets and astronomical numbers of pure strategies can be solved exactly, in polynomial time, for the two-player zero-sum case.

Common Pitfalls

- Sequence form requires perfect recall. Without it, realization plans are not well-defined because the bijection with behavioral strategies breaks down.

- The sequence-form LP solves two-player zero-sum games exactly. For general-sum or multiplayer games, the equilibrium problem remains hard (PPAD-complete for general-sum, and even the definition of the appropriate solution concept is debated for multiplayer).
- Do not confuse the game tree size (which determines the sequence-form LP size) with the number of pure strategies (which determines the normal-form LP size). These can differ exponentially.

Connection: Course Project

The sequence form is the mathematical foundation for the course project topics involving poker-playing agents. If you pursue CFR beyond Kuhn Poker or Deep CFR replication, understanding the sequence form is essential for appreciating how the algorithm exploits the game tree structure.

4 Lecture 13: Counterfactual Regret Minimization — CFR (71 min)

The sequence-form LP solves two-player zero-sum extensive-form games in polynomial time, but "polynomial" can still be impractical when the game tree has billions of information sets. Modern poker AI relies instead on *Counterfactual Regret Minimization* (CFR), an iterative, anytime algorithm that decomposes the global regret-minimization problem into independent subproblems at each information set. CFR is the workhorse algorithm of computational game theory for imperfect-information games, and understanding it is essential for the course project.

4.1 Reach Probabilities

Definition: Reach Probability

Given a strategy profile σ, the **reach probability** of a history (node) h is:

$$\pi^\sigma(h) = \prod_{i \in N \cup \{c\}} \pi_i^\sigma(h),$$

where the product is over all players and chance (c). We decompose this into player i's contribution $\pi_i^\sigma(h)$ (the product of i's action probabilities along the path to h) and the complementary term $\pi_{-i}^\sigma(h) = \pi^\sigma(h)/\pi_i^\sigma(h)$, which includes the contributions of all other players and chance.

This decomposition is central to CFR. By isolating player i's contribution to the reach probability, we can define *counterfactual values* that cleanly attribute regret to individual information sets—disentangling player i's past decisions from the parts of the reach probability outside their control.

4.2 Counterfactual Values

Definition: Counterfactual Value

The **counterfactual value** of information set I for player i under strategy profile σ is:

$$v_i^\sigma(I) = \sum_{h \in I} \pi_{-i}^\sigma(h) \cdot v_i^\sigma(h),$$

where $v_i^\sigma(h)$ is the expected utility of player i starting from history h under σ. The weighting

by $\pi^{\sigma}_{-i}(h)$—the probability that opponents and chance reach h—is what makes the value "counterfactual": it asks "what would i's expected payoff be at I if i played to reach I?"

The **counterfactual value of an action** a at I is denoted $v_i^{\sigma}(I, a)$ and measures the expected value when player i plays action a at I and follows σ everywhere else.

The **instantaneous counterfactual regret** for not playing action a at information set I on iteration t is:

$$r_i^t(I, a) = v_i^{\sigma^t}(I, a) - v_i^{\sigma^t}(I).$$

The **cumulative counterfactual regret** is:

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

4.3 The CFR Decomposition Theorem

Theorem: CFR Decomposition Theorem

The overall average external regret of player i is bounded by the sum of the per-information-set regrets:

$$\text{Regret}_i^T \;\le\; \sum_{I \in \mathcal{I}_i} \max_{a \in A(I)} R_i^{T,+}(I, a),$$

where $R_i^{T,+}(I, a) = \max(R_i^T(I, a), 0)$. Consequently, if regret matching is run *independently* at each information set, the resulting strategy achieves no overall regret, and the **average strategy** converges to a Nash equilibrium in two-player zero-sum games.

This decomposition is what makes CFR scalable: it reduces the global no-regret problem to a collection of local no-regret problems, one per information set. Each information set runs its own instance of regret matching, oblivious to the others. The overall guarantee emerges from the additive structure of the bound.

4.4 The CFR Algorithm

Regret matching at each information set selects actions in proportion to their positive cumulative regret:

$$\sigma_i^{t+1}(I, a) = \begin{cases} \dfrac{R_i^{t,+}(I, a)}{\sum_{a'} R_i^{t,+}(I, a')} & \text{if } \sum_{a'} R_i^{t,+}(I, a') > 0, \\ \dfrac{1}{|A(I)|} & \text{otherwise.} \end{cases}$$

Algorithm 1 Counterfactual Regret Minimization (CFR)

Initialize cumulative regret $R_i(I, a) \leftarrow 0$ for all i, I, a
Initialize cumulative strategy sum $S_i(I, a) \leftarrow 0$ for all i, I, a
for $t = 1, 2, \ldots, T$ **do**
  for each information set I of each player i **do**
    Compute current strategy $\sigma_i^t(I, \cdot)$ via regret matching
  end for
  Traverse the game tree under σ^t
  for each information set I of each player i **do**
    for each action $a \in A(I)$ **do**
      Compute counterfactual value $v_i^{\sigma^t}(I, a)$
      $r \leftarrow v_i^{\sigma^t}(I, a) - v_i^{\sigma^t}(I)$ ▷ Instantaneous regret
      $R_i(I, a) \leftarrow R_i(I, a) + r$ ▷ Accumulate regret
    end for
    for each action $a \in A(I)$ **do**
      $S_i(I, a) \leftarrow S_i(I, a) + \pi_i^{\sigma^t}(I) \cdot \sigma_i^t(I, a)$ ▷ Accumulate strategy
    end for
  end for
end for
return average strategy $\bar{\sigma}_i(I, a) = S_i(I, a) / \sum_{a'} S_i(I, a')$

4.5 Convergence

Theorem: CFR Convergence Bound

After T iterations of CFR, the average strategy profile $\bar{\sigma}^T$ is an ε-Nash equilibrium with exploitability bounded by:

$$\varepsilon \;\le\; \frac{1}{T} \sum_{I \in \mathcal{I}_i} \sqrt{T \cdot |A(I)|} \cdot \Delta_I \;=\; O\!\left(\frac{1}{\sqrt{T}}\right),$$

where Δ_I bounds the range of counterfactual values at information set I. Each iteration requires a full traversal of the game tree, so the per-iteration cost is $O(|\text{game tree}|)$.

4.6 CFR+

CFR+ (Tammelin, 2014) modifies vanilla CFR by clipping negative cumulative regrets to zero at each iteration, a technique called *regret matching plus* (RM+). Formally, instead of $R_i^{t+1}(I, a) = R_i^t(I, a) + r_i^t(I, a)$, CFR+ uses:

$$R_i^{t+1}(I, a) = \max\!\big(R_i^t(I, a) + r_i^t(I, a),\; 0\big).$$

CFR+ also weights the strategy average linearly by the iteration number, giving more weight to later (presumably better) strategies. Empirically, CFR+ converges dramatically faster than vanilla CFR, and it was instrumental in solving heads-up limit Texas hold'em.

Common Pitfalls

- CFR converges the *average* strategy, not the current strategy. The current strategy may oscillate wildly and is not itself an approximate equilibrium.
- Counterfactual values weight by *opponents'* reach probability $\pi^{\sigma}_{-i}(h)$, not the player's own reach. This disentanglement is what makes the decomposition theorem work: player i's regret at an information set is independent of their own strategy at other information sets.
- CFR requires perfect recall. Without it, the decomposition theorem fails because the additive regret bound no longer holds.

Connection: PA2 and Course Project

PA2 uses the multi-agent debate framework, which connects to Week 5. However, understanding CFR gives you the foundational algorithm that the course project topics—CFR beyond Kuhn Poker, Deep CFR replication—build upon. If you are considering a project involving imperfect-information game solving, the material in this lecture is prerequisite.

5 Lecture 14: MCTS and Sampling-based CFR (58 min)

Full-tree CFR traverses the entire game tree on every iteration, which is feasible for games up to about 10^6 nodes but prohibitive beyond that. This lecture covers two strategies for scaling: *Monte Carlo Tree Search* (MCTS), a simulation-based approach originally designed for perfect-information games like Go, and *Monte Carlo CFR* (MCCFR), which adapts sampling ideas to the CFR framework for imperfect-information games.

5.1 Monte Carlo Tree Search (MCTS)

MCTS builds a partial search tree incrementally, guided by random simulations. It is the algorithm behind AlphaGo and its successors. The algorithm proceeds in four phases, repeated until a computational budget is exhausted.

Algorithm 2 Monte Carlo Tree Search (MCTS)

1: **while** within computational budget **do**
2: **Selection**: Starting from the root, descend through the tree by selecting child nodes according to a *tree policy* (e.g., UCB1/UCT) until reaching a node v that has unexpanded children.
3: **Expansion**: Add one (or more) unexpanded children of v to the tree.
4: **Simulation (Rollout)**: From the newly expanded node, play a *random simulation* to a terminal node using a default policy (e.g., uniform random actions).
5: **Backpropagation**: Propagate the simulation result back up the path from the expanded node to the root, updating visit counts and value estimates at each ancestor.
6: **end while**
7: **return** the action at the root with the highest visit count (or value).

5.1.1 UCB1 and UCT

The selection phase uses the **Upper Confidence Bound** (UCB1) formula to balance exploration and exploitation:

$$\text{UCB1}(v) = \bar{X}_v + C\sqrt{\frac{\ln N(\text{parent}(v))}{N(v)}},$$

where $\bar{X}_v$ is the average reward at node v, $N(v)$ is the visit count, and C is an exploration constant. **UCT** (Upper Confidence bounds applied to Trees) applies UCB1 recursively at each level of the tree, treating child selection as a multi-armed bandit problem.

5.1.2 MCTS and Imperfect Information

Straightforward MCTS fails in imperfect-information games because it conflates nodes that belong to the same information set. The tree search assumes the acting player knows exactly which node they are at, but in an imperfect-information game, nodes in the same information set are indistinguishable. Naive modifications—such as *determinization* (sampling a single possible world and solving the resulting perfect-information game) or averaging over sampled worlds—introduce systematic biases. These shortcomings motivate the sampling-based CFR variants described below, which correctly handle information sets.

5.2 Monte Carlo CFR (MCCFR)

MCCFR replaces the full tree traversal of vanilla CFR with sampled trajectories, updating only the information sets visited along the sample path. This reduces the per-iteration cost at the expense of introducing variance. Three main variants differ in what they sample.

5.2.1 Chance Sampling

Chance sampling samples only the chance events (e.g., card deals) but traverses all player actions given the sampled chance outcome. This eliminates the cost of iterating over chance branches, which can be enormous in card games, while retaining exact computation of counterfactual values for the sampled chance realization.

5.2.2 External Sampling

External sampling samples both chance events and opponent actions but traverses all of the updating player i's actions. This yields an unbiased estimator of counterfactual values for player i and is the most commonly used MCCFR variant in practice. It offers a favorable balance between per-iteration cost and variance: the update at each information set considers all of i's actions (avoiding high-variance importance-sampling corrections) while dramatically reducing the tree traversal by sampling the opponents.

5.2.3 Outcome Sampling

Outcome sampling samples a single complete trajectory through the game tree—chance, both players' actions, everything. Updates are weighted by importance-sampling ratios to maintain unbiasedness. This is the cheapest variant per iteration (cost proportional to the path length) but has the highest variance, requiring many more iterations to converge to the same accuracy.

5.3 Pruning

Regret-based pruning skips actions whose cumulative regret is very negative, on the grounds that these actions will have near-zero probability under regret matching and contribute little to the strategy update. Pruning is *safe* in the sense that it does not affect convergence guarantees: any action pruned will eventually accumulate enough regret to be reconsidered. In practice, pruning can reduce the per-iteration cost by an order of magnitude in the later stages of training, when many actions have been identified as clearly suboptimal.

Common Pitfalls

- Standard MCTS (designed for perfect-information games) is *not* sound for imperfect-information games without modifications. Applying it naively produces strategies that exploit knowledge the player does not actually have.
- MCCFR convergence is of the *average* strategy, same as vanilla CFR. The current iterate is not an equilibrium.
- Pruning speeds up computation but does not change the theoretical convergence rate. It is a practical optimization, not an algorithmic improvement.

Connection: Course Project

If your project involves scaling CFR to games larger than Kuhn Poker—such as Leduc hold'em or abstractions of Texas hold'em—you will almost certainly use MCCFR (likely external sampling) rather than vanilla CFR. Understanding the variance–cost tradeoff across the sampling variants is essential for making informed implementation decisions.

Key Algorithms Summary

Algorithm	Game Type	Cost/Iteration	Convergence
Sequence-form LP	2P zero-sum	Polynomial (exact)	Exact NE
Vanilla CFR	2P zero-sum	$O(\|\text{tree}\|)$	$O(1/\sqrt{T})$ to NE
CFR+	2P zero-sum	$O(\|\text{tree}\|)$	Empirically faster
MCCFR (external)	2P zero-sum	$O(\|\text{sampled tree}\|)$	$O(1/T^{1/3})$ to NE
MCCFR (outcome)	2P zero-sum	$O(\|\text{path}\|)$	$O(1/T^{1/4})$ to NE

Key Definitions Summary

Term	Definition
Extensive-form game	Game tree with players, actions, information sets, chance nodes, and payoffs
Information set	Nodes indistinguishable to the acting player; same actions available
Perfect recall	Player remembers all own past actions and information sets
Behavioral strategy	Independent probability distribution over actions at each information set
Subgame	Self-contained subtree rooted at a singleton information set
SPE	Nash equilibrium in every subgame
Sequential equilibrium	Strategy profile + belief system satisfying sequential rationality and consistency
Sequence	Ordered list of a player's actions along a root-to-node path
Realization plan	Probability map over sequences satisfying flow conservation
Reach probability	$\pi^\sigma(h) = \prod_i \pi_i^\sigma(h)$; probability of reaching history h
Counterfactual value	$v_i^\sigma(I) = \sum_{h \in I} \pi_{-i}^\sigma(h) \cdot v_i^\sigma(h)$
CFR	Iterative algorithm: regret matching per information set, average strategy converges

Key Takeaways

1. Extensive-form games generalize normal-form games to sequential settings with game trees, information sets, and chance nodes.
2. Kuhn's theorem guarantees that behavioral and mixed strategies are equivalent under perfect recall, enabling the compact behavioral representation.
3. Subgame perfect equilibrium eliminates non-credible threats by requiring Nash equilibrium play in every subgame; sequential equilibrium extends this to imperfect-information games via beliefs.
4. The sequence form avoids the exponential blowup of the normal form: the LP has variables linear in the game tree size, enabling polynomial-time NE computation for two-player zero-sum games.
5. CFR decomposes global regret into per-information-set regrets, running regret matching independently at each information set. The average strategy converges to an ε-NE at rate $O(1/\sqrt{T})$.
6. MCCFR variants trade per-iteration cost for variance, enabling CFR to scale to games too large for full tree traversal.

Suggested Reading

- Zinkevich, Johanson, Bowling, Piccione (2007): "Regret Minimization in Games with Incomplete Information" (the original CFR paper).
- Lanctot, Waugh, Zinkevich, Bowling (2009): "Monte Carlo Sampling for Regret Minimization in Extensive Games."
- Tammelin (2014): "Solving Large Imperfect Information Games Using CFR+."
- Shoham & Leyton-Brown: *Multiagent Systems*, Chapters 5–6 (extensive-form games, sequential equilibrium).