Foundations of Game Abstraction

Information & Action Abstraction in Extensive-Form Games

Intelligent Agents: Computational Game Solving

October 28, 2025

Today's Roadmap

Big picture: Real games are too large to solve exactly. We need *abstraction*.

Learning objectives:

1. Understand why abstraction is necessary (computational limits)
2. Define abstraction formally: information, action, state aggregation
3. Distinguish perfect-recall vs. imperfect-recall abstractions
4. See how CFR fits with abstraction (solve abstract game, lift strategy)
5. Understand approximation error and evaluation challenges

Running example: Texas Hold'em poker (we'll define it step-by-step)

Motivation: The Computational Wall

What we've done so far:

- Implemented CFR for Kuhn poker (12 information sets)
- Saw convergence to Nash equilibrium in minutes

What about real games?

Game	**Information Sets**	**CFR Feasible?**
Kuhn Poker	$\sim$12	✓
Leduc Poker	$\sim$10,000	✓
Limit Hold'em (Heads-Up)	$\sim 10^{14}$	$\times$
No-Limit Hold'em	$\sim 10^{161}$	$\times$

The problem: Even with sampling CFR, we can't store strategy/regret tables for 10^{14} infosets!

Texas Hold'em: Rules Overview (Part 1)

Setup:

- 2–10 players (we'll focus on 2-player *heads-up*)
- 52-card deck
- Each player antes (forced bet) to create a pot
- Each player starts with a **stack** (total chips available)

Goal: Win chips by having the best 5-card hand at showdown, or by making opponents fold

Hand rankings (best to worst):

- Royal Flush > Straight Flush > Four of a Kind > Full House > Flush > Straight > Three of a Kind > Two Pair > Pair > High Card

Texas Hold'em: Rules Overview (Part 2)

Game structure (4 betting rounds):

1. **Preflop:** Each player dealt 2 private cards (*hole cards*)
   - Betting round: fold / call / raise
2. **Flop:** 3 community cards dealt face-up
   - Players make best hand from 2 hole + 5 community cards
   - Betting round
3. **Turn:** 1 additional community card
   - Betting round
4. **River:** Final community card (5 community cards total)
   - Final betting round

Showdown: If ≥ 2 players remain, best hand wins the pot

Example Betting Terms

Common actions:

- **Fold:** Give up, lose any chips already in pot
- **Check:** Pass action (only if no bet to call)
- **Call:** Match current bet
- **Raise:** Increase bet (forces others to call higher amount or fold)

Example hand terms:

- **Flush draw:** 4 cards of same suit; need 1 more for flush
- **Backdoor flush:** 2 cards of same suit on flop; need 2 more (turn + river) for flush
- **Straight draw:** 4 cards in sequence; need 1 more for straight
- **Outs:** Cards that improve your hand (e.g., 9 outs for flush draw = 9 remaining cards of that suit)

Why Is Hold'em So Large?

Combinatorial explosion:

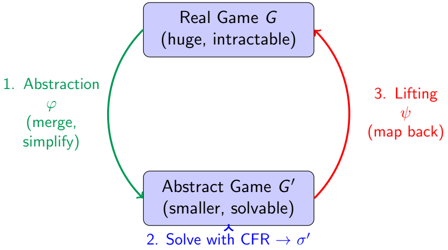
- **Preflop:** $\binom{52}{2} \times \binom{50}{2} \approx 1.3 \times 10^6$ private card deals
- **Flop:** $\binom{48}{3} \approx 17,000$ possible flops per preflop deal
- **Turn:** $\times 45$ possible turn cards
- **River:** $\times 44$ possible river cards
- **Total card combinations:** $\sim 2.4 \times 10^9$

Betting sequences:

- No-limit: any bet size from 1 chip to full stack
- Even discretized (e.g., 10 bet sizes per round), action tree is enormous
- Multiple betting rounds compound the branching

Result: $\sim 10^{161}$ game states in full No-Limit Hold'em (2-player)

Core principle: We can't solve the full game, so solve a smaller *abstract* game



z

Process:

1. **Abstraction:** Map $G \to G'$ (merge similar states, limit actions)
2. **Solve:** Run CFR on G' to get strategy σ'
3. **Lift:** Map σ' back to G to get playable strategy σ

Trade-off: Smaller G' is faster to solve, but σ may be weaker (approximation error)

Three Types of Abstraction

1. **Information Abstraction:**
   - Merge similar information sets
   - Example: Group all "King-high flush draw on turn" hands into one bucket
   - Reduces number of infosets player must track
2. **Action Abstraction:**
   - Restrict available actions
   - Example: Only allow bets of size {fold, call, $0.5 \times$pot, $1 \times$pot, $2 \times$pot}
   - Reduces branching factor at each decision node
3. **State Aggregation (MDP view):**
   - Feature-based grouping of states
   - Example: Use hand strength + pot odds as features
   - Common in RL-based approaches

Often combined: Real poker AIs use both information and action abstraction

Formal Definition: Abstract Game

Original game: $G = (\mathcal{H}, Z, \mathcal{I}, A, \rho, \sigma_c, u)$

- $\mathcal{H}$ = histories, Z = terminals
- $\mathcal{I}_i$ = information sets for player i
- $A(I)$ = actions at infoset I

Abstraction mapping: $\varphi : \mathcal{I}_i \to \mathcal{I}'_i$

- Maps each original infoset to an abstract infoset
- Many-to-one: $\varphi(I_1) = \varphi(I_2) = I'$ (merge I_1, I_2)

Abstract game: $G' = (\mathcal{H}', Z', \mathcal{I}', A', \rho', \sigma'_c, u')$

- $|\mathcal{I}'_i| < |\mathcal{I}_i|$ (fewer infosets)
- Histories in G' correspond to equivalence classes in G
- Payoffs u' aggregate over merged histories (weighted average or representative)

Example: Simple Information Abstraction

Toy poker game: 4 possible hands for each player: {J, Q, K, A}

Original infosets (player 1):

- I_J: holding Jack
- I_Q: holding Queen
- I_K: holding King
- I_A: holding Ace

Abstraction: Merge based on hand strength

- $\varphi(I_J) = \varphi(I_Q) = I'_{\text{weak}}$
- $\varphi(I_K) = \varphi(I_A) = I'_{\text{strong}}$

Result: 2 abstract infosets instead of 4
What we lose: Can't distinguish Queen from Jack (both play same strategy)

Action Abstraction: Example

Original game: No-limit Hold'em allows any bet size from 1 chip to full stack
Suppose stacks = 100 chips, pot = 10 chips

Full action space: $\{\text{fold, call, raise to } x\}$ where $x \in [11, 110]$
$\Rightarrow$ 102 possible actions!

Action abstraction: Discretize to fixed sizes

- Fold
- Call (match current bet)
- Raise to 0.5×pot (raise to 15)
- Raise to 1×pot (raise to 20)
- Raise to 2×pot (raise to 30)
- All-in (raise to 110)

Result: 6 actions instead of 102
Mapping back: If opponent bets 17 chips, map to nearest abstract action (1×pot = 20)

Perfect Recall vs. Imperfect Recall

Perfect recall: Player remembers all their past actions and observations

- Every path through the game tree to an infoset has the same sequence of player i's actions
- Kuhn poker, Texas Hold'em: perfect recall
- CFR guarantees apply with perfect recall

Imperfect recall abstraction: Merging infosets can violate perfect recall
Example: Merge infosets based only on current hand strength, ignoring betting history

- I_1: Player bet on flop, now on turn with K-high flush draw
- I_2: Player checked on flop, now on turn with K-high flush draw
- If we merge I_1 and I_2 (both "K-high flush draw"), player "forgets" what they did on flop

Consequence: Strategy space is restricted; may not contain Nash equilibrium!

Why Use Imperfect Recall?

Problem with perfect recall abstractions:

- To preserve perfect recall, must track full action history
- Exponential blowup in infosets with betting rounds
- Example: After 3 rounds with 5 actions each, $5^3 = 125$ histories to track

Imperfect recall advantage: Much smaller abstract game

- Can merge based on features only (hand strength, pot odds)
- Achieves dramatic compression (e.g., 1 million $\rightarrow$ 10,000 buckets)

The catch:

- CFR convergence guarantees don't directly apply (may not find Nash in abstract game)
- Strategy can be *non-monotonic*: more iterations $\not\Rightarrow$ better strategy
- Requires careful design and empirical validation

Practical stance: Most poker AIs use imperfect recall; test heavily to ensure quality

Solving the Abstract Game

Once we have G':

1. Run CFR (or sampling CFR) on G'
2. Obtain approximate Nash equilibrium strategy σ' for G'
3. Track regrets and average strategy at abstract infosets $I' \in \mathcal{I}'$

Key point: CFR operates on G' exactly as before

- Compute counterfactual values $v_i^{\sigma'}(I', a)$ in abstract game
- Update regrets: $r(I', a) = v_i(I', a) - v_i(I')$
- Regret matching: $\sigma'^{t+1}(I', a) \propto R^+(I', a)$

Convergence in G':

- If G' has perfect recall: CFR converges to Nash in G'
- If G' has imperfect recall: CFR may still converge, but no Nash guarantee

Strategy Lifting: From G' to G

Goal: Play in the real game G using strategy σ' learned in G'

Lifting mapping: $\psi : \mathcal{I}_i \to \mathcal{I}'_i$ (same as abstraction φ in most cases)

Process:

1. During play, observe real infoset $I \in \mathcal{I}_i$
2. Map to abstract infoset: $I' = \psi(I)$
3. Look up strategy: $\sigma'(I', \cdot)$ (distribution over abstract actions)
4. Map abstract action to real action (projection or nearest-neighbor)

Action mapping: If abstract action not legal in real game, choose closest legal action

- Example: Abstract says "bet $1\times$pot," but pot size changed $\to$ adjust proportionally
- Or: Abstract says "raise to 20," but min raise is 22 $\to$ raise to 22

Lifting: Example

Scenario: Playing No-Limit Hold'em with 5-action abstraction

Real infoset: Hold K♠ Q♠, board is A♠ 7♠ 2♣ (flop), pot = 40, opponent bet 30

Step 1: Map to abstract infoset

- Compute hand features: flush draw (9 outs), overcards (6 outs), total 15 outs
- Map to abstract bucket: "strong draw, facing bet"
- I' = abstract infoset for "strong draw" category

Step 2: Look up abstract strategy

- $\sigma'(I', \text{fold}) = 0.1$
- $\sigma'(I', \text{call}) = 0.5$
- $\sigma'(I', \text{raise } 1\times\text{pot}) = 0.4$

Step 3: Sample action and map to real game

- Sample: choose "raise 1×pot" with 40% probability
- Real action: raise to $40 + 30 = 70$ chips

Sources of Approximation Error

Where does quality degrade?

1. **Abstraction design:**
   - Merging dissimilar infosets loses information
   - Example: Grouping "flush draw + pair" with "flush draw only"
2. **Imperfect recall:**
   - Abstract game may not have Nash equilibrium
   - CFR may cycle or converge to suboptimal strategy
3. **Lifting mismatch:**
   - Real action space differs from abstract
   - Opponent plays differently than assumed in G'
4. **Sampling variance:**
   - Using sampling CFR in G' introduces noise
   - More pronounced with aggressive abstraction

Approximation Error: Intuition

Informal bound: If abstraction preserves "similar" infosets, error is small

Key factors:

- **Infoset similarity:** How different are merged infosets?
  - Measure: distance in counterfactual value or expected payoff
  - Lipschitz continuity: if infosets are close in features, values are close
- **Action overlap:** Do abstract actions cover real action space well?
- **Opponent model:** Does opponent's strategy match what we assumed in G'?

Formal bounds exist (Waugh et al., Ganzfried & Sandholm), but:

- Require strong assumptions (perfect recall, Lipschitz payoffs)
- Constants are loose; not useful for practical prediction
- **Empirical evaluation is the gold standard**

Evaluating Abstraction Quality

How do we measure if σ (lifted from G') is good?

Method 1: Exploitability

- Compute best response to σ in G (or a finer abstract game)
- Exploitability = value of BR vs. σ minus game value
- **Pro:** Objective, doesn't require opponent
- **Con:** Computing BR in full G may be intractable (use proxy game)

Method 2: Head-to-head play

- Play σ vs. baseline strategies (e.g., Nash from finer abstraction)
- Measure: win rate (chips won per hand) over many games
- **Pro:** Reflects real performance
- **Con:** High variance; needs many samples

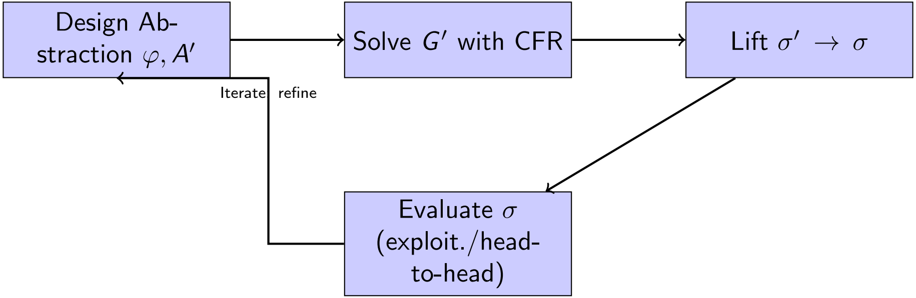
Method 3: Cross-abstraction robustness

- Train with abstraction A, test against strategies trained with abstraction B
- Check sensitivity to abstraction choices

Common Pitfalls in Abstraction

1. **Over-aggregation:**
   - Merging too many infosets $\rightarrow$ loses critical distinctions
   - Example: Treating all "pair" hands the same (pocket aces vs. pocket twos)
2. **Action granularity mismatch:**
   - Too few bet sizes $\rightarrow$ can't represent nuanced strategies
   - Too many $\rightarrow$ abstract game still too large
3. **Ignoring dynamic features:**
   - Static hand strength may not capture draw potential
   - Example: Flush draw on turn (2 cards to come) vs. river (no cards left)
4. **Imperfect recall non-monotonicity:**
   - More CFR iterations may hurt performance due to cycling
   - Need to monitor exploitability and checkpoint best strategies
5. **Opponent adaptation:**
   - If opponent learns your abstraction, they can exploit it
   - Need robustness or real-time re-solving (next lectures)

Abstraction in Context: Where CFR Fits



Iterative process:

1. Design initial abstraction (information + action)
2. Solve abstract game with CFR/MCCFR
3. Lift and evaluate
4. If quality insufficient, refine abstraction (add buckets, actions) and repeat

Next lectures:

- Information abstraction in detail (features, buckets, poker-specific). Action abstraction, real-time re-solving. Evaluation methods, case studies (Libratus, Pluribus)

Key Takeaways

Why abstraction?

- Real games (Hold'em) have 10^{14}–10^{161} states $\rightarrow$ can't solve exactly
- Abstraction reduces game size to enable CFR

Two main types:

- **Information abstraction:** Merge similar infosets (bucketization)
- **Action abstraction:** Discretize action space (bet sizes)

Perfect vs. imperfect recall:

- Perfect recall: CFR guarantees apply, but larger abstract game
- Imperfect recall: Much smaller, but no Nash guarantee (used in practice)

Pipeline: Design φ $\rightarrow$ Solve G' with CFR $\rightarrow$ Lift σ' to G $\rightarrow$ Evaluate $\rightarrow$ Iterate

Coming next: Deep dive into information abstraction for poker (features, bucketing algorithms)

Preview: Information Abstraction Deep Dive

Next topics:

- **Hand strength features:**
  - Effective Hand Strength (EHS)
  - Expected Hand Strength (E[HS])
  - Hand potential (positive potential, negative potential)
- **Bucketing algorithms:**
  - k-means clustering
  - Earth Mover's Distance (EMD) for hand distributions
  - Hierarchical bucketing (coarse preflop, fine river)
- **Potential-aware abstraction:**
  - Incorporate future card distributions
  - Public belief states (PBS)
- **Worked example:** Leduc poker abstraction with CFR convergence comparison

References and Further Reading

Key papers:

- **Waugh et al. (2009):** "A Practical Use of Imperfect Recall" (AAMAS)
  - Imperfect recall abstractions, convergence issues
- **Johanson et al. (2013):** "Finding Optimal Abstract Strategies in Extensive-Form Games" (AAAI)
  - Comprehensive framework for abstraction and evaluation
- **Ganzfried & Sandholm (2014):** "Potential-Aware Imperfect-Recall Abstraction" (AAAI)
  - Incorporating future card distributions
- **Moravčík et al. (2017):** "DeepStack: Expert-level AI in Heads-Up No-Limit Poker" (Science)
  - Real-time re-solving, continual abstraction refinement