Module 3 — Extensive Form Games

Game Theory

hbar.university

Game Trees and Sequential Moves

Position in Knowledge Graph

System: Game Theory **Structural Domain:** Extensive Form Games **Node:** 3.1

The strategic form of Module 1 treats a game as a single simultaneous-choice matrix. This is a conscious flattening: most real strategic situations unfold over time, with players moving in sequence, sometimes observing past moves, sometimes not, sometimes with chance events interleaved. The **extensive form** restores this temporal and informational structure. A game tree is a rooted tree whose nodes are decision points, whose edges are actions, and whose leaves are payoff outcomes. This lesson defines the tree formally and sets the stage for information sets (3.2), sequential strategies (3.3), backward induction (3.4), and the refinement theory of subgame perfection (3.5). Everything about poker — the entire betting tree of a hand — lives in the extensive form.

Formal Definition

An **extensive form game** is a tuple $\Gamma = (N, T, P, A, u)$ where:

- $N = \{1, 2, \ldots, n\}$ is the set of players (plus possibly a special "chance" player 0).
- T is a finite rooted tree with root t_0 and leaves Z. Interior nodes are **decision nodes**; leaves are **terminal nodes**.
- $P : T \setminus Z \to N \cup \{0\}$ assigns each decision node to the player who moves there. $P(t) = 0$ means "chance moves" — outcomes at t are drawn from a fixed probability distribution.
- $A(t)$ is the set of actions available at decision node t. Edges from t are labeled with elements of $A(t)$.
- $u = (u_1, \ldots, u_n)$ where each $u_i : Z \to \mathbb{R}$ gives player i's payoff at each terminal node.

A **path** from root to leaf is a sequence of nodes $t_0, t_1, \ldots, t_k$ where each t_{j+1} is a child of t_j, and $t_k \in Z$. A path corresponds to a complete play of the game.

A **perfect-information** extensive-form game is one where, at each decision node t, the acting player knows the entire history of past moves — i.e., knows t itself. Module 3.2 generalizes to **imperfect information** using *information sets*.

A game with chance nodes ($P(t) = 0$) is a **game with chance moves**; the distribution at a chance node is part of the game specification.

Intuition

Picture the game as a decision tree you can walk through. Starting from the root, the designated player chooses an action, producing a new node; then the next player moves, and so on, until a leaf is reached. At each leaf, the payoffs to all players are written down.

In a perfect-information game, each player sees everything — every past move is visible. Chess is the canonical example (ignoring time pressure): both players see the full history of the game at every move. Backward induction (Node 3.4) solves such games by working from the leaves to the root.

Chance nodes introduce randomness. In poker, the dealing of hole cards is a chance node; a subsequent betting decision by player 1 is a decision node; the dealing of the flop is another chance node; a betting decision by player 2 is another decision node, and so on. The tree alternates between player and chance nodes, with payoffs only at the terminal (showdown or fold) leaves.

The tree is rigid about order. Every play of the game is a sequence, not a set of simultaneous choices. Even "simultaneous" games can be written in the extensive form by using information sets (Node 3.2) to hide the order of moves from one of the players.

Structural Role

Game trees are the structural foundation for every subsequent dynamic concept:

- **3.1 ⇒ 3.2.** Information sets partition decision nodes by what the acting player knows.
- **3.1 ⇒ 3.3.** Strategies in extensive form assign an action at every information set.
- **3.1 ⇒ 3.4.** Backward induction computes optimal play by recursing from leaves to root.
- **3.1 ⇒ 3.5.** Subgame perfect equilibrium is Nash equilibrium of the extensive form that survives backward-induction-like pruning.
- **3.1 ⇒ 4.1.** Bayesian games add type-level chance nodes at the root.
- **3.1 ⇒ 5.1.** Repeated games are infinite trees that unfold over time with each stage.
- **3.1 ⇒ 9.3.** Counterfactual regret minimization operates on the extensive-form tree, computing regret at each information set.

Without the extensive form, we could not distinguish credible from incredible threats, could not define "what a player knows when they move," and could not formalize poker as a game. The tree is the unifying data structure for Modules 3, 4, and 9.

Mathematical Development

Paths and histories

A **history** at node t is the sequence of actions taken to reach t from the root. In perfect-information games, the history uniquely identifies t. In imperfect-information games, several histories may share the same node-ID (they become indistinguishable for the acting player).

Ex-ante strategy of player i: a function assigning an action to every decision node owned by i (perfect information) or to every information set of i (imperfect information, Node 3.2).

Chance nodes and probability distributions

At a chance node t with $P(t) = 0$, a distribution $p_t \in \Delta(A(t))$ is given as part of the game description. When computing expected payoffs, one averages over chance draws. In a finite perfect-information game with chance, the expected payoff of a strategy profile is

$$\mathbb{E}[u_i \mid s] = \sum_{z \in Z} \Pr[z \mid s] \cdot u_i(z),$$

where $\Pr[z \mid s]$ is the product of chance probabilities along the path to z induced by s.

Tree structure: Kuhn's axiomatization

An extensive-form game is a rooted tree (connected, acyclic graph). Each non-terminal node has a player assignment, a set of actions, and a set of child nodes in bijection with its actions. The leaves carry payoff vectors. This structure is minimal and combinatorial — no topology, no calculus, just a finite data structure.

Kuhn (1953) formalized this structure and proved several foundational results, including Kuhn's theorem (Node 1.6) on the equivalence of mixed and behavioral strategies under perfect recall.

Branching and size

A depth-d tree with branching factor b has $O(b^d)$ leaves. Game trees for real games are enormous:

- **Tic-tac-toe:** ~255,000 leaves (fully solvable).
- **Checkers:** ~5 × 10^{20} reachable positions (solved by Schaeffer 2007).
- **Chess:** ~10^{46} positions, well beyond exact solution.
- **Go:** ~10^{170} positions.
- **HUNL Poker:** ~10^{161} information sets (after reasonable abstraction), solved approximately by Libratus/DeepStack 2017.

Backward induction preview

For finite perfect-information games, **backward induction** computes an optimal strategy by recursing from leaves:

1. At each terminal node, the payoff is known.
2. At a decision node whose children are all terminal, the acting player chooses the action maximizing their payoff.
3. Replace the decision node with the payoff of the optimal child.
4. Repeat until the root is reached.

The result is the **backward induction equilibrium** — a specific pure strategy profile that is a Nash equilibrium of the game and, moreover, is subgame perfect (Node 3.5).

Worked Example

Two-stage entry game

Player 1 (Entrant) decides to Enter (E) or Stay Out (O). If Out, payoff $(0, 2)$. If Enter, player 2 (Incumbent) decides to Fight (F) or Accommodate (A). Payoffs: $(E, F) \to (-1, -1); (E, A) \to (1, 1)$.

Tree structure.

```
          root (P1)
         /        \
        O          E
     (0,2)         |
                P2 node
                 /    \
                F      A
             (-1,-1) (1,1)
```

Backward induction. At the P2 node, P2 prefers A $(1 > -1)$; replace the P2 node with payoff $(1, 1)$. At the root, P1 compares O (0) to E (1); prefers E. Backward induction outcome: (E, A) with payoff $(1, 1)$.

Contrast with normal form. The normal form (Node 1.6) has two Nash equilibria: (E, A) and (O, F). The second relies on the incredible threat "Incumbent fights." Backward induction eliminates it because it never reaches the P2 node in equilibrium path, but the threat would be suboptimal if the node *were* reached. Subgame perfection (3.5) formalizes this distinction.

A three-level tree

P1 moves, then P2 (observing P1), then P1 again (observing both past moves). Each player has two actions. The tree has 8 leaves. Enumerating pure strategies: P1 has $2 \times 2^2 = 8$ pure strategies (action at root $\times$ action at each of 2 follow-up nodes); P2 has $2^2 = 4$. Normal form is 8×4. For any specific payoff assignment, backward induction gives a unique outcome in generic position.

Poker Anchor

Concept mapping

Formal object	Poker instantiation
Root of game tree	Start of a hand — blinds posted, seats labeled
Chance node (at root)	Hole card dealing — each player receives 2 private cards
Decision node	A player's action point on some street: check/bet/call/fold/raise, and bet sizes
Chance node (mid-game)	Flop, turn, river dealing
Terminal node	Showdown or fold — the hand ends and chips are distributed
Payoff at leaf	Chip delta for each player (winner takes pot; losers pay their contribution)
Path from root to leaf	One complete hand of poker

The entire betting tree of a poker hand is an extensive-form game. Every fact about solver output, every hand-history review, every GTO drill — all of it lives in the tree. The tree structure is what makes poker game theory work.

Concrete situation

HU no-limit hold'em, 100 BB effective, single-raised pot. Sketch the first few levels of the tree:

1. Root: chance node, deals hole cards. Branching factor = $\binom{52}{2}\binom{50}{2} \approx 1.7 \times 10^6$.
2. Depth 1: BTN decision. Actions: Fold, Limp (size 1x), Raise 2x, Raise 2.5x, Raise 3x, . . . , All-in. For a typical solver abstraction, ~5 actions.
3. Depth 2 (if BTN raised): BB decision. Actions: Fold, Call, Raise (several sizes). ~5 actions.
4. Depth 3 (if BB called): chance node, deals flop. Branching = $\binom{48}{3} = 17296$.
5. Depth 4: first postflop decision. ~5 actions. . . . and so on through turn, river, and each betting round.

By the time you reach the river with significant betting history, the path has ~10 decision nodes and ~3 chance nodes. Each information set has many histories mapped to it (hole cards and positional info create the indistinguishability). The total tree size is astronomical, but the *structure* is the same as any other extensive-form game: rooted tree, decision/chance nodes, terminal leaves, payoffs.

What this understanding buys you

"Solving a spot" means solving a subtree. When a solver tool asks for "a specific spot" (e.g., "turn after flop c-bet call"), what it's doing is taking the *subtree* rooted at that point and solving that subtree independently, assuming the parent game context is fixed. This is legitimate because of subgame perfection (Node 3.5): optimal play in a subtree depends only on the subtree, given consistent beliefs at its root.

Hand histories are walks in the tree. Every hand you play traces a single root-to-leaf path through the game tree. Hand review amounts to asking: "at each of my decision nodes along this path, was I taking the best-response action against my opponent's equilibrium distribution over the rest of the subtree?" This is a node-by-node inspection of the actual walk.

Explosion is the real problem. The reason poker is hard is that the tree is enormous. Every solver technique — abstraction, subgame re-solving, CFR iteration — is a method of managing the astronomical size of the tree without literally traversing it. Node 9.3 (CFR) is the central algorithm here.

Cross-Links

- **1.1 Players, strategies, payoffs** — the normal form that extensive form will flatten to.
- **1.6 Normal form as universal representation** — the flattening operation.
- **3.2 Information sets** — how imperfect information is modeled in the tree.
- **3.3 Strategies in extensive form** — functions from information sets to actions.
- **3.4 Backward induction** — the solution algorithm for perfect-information trees.
- **3.5 Subgame perfect equilibrium** — the refinement of Nash in extensive form.
- **9.3 Counterfactual regret minimization** — the algorithm that exploits tree structure.
- **AI systems — game tree search** — the CS analogue.

Structural Compression

Invariant: A sequential strategic situation is represented as a rooted tree where interior nodes are player or chance decisions, edges are actions, and leaves are payoff outcomes. Every play of the game is a root-to-leaf path, and every payoff is computed at a leaf.

Mechanism: Finite tree data structure with player labels on decision nodes, action labels on edges, probability distributions on chance nodes, and real-valued payoff vectors on leaves.

Cross-System Echo: Game trees are the game-theoretic form of computation trees (CS theory), decision trees (machine learning), ancestry trees (phylogenetics), and search trees (AI). In each case, a rooted tree represents a sequence of choices leading to a terminal outcome; the specific structure makes different questions tractable.

Exercises

1. Draw the complete game tree for Rock–Paper–Scissors, treating it as simultaneous — what device must you use to represent the simultaneity on a tree? (Preview of information sets.)

2. Draw the game tree for the two-stage entry game in the worked example and verify the backward-induction outcome.

3. Count the pure strategies of player 1 and player 2 in a two-stage perfect-information game where each player moves once and each has 3 actions.

4. Construct a 3-level perfect-information tree and compute its backward induction equilibrium. Verify that the backward induction profile is a Nash equilibrium of the normal form.

5. Describe the game tree of a one-street poker game where P1 chooses to Bet or Check, and if Bet, P2 chooses to Call or Fold. Include a chance node at the root dealing a hand category ({strong, weak}) to P1. How many leaves are there?

6. Show that the size of the game tree (number of leaves) can be exponential in the depth of the tree even when the branching factor is small.

7. **Argue, structurally,** why the extensive form is a strictly more expressive representation of a strategic situation than the normal form. Identify a specific strategic fact (not just "the order of moves") that is directly visible in the tree but invisible in the flattened matrix.

Information Sets and Imperfect Information

Position in Knowledge Graph

System: Game Theory **Structural Domain:** Extensive Form Games **Node:** 3.2

An information set is a partition block of decision nodes that the acting player cannot tell apart. It is the formal device that encodes "what the player knows when they move." Without information sets, extensive-form games would be restricted to perfect information — no hidden cards, no simultaneous moves, no private types — and the whole of poker, sealed-bid auctions, and signaling would be out of reach. This lesson is where the game-theoretic treatment of *uncertainty about the state of the game* enters, and it is the structural prerequisite for strategies (3.3), beliefs (4.x), and sequential equilibrium (4.6).

Formal Definition

Let $\Gamma = (N, T, P, A, u)$ be an extensive-form game. For each player i, let $T_i = \{t \in T \setminus Z : P(t) = i\}$ be the set of decision nodes owned by i.

An **information partition** for player i is a partition $\mathcal{I}_i$ of T_i into **information sets** such that:

1. **Same action set within a block.** For every $I \in \mathcal{I}_i$ and every $t, t' \in I$, $A(t) = A(t')$. (The acting player has the same set of available actions at every node in I; otherwise they could tell them apart from the choices they face.)
2. **No node precedes another in the same block.** No node $t \in I$ is an ancestor of any other $t' \in I$. (Otherwise the acting player would know the game had already visited t, and could distinguish.)

An information set with $|I| = 1$ is a **singleton** — the acting player knows exactly where they are. A game with only singleton information sets for every player has **perfect information**. An information set with $|I| > 1$ represents genuine uncertainty: the acting player knows the game is "somewhere in I" but not where.

The game is said to have **perfect recall** if each player, at any information set, remembers everything they knew and did at every previous own-information-set along the path. Formally: for every player i and every $I \in \mathcal{I}_i$, all nodes in I share the same sequence of i's past information sets and chosen actions.

Intuition

An information set is a "fog" over a group of nodes. When you reach the fog, you know you're somewhere in it, but you can't tell where. Since you have the same actions available at each node in the fog, you must choose one action for the whole information set — your action cannot depend on which specific node you're at, because you don't know.

Two canonical ways information sets arise:

Hidden type. A chance node deals you a private card. Subsequent decision nodes that differ only in which card you hold are in the same information set for the opponent (who can't see your card). For you, they are in different information sets (you can see your card). Poker is the quintessential example.

Simultaneous moves. Two players move at "the same time" — that is, neither observes the other's move before making their own. This is modeled as a sequential game where the second mover's decision nodes are all in a single information set (they can't tell which action the first mover took). Rock–Paper–Scissors becomes an extensive-form game: P1 picks, then P2 picks, but P2 has a 3-node information set that they can't distinguish.

Perfect recall is a technical but essential property: it says you don't forget your own past moves or what you knew at them. Kuhn's theorem (1.6) requires perfect recall. Most natural games have it; some abstractions (e.g., poker information-set bucketing) can introduce imperfect recall, which breaks the equivalence of behavioral and mixed strategies and the nice properties of CFR.

Structural Role

Information sets are the bridge between perfect-information games and the full generality of strategic interaction:

- **3.1 ⇒ 3.2.** Information sets refine the tree structure.
- **3.2 ⇒ 3.3.** Strategies are defined on information sets, not nodes. A pure strategy is a function $s_i : \mathcal{I}_i \to A$.
- **3.2 ⇒ 3.5.** Subgame perfection requires that "subgames" respect information sets — a subgame can only be rooted at a singleton information set.
- **3.2 ⇒ 4.1.** Bayesian games introduce type-dependent information sets via an initial chance move.
- **3.2 ⇒ 4.4.** Perfect Bayesian equilibrium adds beliefs — distributions over nodes within an information set — to handle play at non-singleton sets.
- **3.2 ⇒ 9.3.** CFR computes regret at each information set, not at each node.

Information sets are also the reason poker is a game rather than a decision problem: the opponent's hole cards are hidden, creating non-singleton information sets that force strategic reasoning about distributions over the unseen state.

Mathematical Development

Counting information sets

For a player whose decision nodes lie at d depths in the tree, and who can distinguish k_j information states at depth j, the number of information sets is $\sum_j k_j$. Strategies are functions from information sets to actions, so the strategy-space size is $\prod_{I \in \mathcal{I}_i} |A(I)|$, potentially much smaller than the raw node count.

Perfect recall condition

Formally: player i has perfect recall iff for every information set $I \in \mathcal{I}_i$ and every pair $t, t' \in I$, the sequences of i's own information sets and chosen actions along the path from the root to t and to t' are identical.

Perfect recall guarantees Kuhn's theorem: behavioral and mixed strategies yield the same outcome distributions. Without perfect recall, the two can differ: a mixed strategy can correlate actions at "the same" information set visited at different points, which a behavioral strategy cannot.

Simultaneous moves via information sets

To model a two-player simultaneous game in extensive form: P1 moves first (root decision node), then P2 moves. All of P2's decision nodes (one per P1 action) are placed in a single information set — P2 cannot distinguish them, which is what "simultaneity" means. P1's strategy is a single action; P2's strategy is also a single action (because P2 has only one information set). The extensive form reproduces the simultaneous game exactly.

Signaling games

In a signaling game, nature draws a type for the Sender, who then sends a message. The Receiver observes the message but not the type. This creates an information partition for the Receiver: messages that could have come from multiple types put the Receiver's decision nodes into the same information set. Node 4.5 develops signaling games in detail.

Example: simple poker

Two players each ante 1. Nature deals each player a card from $\{H, L\}$ uniformly. P1 sees only their own card, then decides Bet or Fold. If P1 Bets, P2 sees P1's action but not P1's card, then decides Call or Fold. If P2 Calls, cards are revealed and the higher card wins.

- P1's decision nodes: 2 nodes (one per P1's card). Each is its own information set (P1 knows their own card).
- P2's decision nodes: after each combination of (P1's card, P2's card, P1's action), but P2 doesn't see P1's card. P2 has an information set per (P2's card, P1's action = Bet) combination — 2 information sets total (one per P2's card, only reached after Bet).
- Total information sets: 2 for P1, 2 for P2.

Strategy space: P1's pure strategies are $\{B, F\}^2$ — 4 plans. P2's pure strategies are $\{C, F\}^2$ — 4 plans. Total profile space: $4 \times 4 = 16$.

Worked Example

The AKQ game

A classic toy poker game. Two players ante 1 each. Deck has 3 cards $\{A, K, Q\}$; each player gets one. P1 acts first: Bet 1 or Check. If P1 Bets, P2 may Call or Fold; cards revealed, higher card wins. If P1 Checks, cards revealed immediately at showdown.

Information structure. - P1's cards: $\{A, K, Q\}$, each with conditional prob 1/3, and given P1's card, P2's card is uniform on the remaining 2. - P1's information sets: 3 (one per P1's card). - P2's information sets: 3 (one per P2's card), but only reached after a P1 Bet.

Strategies. - P1 pure strategies: $2^3 = 8$ (action at each of 3 P1 information sets). - P2 pure strategies: $2^3 = 8$ (action at each of 3 P2 information sets, all conditional on P1 having bet).

Solution sketch. Equilibrium analysis on this game: P1 bets A always, mixes on Q (as a bluff), checks K; P2 calls with A always, mixes on K, folds Q. The mixing frequencies satisfy the indifference conditions. This is the smallest example where polarized value betting + bluff catching + bluffing are all present, and is the standard "first poker game" in game-theory courses.

Information-set constraint: same actions

Consider a tree where at node t player 1 has actions $\{A, B, C\}$ and at node t' player 1 has actions $\{A, B\}$. These two nodes **cannot** be placed in the same information set for player 1, because the action sets differ. The constraint $A(t) = A(t')$ within an information set is the way the formalism encodes that the player cannot distinguish by "what choices are presented."

In an information set, you know it is your turn and you know your action menu. What you don't know is the full history.

Poker Anchor

Concept mapping

Formal object	Poker instantiation
Player's information set	"What you know right now": your hole cards + public board + betting history

Formal object	Poker instantiation
Hidden hole cards	Opponent's 2 private cards are unknown to you → opponent's decision points with different hole cards are in the *same* information set from your perspective, different ones from theirs
Singleton information set	Your own decision points — you know your own cards
Non-singleton information set	Opponent's decision points (from your viewpoint)
Perfect recall	"I remember my own cards and past actions" — true in standard poker
Imperfect recall (pathological)	Only arises in abstractions that bucket "similar hands" together, collapsing some distinctions
Information set count (HUNL)	~10^{161} after a reasonable abstraction — the number a solver has to store regret values for

Concrete situation

Flop information set in HUNL. BTN raises preflop, BB calls. Flop comes 9♠7♡2♣. BB checks, BTN must decide a c-bet action.

What is BTN's information set here? BTN knows: (a) their own hole cards, (b) the three flop cards, (c) the full preflop betting history (BTN raise, BB call), (d) BB's check action on the flop. BTN does **not** know BB's hole cards.

BTN's information set is the set of all game histories consistent with this observable information — it contains one node per possible BB hole-card combination, since BTN cannot distinguish between them. There are $\binom{47}{2}$ − card conflicts ≈ 1000+ such nodes, all in the same information set from BTN's perspective.

A pure strategy for BTN at this information set must specify a single action — BTN cannot say "c-bet if BB has AK, check if BB has 72." BTN does not know BB's hand. The strategy is a single action (or a mixed distribution over actions) that applies to all of BB's possible holdings.

But BTN's own range matters: BTN knows their own hole cards, so each of BTN's possible hands has its own separate information set. If BTN holds AA, that is information set 1; if $72o$, that is information set 2. The solver outputs a different mixed strategy at each.

What this understanding buys you

You strategize over information sets, not nodes. When you think "what should I do with AQ on this flop," you are thinking about your strategy at a specific information set. You are not thinking about the specific opponent holding; that is unknown to you. The fog in the tree is exactly the content of strategic uncertainty.

Range vs hand. The distinction between "my range" and "my hand" maps directly to: my range is the distribution over my possible hole cards (the information partition opponents face), and my hand is the specific realization (which selects my own information set). Good players think about their range at every decision because that is what the opponent sees; they take an action from their specific hand's information set because that is what they personally face.

Abstraction and imperfect recall. When solvers abstract information sets (e.g., merging similar board textures to save compute), they risk introducing imperfect recall, which can subtly break optimality. This is one of the deep technical issues in modern poker solving, and it is solvable only via careful subgame resolving (Libratus, DeepStack).

Cross-Links

- **3.1 Game trees** — the structural carrier of information sets.
- **3.3 Strategies in extensive form** — functions on information sets.
- **3.4 Backward induction** — works cleanly only on singleton information sets.
- **3.5 Subgame perfection** — subgames must respect information sets.
- **4.1 Bayesian games** — type chance node generates non-singleton information sets.
- **4.4 Perfect Bayesian equilibrium** — adds beliefs at non-singleton sets.
- **4.6 Sequential equilibrium** — makes beliefs fully consistent.
- **9.3 CFR** — iterates regret at each information set.

Structural Compression

Invariant: An information set is a maximal set of decision nodes indistinguishable to the acting player — containing nodes with the same action menu, none of which is an ancestor of another — and pure strategies assign one action to each information set.

Mechanism: Partition structure on the set of decision nodes owned by each player; the perfect-recall condition ties the partition to the history of own-information and own-actions along each path.

Cross-System Echo: Information sets are the game-theoretic version of "observation histories" in partially observable Markov decision processes (POMDPs). In both settings, the decision-maker's effective state is not the world state but the equivalence class of world states consistent with their observations. Belief states, sufficient statistics, and filtration structures all generalize the information-set construction.

Exercises

1. Draw the extensive-form representation of Matching Pennies using information sets to model simultaneity.
2. Draw the extensive-form representation of the AKQ poker game, labeling each player's information sets.
3. Prove that the "same action set within an information set" axiom is necessary: give a game where violating it produces an ill-defined strategy.
4. Construct a game with perfect recall by giving a path and verifying the condition at each information set along it.
5. Construct a minimal imperfect-recall game and give a mixed strategy whose outcome distribution is not achievable by any behavioral strategy.
6. In a two-stage game where player 1 moves first but player 2 cannot see player 1's move, how many information sets does player 2 have? What is player 2's strategy space?
7. **Argue, structurally,** why information sets are the unique mechanism by which extensive-form games capture both hidden information (private types, hole cards) and simultaneous choice. What feature of tree structure makes this unification possible?

Strategies in Extensive Form

Position in Knowledge Graph

System: Game Theory **Structural Domain:** Extensive Form Games **Node:** 3.3

In the normal form, a strategy is a single choice from S_i. In the extensive form, it is a *function* mapping information sets to actions — a complete contingent plan of action that specifies what the player would do at every decision point. This node makes the strategy object formal and introduces the three representations — **pure**, **mixed**, and **behavioral** — that will be used throughout dynamic game theory. Kuhn's theorem ties them together under perfect recall. This is the node that lets us formally write down "what a poker solver stores," which is a behavioral strategy.

Formal Definition

Let Γ be an extensive-form game with information partitions $\mathcal{I}_i$ and action set $A(I)$ at each information set I.

Pure strategy. A function

$$s_i : \mathcal{I}_i \to \bigcup_{I \in \mathcal{I}_i} A(I) \quad \text{with } s_i(I) \in A(I).$$

The set of pure strategies is the Cartesian product $S_i = \prod_{I \in \mathcal{I}_i} A(I)$. Its cardinality is $\prod_{I \in \mathcal{I}_i} |A(I)|$, which is generally exponential in the number of information sets.

Mixed strategy. A probability distribution over pure strategies:

$$\sigma_i \in \Delta(S_i).$$

A mixed strategy can be implemented by randomizing once, at the start of the game, over a complete contingent plan.

Behavioral strategy. A collection of independent action distributions, one per information set:

$$\beta_i = (\beta_i(I))_{I \in \mathcal{I}_i}, \quad \beta_i(I) \in \Delta(A(I)).$$

A behavioral strategy can be implemented by randomizing independently at each information set. The space of behavioral strategies is $\prod_{I \in \mathcal{I}_i} \Delta(A(I))$, which has dimension only linear in the number of information sets.

Kuhn's theorem (perfect recall). Under perfect recall, every mixed strategy is equivalent (in outcome distribution) to some behavioral strategy, and vice versa. The two representations yield the same expected payoff against any opponent strategy.

Intuition

A pure strategy in extensive form is *exhaustive*: it tells you what you would do in every possible situation you could be in, including situations you will never actually reach in any specific play. A pure strategy is a complete contingency plan.

A mixed strategy randomizes *once* at the start of the game over complete contingency plans. You commit to a single plan but the plan itself is random.

A behavioral strategy randomizes *at each information set*, independently. When you reach an information set, you flip a coin (possibly biased) and choose an action based on the outcome.

These feel different but, under perfect recall, they are equivalent. Kuhn's theorem says: for any mixed strategy, you can construct a behavioral strategy with the same outcome, and vice versa. The proof uses perfect recall essentially: since you remember your own past actions, the behavioral decomposition can be done information-set-by-information-set without losing strategic content.

The practical consequence: solvers store behavioral strategies (linear space) rather than mixed strategies (exponential space). Everything we call "the GTO frequency at this info set" is the behavioral strategy's conditional distribution at that info set. Without Kuhn's theorem, we would be stuck storing mixed strategies, and the whole computation would be infeasible.

Structural Role

Extensive-form strategies underpin every downstream concept:

- **3.1–3.2 ⇒ 3.3.** The tree and information sets set the domain of strategies.
- **3.3 ⇒ 3.4.** Backward induction computes optimal actions at each information set.
- **3.3 ⇒ 3.5.** Subgame perfection requires optimality at every information set, not just on the equilibrium path.
- **3.3 ⇒ 4.4.** Perfect Bayesian equilibrium adds beliefs at information sets.
- **3.3 ⇒ 9.3.** CFR iterates over information sets, updating the behavioral strategy based on counterfactual regret at each.

Kuhn's theorem is also the reason extensive-form solutions are computationally feasible at all: the behavioral-strategy space is $O(|\mathcal{I}_i| \cdot |A|)$, whereas the mixed-strategy space is $O(|A|^{|\mathcal{I}_i|})$ — an exponential difference.

Mathematical Development

Equivalence of mixed and behavioral under perfect recall

Let σ_i be a mixed strategy. Define the **behavioral representation** β_i^σ as follows: at information set I, let $\Pr[\text{reach } I \mid \sigma_i]$ be the probability of reaching I when player i plays σ_i and the other players move to allow I to be reached. Set

$$\beta_i^\sigma(I)(a) = \frac{\Pr[\text{reach } I \text{ and play } a \text{ at } I \mid \sigma_i]}{\Pr[\text{reach } I \mid \sigma_i]} \quad \text{if the denominator is positive.}$$

Under perfect recall, this conditional distribution is well-defined (independent of opponent play through I), and the induced outcome distribution of β_i^σ against any opponent profile matches that of σ_i. This is the content of Kuhn's theorem.

Pure, mixed, behavioral in a small tree

Consider a tree where player 1 has 2 information sets, each with 2 actions.

- **Pure strategies:** $2 \times 2 = 4$. $((A, A), (A, B), (B, A), (B, B))$
- **Mixed strategies:** $\Delta(\{(A, A), (A, B), (B, A), (B, B)\})$, a 3-dimensional simplex.
- **Behavioral strategies:** $\Delta(\{A, B\}) \times \Delta(\{A, B\})$, a $1 \times 1 = 2$-dimensional rectangle (parametrized by two probabilities).

Under perfect recall, any mixed strategy σ_1 can be represented by a behavioral strategy β_1 with the same outcome distribution, even though σ_1 has 3 parameters and β_1 has only 2. The "missing" parameter is the correlation between information sets — which mixed can capture but behavioral cannot — but perfect recall makes such correlation strategically irrelevant.

Imperfect recall counterexample

Consider a game where player 1 moves at two information sets in sequence, but at the second information set player 1 has forgotten the first action. This is absent memory. A mixed strategy can correlate the two actions (e.g., always play AA or always play BB, 50/50 each). A behavioral strategy cannot: it mixes independently at each information set, so it would produce AA, AB, BA, BB each with probability 1/4.

The outcome distributions differ. Kuhn's theorem fails. This is why perfect recall is essential.

Reach probabilities

For a behavioral strategy β_i, the probability of reaching information set I is

$$\pi_i(I \mid \beta_i) = \prod_{(I',a) \preceq I} \beta_i(I')(a),$$

where the product is over i's own information sets on the path to I and the actions taken at them. The total reach probability of I, factoring in opponents and chance, is $\pi_i(I \mid \beta_i) \cdot \pi_{-i}(I)$, where $\pi_{-i}(I)$ is independent of β_i under perfect recall. This factorization is what makes CFR work.

Worked Example

AKQ game strategies

From the prior node's AKQ game. P1 has 3 information sets (one per P1's card), each with 2 actions (B, C). P2 has 3 information sets (one per P2's card, reached after P1 bets), each with 2 actions (C, F).

P1 pure strategies: $2^3 = 8$. Example: $s_1 = (B, C, B)$ means "bet A, check K, bet Q."

P1 behavioral strategy: 3 numbers in $[0, 1]$, one per information set, giving the bet probability.

P1 mixed strategy: distribution over 8 pure strategies — 7-dimensional simplex.

Equivalence check. A mixed strategy $\sigma_1 = 0.5 \cdot (B, C, B) + 0.5 \cdot (B, B, C)$ randomizes over two plans, each chosen with probability 0.5. Its behavioral representation: at the A info set, both plans play B, so $\beta_1(A)(B) = 1$. At the K info set, one plan plays C, the other plays B, each with prob 0.5, so $\beta_1(K)(B) = 0.5, \beta_1(K)(C) = 0.5$. At the Q info set, one plan plays B, the other plays C, each with prob 0.5, so $\beta_1(Q)(B) = 0.5, \beta_1(Q)(C) = 0.5$.

Behavioral strategy: $\beta_1 = (B \text{ prob } 1, B \text{ prob } 0.5, B \text{ prob } 0.5)$. Outcome distribution: same as σ_1 (under perfect recall).

Poker Anchor

Concept mapping

Formal object	Poker instantiation
Pure strategy s_i	A complete deterministic plan: "I will do X with every hand at every spot in every betting history"
Mixed strategy σ_i	A randomization over complete plans — a distribution over "whole strategies"
Behavioral strategy β_i	The object the solver outputs: a distribution over actions at *each* info set, independent per info set
Reach probability of an info set	How often, given your whole strategy, you end up at this exact decision point

Formal object	Poker instantiation
Perfect recall assumption	"I remember my own cards and my own past actions" — always true in real poker

Kuhn's theorem is why you can memorize solver output. Solver tables look like "at this info set, bet 62% / check 38%" — that is a behavioral strategy (local probabilities per info set). Without Kuhn's theorem, you would need to memorize a joint distribution over the astronomical pure-strategy space. With Kuhn's theorem, you only need one distribution per info set. This reduces the memorization problem from "impossible" to "merely huge."

Concrete situation

Solver output interpretation. You are drilling a flop spot with $AhKh$ on $Kd7s2c$ in a BTN vs BB single-raised pot. PioSolver tells you: - bet 33% pot: 73% - check: 27%

This is a **behavioral strategy** at your information set. The interpretation is "independently randomize at this spot with these frequencies." To implement, you need a real random device (second hand of a watch, suit of a lower hole card in a different range, an app).

Notice the number 73% is *not* a probability over complete strategies. It is the conditional probability of betting 33% *at this specific info set*. The solver also outputs independent mixes at every other info set. Your complete strategy is the product of all these conditional mixes, and — by Kuhn's theorem — this product is strategically equivalent to some mixed strategy over complete plans.

What this understanding buys you

Solver outputs are per-info-set, not joint. You do not need to correlate your flop check frequency with your turn mixing. Each info set is independent. This is the crucial simplification that makes memorization possible.

Range is a consequence of behavioral strategy + reach probabilities. Your c-bet range on a given board is the set of your hole cards for which the behavioral strategy at the corresponding info set puts positive probability on betting. This is a derived quantity, not an independent object. When a coach says "your betting range should be X," they are saying "your behavioral strategy at this info set should put weight on betting with these hole cards."

Imperfect recall is a real risk in abstractions. Solvers that bucket hands together (treating AKs and AQs identically at some info sets, e.g.) can introduce imperfect recall. When this happens, Kuhn's theorem fails, and the behavioral output is no longer equivalent to a mixed strategy, which means the "solution" may not be a Nash equilibrium of the original game. Libratus and DeepStack handled this with careful re-solving at abstraction boundaries.

Cross-Links

- **1.6 Normal form as universal representation** — extensive strategies flatten to normal-form pure strategies.
- **3.1 Game trees** — the domain.
- **3.2 Information sets** — the argument domain of strategies.
- **3.4 Backward induction** — computes optimal pure strategies.
- **3.5 Subgame perfect equilibrium** — requires optimality at every info set.
- **4.4 Perfect Bayesian equilibrium** — adds belief systems.
- **9.3 CFR** — iterates on behavioral strategies directly.
- **POMDPs** — the computer-science analogue.

Structural Compression

Invariant: A strategy in an extensive-form game is a function from information sets to actions (pure), a distribution over such functions (mixed), or a collection of independent conditional distributions at each information set (behavioral). Under perfect recall, all three representations yield the same outcome distributions.

Mechanism: Factorization of strategic randomization across information sets; Kuhn's theorem establishes equivalence between local (behavioral) and global (mixed) randomization under memory-preservation constraints.

Cross-System Echo: The equivalence of mixed and behavioral strategies under perfect recall mirrors the equivalence of Markov policies and history-dependent policies in reinforcement learning under appropriate informational conditions. In both cases, a "local" rule (behavioral, Markov) suffices when the agent has adequate memory.

Exercises

1. Count the pure strategies for each player in the two-stage entry game (Node 3.1).
2. For the AKQ game, write out a pure strategy for P1 that dominates a specific alternative pure strategy. Verify the dominance.
3. Prove Kuhn's theorem in the special case of a game with two information sets.
4. Construct a game with imperfect recall and exhibit a mixed strategy whose outcome distribution is not achievable by any behavioral strategy.
5. For a given behavioral strategy on a tree of depth 4, compute the reach probability of a specific leaf.
6. Show that the behavioral-strategy space has dimension linear in the number of information sets while the mixed-strategy space has dimension exponential in it.
7. **Argue, structurally,** why perfect recall is the minimal assumption that equates mixed and behavioral strategies, and why this equivalence is the single most important reason extensive-form game theory is computationally feasible.

Backward Induction

Position in Knowledge Graph

System: Game Theory **Structural Domain:** Extensive Form Games **Node:** 3.4

Backward induction is the dynamic-programming solution of finite perfect-information games. Starting from terminal nodes and working toward the root, at each decision node you compute the action that maximizes the acting player's payoff given the optimal continuation strategies already computed for the subtree rooted at that node. The result is a pure strategy profile — the **backward induction equilibrium** — that is guaranteed to be a Nash equilibrium and, further, is subgame perfect (Node 3.5). For games without perfect information, backward induction on node values fails, but a generalization — backward induction on *information sets*, with belief updates and value recursion — underlies every modern poker solver. "River-first solving" is the poker-specific implementation of backward induction.

Formal Definition

Let Γ be a finite perfect-information extensive-form game (all information sets are singletons). Define the **backward induction value** $V_i(t)$ for every node $t \in T$ recursively:

Base case (terminal nodes). If $t \in Z$, $V_i(t) = u_i(t)$ for every player i.

Recursive case (decision nodes). If $P(t) = j$ and t's children are $\{t_a\}_{a \in A(t)}$ (with t_a reached by action a):

$$a^*(t) = \arg\max_{a \in A(t)} V_j(t_a), \qquad V_i(t) = V_i(t_{a^*(t)}) \text{ for every } i.$$

Chance nodes. If $P(t) = 0$ with distribution $p_t \in \Delta(A(t))$:

$$V_i(t) = \sum_{a \in A(t)} p_t(a) V_i(t_a).$$

The **backward induction equilibrium** is the pure strategy profile $s^{BI} = (a^*(t))_t$ restricted to each player's decision nodes, and the **backward induction outcome** is the play of the game under s^{BI} (from root).

Theorem (Zermelo–Kuhn). Every finite perfect-information game has at least one backward induction equilibrium, and every such equilibrium is a Nash equilibrium of the game.

Intuition

Think of backward induction as reasoning "from the end." Start at the leaves where payoffs are known. At the last-to-move player's last decision, they will maximize their own payoff among their available continuations. Since you know what they'll do, the node *above* that is effectively simpler — its value is the value of the choice you just determined. Work your way up the tree one level at a time.

At the root, you have the value of the game under optimal play. Moreover, the specific actions chosen at each step constitute a pure strategy profile that is the optimal play.

Two key features:

Credibility. At each node, the player whose turn it is chooses the action they would *actually take if the node were reached.* This rules out incredible threats: no player commits to suboptimal off-path behavior, because the backward induction explicitly computes the optimal thing to do if the node were reached. This is exactly the refinement that subgame perfection (3.5) captures in general.

Complete information required. Backward induction on *nodes* works only in perfect-information games. If there are information sets with multiple nodes, you cannot compute "the value of node t" because the acting player doesn't know they're at t rather than at some other node in the same info set. In this case, backward induction must be generalized to work on information sets with belief updates (previewing Node 4.4).

Structural Role

Backward induction is the computational engine of dynamic game theory:

- **3.4 ⇒ 3.5.** Backward induction equilibria are subgame perfect.
- **3.4 ⇒ 3.6.** The centipede game reveals the puzzle of backward induction in finitely repeated interaction.
- **3.4 ⇒ 5.1.** In finitely repeated games, backward induction gives the unique unraveling result (the stage Nash repeated).
- **3.4 ⇒ 4.4.** Perfect Bayesian equilibrium generalizes backward induction to imperfect information.
- **3.4 ⇒ 9.3.** CFR is a learning-theoretic analogue of backward induction: it iteratively updates counterfactual values at each information set.
- **3.4 ⇒ Poker solving.** "River-first solving" is backward induction with beliefs at each information set.

Mathematical Development

Termination and correctness

In a finite tree, backward induction terminates in $O(|T|)$ node evaluations. At each step, the acting player's expected payoff is computed under the assumption that the continuation subtree plays optimally (which was computed in previous steps). Induction on tree depth establishes that the result is a Nash equilibrium.

Zermelo's theorem (1913)

Zermelo proved that in any finite two-player zero-sum game of perfect information without chance, one of the following holds: - Player 1 has a winning strategy. - Player 2 has a winning strategy. - Both have drawing strategies.

The proof is backward induction applied to tic-tac-toe, chess, checkers, etc. The existence of a winning/drawing strategy follows from finiteness alone. Computing the strategy may be infeasible (chess has ~10^{46} positions).

Centipede puzzle (preview)

Consider a finitely repeated Centipede game: two players alternate, at each stage deciding to "Take" (end the game with a split) or "Pass" (double the pot and hand to the opponent). Backward induction predicts that at the last stage, the player who is to move Takes (to get more than passing would yield after the opponent takes). At the second-to-last stage, knowing the opponent will Take, you Take. Iterating, the unique backward induction equilibrium is "Take immediately" — a Pareto-inefficient outcome.

Empirically, people do not play "Take immediately." This is the puzzle of Node 3.6.

Extensions to imperfect information

For games with non-singleton information sets, backward induction must be replaced by a belief-augmented recursion. The value at an information set depends on:

1. Beliefs about which node within the information set the game is actually at.
2. Optimal play in the subtree beyond the information set.

This is the structure of perfect Bayesian equilibrium (Node 4.4) and sequential equilibrium (Node 4.6). Solver algorithms like CFR compute this recursively.

Worked Example

A three-level perfect-information game

P1 moves at root: A or B. - If A, P2 moves: a or b. - If a, payoff $(3, 1)$. - If b, payoff $(1, 3)$. - If B, P2 moves: c or d. - If c, payoff $(2, 2)$. - If d, payoff $(0, 4)$.

Backward induction.

Level 3 (terminal nodes): values are as given.

Level 2 (P2 nodes): - At the a/b node: P2 compares $b \to 3$ vs $a \to 1$, chooses b. Value $(1, 3)$. - At the c/d node: P2 compares $c \to 2$ vs $d \to 4$, chooses d. Value $(0, 4)$.

Level 1 (P1 root): - A leads to value $(1, 3)$, so P1 gets 1. - B leads to value $(0, 4)$, so P1 gets 0. - P1 chooses A.

Backward induction equilibrium: P1 plays A; P2 plays b (if reached) and d (if reached). Outcome: $(1, 3)$.

Normal form. P1 has 2 pure strategies: $\{A, B\}$. P2 has 4 pure strategies: $\{ab \text{ conditionals}\}, \{ac\}, \ldots$ — actually $2 \times 2 = 4$ pure plans. Normal form is 2×4 with entries determined by the combination of plans. The backward induction equilibrium is one of its Nash equilibria; there may be others that rely on incredible threats at off-path P2 nodes.

Stackelberg duopoly via backward induction

Two firms, Leader and Follower. Leader picks quantity $q_1 \geq 0$. Follower observes q_1 and picks $q_2 \geq 0$. Demand $P = a - q_1 - q_2$, marginal cost c.

Stage 2 (Follower). $\max_{q_2}(a - q_1 - q_2 - c)q_2 \Rightarrow q_2 = (a - c - q_1)/2$.

Stage 1 (Leader). Knowing q_2 as a function of q_1, maximize $\pi_1(q_1) = (a - q_1 - q_2(q_1) - c)q_1 = (a - c - q_1 - (a - c - q_1)/2)q_1 = \frac{(a-c-q_1)q_1}{2}$. FOC: $q_1 = (a - c)/2$.

Then $q_2 = (a - c - (a - c)/2)/2 = (a - c)/4$. Total output $3(a - c)/4$, price $P = a - 3(a - c)/4 = (a + 3c)/4$, Leader profit $\pi_1 = (a - c)^2/8$, Follower profit $\pi_2 = (a - c)^2/16$.

Comparison to Cournot. Cournot (simultaneous) equilibrium is $q_1 = q_2 = (a - c)/3$, each earning $(a - c)^2/9$. Stackelberg Leader does better $(1/8 > 1/9)$, Stackelberg Follower does worse $(1/16 < 1/9)$. This is the first-mover advantage: by committing credibly to a quantity, Leader changes Follower's best response.

Poker Anchor

Concept mapping

Formal object	Poker instantiation
Backward induction	"River-first solving" — compute optimal river play, then turn given river play, then flop given turn play
Leaf payoffs	Showdown outcomes — chip delta at the end of a hand
Value at node	Expected chip EV of a subtree under optimal continuation play

Formal object	Poker instantiation
Credible continuation	A solver's "tree-exploitability" check — would opponent actually play the prescribed line if reached?
Chance nodes	Flop, turn, river deals, averaged with uniform weight

River-first solving = backward induction. Every modern poker solver operates by computing optimal play street-by-street, backward. First, it computes the optimal river play given the range distributions that reach the river. Then, knowing river EVs, it computes optimal turn play. Then flop. Then preflop. Each step replaces a subtree with a value computed from the one below.

This is literally backward induction in the sense of this node. The added wrinkle is the information-set structure: at each decision, you have a *distribution* over hole cards (your range), and the solver must compute value per range element, not per single hand. The recursion is the same; the object at each node is a vector (values for each hand in the range) rather than a scalar.

Concrete situation

Solving a HUNL turn decision. You are on the turn with your entire range vs opponent's range. The board is $9\heartsuit 8\heartsuit 3\clubsuit K\diamondsuit$ after BTN c-bet on the flop was called. BTN decides on a turn barrel.

River values (computed first). For each possible river card (44 cards), and for each pair (your hand, villain's hand), the solver computes the EV of the river decision. This gives a lookup table: "if river is Ah and my hand is $A\heartsuit T\heartsuit$ and villain's hand is $K\clubsuit Q\clubsuit$, my river-optimal EV is X chips."

Turn values (computed next). Given the river lookup table, the solver computes the turn decision EV: for each turn action (bet/check/bet size), average over river cards with equity-weighted probabilities, using the river lookup for each. This produces a turn-optimal action distribution for each hand combination at the current turn info set.

Iterate up to flop and preflop. The flop decision is computed knowing the turn-optimal values. Preflop is computed knowing the flop-optimal values. The whole thing is a bottom-up (leaf-to-root) dynamic programming computation on the game tree.

What this understanding buys you

"River-first" is not a heuristic; it's the algorithm. When a solver tells you "bet 62%" on the turn, it computed that frequency by first solving the river under every possible turn outcome and then propagating those values backward. You cannot solve the turn in isolation because turn values *depend on* river values, which depend on the range distributions that reach the river, which depend on your turn action. The circular dependency is resolved by computing from the leaves (river) up.

Subgame re-solving = re-running backward induction from a specific node. If you want to solve just "the turn in this exact spot," you take the turn as your root, fix the input range distributions, and run backward induction on the turn-and-river subtree. This is standard practice in modern poker study.

Centipede-like effects in tournaments. In tournament situations with ICM pressures, iterated backward induction can predict "everyone folds forever" at the bubble, which is manifestly wrong in practice. This is the same puzzle as the Centipede game (Node 3.6): empirical play deviates from backward induction because players are not common-knowledge rational, and ICM distortions further compound this.

Cross-Links

- **3.1 Game trees** — the object backward induction operates on.
- **3.2 Information sets** — singleton info sets for pure backward induction.
- **3.3 Strategies in extensive form** — backward induction outputs a pure strategy.

- **3.5 Subgame perfect equilibrium** — backward induction equilibria are SPE.
- **3.6 Centipede** — the puzzle of backward induction's behavioral predictions.
- **4.4 Perfect Bayesian equilibrium** — belief-augmented generalization.
- **9.3 CFR** — learning-theoretic analogue.
- **Dynamic programming (CS)** — the algorithmic family.

Structural Compression

Invariant: In a finite perfect-information game, the optimal action at every decision node depends only on the subtree rooted at that node; this locality enables bottom-up computation of optimal continuation values and produces a pure Nash equilibrium that is credible at every node.

Mechanism: Recursive value function $V_i(t)$ defined by maximization over child values at decision nodes and expectation over child values at chance nodes, computed from leaves to root.

Cross-System Echo: Backward induction is the same algorithm as value iteration in dynamic programming, the Bellman equation in Markov decision processes, and the minimax search in game-tree AI. Each computes optimal policy bottom-up using the principle of optimality (Bellman 1957).

Exercises

1. Apply backward induction to a 4-stage centipede game with doubling payoffs. Verify the unique backward induction equilibrium is "Take at stage 1."
2. For the Stackelberg duopoly, verify the profits $(a-c)^2/8$ and $(a-c)^2/16$ by direct substitution.
3. Show that in any finite perfect-information game, every backward induction equilibrium is a Nash equilibrium.
4. Construct a perfect-information game with multiple backward induction equilibria. What causes the non-uniqueness?
5. Apply backward induction to a 3-stage sequential auction (two bidders alternate bids), with valuations drawn at the start.
6. In the Stackelberg setup, what changes if the Leader cannot credibly commit to q_1? Derive the equilibrium of the simultaneous version and compare.
7. **Argue, structurally,** why backward induction fails in games with non-singleton information sets. What replaces the pointwise "optimal action at each node" rule? (Hint: beliefs.)

Subgame Perfect Equilibrium

Position in Knowledge Graph

System: Game Theory **Structural Domain:** Extensive Form Games **Node:** 3.5

The entry deterrence game (Node 1.6) revealed that the normal form of an extensive-form game can contain Nash equilibria supported by **incredible threats** — off-path behavior that is not individually rational but that equilibrium conditions cannot punish because it is never reached. Subgame perfect equilibrium (Selten, 1965) is the refinement that eliminates such equilibria. It requires that the strategy profile induce a Nash equilibrium not only in the whole game but in every **subgame** — every part of the tree rooted at a singleton information set. It is the extensive-form refinement of Nash equilibrium, and it is what backward induction computes in perfect-information games. For imperfect-information games, the refinement extends to perfect Bayesian equilibrium (Node 4.4) and sequential equilibrium (Node 4.6).

Formal Definition

Let Γ be an extensive-form game. A **subgame** of Γ is a subtree Γ' rooted at some decision node t such that:

1. The root t is a **singleton information set** (the acting player knows they are at t and nowhere else).
2. For every information set I that meets the subtree rooted at t, either $I \subseteq \Gamma'$ or $I \cap \Gamma' = \emptyset$. (Information sets cannot be "split" by the subgame.)
3. The subgame inherits players, actions, chance probabilities, and terminal payoffs from Γ.

A **subgame perfect equilibrium (SPE)** of Γ is a strategy profile σ^* such that the restriction of σ^* to every subgame is a Nash equilibrium of that subgame.

Theorem (Selten). In every finite perfect-information game, backward induction produces a subgame perfect equilibrium. Every perfect-information game has at least one SPE; in generic games it is unique.

Nash $\supseteq$ SPE. Every SPE is a Nash equilibrium (take the subgame to be Γ itself). Not every Nash equilibrium is SPE; the non-SPE Nash equilibria are precisely those relying on incredible off-path behavior.

Intuition

Nash equilibrium requires "nobody has a profitable deviation at the root of the game." SPE requires "nobody has a profitable deviation at the root of *any subgame*." The extra condition is that, even at nodes that would not be reached in equilibrium, the strategy prescribes optimal play conditional on being there.

The motivation is **credibility**. If a player commits to a threat ("I will fight if you enter") that would be suboptimal to execute if the threat were tested, the threat is not credible. A rational opponent should not fear it, and equilibrium reasoning that relies on fear of the threat is not robust. SPE formalizes this by requiring optimality everywhere in the tree, not just on the equilibrium path.

A second way to see it: SPE is "Nash equilibrium in every subtree." The normal form loses subtree-level information (Node 1.6), and SPE is the refinement that recovers it.

The canonical SPE computation algorithm is backward induction: at each subtree, the optimal action is chosen given optimal continuations. For perfect-information games this is exact; for imperfect-information games, subgames only exist at singleton information sets, which may be rare, and the refinement must be further strengthened (Node 4.4, 4.6).

Structural Role

SPE is the extensive-form strengthening of Nash equilibrium:

- **2.1 Nash ⇒ 3.5 SPE.** SPE is a strict refinement: fewer equilibria, eliminating incredible threats.
- **3.4 Backward induction ⇒ 3.5.** Backward induction produces SPE in perfect-info games.
- **3.5 ⇒ 3.6 Centipede.** SPE is the solution concept that predicts "Take immediately" in Centipede, setting up the empirical puzzle.
- **3.5 ⇒ 5.x Repeated games.** In finitely repeated games, SPE unraveling forces the unique stage-Nash profile; in infinitely repeated games, the Folk Theorem characterizes the SPE payoff set.
- **3.5 ⇒ 4.4 PBE.** For imperfect-information games, SPE is too weak; PBE extends it.
- **3.5 ⇒ 10.4 Commitment.** Credibility of commitment is directly about whether a threat survives SPE refinement.

Mathematical Development

Subgame identification

A subgame is rooted at a singleton information set. In perfect-information games, every decision node is a singleton, so every subtree is a subgame. In games with non-singleton information sets, most subtrees are *not* subgames — their root is inside an information set with other nodes, so the acting player doesn't know they are at the root of the subtree.

Example: AKQ poker. After the chance deals cards, P1 has 3 singleton info sets (one per card). Each P1-level decision is the root of a subgame (the subtree starting from P1's decision). After P1's action, P2's info set (one per P2's card, conditioned on P1 betting) is also a singleton. So there are subgames rooted at each P2 decision. But the *whole game* — starting from chance — is the only global subgame; the chance node is the root, and it has multiple "positions" (for each (P1 card, P2 card) pair) but chance is distinct from players so the subgame definition still works at the chance-root level.

Imperfect-information games have fewer subgames than decision nodes, and SPE is consequently weaker.

One-shot deviation principle

Theorem. In a finite extensive-form game, a strategy profile is SPE iff no player can profitably deviate at any single information set holding all other play (their own and opponents') fixed.

This reduces the SPE check from "consider all possible deviations" to "check one-shot deviations at each information set." It is the discrete-time analogue of the Bellman principle: optimality is preserved under single-step deviations. Proof is by induction on the tree.

Selten's example

Selten (1965) introduced the chain store paradox to motivate SPE. A chain store operates in many small towns; in each, an entrant may enter or stay out. The chain store prefers to "fight" (drive out entrants) to build a reputation, even at a one-town loss, so that future entrants fear entry. But at the last town, the store has no future reputation to protect, so fighting is irrational. Backward induction unravels: at town k, fighting is irrational; knowing this, town $k-1$ entrant enters; knowing this, fighting at $k-1$ is irrational too; iterating, the store never fights, and every entrant enters. The "reputation strategy" fails SPE at every subgame.

The paradox is that this backward induction prediction contradicts intuition and experiment. It is resolved only by introducing incomplete information about the store's type (Node 5.5).

SPE in stochastic games and bargaining

Rubinstein's alternating-offer bargaining game (1982) has a unique SPE computable by backward induction on the continuation values: with discount factor δ, the proposer offers $\frac{1-\delta}{1-\delta^2} = \frac{1}{1+\delta}$ to themselves. This uniqueness is the famous Rubinstein result and shows how SPE can pin down a single outcome in games with continuous action spaces.

Worked Example

Entry deterrence with SPE check

Entry deterrence (Node 1.6): Entrant Out $\rightarrow (0, 2)$; Entrant In, Incumbent Accommodate $\rightarrow (1, 1)$; Entrant In, Incumbent Fight $\rightarrow (-1, -1)$.

Nash equilibria of normal form. Two pure: (In, Accommodate) and (Out, Fight).

SPE check. The whole game is one subgame. Is (In, Accommodate) SPE? The only non-root decision is the Incumbent's — and in this subgame (rooted at Incumbent's node), Accommodate gives 1 vs Fight giving -1. Accommodate is the optimal action. ✓

Is (Out, Fight) SPE? The subgame rooted at the Incumbent's node has Fight as the strategy. But Fight is suboptimal in that subgame: Accommodate gives 1 vs Fight giving -1. Not a Nash of the subgame. Not SPE. $\times$

Conclusion. Unique SPE: (In, Accommodate). The equilibrium (Out, Fight) is eliminated.

Rubinstein bargaining (brief)

Two players alternate making offers to split 1 unit. Each period without agreement discounts by $\delta \in (0, 1)$. P1 offers, P2 accepts/rejects; if rejects, P2 offers, and so on.

Backward induction via stationary SPE. Let x^* be P1's equilibrium offer to themselves. In the SPE, P1's offer gives P2 exactly the discounted continuation value $\delta(1 - x^*)$. P2 accepts iff $1 - x^* \geq \delta(1 - x^*)$, which holds. P2's counteroffer (if reached) would give themselves x^* (by symmetry), so P2's continuation value is δx^*. P1's indifference condition: $x^* = 1 - \delta \cdot (\delta x^*) \Rightarrow x^* = \frac{1}{1+\delta}$.

Unique SPE: P1 offers $(x^*, 1 - x^*) = (1/(1 + \delta), \delta/(1 + \delta))$ and P2 accepts immediately.

Poker Anchor

Concept mapping

Formal object	Poker instantiation
Subgame	A subtree of a hand — e.g., "turn onward given this board and betting history"
SPE	Every subgame's play is Nash of that subgame — i.e., the strategy is credible at every decision point
Non-SPE Nash	A Nash equilibrium supported by incredible off-path threats — rare in solver output because solvers enforce SPE-like optimality
One-shot deviation principle	Check your play at each info set: would a single deviation, holding everything else fixed, improve EV? If no for every info set, it's SPE.
Credible commitment	"I will always 3-bet aggressively" — not credible without a mechanism enforcing it

Concrete situation

Subgame decomposition for study. When you load a flop spot into a solver and "solve" it, you are solving a subgame: the subtree rooted at the flop decision, given fixed preflop ranges (the input). The solver output is an SPE of this subgame — it is Nash within the subgame, and because the subtree is perfectly-information-set at its root (the acting player knows their own hand and the board), the subgame root is a valid singleton info set in the usual sense.

Why this decomposition is legitimate. SPE guarantees that if you know the correct input ranges at the root of a subgame, the optimal play *within* the subgame can be computed independently of the larger game. This is exactly what "solving a flop in isolation" relies on. Libratus and DeepStack used this to do "re-solving" during play: when a new state was reached, they solved the subgame rooted there, using the opponent's range as previously computed.

Incredible threats in poker. A pure Nash equilibrium where a player commits to "I will always 3-bet shove with 72o preflop, never bluff with suited connectors" can exist in the normal form if the opponent's belief is consistent with it. But it fails SPE: at the 3-betting information set, the optimal action is not to shove 72o — it is to fold. Any realistic solver refuses such an equilibrium because it is not SPE.

What this understanding buys you

"Solving a subgame" is rigorous because of SPE. When you freeze the preflop ranges and solve the flop in isolation, you are relying on the SPE property: optimal play in a subtree depends only on the subtree plus the input ranges at its root. This is why subgame solvers work — and also why getting the input ranges wrong is catastrophic (the "subgame" you're solving isn't actually a subgame of the right game).

Every solver output is SPE, by construction. CFR iterates toward an SPE by ensuring that no one-shot deviation at any information set is profitable. The output is not just Nash; it is subgame perfect. Any claim "this is the GTO play at this info set" is implicitly a claim that the full strategy is SPE.

Incredible commitments do not work in poker. If you try to adopt a strategy like "I always shove with this" that would be suboptimal at the info set in question, the solver (and a good human opponent) will exploit you. There is no hidden commitment mechanism; your strategy is always executing at the info set you are at, and it must be optimal there.

Cross-Links

- **1.6 Normal form as universal representation** — the flattening that loses SPE information.
- **2.1 Nash equilibrium** — SPE is a refinement.
- **3.1–3.4** — the extensive-form machinery.
- **3.6 Centipede game** — puzzle of SPE empirical predictions.
- **4.4 Perfect Bayesian equilibrium** — extension to imperfect information.
- **5.x Repeated games** — finitely repeated unraveling via SPE.
- **10.4 Commitment and credibility** — SPE formalizes credibility.

Structural Compression

Invariant: A subgame perfect equilibrium is a strategy profile whose restriction to every subgame is a Nash equilibrium of that subgame — equivalently, no one-shot deviation at any information set is profitable.

Mechanism: Recursive optimality on the tree: backward induction or its generalization to imperfect-information games, enforced by the one-shot deviation principle.

Cross-System Echo: Subgame perfection is the game-theoretic analogue of the principle of optimality in dynamic programming (Bellman) and the markov property in stochastic processes. In all three, local optimality at every state/time implies global optimality, and vice versa.

Exercises

1. Verify that the Stackelberg duopoly equilibrium is SPE.
2. Construct a 2-player extensive-form game with two Nash equilibria, exactly one of which is SPE. Identify the incredible threat.
3. Apply the one-shot deviation principle to verify that the backward induction equilibrium of a 3-stage perfect-info game is SPE.
4. In Rubinstein bargaining, derive the unique SPE by writing the two indifference conditions at the proposer's info set and the responder's.
5. Show that in a perfect-information game, every SPE is a backward induction equilibrium.
6. Construct an imperfect-information extensive-form game where SPE fails to pin down a unique equilibrium. What refinement would be needed?
7. **Argue, structurally,** why subgame perfection is the "right" refinement of Nash in extensive-form games with perfect information. What specific strategic pathology does it rule out, and why is that pathology a problem?

The Centipede Game and Common Knowledge

Position in Knowledge Graph

System: Game Theory **Structural Domain:** Extensive Form Games **Node:** 3.6

The Centipede game (Rosenthal, 1981) is a finite-horizon perfect-information game whose unique backward induction / SPE prediction is "Take immediately, at the very first stage." Empirically, people do not play this way — they pass for several rounds before taking, generating a collective surplus that SPE says should not arise. The Centipede is the paradigmatic puzzle of backward induction: it shows that the prediction of a well-defined solution concept can fail in experiments, and it motivates refinements involving bounded rationality, incomplete information about rationality, and common-knowledge failures. This lesson closes Module 3 by confronting the limits of equilibrium reasoning and motivating Modules 4 (incomplete information) and 5 (repeated games with reputation).

Formal Definition

Centipede game (Rosenthal, 1981). Two players alternate over n stages. At each stage $k \in \{1, \ldots, n\}$, the player whose turn it is decides to **Take** (T) or **Pass** (P).

- If Take at stage k, the game ends with payoffs that depend on k and who moves (specified below).
- If Pass, the pot doubles and the next stage begins.
- If all n stages Pass, the game ends with the maximum pot, split according to some fixed rule.

Concretely (standard version): at stage k, if the moving player Takes, they receive $2^{k-1} + c$ and the other receives $2^{k-1} - c$ for some $c > 0$. Passing doubles the pot. So the immediate Take is always marginally better than passing for the mover *at that specific stage*, because after passing, the opponent will Take next and leave the mover with less. But if both players passed throughout, the total surplus would be much larger.

SPE / backward induction prediction. Stage n: mover Takes (higher than passing leaves them). Stage $n-1$: knowing stage n mover will Take, mover at $n-1$ compares Take vs Pass (which leads to being Taken on by opponent); Take is better. Iterating: at every stage, Take is the unique SPE action. Unique SPE outcome: mover at stage 1 Takes immediately, receiving $1 + c$, other gets $1 - c$.

Empirical observation. McKelvey and Palfrey (1992) ran experiments: ~75% of pairs pass at the first stage, ~40% pass at the second, ~20% pass at the third, and about 1% make it to the last stage before one takes. Nobody plays "Take immediately."

Intuition

The Centipede game is strategically bizarre: the SPE prediction is Pareto-dominated by almost every other possible outcome. Both players would be better off if both Passed several times, but neither can credibly commit to Passing, because at any stage Taking is strictly individually rational (given the opponent will Take next).

The backward induction unravels from the last stage. You can't pass at the last stage because the game ends with the pot split badly for you. You can't pass at the second-to-last, because knowing the opponent will Take, you get less. Iterating, unraveling propagates all the way to stage 1, and you Take immediately.

Why experiments diverge. Several explanations compete:

1. **Bounded rationality.** Subjects cannot reason through all n steps of backward induction. Evidence: deeper centipedes have faster unraveling than shallow ones, inconsistent with full backward induction.

2. **Incomplete information about rationality.** If a player thinks there is even a small probability that the opponent is irrational (plays Pass even when Taking is better), then early Passing becomes rational: you Pass to "test" whether the opponent will Pass too, with the hope of a large future reward. Kreps–Milgrom–Roberts–Wilson (1982) formalized this in the context of the chain store paradox.
3. **Other-regarding preferences.** Players care about fairness or reciprocity, not just own payoff. Passing is cooperative, Taking is defection; the true payoff function includes a fairness term.
4. **Common knowledge failure.** Aumann (1995) showed that SPE requires common knowledge of rationality, and this is a strong assumption. Relaxing it allows more equilibria.

Any of these resolves the paradox. All are interesting. All motivate topics in later modules.

Structural Role

The Centipede game is the bridge between Module 3 (SPE) and Modules 4, 5, 10:

- **3.4, 3.5 ⇒ 3.6.** SPE gives the unravel prediction.
- **3.6 ⇒ 4.x.** Incomplete information about rationality is developed via Bayesian games.
- **3.6 ⇒ 5.5 Chain store paradox.** The same unravel logic, with reputation as the mechanism to escape it.
- **3.6 ⇒ 10.4 Commitment.** If players could commit to passing for k stages, both would do strictly better.
- **3.6 ⇒ 9.x Learning.** Learning dynamics can converge to non-SPE equilibria, especially when common knowledge of rationality is relaxed.

Mathematical Development

Backward induction on the standard centipede

Consider $n = 6$ centipede with starting pot 2 and doubling. Payoffs at stage k (mover takes): mover gets $(3/4) \cdot 2^k$, other gets $(1/4) \cdot 2^k$. Passing doubles the pot.

Stage	Mover	Take payoff (mover, other)	Pass leads to
1	P1	(1.5, 0.5)	Stage 2 with pot 4
2	P2	(3, 1)	Stage 3 with pot 8
3	P1	(6, 2)	Stage 4 with pot 16
4	P2	(12, 4)	Stage 5 with pot 32
5	P1	(24, 8)	Stage 6 with pot 64
6	P2	(48, 16)	End with pot 128 split (96, 32)

Backward induction from stage 6: P2 compares Take (48) to Pass (32 from end) → Take.

Stage 5: P1 compares Take (24) to Pass leading to stage 6 where P2 Takes, leaving P1 with 16 → Take ($24 > 16$).

Stage 4: P2 compares Take (12) to Pass leading to stage 5 where P1 Takes (leaving P2 with 8) → Take ($12 > 8$).

Stage 3: P1 Take (6) vs Pass → P2 Takes at stage 4 leaving P1 with 4 → Take ($6 > 4$).

Stage 2: P2 Take (3) vs Pass → P1 Takes at stage 3 leaving P2 with 2 → Take ($3 > 2$).

Stage 1: P1 Take (1.5) vs Pass → P2 Takes at stage 2 leaving P1 with 1 → Take ($1.5 > 1$).

Unique SPE outcome: Take at stage 1. Payoff (1.5, 0.5). The reachable surplus (128) is never realized.

Pareto inefficiency

The SPE outcome dominates no other outcome — it has total surplus 2, while "both pass to the end" gives 128, with both players much better off than in SPE. The SPE outcome is Pareto-dominated by "pass all 6" and by "pass for several stages then take."

Kreps–Milgrom–Roberts–Wilson: tiny irrationality resolves the paradox

Add $\epsilon > 0$ probability that the opponent is an "irrational always-Pass type." Then Passing at stage 1 gives some chance of reaching stage 2 where the rational player might also Pass (to maintain the pretense of being irrational, for large future gains). The equilibrium of the perturbed game has substantial early passing, converging to the observed empirical behavior as $\epsilon \to 0$.

This shows the SPE of the full-rationality game is *not continuous in the common-knowledge assumption.* A tiny epistemic perturbation changes predictions drastically.

Worked Example

Empirical data from McKelvey–Palfrey (1992)

In a 4-stage centipede with increasing doublings: - ~80% of pairs Pass stage 1. - ~50% Pass stage 2. - ~25% Pass stage 3. - ~5% Pass stage 4 (reach the end).

SPE prediction: 0% Pass stage 1, 100% Take. The gap is dramatic.

Interpretation. Players cooperate for several rounds, generating surplus, until one defects. The modal outcome is not "Take immediately" but "Take around stage 2 or 3." Replicated across many experimental studies and subject populations.

Quantal response equilibrium explanation

A quantal response equilibrium (McKelvey–Palfrey, 1995) is a "logit"-smoothed Nash equilibrium where players don't best-respond deterministically but with probabilities proportional to $e^{\lambda \cdot \mathrm{EV}}$ for some rationality parameter λ. For small λ (bounded rationality), the equilibrium of a centipede game involves substantial early passing, matching experimental data.

This is a formal model of deviations from SPE. As $\lambda \to \infty$, QRE $\to$ SPE. For finite λ, QRE predicts the Pass-heavy early-stage behavior observed experimentally.

Poker Anchor

Concept mapping

Formal object	Poker instantiation
Centipede unraveling	The theoretical argument that in a pot with multiple streets of betting, one should fold earlier than feels right
SPE prediction vs empirical play	GTO "folds too much" for humans; humans "pass" (call down light) beyond what solver recommends
Irrationality epsilon	Uncertainty about opponent's type (fish, thinking player, GTO bot)
Pareto-dominated equilibrium	The SPE prediction gives lower EV than mutually passive play

Concrete situation

Multi-street pot with polarized opponent. You hold a mid-strength hand (e.g., top pair with weak kicker) and are facing bets on every street. The solver says to fold on the turn (because opponent's turn barrel is too polarized — mostly value, not enough bluffs). You call anyway, figuring you might catch a bluff on the river.

The centipede analogy. Each street is a stage. "Call" is roughly "Pass" (continue to next stage); "Fold" is roughly "Take" (end the game with your current EV). GTO says fold earlier than feels right, because the opponent's betting range gets more polarized on each street, and your bluff-catcher is "unraveled" from the river back.

Empirical bias. Most players hate folding, especially on turn and river, because the immediate impulse is "I already have money in the pot." The SPE prediction (fold the turn) is correct under equilibrium assumptions, but if the opponent is irrational in a specific way (over-bluffs), then calling is better. The Kreps–Milgrom–Roberts–Wilson resolution applies: if you believe there's a non-trivial probability the opponent is over-bluffing (many live recreational players do), then passing (calling) has positive EV despite the SPE prediction.

Trust in backward induction vs empirical adaptation. A strong GTO-trained player folds according to the equilibrium prediction even when it feels wrong; a weaker player "hero-calls" based on the suspicion that the opponent is irrational. Both can be correct — the question is whether the epsilon is actually there.

What this understanding buys you

The bluff-catcher dilemma is a centipede in miniature. GTO tells you to fold more than emotionally feels right. Empirically, most amateur opponents bluff less than GTO-optimal frequency, so the right answer is to fold *even more* than GTO says. But the opposite mistake — hero-calling against a polarized range — is also common. The centipede framing says: trust the backward induction, and fold when you should fold.

Reputation matters across sessions. In a single hand, you play the equilibrium of the hand. Across hands, your history affects opponent beliefs. Playing a non-SPE "always call down" strategy builds a reputation that opponents exploit by bluffing less and value-betting thinner — exactly the KMRW dynamic. Module 5.5 (chain store paradox) makes this precise.

Common knowledge failures are normal. You do not actually know whether your opponent knows you know they know about GTO. This is a chain of beliefs that, at some level, fails. The centipede game is the formal demonstration that when this chain is short, the predictions of full-rationality game theory diverge from reality.

Cross-Links

- **3.4 Backward induction** — the computation that produces the Centipede SPE.
- **3.5 Subgame perfection** — the refinement that uniquely pins down Take-immediately.
- **4.3, 4.4 Bayesian Nash, PBE** — how incomplete information rescues the empirical predictions.
- **5.5 Chain store paradox** — reputation in a similarly unraveling game.
- **5.6 Cooperation in repeated PD** — empirical cooperation vs SPE prediction.
- **10.4 Commitment** — Pareto improvements from commitment devices.
- **QRE, bounded rationality** — formal models of the gap.

Structural Compression

Invariant: In a finite-horizon perfect-information game with unilaterally dominant "defection" actions at each stage, backward induction unravels all cooperation, producing a unique SPE that is Pareto-dominated by almost every other outcome. Empirically, this prediction fails.

Mechanism: Backward recursion from the terminal stage through each decision node, with the common-knowledge-of-rationality assumption propagating the unraveling all the way to the first stage.

Cross-System Echo: The centipede game is the game-theoretic instance of a broader pattern: "logically forced defection" appears in the chain store paradox, finitely repeated prisoner's dilemma, public goods provision, and pollution games. In each case, the equilibrium unraveling argument gives a Pareto-dominated prediction, and experimental subjects routinely fail to unravel. The discrepancy is a pointer to the limits of common-knowledge reasoning.

Exercises

1. Verify the backward induction of a 4-stage centipede with payoffs $(1, 0.5), (0.5, 2), (4, 1), (2, 8)$, and terminal split $(16, 16)$.
2. Show that every stage of a generic centipede game has Take as the unique SPE action.
3. Compute the payoff gap (SPE vs "Pass all") for a centipede of length n with a doubling rule.
4. In the KMRW setup with ϵ irrationality probability, derive the first stage at which Taking becomes optimal.
5. Compute the quantal response equilibrium of a 4-stage centipede for $\lambda = 1$. Does it predict substantial early passing?
6. Give an example of a perfect-information game where SPE coincides with an intuitive cooperative outcome rather than defecting. What structural feature distinguishes it from the centipede?
7. **Argue, structurally,** why the Centipede game is a proof that backward induction requires common knowledge of rationality, not just individual rationality. Identify the exact point in the unraveling argument that fails if one player only knows they are rational but is not sure the opponent knows they know.