University of Alberta

Regret Minimization in Games and the Development of Champion Multiplayer Computer Poker-Playing Agents

by

Richard Gibson

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

Spring 2014
Edmonton, Alberta

Abstract

Recently, poker has emerged as a popular domain for investigating decision problems under conditions of uncertainty. Unlike traditional games such as checkers and chess, poker exhibits imperfect information, varying utilities, and stochastic events. Because of these complications, decisions at the poker table are more analogous to the decisions faced by humans in everyday life.

In this dissertation, we investigate regret minimization in extensive-form games and apply our work in developing champion computer poker agents. Counterfactual Regret Minimization (CFR) is the current state-of-the-art approach to computing capable strategy profiles for large extensive-form games. Our primary focus is to advance our understanding and application of CFR in domains with more than two players. We present four major contributions. First, we provide the first set of theoretical guarantees for CFR when applied to games that are not two-player zero-sum. We prove that in such domains, CFR eliminates strictly dominated plays. In addition, we provide a modification of CFR that is both more efficient and can lead to stronger strategies than were previously possible. Second, we provide new regret bounds for CFR, present three new CFR sampling variants, and demonstrate their efficiency in several different domains. Third, we prove the first set of sufficient conditions that guarantee CFR will minimize regret in games with imperfect recall. Fourth, we generalize three previous game tree decomposition methods, present a new decomposition method, and demonstrate their improvement empirically over standard techniques. Finally, we apply the work in this thesis to construct three-player Texas hold'em agents and enter them into the Annual Computer Poker Competition. Our agents won six out of the seven three-player events that we entered from the 2010, 2011, 2012, and 2013 computer poker competitions.

Acknowledgements

Firstly, I would like to thank my supervisor, **Duane Szafron**, for mentoring me throughout my entire Ph.D. studies at the University of Alberta. Duane guided me into poker research and gave me the freedom to explore the problems in the areas that I found most interesting. Duane's career advice has also helped me as I prepare for life after graduate school.

Secondly, much of my work would not have been possible without the support of the **Computer Poker Research Group (CPRG)**. I would like to thank all of its members during my time in the group: **Michael Johanson**, **Nolan Bard**, **Neil Burch**, **Johnny Hawkin**, **Josh Davidson**, **Parisa Mazrooi**, **Kevin Waugh**, **Nick Abou Risk**, **Dave Schnizlein**, **Trevor Davis**, **Dustin Morrill**, **Chris Archibald**, **Michael Bowling**, and **Rob Holte**. In particular, my programming skills have definitely sharpened thanks to code sharing with Neil Burch and Michael Johanson. By building off of their work, I was able to produce very efficient implementations with moderate ease for the experiments performed throughout this dissertation. Furthermore, I would like to give a special thanks to **Marc Lanctot**. Marc's similar interests and complementary expertise in games gelled well with my own skill set and led us to coauthor several papers in top international artificial intelligence conferences.

Thirdly, thank-you to **NSERC**, **Alberta Innovates – Technology Futures**, and the **University of Alberta** for awarding me with very generous scholarships throughout my Ph.D. studies. These scholarships have allowed me to focus primarily on my research without the distractions of worrying about my financial situation. In addition, thank-you to **WestGrid** and **Compute Canada** for granting myself and the rest of the CPRG access to the clusters of supercomputers that they maintain. Nearly all of the experiments performed in this dissertation used these computational resources and would not have been possible without them.

Fourthly, I would like to thank my wife, **Melissa Gibson**, for moving to Edmonton for me, supporting me, and keeping me sane while I slowly amassed the contents of this dissertation. Finally, thank-you to all of my other friends and family that I have not already mentioned for continuing to make my life outside of academics very enjoyable.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Autonomous computer agents capable of meaningful interactions with the world is the holy grail of artificial intelligence research. Every day, humans must make decisions based on an incomplete view of the world, experienced only through limited observations. In addition, the consequences of these decisions often cannot be fully known at the time of the decision-making. These decisions can be short term, such as quick decisions made while driving, or long term, such as whether or not to move to a new city. To further complicate matters, the world is full of millions of people and the actions of others often affect our own actions. For instance, if we see a lot of vehicles on the highway nearby, we may believe that it will be faster to take a detour to work rather than potentially being slowed by the traffic. We would expect any fully autonomous robot to be capable of making such decisions, despite lacking all pertinent information (like a full traffic report). Unfortunately, current science and technology is far from achieving such complete autonomous behaviour in artificially intelligent agents.

For at least the past few decades, games have been an exceptional platform for artificial intelligence research. Many successful computer programs have been developed that play at expert levels in games such as Othello [11], checkers [63], chess [12], and Go [22]. These examples are all two-player, zero-sum, deterministic games where both players share perfect information about the game state. Such games can be handled reasonably well by classic artificial intelligence techniques, such as heuristic search. In many real-life problems, however, multiple players exist, non-determinism (chance) abounds, and we often have imperfect information about the state of the game.

Recently, poker has become a popular domain for artificial intelligence research [3, 5, 6, 8, 31, 40]. Poker involves multiple players, random card dealings, and some information hidden from the players (the other players' hole cards). Texas hold'em is a tremendously popular version of poker played around the world, and many professional players have earned millions of dollars playing the game. The game is considered to be very strategic, making it an excellent showcase for artificial intelligence research.

The Computer Poker Research Group (CPRG) at the University of Alberta uses poker as a testbed for conducting artificial intelligence research in games. Since 2006, the CPRG has competed in the Annual Computer Poker Competition (ACPC) [4] that evaluates computer poker agents developed by teams of academics and hobbyists to help drive research in this area. In addition, the CPRG competed against teams of human professionals in two Man vs. Machine Poker Competitions [1, 2], where the CPRG edged the humans at heads-up (two-player) limit Texas hold'em in 2008. Since joining the research group in 2009, my role in the CPRG has been to focus on research in games with three or more agents and to evaluate the research results in poker. My activities have included managing the team's three-player entries of the 2010, 2011, 2012, and 2013 ACPC and building on the group's previous work in ring (more than two players) poker.

1.1 Previous State of the Art in Ring Poker

Three-player limit Texas hold'em events were first included in the ACPC in 2009. For these events, Abou Risk [62] of the CPRG built agents using the Counterfactual Regret Minimization (CFR) algorithm [75]. CFR is an iterative algorithm described later in Section 2.2.2 and is the current state-of-the-art approach to computing approximate Nash equilibria in large two-player, zero-sum games. To our knowledge, Abou Risk was the first to apply CFR to a game with more than two players and his agents won the three-player events by a significant margin.

Despite winning the ACPC events, there are four major issues with regards to the CPRG's 2009 ring poker agents:

1. There is no theory to explain why CFR is successful in games with more than two players. While CFR is guaranteed to converge to an equilibrium strategy profile in two-player zero-sum games, the algorithm does not, in general, produce equilibrium profiles for even small three-player games [3, Table 2]. In games with more than two players, a Nash equilibrium profile may not be our best choice of strategy anyways. A Nash equilibrium only guarantees that no player can benefit from unilaterally deviating from the profile; if more than two players deviate, all guarantees are lost. Even worse, if each player is playing a strategy from a different Nash equilibrium profile, the combination of the players' strategies might not comprise an equilibrium profile. In games with more than two players, it is not clear what properties a strategy should have to be considered *optimal* or whether such a notion even exists.

2. Chance Sampling is very slow in three-player limit Texas hold'em. *Chance Sampling* [75] is a CFR variant that uses Monte Carlo sampling to reduce the algorithm's per iteration time cost. This is achieved by considering just a single outcome for the card deals on each iteration and only traversing the portion of the game tree associated with the given card outcome. For two-player limit poker and other games with many chance branches, Chance Sampling can significantly reduce computation time to reach a given solution quality. Abou Risk used Chance Sampling to generate the CPRG's 2009 ACPC entries for the three-player events. However, in three-player Texas hold'em and other games with many player decision states, Chance Sampling can be quite slow as we demonstrate later in Section 5.6.

3. CFR's original analysis only applies to games with perfect recall. CFR uses space linear in the number of information sets in the game. However, even heads-up limit Texas hold'em, the smallest variant of Texas hold'em, is too large to feasibly solve. To address this limitation, an abstract version of the game is created that groups similar states together into single abstract states. If players never forget information that was revealed to them in the abstract game, nor the order in which the information was revealed, the abstract game exhibits *perfect recall*; otherwise, the abstract game exhibits *imperfect recall*. We can then apply CFR to the abstract game and use the resulting abstract strategy to play the real game. In practice, agents built with imperfect recall abstractions often perform better than agents employing perfect recall abstractions of the same size [45, 73]. However, regret minimization is only guaranteed by CFR in games with perfect recall [75]. In games with imperfect recall, it is unclear what theoretical guarantees are provided by CFR. This is true for all types of imperfect recall games, even those that are two-player zero-sum.

4. Given a finite amount of computing resources, ring poker strategies typically suffer from more card ambiguity than two-player strategies. As mentioned previously, limit Texas hold'em is too large to feasibly solve with CFR or other strategy computation techniques, even in the two-player case. Extending to more players further complicates the problem because the number of game states increases. As a result, under a fixed memory limitation, three-or-more-player poker strategies must employ much coarser abstractions compared to two-player strategies, leaving many different states indistinguishable. In an attempt to alleviate this problem, Abou Risk and Szafron [3] created heads-up experts by applying CFR to a selection of two-player subtrees using an alternative abstraction. Their results from combining these experts with a *base strategy*, however, were mixed.

1.2 Main Contributions

This dissertation presents my research results that address each of the four major impediments enumerated in the previous section. My work here advances the state of the art in developing automated agents for imperfect information domains containing two or more adversaries and particularly targets the development of three-player Texas hold'em poker agents. My contributions are the following:

1. Proof that CFR avoids strict domination in games that are not two-player zero-sum. *Dominated actions* are formally defined in extensive-form games and it is shown that CFR avoids iteratively strictly dominated actions and strategies. These are the first theoretical results to suggest that CFR may lead to good performance in games with more than two players. In addition, for two-player non-zero-sum games, worst case performance is bounded and regret minimization is shown to yield strategies very close to equilibrium in practice. These theoretical advancements lead us to a new variant of CFR for games with more than two players that is more efficient and may be used to generate stronger strategies than previously possible.

2. A general analysis, both theoretical and empirical, of CFR and its sampling variants. Monte Carlo CFR (MCCFR) [54], a family of algorithms to which Chance Sampling belongs, is generalized and it is shown that any bounded, unbiased estimator of the sought values can be used to probabilistically minimize regret. In addition, the variance of the estimator is shown to bound the convergence rate of an algorithm that calculates regret directly from the estimator. Furthermore, three new sampling variants for CFR are presented, two of which are now the best known algorithms for abbreviated no-limit games and for three-player limit Texas hold'em respectively.

3. The first set of regret bounds for CFR when applied to a general class of imperfect recall games. *Well-formed games* are defined as a special class of games containing all perfect recall games, but also many games with imperfect recall. CFR is shown to minimize regret in well-formed games. In addition, *skew well-formed games*, a generalization of well-formed games, are introduced where an additional regret bound for these games is also derived.

4. A general framework for strategy stitching in large games. We discuss *strategy stitching*, a family of techniques for combining a base strategy in a coarse abstraction of the full game tree to expert strategies in fine abstractions of smaller subtrees. Two techniques are analyzed. Firstly, *static experts*, an approach that generalizes some previous strategy stitching efforts, are defined and used to create entries to the 2010 and 2011 ACPC three-player events. Secondly, *dynamic experts* are proposed that combine multiple abstractions simultaneously within one abstract game. Dynamic experts were used to win the three-player competitions of the 2012 and 2013 ACPC.

Portions of these research contributions have appeared in the following refereed papers:

- ”Regret Minimization in Multiplayer Extensive Games” (extended abstract) appearing at IJCAI 2011 [26],
- ”On Strategy Stitching in Large Extensive Form Multiplayer Games” appearing at NIPS 2011 [27],
- ”Generalized Sampling and Variance in Counterfactual Regret Minimization” appearing at AAAI 2012 [25],
- ”No-Regret Learning in Extensive-Form Games with Imperfect Recall” appearing at ICML 2012 [53], and
- ”Efficient Monte Carlo Counterfactual Regret Minimization in Games with Many Player Actions” appearing at NIPS 2012 [24].

The rest of this dissertation is organized as follows. First, fundamental background material in game theory, including normal and extensive-form games, CFR, and abstraction, is presented in Chapter 2. Next, Chapter 3 describes several poker games, including Kuhn Poker, Leduc hold'em, and Texas hold'em, and touches on related work in poker. Chapters 4 through 7 present the main research contributions in domination, sampling, imperfect recall, and strategy stitching respectively. This is followed by Chapter 8 that presents the 2010, 2011, 2012, and 2013 ACPC three-player agents that won all but one of the seven events competed in from these four years. Finally, Chapter

9 concludes the dissertation and discusses possible future directions and extensions of this research. An extended example of CFR is provided in Appendix A, while full proofs for Theorems presented in Chapters 4, 5, and 6 appear in Appendices B, C, and D respectively.

Chapter 2

Background

In this chapter, we formally define normal-form and extensive-form games and describe the difference between perfect recall and imperfect recall extensive-form games. We also define Nash equilibrium and dominated strategies, two main solution concepts of particular interest to us. This is followed by a thorough explanation of Counterfactual Regret Minimization (CFR) and Monte Carlo CFR (MCCFR). We then discuss other solution concepts and techniques that have appeared in the literature before ending this chapter with a formal introduction to abstraction in games.

2.1 Normal-Form and Extensive-Form Games

Normal-form games are a common and general framework useful for modelling problems involving single, simultaneous decisions made by multiple agents. Two-player normal-form games are often represented by a matrix with rows denoting the row player's actions, columns denoting the column player's actions, and entries denoting payoffs resulting from the row player's and column player's actions respectively. More formally, a normal-form game is defined as follows:

Definition 2.1. *A finite* ***normal-form game*** *G is a tuple $\langle N, A, u\rangle$ containing the following components:*

- *A finite set $N = \{1, 2, ..., n\}$ of* ***players****.*
- *A set of* ***action profiles*** *$A = A_1 \times A_2 \times \cdots \times A_n$, with A_i being a finite set of* ***actions*** *available to player i.*
- *For each $i \in N$, a* ***utility function*** *$u_i : A \rightarrow \mathbb{R}$ that denotes the payoff for player i under each possible action profile.*

If $n = 2$ and $u_1 = -u_2$, the game is **zero-sum**. Otherwise, we say the game is **non-zero-sum**. We emphasize here that zero-sum only refers to two-player games. For example, a three-player game with $u_1 + u_2 + u_3 = 0$ is considered a non-zero-sum game. Note that two-player constant-sum

$$\begin{array}{c c}
 & \begin{array}{ccc} a & b & c \end{array} \\
\begin{array}{c} A \\ B \\ C \end{array} & \begin{pmatrix} 3 & -1 & -3 \\ 1 & 0 & 1 \\ -3 & -1 & 3 \end{pmatrix}
\end{array}$$

Figure 2.1: A zero-sum normal-form game. Entries represent utilities for the row player; utilities for the column player are the negation.

games where $u_1 + u_2 = C$ for some constant C can easily be translated to a zero-sum game without changing the strategic properties by simply subtracting C from one player's utility.

Figure 2.1 presents a zero-sum normal-form game in matrix form. In this example, both players in $N = \{1, 2\}$ have three actions, $A_1 = \{A, B, C\}$ and $A_2 = \{a, b, c\}$. The utilities for the players range between -3 and $+3$ depending on the actions selected.

A player's strategy defines how likely the player is to play each of the available actions. More precisely, a **mixed strategy** σ_i for player i is a probability distribution over A_i, where $\sigma_i(a)$ is the probability that action a is taken under σ_i. For example,

$$\sigma_1(a) = \begin{cases} 3/4 & \text{if } a = A \\ 1/4 & \text{if } a = B \\ 0 & \text{if } a = C \end{cases}$$

is a mixed strategy for player 1 (row player) in the game shown in Figure 2.1. The set of all such strategies for player i is denoted Σ_i. A **pure strategy** for player i, $s_i \in \mathcal{S}_i$, assigns a probability of 1 to a single action. Define the **support** of σ_i, supp(σ_i), to be the set of actions assigned positive probability by σ_i. For instance, the support of the example strategy above is supp$(\sigma_1) = \{A, B\}$, the possible actions that player 1 might take when following σ_1. A **strategy profile** $\sigma \in \Sigma$ is a collection of strategies $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n)$, one for each player. We let σ_{-i} refer to the strategies in σ excluding σ_i, and $u_i(\sigma)$ to be the expected utility for player i when players play according to σ. If we define $\sigma_2 = \{(a, 1), (b, 0), (c, 0)\}$, then the expected utility for player 1 under σ in the game shown in Figure 2.1 is

$$u_1(\sigma) = 3 \cdot 3/4 + 1 \cdot 1/4 + (-3) \cdot 0 = 5/2.$$

When decisions are sequential rather than simultaneous, or when the game involves imperfect information or stochastic events, extensive-form games are generally more applicable than normal-form games. An extensive-form game is a rooted directed tree with nodes representing decision states (possibly belonging to *chance*), edges representing actions, and terminal nodes holding end-game utility values for the players. For each player, the decision states are partitioned into *information sets* such that game states within an information set are indistinguishable to the player. Non-singleton information sets arise due to hidden information that is only available to a subset of the players, such as private cards in poker. We now provide a formal definition of extensive-form games:

Definition 2.2 (**Osborne and Rubenstein [59, p. 200]).** *A finite* ***extensive-form game*** Γ *with imperfect information is a tuple* $\langle N, H, P, \sigma_c, u, \mathcal{I} \rangle$ *containing the following components:*

- *A finite set* $N = \{1, 2, ..., n\}$ *of* ***players****.*
- *A finite set* H *of sequences, the possible* ***histories*** *of* ***actions****, such that the empty sequence is in* H *and every prefix* h *of a sequence* $h' \in H$*, denoted* $h \sqsubseteq h'$*, is also in* H*. For* $h \in H$*, we denote* $A(h) = \{a \mid ha \in H\}$ *to be the set of actions available at* h*. In addition,* $Z \subseteq H$ *denotes the set of* ***terminal histories*** *that are not a prefix of any other sequence.*
- *A* ***player function*** P *that assigns to each nonterminal history* $h \in H \backslash Z$ *a member* $P(h) \in N \cup \{c\}$*.* $P(h)$ *is the player who acts after the history* h*. If* $P(h) = c$*, then* ***chance*** *generates the action taken after history* h*. We denote* $H_i = \{h \in H \mid P(h) = i\}$ *to be the set of histories belonging to player* i*.*
- *A function* σ_c *that associates with every history in* H_c *a probability measure* $\sigma_c(h, \cdot)$ *on* $A(h)$*, where each such probability measure is independent of every other such measure. For* $h \in H_c$ *and* $a \in A(h)$*,* $\sigma_c(h, a)$ *is the probability that action* a *occurs after history* h*.*
- *For each player* $i \in N$*, a* ***utility function*** $u_i : Z \to \mathbb{R}$ *that denotes the payoff for player* i *at each possible terminal history. We denote* $\Delta_i = \max_{z,z' \in Z} u_i(z) - u_i(z')$ *to be the range of utilities for player* i*.*
- *For each player* $i \in N$*, an* ***information partition*** $\mathcal{I}_i$ *of* H_i *with the property that* $A(h) = A(h')$ *whenever* h *and* h' *are in the same member of the partition. For any* ***information set*** $I \in \mathcal{I}_i$*, we denote* $A(I)$ *to be the set of actions* $A(h)$ *and denote* $P(I)$ *to be the player* $P(h)$ *for any* $h \in I$*. In addition, for* $h \in H_i$*, let* $I(h)$ *denote the information set containing* h*. Finally, let* $|A(\mathcal{I}_i)| = \max_{I \in \mathcal{I}_i} |A(I)|$ *denote the maximum number of actions available to player* i *at any information set.*

Analogous to normal-form games, an extensive-form game is **zero-sum** if $n = 2$ and $u_1 = -u_2$, and is **non-zero-sum** otherwise.

Figure 2.2 shows an example of an extensive-form game tree with one chance action followed by one decision for each of two players, $N = \{1, 2\}$. The game has 22 histories with terminal histories $Z = \{all, alr, arl, arr, bll, blr, brl, brr, dll, dlr, drl, drr\}$ and non-terminal histories $H \backslash Z = \{\emptyset, a, b, d, al, ar, bl, br, dl, dr\}$. The empty sequence, $\emptyset$, represents the initial game state (the root of the tree) where it is chance's action. Since player 1 cannot distinguish between whether chance generated b or d, player 1 has two information sets, namely $\mathcal{I}_1 = \{\{a\}, \{b, d\}\}$. Player 2, on the other hand, has perfect information regarding the current game state and thus has six information sets in $\mathcal{I}_2 = \{\{al\}, \{ar\}, \{bl\}, \{br\}, \{dl\}, \{dr\}\}$.

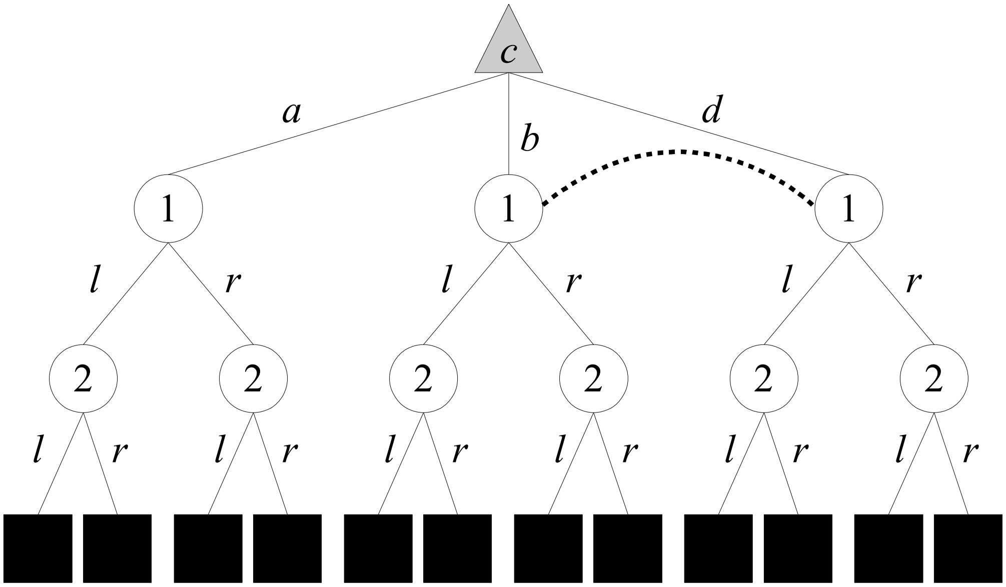


Figure 2.2: A two-player extensive-form game, where game states connected by a bold dashed curve are in the same information set. In this game, player 1 cannot distinguish between whether chance generated b or d. Utilities at terminal nodes are not shown.

While extensive-form games are primarily used to model stochastic games with imperfect information, they are versatile enough to also represent deterministic games and games where no information is hidden from the players. For example, checkers can be modelled in extensive form where chance does not occur (*i.e.* $P(h) \neq c$ for all $h \in H$) and every information set for each player contains a single history. In addition, backgammon can also be represented as an extensive-form game where chance's actions represent the possible dice rolls. Like in checkers, the information sets in backgammon are also singleton sets since both players know the exact state of the game at all times. However, in most poker games, a player cannot see the cards held by the opponent(s). An information set represents this imperfect information by containing the histories for every possible hand the opponent(s) might be holding, given the cards our player can see. In other words, $\mathcal{I}_i$ partitions the histories according to the cards seen by player i and the public actions taken by the players, but not according to the private cards held by players other than i.

A **behavioral strategy** σ_i for player i is a function that maps each information set $I \in \mathcal{I}_i$ to a probability distribution over $A(I)$. This defines how likely player i is to take each of the available actions at each information set belonging to player i. The set of all possible behavioral strategies for player i is again denoted by Σ_i and a **strategy profile** $\sigma \in \Sigma$ is a collection of strategies $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n)$. We overload the notation here with that of mixed strategies in normal-form games as we will only consider strategies in normal-form games to be mixed and strategies in extensive-form games to be behavioral.

Let $\pi^\sigma(h)$ denote the probability of history h occurring if all players play according to $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n)$. With this notation, it follows that the expected utility for player i when all players play according to σ is $u_i(\sigma) = \sum_{z \in Z} \pi^\sigma(z) u_i(z)$. We can decompose $\pi^\sigma(h) = \prod_{i \in N \cup \{c\}} \pi_i^\sigma(h)$ into each player's and chance's contribution to this probability. Hence, $\pi_i^\sigma(h) = \prod_{\substack{h'a \sqsubseteq h \\ P(h')=i}} \sigma_i(h', a)$ is the probability that player i **plays to reach h** under σ_i. Let $\pi_{-i}^\sigma(h) = \prod_{j \in (N-i) \cup \{c\}} \pi_j^\sigma(h)$ be the product of all players' contributions (including chance) except that of player i. In addition, let $\pi^\sigma(h, h')$ be the probability of history h' occurring after h, given that h has occurred. Let $\pi_i^\sigma(h, h')$ and $\pi_{-i}^\sigma(h, h')$ be defined similarly.

Consider again the example game tree in Figure 2.2. A possible strategy for player 1 is

$$\sigma_1(I) := \begin{cases} \{(l, 1), (r, 0)\} & \text{if } I = \{a\} \\ \{(l, 1/2), (r, 1/2)\} & \text{if } I = \{b, d\}, \end{cases}$$

which always takes action l after chance generates a, but is equally likely to take l or r if chance generates b or d. Since the histories b and d fall into the same information set for player 1, any strategy in Σ_1 must act under a single set of action probabilities for both of the game states.

Furthermore, assume now that both chance and player 2 take actions uniformly at random. Thus, $\sigma_c(\emptyset, \hat{a}) = 1/3$ for all $\hat{a} \in \{a, b, d\}$ and $\sigma_2(I, \hat{a}) = 1/2$ for all $I \in \mathcal{I}_2$, $\hat{a} \in A(I)$. We can then determine the probability of reaching any history h by simply decomposing the probabilities according to each player's contribution, $\pi^\sigma(h) = \pi_c^\sigma(h) \cdot \pi_1^\sigma(h) \cdot \pi_2^\sigma(h)$. For example,

$$\pi^\sigma(alr) = 1/3 \cdot 1 \cdot 1/2 = 1/6,$$

$$\pi^\sigma(brl) = 1/3 \cdot 1/2 \cdot 1/2 = 1/12,$$

and

$$\pi^\sigma(dl) = 1/3 \cdot 1/2 \cdot 1 = 1/6.$$

A strategy s_i is **pure** if a single action is assigned probability 1 at every information set; for each $I \in \mathcal{I}_i$, let $s_i(I)$ be this action. We denote by $\mathcal{S}_i$ the (finite) set of all pure strategies for player i. For a behavioral strategy σ_i, define the **support** of σ_i to be $\text{supp}(\sigma_i) = \{s_i \in \mathcal{S}_i \mid \sigma_i(I, s_i(I)) > 0 \text{ for all } I \in \mathcal{I}_i\}$, the set of pure strategies that σ_i follows with positive probability at each information set.

A **best response** to an opponent profile σ_{-i} is a player i strategy $\sigma_i^* \in \arg\max_{\sigma_i} u_i(\sigma_i, \sigma_{-i})$ that maximizes player i's expected utility against σ_{-i}. The **best response value** for player i is the value of that strategy, $br_i(\sigma_{-i}) = u_i(\sigma_i^*, \sigma_{-i})$. The **exploitability** of a strategy profile σ, $e(\sigma) = -\sum_{i \in N} \min_{\sigma'_{-i}} u_i(\sigma_i, \sigma'_{-i})/n$, measures how much σ loses on average to a set of worst-case opponents when players rotate positions. In zero-sum games, exploitability is the average amount lost to a best response, $e(\sigma) = (br_1(\sigma_2) + br_2(\sigma_1))/2$. In non-zero-sum games, however, exploitability becomes less meaningful. For these games, the worst-case scenario for player i would be for all other players to minimize player i's expected utility while disregarding their own individual utilities. Because this is unrealistic behavior, we will rarely consider exploitability in non-zero-sum games.

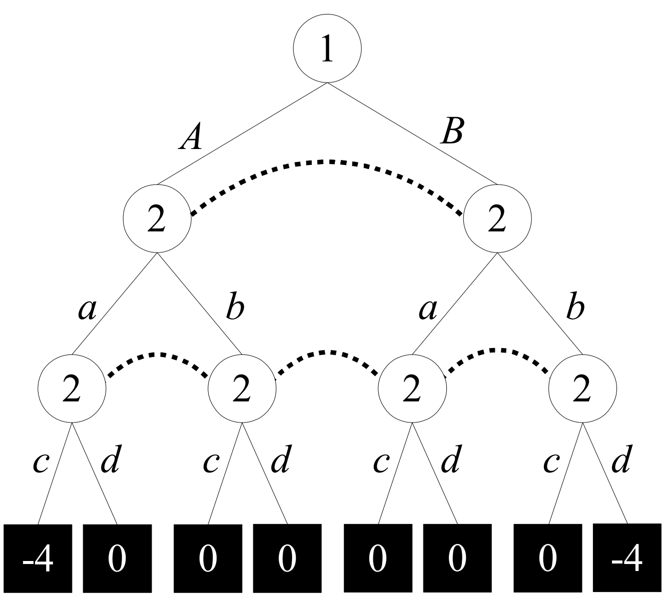
2.1.1 Perfect Vs. Imperfect Recall

In an extensive-form game with perfect recall, players never forget any information that was revealed to them, nor the order in which the information was revealed. To formally define perfect recall, we need some additional notation. First, for a history $h \in H$, let $X_i(h)$ be the sequence of information set, action pairs $(I_1, a_1), (I_2, a_2), ..., (I_\ell, a_\ell)$ such that player i visited each $I_1, I_2, ..., I_\ell$ in turn and took actions $a_1, a_2, ..., a_\ell$ in turn before reaching h. More formally, define $X_i(h)$ to be the sequence of information set, action pairs such that $(I, a) \in X_i(h)$ if $I \in \mathcal{I}_i$ and there exists $h' \sqsubseteq h$ such that $h' \in I$ and $h'a \sqsubseteq h$. The order of the pairs in $X_i(h)$ is the order in which they occur in h. Define $X(h)$ to be the sequence of information set, action pairs belonging to all players in the order in which they occur in h, and $X_{-i}(h)$ similarly, by removing player i's information set, action pairs from $X(h)$. Also, define $X(h, h')$ to be the sequence of information set, action pairs belonging to all players that start at h and end at h' when $h \sqsubseteq h'$; if $h \not\sqsubseteq h'$, $X(h, h')$ is defined to be the empty sequence. We will make use of $X(h, h')$ in Chapter 6. Let $X_i(h, h')$ and $X_{-i}(h, h')$ be similarly defined.

Definition 2.3. *An extensive-form game has* ***perfect recall*** *if for every player $i \in N$, for every information set $I \in \mathcal{I}_i$, and for any $h, h' \in I$, $X_i(h) = X_i(h')$. Otherwise, the game has* ***imperfect recall****.*

Intuitively, with perfect recall every player has an infallible memory and cannot "forget" anything during a play of the game that they once knew. Hence, what a player knows at I is a composition of what the player has discovered in the past up to this point and the precise order in which information was discovered. For games with perfect recall, we denote $X_i(I) = X_i(h)$ for any $h \in I$.

Perfect recall presents a nice connection between normal-form and extensive-form games. In particular, any extensive-form game Γ, with perfect recall or not, can be represented in normal form G by setting the action set in G for player i to be all pure strategies in Γ and for all $s \in \mathcal{S}$, assigning utility $u_i(s) = \sum_{z \in Z} \pi^s(z) u_i(z)$. Kuhn [51] proved that if Γ has perfect recall, then any mixed strategy in G has a utility-equivalent behavioral strategy in Γ; that is, for every mixed strategy σ_i^G, there exists an equivalent behavioral strategy σ_i^Γ such that for all mixed opponent profiles σ_{-i}^G with corresponding equivalent behavioral opponent profiles σ_{-i}^Γ, we have $u_i(\sigma_i^G, \sigma_{-i}^G) = u_i(\sigma_i^\Gamma, \sigma_{-i}^\Gamma)$. Similarly, a behavioral strategy σ_i in Γ has a utility-equivalent mixed strategy in G where the probability of selecting the pure strategy (action) s_i is $\prod_{I \in \mathcal{I}_i} \sigma_i(I, s_i(I))$. In imperfect recall games, however, Kuhn's theorem does not hold. Intuitively, this is because a mixed strategy in G provides a probability of following each possible sequence of actions in Γ, whereas behavioral strategies in imperfect recall games cannot condition action probabilities from previously forgotten information. For example, consider the imperfect recall game and its normal-form representation in Figure 2.3. In the normal-form game, player 2 (the column player) can guarantee an expected utility of 2 by playing the mixed strategy $\sigma_2 = \{(ac, 0.5), (ad, 0), (bc, 0), (bd, 0.5)\}$. However, a



(a) Extensive form

$$\begin{array}{c} \\ A \\ B \end{array}\begin{array}{c} \begin{array}{cccc} ac & ad & bc & bd \end{array} \\ \begin{pmatrix} -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix} \end{array}$$

(b) Normal form

Figure 2.3: **(a)** A zero-sum extensive-form game with imperfect recall, where game states connected by a bold dashed curve are in the same information set. Values at terminal nodes indicate utilities for player 1; player 2's utilities are the negation. The game has imperfect recall since player 2 takes an action, a or b, and then immediately forgets which action they took. Originally found in Koller and Megiddo [48, Figure 1]. **(b)** The game's normal-form representation created by assigning each pure strategy in the extensive-form game as an action in the normal-form game.

behavioral strategy for player 2 in the extensive-form game can only guarantee an expected utility of at most 1. To see this, let w, x, and y be the probabilities that player 1 plays A, player 2 plays a, and player 2 plays c respectively. Then player 2's best worst-case expected utility is $\max_{x,y \in [0,1]} \min_{w \in [0,1]} 4wxy + 4(1-w)(1-x)(1-y) = 1$ by choosing $x = y = 0.5$.

For an information set $I \in \mathcal{I}_i$, in perfect recall games we let $\pi_i^\sigma(I) = \pi_i^\sigma(h)$ for any $h \in I$. As the sequence of information set, action pairs taken by player i to reach h must be identical for all $h \in I$, this value is well-defined. In addition, we define $\pi_{-i}^\sigma(I) = \sum_{h \in I} \pi_{-i}^\sigma(h)$ to be the probability that chance and players other than i play to reach I. Note that in an imperfect recall game, $\pi_{-i}^\sigma(I)$ may not be a valid probability. For example, consider a game where player i acts at the root and takes one of two actions, a or b, and then acts again but forgets the previous action taken. If we consider the information set $I = \{a, b\}$, then $\pi_{-i}^\sigma(I) = \pi_{-i}^\sigma(a) + \pi_{-i}^\sigma(b) = 1 + 1 = 2$, which is not in the valid probability range $[0, 1]$.

We will now assume perfect recall from here on until we investigate imperfect recall games in Chapter 6.

2.1.2 Solution Concepts

In this thesis, we consider the problem of computing a strategy profile to a game for play against a set of unknown opponents. For this problem, the most common solution concept is the Nash

$$\begin{array}{cc} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} A \\ B \end{array} & \begin{pmatrix} 1,1 & 0,0 & 1,0 \\ 1,1 & 1,0 & 0,0 \end{pmatrix} \end{array}$$

Figure 2.4: A two-player normal-form game where every row player strategy is iteratively weakly dominated. Originally found in Conitzer and Sandholm [15].

equilibrium. For $\epsilon \geq 0$, a strategy profile σ is an ϵ-Nash equilibrium if no player can unilaterally deviate from σ and gain more than ϵ in expected utility.

Definition 2.4. *For $\epsilon \geq 0$, a strategy profile σ is an **ϵ-Nash equilibrium** if*

$$\max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i}) \leq u_i(\sigma) + \epsilon \textit{ for all } i \in N.$$

*A 0-Nash equilibrium is simply called a **Nash equilibrium**.*

This concept is due to John Nash [58], who showed that any finite game must have a Nash equilibrium. In equilibrium, all players are playing a best response to the opponents' strategies.

A Nash equilibrium is most meaningful in zero-sum games. In this case, if $\sigma = (\sigma_1, \sigma_2)$ is a Nash equilibrium, then by playing σ_1, player 1 is guaranteed to earn no worse than $u_1(\sigma)$; if player 2 deviates from σ_2, player 1 may only do the same or better. As a consequence of this, Nash equilibria in zero-sum games are *interchangeable*: if (σ_1^1, σ_2^1) and (σ_1^2, σ_2^2) are both Nash equilibria, then $(\sigma_1^k, \sigma_2^\ell)$ is a Nash equilibrium for any $k, \ell \in \{1, 2\}$. Note that an arbitrary extensive-form game can have an arbitrary number of Nash equilibria associated with it; however, for zero-sum games, all Nash equilibria have the same *game value*. In other words, if σ and σ' are Nash equilibria of a zero-sum game, then

$$u_1(\sigma) = u_1(\sigma') = -u_2(\sigma') = -u_2(\sigma).$$

While a Nash equilibrium of a zero-sum game can be computed in polynomial time, computing a Nash equilibrium of a non-zero-game is hard and belongs to the PPAD-complete class of problems [13, 14, 16, 17].

Nash equilibria also guarantee avoidance from making strictly dominated errors: mistakes where there exists an alternative that is guaranteed to do better, regardless of what the opponents do.

Definition 2.5. *A strategy σ_i for player i is a **weakly dominated strategy** if there exists another player i strategy σ_i' such that*

(i) $u_i(\sigma_i, \sigma_{-i}) \leq u_i(\sigma_i', \sigma_{-i})$ *for all opponent profiles $\sigma_{-i} \in \Sigma_{-i}$, and*

(ii) $u_i(\sigma_i, \sigma_{-i}) < u_i(\sigma_i', \sigma_{-i})$ *for some opponent profile $\sigma_{-i} \in \Sigma_{-i}$.*

*If $u_i(\sigma_i, \sigma_{-i}) < u_i(\sigma_i', \sigma_{-i})$ for all opponent profiles $\sigma_{-i} \in \Sigma_{-i}$, then σ_i is a **strictly dominated strategy**.*

$$\begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{cc} \;\;a & \;\;\;\;b \end{array} \\ \begin{pmatrix} 1,0 & 0,0 \\ 0,0 & 2,0 \\ -1,0 & 1,0 \end{pmatrix} \end{array}$$

Figure 2.5: A two-player non-zero-sum normal-form game, where the column player's utility is always zero.

In addition, **very weakly dominated strategies** have been studied [36, 47] that only require part (i) of Definition 2.5 to hold, but we do not consider very weak dominance here.

For each type of dominance, an **iteratively dominated strategy** can be defined recursively as any strategy that is either dominated or becomes dominated after successively removing iteratively dominated strategies from the game. It is well-known that iterated removal of strictly dominated strategies always results in the same nonempty set of remaining strategies, regardless of the order of removal [28]. However, it is possible to have every strategy for a player be iteratively weakly dominated, such as for the row player in the game in Figure 2.4. Here, any column player strategy that plays b or c with positive probability is iteratively weakly dominated by the strategy that always plays a. If we first remove just b, then B is iteratively weakly dominated for the row player. Likewise, removing just c results in A for the row player being iteratively weakly dominated. This demonstrates that the set of remaining strategies resulting from iterated removal of weakly dominated strategies can be different depending on the order in which they are removed.

Two generalizations of Nash equilibria, correlated and coarse correlated equilibria, require a mechanism for correlation among the players. Suppose an independent moderator selects a profile σ^k from $E = \{\sigma^1, ..., \sigma^K\}$ according to distribution q and privately recommends each player i play strategy σ_i^k. Then (E, q) is a **correlated equilibrium** if no player has an incentive to unilaterally deviate from any recommendation. A **coarse correlated equilibrium** is similar but even more general, where for all $i \in N$, we only require that

$$\sum_{k=1}^{K} q(k)u_i(\sigma^k) \geq \max_{\sigma_i' \in \Sigma_i} \sum_{k=1}^{K} q(k)u_i(\sigma_i', \sigma_{-i}^k). \tag{2.1}$$

To not be in a coarse correlated equilibrium, a player would need incentive to deviate even before receiving a recommendation and the deviation must be independent of the recommendation. For extensive-form games, von Stengel and Forges [69] introduced an alternative generalization of correlated equilibria called **extensive-form correlated equilibria**. Here, recommended strategies are only revealed action-by-action at the players' information sets until a deviation occurs, rather than revealing the entire strategy before play. Extensive-form correlated equilibria can be efficiently approximated even in large games [18].

Without a mechanism for correlation, it is unclear how a practitioner should use a correlated equilibrium. In addition, while correlated equilibria remove strictly dominated strategies [9], a coarse correlated equilibrium may lead to the recommendation of a strictly dominated strategy. For example, in the normal-form game in Figure 2.5, the distribution $q = \{(A,a) = 0.5, (B,b) = 0.25, (C,b) = 0.25\}$ is a coarse correlated equilibrium with the row player's expected utility being $5/4$, yet the strictly dominated row player strategy that always plays C is recommended 25% of the time.

2.2 Regret Minimization

Given a sequence of strategy profiles $\sigma^1, \sigma^2, ..., \sigma^T$, the **(external) regret** for player i is

$$R_i^T = \max_{\sigma_i' \in \Sigma_i} \sum_{t=1}^{T} \left(u_i(\sigma_i', \sigma_{-i}^t) - u_i(\sigma^t) \right). \tag{2.2}$$

R_i^T measures the amount of utility player i could have gained by following the single best fixed strategy in hindsight over all time steps $t = 1, ..., T$. We say that an algorithm is **regret minimizing** for player i if it generates player i strategies such that $R_i^{T,+}/T \to 0$ as $T \to \infty$, where $x^+ = \max\{x, 0\}$. The following well-known "folk theorem" states the connection between regret minimization and Nash equilibrium in zero-sum games:

Theorem 2.1. *Let $\epsilon \geq 0$. In a zero-sum game with perfect recall, if $R_i^T/T \leq \epsilon$ for both players $i = 1, 2$, then the average of the strategy profiles $\bar{\sigma}^T$ (defined below) is a 2ϵ-Nash equilibrium.*

A proof is provided by Waugh [70, p. 11], while we provide a proof to a more general result in Chapter 4. It is also well-known that in any game, minimizing internal regret, a stronger notion of regret, for all players leads to a correlated equilibrium, but we only consider external regret here.

In a normal-form game with mixed strategy profiles $\sigma^1, \sigma^2, ..., \sigma^T$, the average profile $\bar{\sigma}^T$ is defined in the obvious way, where $\bar{\sigma}_i^T(a) = \sum_{t=1}^{T} \sigma_i^t(a)/T$ for all $i \in N$, $a \in A_i$. For extensive-form games and behavioral strategy profiles $\sigma^1, \sigma^2, ..., \sigma^T$, the corresponding definition of the average profile is

$$\bar{\sigma}_i^T(I,a) = \frac{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I) \sigma_i^t(I,a)}{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)}$$

for all $i \in N$, $I \in \mathcal{I}_i$, and $a \in A(I)$.

2.2.1 Regret Matching

Regret matching [34, 37] is a very simple, iterative procedure that minimizes regret in a normal-form game. First, the initial profile σ^1 is chosen arbitrarily. For each action $a \in A_i$, we store the accumulated regret $R_i^T(a) = \sum_{t=1}^{T} \left(u_i(a, \sigma_{-i}^t) - u_i(\sigma_i^t, \sigma_{-i}^t) \right)$ that measures how much player i would rather have played a at each time step t than follow σ_i^t. Successive strategies are then determined according to

$$\sigma_i^{T+1}(a) = \frac{R_i^{T,+}(a)}{\sum_{b \in A_i} R_i^{T,+}(b)}, \tag{2.3}$$

where actions are chosen arbitrarily when the denominator is zero.

Algorithm 1 Counterfactual Regret Minimization (Vanilla CFR) [75]

```
Initialize regret: $\forall I, a \in A(I) : R(I, a) \leftarrow 0$
Initialize cumulative profile: $\forall I, a \in A(I) : s(I, a) \leftarrow 0$
Initialize current profile: $\forall I, a \in A(I) : \sigma(I, a) = 1/|A(I)|$
for $t \in \{1, 2, ..., T\}$ do
    for $i \in N$ do
        for $I \in \mathcal{I}_i$ do
            $\sigma_i(I, \cdot) \leftarrow$ RegretMatching($R(I, \cdot)$)
            for $a \in A(I)$ do
                $R(I, a) \leftarrow R(I, a) + v_i(I, \sigma_{(I \to a)}) - v_i(I, \sigma)$
                $s(I, a) \leftarrow s(I, a) + \pi_i^{\sigma}(I)\sigma_i(I, a)$
            end for
        end for
    end for
end for
```

The following theorem shows that when using regret matching, a player's regret grows proportional to the square root of the number of time steps. A proof of a more general result is provided by Gordon [33], but for completeness, we provide a simple proof of Theorem 2.2 in Appendix A.

Theorem 2.2. *If player i uses regret matching, then after T time steps,*

$$R_i^T \leq \Delta_i \sqrt{T|A_i|}.$$

2.2.2 Counterfactual Regret Minimization (CFR)

Consider now minimizing regret in an extensive-form game. One possibility is to apply regret matching to the derived normal-form game, such as the game in Figure 2.3b. However, regret matching requires storage of $R_i^T(a)$ for all pure strategies $a \in A_i = \mathcal{S}_i$ in the extensive-form game, and $|\mathcal{S}_i|$ is exponential in $|\mathcal{I}_i| \cdot |A(\mathcal{I}_i)|$. Thus, this possibility is infeasible for even moderately-sized extensive-form games due to the resulting exponential size of the storage required and exponential time required to update strategy profiles on each iteration.

Alternatively, **Counterfactual Regret Minimization (CFR)** [75] is a state-of-the-art algorithm that minimizes regret while only requiring storage proportional to $|\mathcal{I}_i| \cdot |A(\mathcal{I}_i)|$ in the extensive-form game. To distinguish CFR from its variants presented later in Section 2.2.3 and Chapter 5, we often refer to CFR as **Vanilla CFR**. Pseudocode is provided in Algorithm 1. On each iteration t and for each player i, the expected utility for player i is computed at each information set $I \in \mathcal{I}_i$ under the current profile σ^t, assuming player i plays to reach I. This expectation is the **counterfactual value** for player i,

$$v_i(I, \sigma) = \sum_{z \in Z_I} u_i(z) \pi_{-i}^{\sigma}(z[I]) \pi^{\sigma}(z[I], z),$$

where Z_I is the set of terminal histories passing through I and $z[I]$ is the history leading to z contained in I. For each action $a \in A(I)$, these values determine the **counterfactual regret** at iteration t, $r_i^t(I, a) = v_i(I, \sigma_{(I \to a)}^t) - v_i(I, \sigma^t)$, where $\sigma_{(I \to a)}$ is the profile σ except at I, action

a is always taken. The regret $r_i^t(I,a)$ measures how much player i would rather play action a at I than follow σ_i^t at I, assuming player i plays to reach I. These regrets are accumulated to obtain the **cumulative counterfactual regret**, $R_i^T(I,a) = \sum_{t=1}^T r_i^t(I,a)$, that define the **current strategy profile** via regret matching at I,

$$\sigma_i^{T+1}(I,a) = \frac{R_i^{T,+}(I,a)}{\sum_{b \in A(I)} R_i^{T,+}(I,b)}. \tag{2.4}$$

When the denominator of equation (2.4) is zero, we simply set $\sigma_i^{T+1}(I,\cdot)$ to play uniformly random at I. During computation, CFR stores a **cumulative profile** $s_i^T(I,a) = \sum_{t=1}^T \pi_i^{\sigma^t}(I)\sigma_i^t(I,a)$. Once CFR is terminated after T iterations, the output is the average strategy profile $\bar{\sigma}_i^T(I,a) = s_i^T(I,a)/\sum_{b\in A(I)} s_i^T(I,b)$. We provide a walk-through of Vanilla CFR on a small extensive-form game in Appendix A.

A key result in the original CFR analysis shows that player i's regret is bounded by the sum of the positive parts of the cumulative counterfactual regrets.

Theorem 2.3 (Zinkevich *et al.* [75]). *In an extensive-form game with perfect recall,*

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

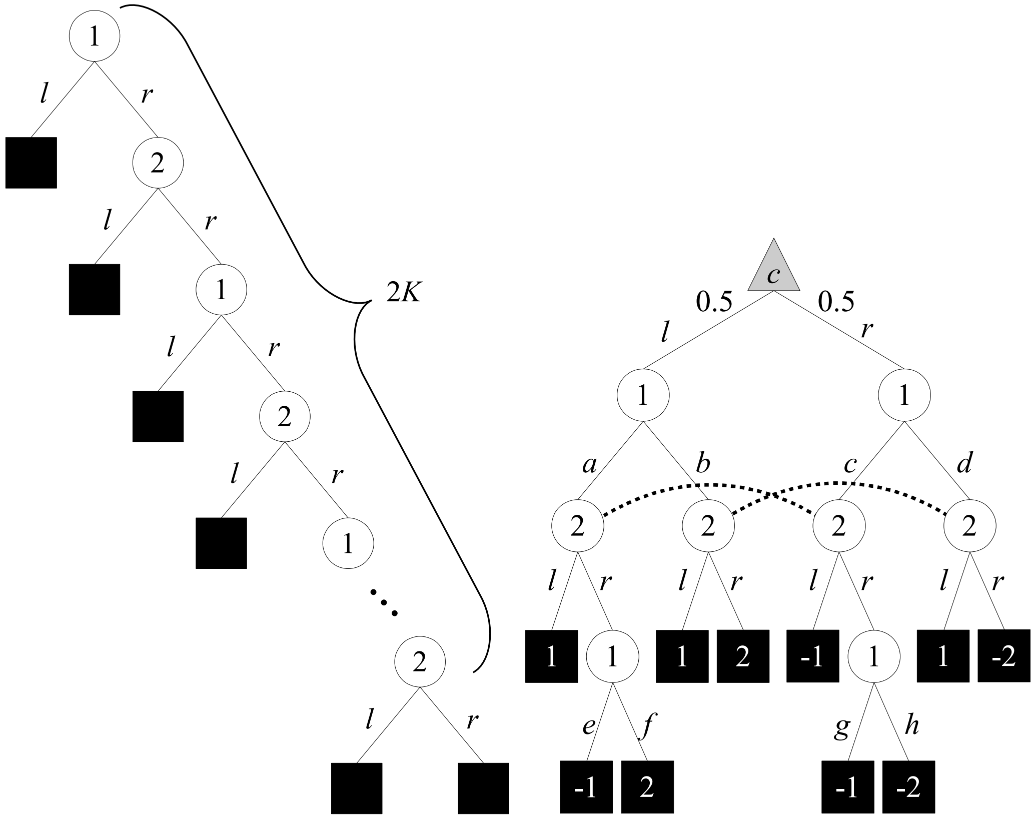
Thus, since each $R_i^T(I,a)$ is minimized via regret matching at I, it follows by Theorem 2.3 that player i's regret is also minimized. This in turn implies that, by Theorem 2.1, the average strategy profile in a zero-sum game converges to a Nash equilibrium.

Prior to our work in Chapter 5, the best known bound on the average regret was due to Lanctot *et al.* [54] and requires some addition notation to present. For each player i, let $\mathcal{B}_i$ be the partition of $\mathcal{I}_i$ such that two information sets I, I' are in the same part $B \in \mathcal{B}_i$ if and only if the sequence of player i's actions leading to I is the same as the sequence of player i's actions leading to I'. $\mathcal{B}_i$ is well-defined due to perfect recall, as player i's actions leading to an information set $I \in \mathcal{I}_i$ cannot be different among the histories $h \in I$. Next, define the **M-value of the game to player i** to be $M_i = \sum_{B \in \mathcal{B}_i} \sqrt{|B|}$.

Theorem 2.4 (Lanctot *et al.* [54]). *When using Vanilla CFR in a game with perfect recall, average regret is bounded by*

$$\frac{R_i^T}{T} \le \frac{\Delta_i M_i \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}}.$$

The M-value can range anywhere between $\sqrt{|\mathcal{I}_i|} \le M_i \le |\mathcal{I}_i|$ with each side of the bound being realized by some game. For example, in the game shown in Figure 2.2, for each $i = 1,2$, $\mathcal{B}_i$ has just one part B that contains all of player i's information sets, and thus $M_i = \sqrt{|\mathcal{I}_i|}$. However, in the game in Figure 2.6a, each part of $\mathcal{B}_i$ is a singleton. This is because every information set is reached by an action sequence of different length. Thus, $M_i = \sum_{B \in \mathcal{B}_i} \sqrt{|B|} = \sum_{I \in \mathcal{I}_i} \sqrt{1} = |\mathcal{I}_i|$.



(a)

(b)

Figure 2.6: **(a)** A two-player extensive-form game where both players have K information sets. The M-value for both players is $M_i = K = |\mathcal{I}_i|$. **(b)** A two-player game where the M-value for player 1 is dependent on the labeling of the actions.

A subtle note is that the M-value can be dependent on the labeling of the actions, as in the game in Figure 2.6b. If player 1's actions a and c are labeled differently, then the information sets $\{lar\}$ and $\{rcr\}$ are in different parts of the partition $\mathcal{B}_1$, giving $M_1 = 2 + \sqrt{2}$. However, if a and c are identically labeled by, say, replacing the label c with the label a, then these two information sets are in the same part within $\mathcal{B}_1$ and $M_1 = 2\sqrt{2}$.

2.2.3 Monte Carlo CFR

For large games, CFR's full game tree traversals can be very expensive. Alternatively, one can traverse a smaller, sampled portion of the tree on each iteration using **Monte Carlo CFR (MCCFR)** [54]. Let $\mathcal{Q} = \{Q_1, ..., Q_K\}$ be a set of subsets, or **blocks**, of the terminal histories Z such that the union of $\mathcal{Q}$ spans Z. For example, **Chance Sampling (CS)** [75] is an instance of MCCFR that partitions Z into blocks such that two histories are in the same block if and only if all corresponding chance actions are the same. Thus, CS applied to the game in Figure 2.6b would yield two blocks in $\mathcal{Q}$, $Q_1 = \{lal, lare, larf, lbl, lbr\}$ and $Q_2 = \{rcl, rcrg, rcrh, rdl, rdr\}$. On each iteration, a block Q_j is sampled with probability q_j, where $\sum_{k=1}^{K} q_k = 1$. In CS, we generate a block by sampling a single action a at each history $h \in H$ with $P(h) = c$ according to its likelihood of occurring, $\sigma_c(h, a)$. In general, the **sampled counterfactual value** for player i is

$$\tilde{v}_i(I, \sigma) = \sum_{z \in Z_I \cap Q_j} u_i(z)\pi^{\sigma}_{-i}(z[I])\pi^{\sigma}(z[I], z)/q(z), \tag{2.5}$$

where $q(z) = \sum_{k: z \in Q_k} q_k$ is the probability that z was sampled. For example, in CS, $q(z) = \pi^{\sigma}_c(z)$. When $q(z) > 0$ for all $z \in Z$ satisfying $\pi^{\sigma}_{-i}(z) > 0$, $\tilde{v}_i(I, \sigma)$ is an unbiased estimate of the true counterfactual value $v_i(I, \sigma)$ [54, Lemma 1][1]. Define the **sampled counterfactual regret** for action a at I to be $\tilde{r}^t_i(I, a) = \tilde{v}_i(I, \sigma^t_{(I \to a)}) - \tilde{v}_i(I, \sigma^t)$. Strategies are then generated by applying equation (2.4) to the **sampled cumulative counterfactual regret** $\tilde{R}^T_i(I, a) = \sum_{t=1}^{T} \tilde{r}^t_i(I, a)$.

MCCFR results in faster iterations than Vanilla CFR since we only need to traverse to the histories in Q to compute the sampled counterfactual regrets. In games with many possible chance outcomes, CS significantly reduces traversal time, although more iterations are required before convergence. Nonetheless, this trade-off has been shown to significantly reduce computing time in poker games [74, Appendix A.5.2]. Other instances of MCCFR include **External Sampling (ES)** and **Outcome Sampling (OS)** [54]. ES takes CS one step further by considering only a single action for not only chance, but also for the opponents, where opponent actions are sampled according to the current profile σ^t_{-i}. OS is the most extreme version of MCCFR that samples a single action at every history, walking just a single trajectory through the tree on each traversal ($Q_j = \{z\}$). ES and OS converge to equilibrium faster than Vanilla CFR in a number of different domains [54, Figure 1].

Pseudocode for an implementation of ES is presented in Algorithm 2. ES and other MCCFR instances are conveniently implemented as recursive procedures that obtain utilities by recursing down the tree, and then use these values to update counterfactual regret while recursing back up the tree. In Algorithm 2, the recursive function WalkTree considers four different cases. Firstly, if we have reached a terminal node (line 6), we simply return the utility at that node. Secondly, when at a chance node (line 7), we sample a single action according to σ_c and recurse down that action. Thirdly, at an opponent's choice node (lines 9 to 14), we again sample a single action and recurse, this time according to the opponent's current strategy obtained via regret matching (equation (2.4)). In two-player games, we also update the cumulative profile (line 11). We do so here because the current strategy is conveniently an unbiased estimate of the value $\pi^{\sigma}_{-i}(I)\sigma_{-i}(I, a)$ that we want to add to our cumulative profile. For games with more than two players, UpdateCumulativeProfile is called instead after WalkTree updates the regret. Note that in practice, UpdateCumulativeProfile need not be called on every iteration and can instead be called every $1/p$ iterations on average to save computation time, where $p > 0$ is a probability parameter. Throughout this dissertation, when using MCCFR on a game with more than two players, we use $p = 0.001$ as this choice worked well in preliminary experiments. Later in Chapter 4, we show that in games with more than two players,

[1] When $\pi^{\sigma}_{-i}(z) = 0$, $\tilde{v}_i(I, \sigma)$ is still an unbiased estimate if $q(z) = 0$ and we simply treat z's contribution to the sum in equation (2.5) as zero.

Algorithm 2 External Sampling [54]

```
1: if $n > 2$ then require: Update cumulative profile probability, $p \in (0, 1]$
2: Initialize regret: $\forall I, \forall a \in A(I) : R(I, a) \leftarrow 0$
3: Initialize cumulative profile: $\forall I, \forall a \in A(I) : s(I, a) \leftarrow 0$
4:
5: WalkTree(history $h$, player $i$):
6:     if $h \in Z$ then return $u_i(h)$ end if
7:     if $P(h) = c$ then Sample action $a \sim \sigma_c(h, \cdot)$, return WalkTree($ha$, $i$) end if
8:     $I \leftarrow I(h)$, $\sigma(I, \cdot) \leftarrow$ RegretMatching($R(I, \cdot)$)
9:     if $P(h) \neq i$ then
10:        if $n = 2$ then
11:            for $a \in A(I)$ do $s(I, a) \leftarrow s(I, a) + \sigma(I, a)$ end for
12:        end if
13:        Sample action $a \sim \sigma(I, \cdot)$, return WalkTree($ha$, $i$)
14:    end if
15:    for $a \in A(I)$ do $\tilde{v}(a) \leftarrow$ WalkTree($ha$, $i$) end for
16:    for $a \in A(I)$ do $R(I, a) \leftarrow R(I, a) + \tilde{v}(a) - \sum_{a \in A(I)} \sigma(I, a)\tilde{v}(a)$ end for
17:    return $\sum_{a \in A(I)} \sigma(I, a)\tilde{v}(a)$
18:
19: UpdateCumulativeProfile(history $h$, player $i$)
20:    if $P(h) = c$ then Sample action $a \sim \sigma_c(h, \cdot)$, UpdateCumulativeProfile($ha$, $i$)
21:    else if $h \notin Z$ and $P(h) \neq i$ then
22:        for $a \in A(h)$ do UpdateCumulativeProfile($ha$, $i$) end for
23:    else if $P(h) = i$ then
24:        $I \leftarrow I(h)$, $\sigma(I, \cdot) \leftarrow$ RegretMatching($R(I, \cdot)$)
25:        for $a \in A(I)$ do $s(I, a) \leftarrow s(I, a) + \sigma(I, a)$ end for
26:        Sample action $a \sim \sigma(I, \cdot)$, UpdateCumulativeProfile($ha$, $i$)
27:    end if
28:
29: Solve(iterations $T$):
30:    for $t \in \{1, 2, ..., T\}$ do
31:        for $i \in N$ do WalkTree($\emptyset$, $i$) end for
32:        if $n > 2$ and $Random(0, 1) < p$ then
33:            for $i \in N$ do UpdateCumulativeProfile($\emptyset$, $i$) end for
34:        end if
35:    end for
```

the average strategy can be discarded in favour of the current strategy without any significant loss in theoretical guarantees.

The final case in WalkTree handles choice nodes for player i (lines 15 to 17). For each action $a \in A(I)$, we recurse to obtain the sampled counterfactual value $\tilde{v}(a) = \tilde{v}_i(I, \sigma^t_{(I \to a)})$ (line 15). We then update the regrets at I (line 16) and return the sampled counterfactual value at I, $\sum_{a \in A(I)} \sigma(I, a)\tilde{v}(a) = \tilde{v}_i(I, \sigma^t)$. Note that ES does not need to weight the sampled counterfactual values by the probability of the opponent reaching each terminal history, $\pi^{\sigma}_{-i}(z)$, as this is exactly the probability of sampling z, $q(z)$, and is canceled out in equation 2.5.

ES and OS yield a probabilistic bound on the average regret given below in Theorem 2.5, and thus provide a probabilistic guarantee that $\bar{\sigma}^T$ converges to a Nash equilibrium in zero-sum games. A probabilistic bound can also be established for CS, though it appears to have been overlooked in previous work. We present that bound in Chapter 5. Since both ES and OS generate blocks by sampling actions independently, we can decompose $q(z) = \prod_{i \in N \cup \{c\}} q_i(z)$ so that $q_i(z)$ is the probability contributed to $q(z)$ by sampling player i's actions. For example, when updating player i's regret with ES, we have $q_i(z) = 1$ because all of player i's actions are sampled.

Theorem 2.5 (**Lanctot *et al.* [54]**). *Let X be one of ES or OS (assuming OS also samples chance and opponent actions according to σ_{-i}), let $p \in (0, 1]$, and let $\delta = \min_{z \in Z} q_i(z) > 0$ over all $1 \le t \le T$ when updating player i's regret. When using X in a game with perfect recall, with probability $1 - p$, player i's average regret is bounded by*

$$\frac{R_i^T}{T} \le \left(M_i + \frac{\sqrt{2|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}} \right) \left(\frac{1}{\delta} \right) \frac{\Delta_i \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}}.$$

Theorem 2.5 is presented slightly differently by Lanctot *et al.* [54], but the last step of their proof mistakenly used $M_i \ge \sqrt{|\mathcal{I}_i||\mathcal{B}_i|}$, which is actually incorrect. For example, in the game in Figure 2.6b with $a \ne c$, we have $M_i = 2 + \sqrt{2}$, but $\sqrt{|\mathcal{I}_i||\mathcal{B}_i|} = \sqrt{4 \cdot 3} = \sqrt{12} > 2 + \sqrt{2}$. The bound we present in Theorem 2.5 is correct.

2.3 Other Solution Concepts and Techniques

While MCCFR is currently the best known technique for solving large zero-sum extensive-form games, a number of other methods have been proposed. Koller *et al.* [49] developed a linear programming approach for solving extensive-form games without the need to convert to the derived normal-form game. This is done by representing the players' strategies σ_i in sequence form according to a **realization plan** $\beta_i(I, a) = \pi_i^\sigma(I)\sigma_i(I, a)$. A sparse matrix U is then constructed where each column represents a player 1 sequence and each row represents a player 2 sequence. When a player 1 sequence together with a player 2 sequence correspond to a set of terminal histories, the corresponding entry in U gives the expected utility for player 1 at those terminal histories; all other entries of U are zero. For example, for the game in Figure 2.6b, U is the matrix

$$\begin{array}{c} \\ \emptyset_2 \\ \{lal, rcl\} \\ \{lar, rcr\} \\ \{lbl, rdl\} \\ \{lbr, rdr\} \end{array} \begin{array}{c} \begin{array}{ccccccccc} \emptyset_1 & la & lb & rc & rd & lare & larf & rcrg & rcrh \end{array} \\ \left(\begin{array}{ccccccccc} & & & & & & & & \\ & 0.5 & & -0.5 & & & & & \\ & & & & & -0.5 & 1 & -0.5 & -1 \\ & & 0.5 & & 0.5 & & & & \\ & & 1 & & -1 & & & & \end{array} \right) \end{array}$$

where blank entries are zero. Note that all utilities in Figure 2.6b have been multiplied by one-half in U to account for chance's likelihood for reaching each terminal history. A solution to the optimization problem

$$\min_{\beta_2} \max_{\beta_1} \beta_2^T U \beta_1 \tag{2.6}$$

$$\textbf{subject to } \forall I \in \mathcal{I}, \sum_{a \in A(I)} \beta(I,a) = \rho(I),$$
$$\forall I \in \mathcal{I}, a \in A(I), \beta(I,a) \geq 0$$

is a Nash equilibrium, where $\rho(I) = \beta(J,a)$ with J and a being the last information set visited and action taken respectively by player $P(I)$ before reaching I, or $\rho(I) = 1$ if I is the first information set visited. By taking the dual of the inner maximization problem, one can then solve the resulting minimization problem for an equilibrium in polynomial time. As for memory requirements, note that the number of non-zero entries in U is the number of sequence pairs for the players that result in a terminal history. This number is $|\mathcal{I}_1||A(\mathcal{I}_1)| \cdot |\mathcal{I}_2||A(\mathcal{I}_2)|$ in the worst case, whereas CFR only requires space proportional to $|\mathcal{I}_1||A(\mathcal{I}_1)| + |\mathcal{I}_2||A(\mathcal{I}_2)|$.

While this linear programming approach solves for an exact equilibrium, even the best linear programming implementations can take a long time to solve games with millions of sequences per player. Alternatively, one can relax the optimization problem to instead find an ϵ-Nash equilibrium in much less time. This is done in the *excessive gap technique (EGT)* [38] by augmenting the objective function of (2.6) to make it differentiable and convex so that a solution is suboptimal by at most some value ϵ^0. EGT iteratively generates solutions with suboptimality $\epsilon^0 > \epsilon^1 > \epsilon^2 > ...$ by applying gradient descent along the objective function, with $\epsilon^T \to 0$ as $T \to \infty$. EGT was successfully used to solve a version of poker called *Rhode Island hold'em* [29] that contains over 3.1×10^9 histories. Unfortunately, EGT can only be applied to zero-sum games with perfect recall. CFR, on the other hand, can be applied to non-zero-sum games [3] and games with imperfect recall [45, 73], as we further study in Chapters 4 and 6 respectively.

With regards to elimination of dominated strategies, Conitzer and Sandholm [15] prove that a strictly dominated strategy $\sigma_i \in \Sigma_i$ in a normal-form game can be identified in time polynomial in $|A_i| = |\mathcal{S}_i|$ by showing that the objective of the linear program

$$\begin{aligned} &\textbf{minimize} \sum_{s_i \in \mathcal{S}_i} p_{s_i} \\ &\textbf{subject to } \forall s_{-i} \in \mathcal{S}_{-i}, \sum_{s_i \in \mathcal{S}_i} p_{s_i} u_i(s_i, s_{-i}) \geq u_i(\sigma_i, s_{-i}) \\ &\qquad \forall s_i \in \mathcal{S}_i, p_{s_i} \geq 0 \end{aligned} \tag{2.7}$$

is less than 1. Iteratively strictly dominated pure strategies can then be eliminated by repeatedly solving this program up to $O(|\mathcal{S}|^2)$ times and iteratively removing the dominated pure strategies from $\mathcal{S}_i$ and $\mathcal{S}_{-i}$. An alternative linear program can be used to prove weak dominance, but as mentioned earlier, iterative weak dominance can be dependent on the order in which weakly dominated strategies are removed. However, these methods are infeasible for large extensive-form games as the linear programs would require an exponential number of constraints in the size of the game. In addition, Hansen *et al.* [36] develop a dynamic programming algorithm for partially observable stochastic games, a generalization of normal-form games, that removes iteratively very weakly dominated strategies, but is not practical beyond small problems. In Chapter 4, we will provide evidence that CFR eliminates iterative strict domination in non-zero-sum extensive-form games.

Finally, there are two other solution concepts associated with the notion of regret minimization. Both concepts define the regret of a strategy σ_i to be the maximum amount of utility that could have been gained across all possible alternative strategies σ'_i and opponent profiles σ_{-i},

$$regret_i(\sigma_i) = \max_{\substack{\sigma'_i \in \Sigma_i \\ \sigma_{-i} \in \Sigma_{-i}}} u_i(\sigma'_i, \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}).$$

Firstly, Renou and Schlag [61] define $\sigma^* \in \Sigma$ as a *minimax regret equilibrium* relative to Σ if

$$regret_i(\sigma^*_i) \leq regret_i(\sigma_i) \text{ for all } \sigma_i \in \Sigma_i \text{ and all } i \in N.$$

The authors also define the *ϵ-minimax regret equilibrium* variant where with probability $1 - \epsilon$ the opponents are assumed to play according to the equilibrium, and with probability ϵ no assumption is made. This can lead to an ϵ-minimax regret equilibrium that plays iteratively strictly dominated strategies [61, p. 276]. Secondly, Halpern and Pass [35] introduce *iterated regret minimization*. Much like iterated removal of dominated strategies, the authors iteratively remove all strategies σ_i that do not provide minimal $regret_i(\sigma_i)$. They show that while the set of non-iteratively strictly dominated strategies can be disjoint from those that survive iterated regret minimization, their solutions match closely to those solutions played by real people in a number of small games. Our work in this dissertation is less concerned with understanding how humans arrive at solutions and more concerned with understanding and improving CFR to increase performance in large extensive-form games, such as Texas hold'em.

2.4 Abstraction

A common approach to computing strategies in extensive-form games is summarized in Figure 2.7. For small games, we can simply compute a strategy directly using, for instance, CFR. However, many real-world problems have very large extensive form representations, which makes strategy computation with CFR and other techniques infeasible. Instead, an abstract game can be created by combining information sets together into single abstract states, restricting the actions a player can take, or both. If the abstract game is sufficiently small, we can then compute a strategy for the abstract game. Finally, we must decide how to translate the abstract game strategy in order to employ it in the real game.

Abstraction can be formally described as follows:

Definition 2.6 (Waugh *et al.* [72]). *An **abstraction for player i** is a pair $\alpha_i = \langle \alpha_i^{\mathcal{I}}, \alpha_i^A \rangle$, where*

- *$\alpha_i^{\mathcal{I}}$ is a partition of H_i defining a **state abstraction** or a set of **abstract information sets** coarser than $\mathcal{I}_i$ (i.e., every $I \in \mathcal{I}_i$ is a subset of some set in $\alpha_i^{\mathcal{I}}$), and*

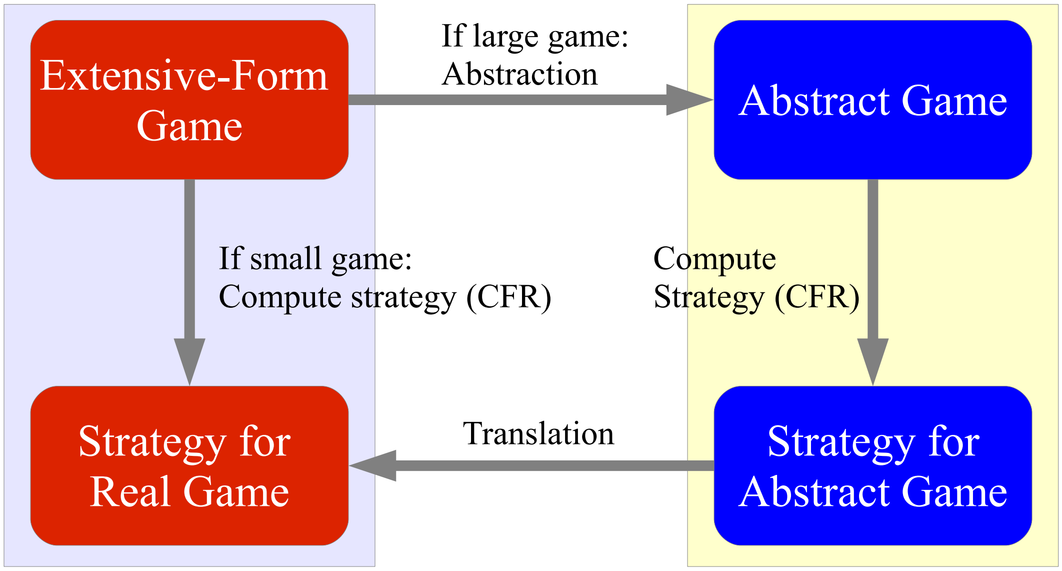

Extensive-Form Game
If large game: Abstraction
Abstract Game
If small game: Compute strategy (CFR)
Compute Strategy (CFR)
Strategy for Real Game
Translation
Strategy for Abstract Game

Figure 2.7: An overview of the process for creating a strategy for playing an extensive-form game. We start with an extensive-form game in the top left and work our way to a strategy for the game in the bottom left.

- *α_i^A is a function on histories where $\alpha_i^A(h) \subseteq A(h)$ and $\alpha_i^A(h) = \alpha_i^A(h')$ for all histories h and h' in the same abstract information set. We will call this the* ***action abstraction*** *or the* ***abstract action set****.*

The ***null abstraction*** *for player i is $\phi_i = \langle \mathcal{I}_i, A \rangle$. An* ***abstraction*** *α is a set of abstractions α_i, one for each player. Finally, for any abstraction α, the* ***abstract game****, Γ^α, is the extensive-form game obtained from Γ by replacing $\mathcal{I}_i$ with $\alpha_i^{\mathcal{I}}$ and $A(h)$ with $\alpha_i^A(h)$ when $P(h) = i$, for all $i \in N$.*

Figure 2.8 shows an example of an abstraction of the game in Figure 2.2. In this example, player 1's two information sets have been merged into a single abstract information set, so $\alpha_1^{\mathcal{I}} = \{\{a, b, d\}\}$. Player 1's action set has not been abstracted, and thus $\alpha_1^A = A$. For player 2, we now have four abstract information sets, namely $\alpha_2^{\mathcal{I}} = \{\{al, bl\}, \{ar, br\}, \{dl\}, \{dr\}\}$. However, at some abstract information sets, player 2's actions have been restricted so that $\alpha_2^A(ar) = \alpha_2^A(br) = \{l\}$ and $\alpha_2^A(dl) = \{r\}$.

Once we have a strategy σ_i^α for playing an abstract game Γ^α, we must translate σ_i^α into a strategy σ_i in the real game Γ. When no action abstraction is applied so that $\alpha_i^A = A$ for all $i \in N$, the translation process is easy; for $I \in \mathcal{I}_i$ with $I \subseteq I^\alpha \in \alpha_i^{\mathcal{I}}$, simply set $\sigma_i(I) = \sigma_i^\alpha(I^\alpha)$. For abstract games with action abstraction, translation can become trickier. For example, one could simply interpret each possible opponent action from the real game as a fixed legal action in the abstract game, or probabilistically map each real game action to several possible legal interpretations [64]. For the majority of this dissertation, we do not apply action abstractions. Intuitively, if σ^α performs well in Γ^α, and if $\alpha_i^{\mathcal{I}}$ is defined such that merged information sets are "strategically similar," then σ

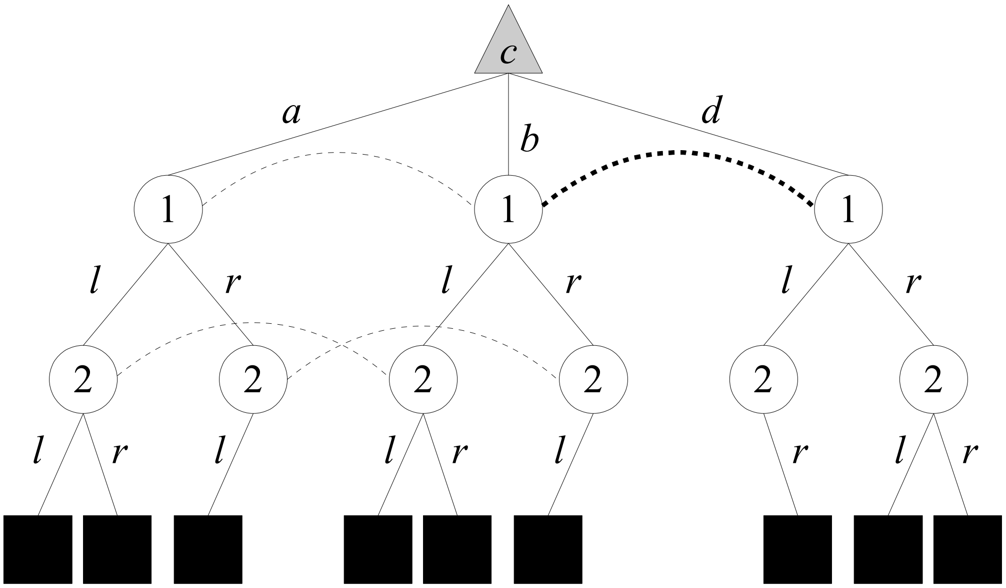


Figure 2.8: An abstraction of the game in Figure 2.2, where bold dashed curves denote game states in the same information set and thin dashed curves denote merged information sets. Here, player 1 cannot distinguish between any of chance's actions and player 2 cannot distinguish between a and b. If chance generates a or b and player 1 takes action r, player 2 can no longer take action r. If chance generates c and player 1 takes action l, player 2 cannot take action l.

is also likely to perform well in Γ. Identifying strategically similar information sets can be delicate though and typically becomes a domain-specific task.

One should also be aware that applying abstraction leads to no guarantees on how well equilibrium strategies of the abstract game perform in the real game. In fact, abstraction pathologies are known where equilibrium strategies in one abstract game are less exploitable in the real game than equilibrium strategies in a second abstract game, where the second abstraction is a strict refinement of the first [72]. On the other hand, Johanson *et al.* [42] recently developed a new regret minimization algorithm called *CFR-BR* that converges to the least exploitable strategy in an unabstracted zero-sum game that is representable in a given abstract space. By keeping the opponent's information unabstracted, their procedure eliminates pathologies, but can be computationally expensive in large games. In this dissertation, we simply compute abstract game strategies using CFR and its variants.

2.4.1 Strategy Grafting

Although these abstraction pathologies exist, we generally would like to have as fine granularity in our abstractions as possible. Larger abstract games allow the abstract strategies to differentiate between more situations that may need to be treated differently in the real game.

A natural approach is to break the game down into subtrees and compute a strategy for each subtree independently. Divide-and-conquer methodologies have been used, for example, in chess

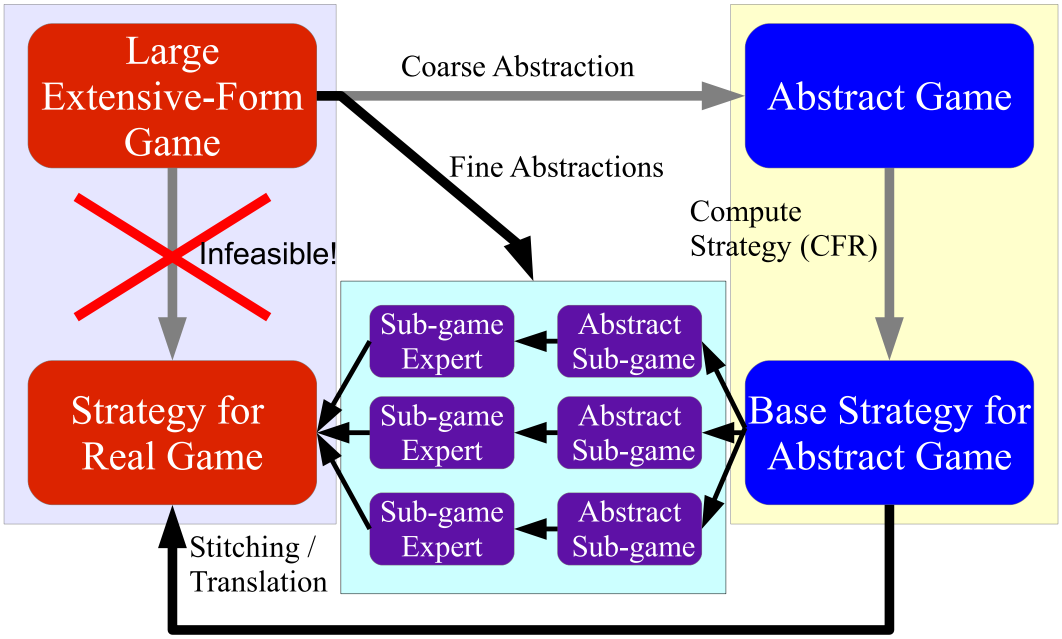


Figure 2.9: An overview of a general procedure for creating a stitched strategy in an extensive-form game. The sub-games depend on both a base strategy and a partition of the game tree.

to solve small sub-games near the leaves of the tree [68]. Using a bottom-up approach, the results can then be backed up to higher places in the tree, reducing the effort required to play optimally. However, in games such as poker with imperfect information, subtrees are often dependent because action probabilities for histories in different subtrees but the same information set must be consistent. In addition, an opponent may choose actions differently according to his or her private information and thus reaches a particular subtree with some distribution over private states. This should be taken into consideration when computing a strategy for a smaller subtree of the entire game.

Strategy grafting is an instance of strategy stitching discussed in Chapter 1 and uses the general procedure shown in Figure 2.9. First, a *base strategy* $\sigma_i \in \Sigma_i$ for player i is computed for playing the undivided game, typically using abstraction as described in Figure 2.7. Secondly, the game is divided into sub-games:

Definition 2.7 (Waugh *et al.* [71]). *$G_i = \{G_{i,0}, G_{i,1}, ..., G_{i,p}\}$ is a **grafting partition for player i** if*

- *G_i is a partition of H_i,*
- *$\forall I \in \mathcal{I}_i$, $\exists j \in \{0, 1, ..., p\}$ such that $I \subseteq G_{i,j}$, and*
- *$\forall j \in \{1, 2, ..., p\}$, if $h \sqsubseteq h' \in H_i$ and $h \in G_{i,j}$, then $h' \in G_{i,j} \cup G_{i,0}$.*

Then, since the sub-games are disjoint, expert strategies for these sub-games, or *grafts*, can be computed and combined without any overlap to the base strategy in the undivided game:

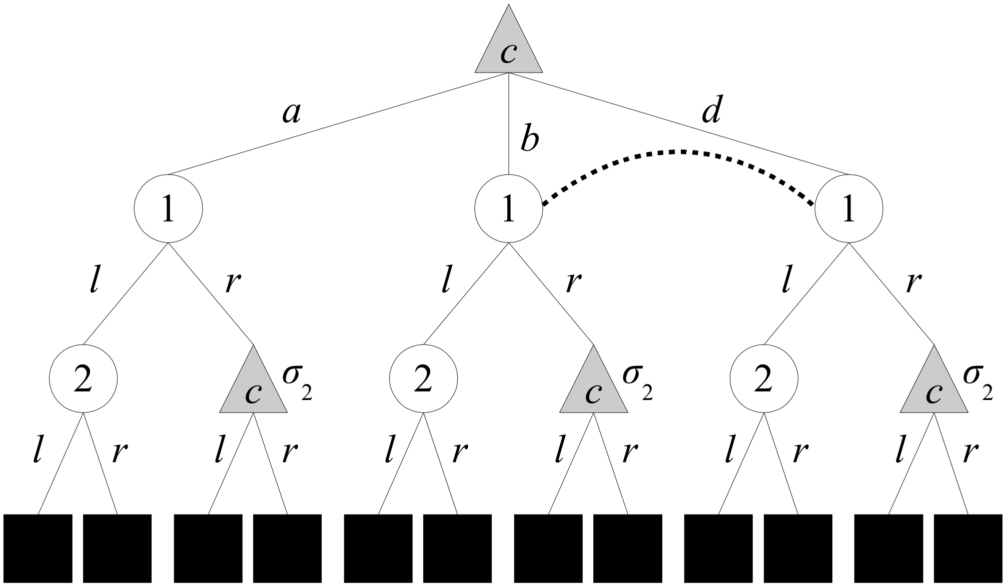


Figure 2.10: An example of a game $\Gamma^{\sigma_2,j}$ for strategy grafting derived from the game Γ in Figure 2.2. Here, if player 1 takes action r, player 2 no longer controls which actions to take. The actions are instead generated by the base strategy σ_2, computed beforehand.

Definition 2.8 (**Waugh *et al.* [71]).** *Let $\sigma_i \in \Sigma_i$ be a strategy for player i and G_i be a grafting partition for player i. For $j \in \{1, 2, ..., p\}$, define $\Gamma^{\sigma_i,j}$ to be an extensive-form game derived from the original game Γ where, for all $h \in H_i \backslash G_{i,j}$, we set $P(h) = c$ and $\sigma_c(h, a) = \sigma_i(h, a)$. That is, player i only controls his or her actions for histories in $G_{i,j}$ and is forced to play according to σ_i elsewhere. Let the **graft** of $G_{i,j}$, $\sigma^{*,j}$, be an ϵ-Nash equilibrium of the game $\Gamma^{\sigma_i,j}$. Finally, define the **grafted strategy for player i**, σ_i^*, as*

$$\sigma_i^*(h,a) = \begin{cases} \sigma_i(h,a) & \text{if } h \in G_{i,0} \\ \sigma_i^{*,j}(h,a) & \text{if } h \in G_{i,j}. \end{cases}$$

*We will call σ_i the **base strategy** and G_i the grafting partition for the grafted strategy σ_i^*.*

Figure 2.10 shows an example of an extensive-form game $\Gamma^{\sigma_2,j}$ for some j derived from the game in Figure 2.2. For this sub-game, we have $G_{2,j} = \{al, bl, cl\}$ as these are the histories where player 2 still has control over which action to take. This may be the only sub-game for which a graft is computed, *i.e.* $j = 1$ and $G_2 = \{G_0 = H_2 \backslash G_{2,1}, G_{2,1}\}$, or there could be more sub-games contained in the grafting partition. A grafted strategy for player 2 would then follow a graft for this game whenever player 2 has observed one of the histories in $G_{2,j}$.

Under a fixed memory limitation, we can employ finer abstractions for the sub-games $\Gamma^{\sigma_i,j}$ than we can in the full game Γ. This is because $\Gamma^{\sigma_i,j}$ removes some of player i's information sets from the game, freeing up valuable memory when running algorithms such as CFR. Note that strategy grafting is only applied to one specific player at a time. An entire strategy grafted profile can be constructed by repeating the process for each individual player.

Chapter 3

Poker

In this chapter, we provide the rules for Kuhn Poker and Leduc hold'em, two very simple poker games that we use throughout this dissertation. We then explain the rules of Texas hold'em poker and a number of shortened variants that require less computation to reach a desired solution quality. Finally, we end this chapter with a discussion about previous work in poker related to our research contributions in Chapters 4 through 8.

3.1 Introduction

We begin this chapter by describing the order of play for a single hand of poker, which can be modelled as an extensive-form game as defined in Section 2.1. To begin a hand, one player is denoted as the **dealer** and each player, including the dealer, pays an **ante** equal to a fixed number of chips from their **stack** to the **pot**. Alternatively, a game may use **blinds** instead of or in addition to antes, where two players must place a **small blind** or a **big blind** equal to a set number of chips in the pot. The big blind is typically twice the size of the small blind and is often used as a unit of measurement for winnings. With more than two players, the player immediately to the left of the dealer posts the small blind and the next player to the left posts the big blind. In a two-player game with blinds, the dealer posts the small blind and the other player posts the big blind.

Once antes or blinds have been posted, private cards or **hole cards** are dealt out to the players. The hole cards give rise to non-singleton information sets in the extensive-form game representation, since hole cards are not seen by other players. Next, the first betting round, or **pre-flop**, begins. In each betting round, play begins with the player to the left of the dealer, with the exception of the pre-flop round when blinds are used. In this case, the player to the left of the big blind starts the pre-flop round. On a player's turn, if a player is not faced with a prior bet (or blind), then that player may:

- **check** - pass without committing any chips to the pot, or
- **bet** - place an amount of chips into the pot.

When faced with a bet or a blind, a player typically has three options:

- **fold** - forfeit the pot, eliminating the player from the remainder of the hand,
- **call** - match the previous bets and raises, placing an amount of chips equal to the sum of the previous unmatched bets or raises into the pot, or
- **raise** - increase the previous bet, achieved by first calling the previous bets and then placing an amount of chips greater than the previous bet or raise into the pot.

All actions are public information. In addition, a **limit** game is where the total number of raises and the sizes of each bet or raise is fixed each round. In contrast, a **no-limit** game is where a bet or raise can be any number of chips from a player's stack greater than or equal to the size of the big blind. In no-limit poker, a raise must also be greater than the previous bet or raise.

Each poker game is made up of a set number of **betting rounds**. In each betting round, each player still in the hand is given a turn to act. After a bet or raise, all players remaining must then respond to that action. A betting round ends once all players have checked or once all other players have responded to the last bet or raise. At the beginning of each betting round after the pre-flop, public **community cards** are revealed before players again take actions in turn. At the end of the last betting round, all players that did not fold enter the **showdown** and reveal their hole cards. The player with the highest ranked poker hand made up of their hole cards and community cards wins the pot. If all players but one fold before the end of the final betting round, then the single remaining player takes the pot and no hole cards are revealed.

3.2 Kuhn Poker

Kuhn Poker [50] is a very basic limit poker game. Though the original rules are only for two players, Abou Risk and Szafron [3] defined a three-player version and we now extend the rules to n players. Kuhn Poker is played with a deck consisting of $n+1$ cards labelled 1, 2, ..., $n+1$, where 1 has the lowest rank and $n+1$ has the highest. Antes of one chip are used and each player is dealt one hole card. Kuhn Poker has just the pre-flop betting round where the bet size is one chip. No raises are allowed, so following a bet, the other players may only fold or call the bet. There are no community cards and the showdown is won by the player holding the highest ranked card among those who did not fold.

Figure 3.1 shows the three-player Kuhn Poker extensive-form game tree for a fixed dealing of the cards. Each player has four information sets per private card, and since the deck contains four cards, this gives $|\mathcal{I}_i| = 16$ for each $i \in \{1, 2, 3\}$. Because the opponents' hole card cannot be seen, each information set contains all $3 \cdot 2 = 6$ possible card dealings consistent with the player's hole card and betting sequence. Hence, $|I| = 6$ for all information sets I. Since raises are never allowed, we also have $|A(I)| = 2$ for all information sets I.

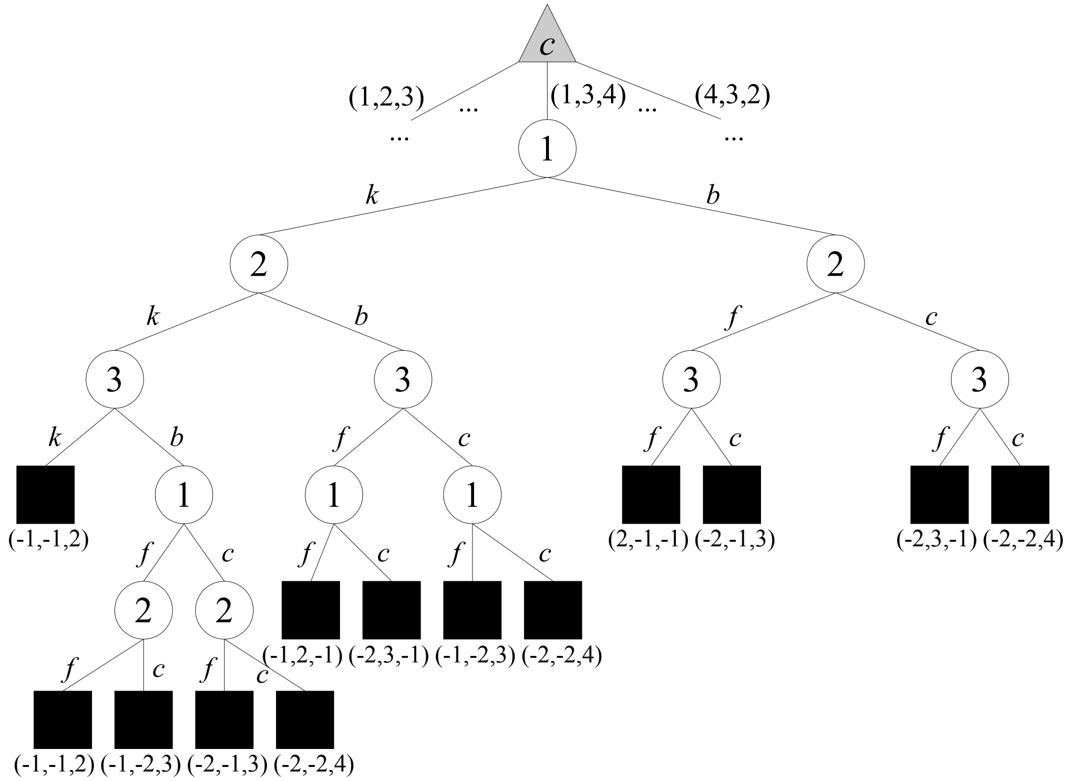


Figure 3.1: The three-player Kuhn Poker game tree for when player 1 is dealt the 1, player 2 is dealt the 3, and player 3 is dealt the 4. Actions check, bet, fold, and call are denoted by k, b, f, and c respectively. The utilities for each player are denoted under the corresponding terminal node. Each decision state belongs to an information set of size six (not shown).

The two-player version of Kuhn Poker is very small and all Nash equilibria can be computed by hand [50]. For the three-player game, Abou Risk and Szafron [3] showed that 100 million iterations of Chance Sampling MCCFR produces an ϵ-Nash equilibrium with $\epsilon \approx 1.69$ milli-antes per hand. This result is surprising since there is no theoretical evidence to suggest that CFR will produce an approximate equilibrium in a three-player game. Recently, the author of this dissertation contributed to work that derived a parameterized family of Nash equilibrium profiles for three-player Kuhn Poker [66]. It is still an open question as to whether more equilibria exist beyond this discovered family.

3.3 Leduc Hold'em

The next poker game that we consider is Leduc hold'em, or simply Leduc, a limit poker game originally described by Southey *et al.* [65]. Again, the original rules for Leduc are only for two players, but we can easily allow more players to participate in the game. In this dissertation, we consider both two-player and three-player Leduc. Leduc is played with a six card deck consisting of two Jacks, two Queens, and two Kings. As in Kuhn Poker, antes of one chip are used and each player is dealt one hole card. However, Leduc has two betting rounds, the pre-flop and the **flop**.

Bets and raises in the pre-flop are fixed at two chips and in the flop are fixed at four chips. In each round, at most one raise is allowed. At the start of the flop, one community card is revealed. In a showdown, a player whose hole card matches the community card (a pair) wins the pot. Otherwise, the pot goes to the player with the highest ranked hole card, where the pot is split in the case of a tie. Note that Abou Risk and Szafron defined a three-player version of Leduc that used a deck of eight cards rather than six cards. In this dissertation, we use a six card deck for both the two-player and three-player games.

While the extensive-form game representation of Leduc is complex enough that we do not show it here, two-player Leduc is still a small enough zero-sum game to rapidly compute an approximate Nash equilibrium with CFR without using abstraction. Alternatively, an exact equilibrium of two-player Leduc can be computed by linear programming using the sequence form representation described in Section 2.3 [72]. For three or more players, however, no equilibrium profiles are known. In contrast to Kuhn Poker, CFR applied to three-player Leduc has not produced an ϵ-Nash equilibrium for small ϵ, and the linear programming approach is not applicable to three-player games. These problems do not deter us, however, as our research contributions listed in Chapter 1 do not focus on Nash equilibrium solutions for non-zero-sum games.

3.4 Texas Hold'em

Texas hold'em, or simply hold'em, is arguably the most popular poker game played around the world today. The game is played in many casinos and in many on-line poker rooms, with millions of dollars at stake in the top level tournaments. Hold'em is not only an interesting domain from a scientific perspective, but also from a financial perspective for players, event sponsors, promoters, and broadcasters.

3.4.1 Rules

Hold'em is typically played by two to ten players with a full fifty-two card deck consisting of thirteen ranks (2, 3, ..., 10, Jack, Queen, King, and Ace) and four suits (diamonds, clubs, hearts, and spades). Blinds are used instead of antes and players are dealt two hole cards each. There are four betting rounds, the pre-flop, flop, **turn**, and **river**. For limit hold'em, a maximum of three raises total are allowed per round, where bets and raises during the pre-flop and flop rounds are equal to the size of the big blind. In the turn and river rounds, bets and raises are doubled to twice the size of the big blind. Three community cards, called the **flop cards**, are revealed at the start of the flop, with one additional community card revealed on the turn (**turn card**) and one final card revealed on the river (**river card**). The player that has the best five-card poker hand using any combination of the two private cards and the five community cards wins the pot in a showdown. Poker hand rankings can be found on-line [57].

We assume throughout this dissertation that in limit hold'em, the players' chip stacks are infinite.

This way, a player always has enough chips to make a bet, call, or raise when desired. In contrast, in no-limit games, a player can go **all-in**, meaning that the player's entire stack is committed to the pot and the player has no chips left for another call, bet, or raise. In this case, the all-in player no longer participates in the betting rounds, but remains in play during a showdown. In no-limit games, we will always specify the size of the players' stacks and reset to the initial stacks after each game. When evaluating in-game performance, we simply average the amount of chips won or lost across many games rather than comparing final stack sizes.

3.4.2 Abstraction

Hold'em is a massive game in terms of the number of information sets. For example, two-player limit hold'em has approximately $|H| \approx 10^{18}$ histories [5, Figure 1] contained in $|\mathcal{I}_i| \approx 3 \times 10^{14}$ information sets per player [41], and three-player has $|\mathcal{I}_i| \approx 5 \times 10^{17}$. Applying CFR to these enormous state spaces necessitates abstraction.

Abstractions in poker typically group many different card dealings into **buckets** so that any deals that fall into the same bucket become indistinguishable. Two common bucketing techniques are **percentile hand strength** [7] and **percentile hand strength squared** [40]. First, for each betting round, these techniques order all possible hands according to expected hand strength (**E**[HS]) and expected hand strength squared (**E**[HS^2]) respectively. **E**[HS] is the probability of the given hand winning against a single random opponent hand in a showdown, averaged over all possible future community cards and all possible opponent hands. **E**[HS^2] squares each term before averaging, giving a bonus to possible straights, flushes, and other hands with high potential. Then, percentile bucketing with k buckets and m hands groups the top m/k hands into one bucket, the next top m/k hands into a second bucket, and so on so that buckets are approximately equal in size.

Recently, Johanson *et al.* [45] proposed a new bucketing technique that groups hands not based on a scalar metric such as **E**[HS] or **E**[HS^2], but instead based on **hand strength distributions**. This is done by discretizing hand strength into a number of small ranges and forming histograms, one for each set of private cards and public board cards for the given round. The histograms represent the number of ways the remaining board cards can be dealt that result in a hand strength value in a given range [45, Figure 2]. Hands are then merged according to how similar the resulting histograms are using k-means clustering over earth mover's distance. While this approach is not applicable on the river round, Johanson *et al.* suggest a second new technique called **opponent cluster hand strength (OCHS)** that does apply to the river. Instead of averaging over all possible opponent hands as is done by **E**[HS], OCHS splits the possible opponent hands into eight clusters using the pre-flop hand strength distribution. For each cluster, hand strength is then computed averaging only over the possible opponent hands within that cluster. This results in a vector of eight hand strength values per hand, which are then grouped using k-means clustering over L_2 distance.

Once a bucketing technique has been chosen in each round for player i, it is straightforward to construct an abstraction α_i for the player. We merge information sets in $\mathcal{I}_i$ that contain histories with card deals falling into the same bucket. In limit hold'em, we do not consider abstractions that hide or remove public player actions, and so $\alpha_i^A(h) = A(h)$ for all $h \in H$. For no-limit hold'em, we will use common action abstractions used by others [32, 64] that limit players to a small number of possible bet sizes relative to the pot size. In addition, we apply the same bucketing techniques to all players unless otherwise noted.

For perfect recall abstractions, we remember which bucket the hand belonged to on all previous rounds before more card(s) were revealed. For example, consider a perfect recall abstraction that separates new card information into five buckets per round. This means that we have 5 pre-flop buckets, $5 * 5 = 5^2$ flop buckets, 5^3 turn buckets, and 5^4 river buckets. As a convention, we write *5/5/5/5*, or simply *5s*, to denote the size of the abstraction. In contrast, an imperfect recall abstraction forgets which bucket the hand belonged to on one or more rounds. For instance, writing *IR5/5/5/5* means that the pre-flop bucket is forgotten in later rounds, leaving us with 5 preflop buckets, 5 flop buckets, 5^2 turn buckets, and 5^3 river buckets.

We apply bucketing to hold'em to produce an abstract game Γ^α that is much more manageable in size. For example, a 5s abstraction in two-player limit hold'em contains 3.6×10^6 information sets between both of the players. If we apply imperfect recall on every round, we can produce even smaller abstract games. For instance, IR5/IR5/IR5/5 contains a total of 3.2×10^4 information sets in two-player limit hold'em. In addition, an IR169/IR569/IR569/569 abstraction is roughly the same size as a 5s abstraction. Imperfect recall abstractions sacrifice previous card information in order to differentiate between more hands on the current round.

3.4.3 2-1 and 2-NL Hold'em

Throughout this thesis, we consider several abstractions of different Texas hold'em games. In addition, we consider two variants of Texas hold'em that reduce the amount of abstraction required to feasibly run algorithms such as CFR. The first variant, **2-1 hold'em** [43], is identical to limit Texas hold'em, except consists of only the first two betting rounds, the pre-flop and flop, and only one bet or raise is allowed per round. Two-player 2-1 hold'em has approximately 1.6×10^7 information sets and can easily be solved with CFR without any abstraction.

The second variant, **2-NL hold'em**, is a new game derived from no-limit Texas hold'em. As in 2-1 hold'em, 2-NL hold'em consists of just the pre-flop and flop betting rounds; otherwise, the game is identical to no-limit Texas hold'em. This means that any number of bets and raises of any size are allowed by the players up to their remaining stack sizes. While we will still employ card abstractions and keep the players' starting stacks small, we will not use any action abstraction with 2-NL hold'em. In Chapter 5, we will use 2-NL hold'em to study the effects of our algorithms on games with a large range of player actions.

3.5 Related Work in Poker

3.5.1 Domination

Since Nash equilibria do not contain iteratively strictly dominated strategies, and because much work in poker and other games has revolved around computing equilibria, very little effort has gone into examining domination in poker. The closest work in this area is by Waugh [70]. He defines the domination value of a strategy σ_i in a zero-sum game to be the amount of utility player i loses relative to the game value when the opponent plays a strategy from a Nash equilibrium profile that best exploits σ_i. More precisely, the **domination value** of a player 1 strategy σ_1 in a zero-sum game is

$$\mathrm{Dom}_1(\sigma_1) = \max_{\sigma_2 \in \Sigma_2^*} u_2(\sigma_1, \sigma_2) + u_1(\sigma^*)$$

(and similarly for a player 2 strategy σ_2), where Σ^* is the set of all Nash equilibrium profiles and σ^* is any Nash equilibrium. Waugh examines the correlation between domination value and actual agent performance by computing exact Nash equilibrium profiles in a number of different abstract games of two-player Leduc. Tournaments between all of the strategy profiles are run and the results are compared to both the exploitability and the domination value of each of the strategies within the real game. Domination value is found to be strongly correlated with actual tournament performance, whereas exploitability is only weakly correlated.

While strictly dominated strategies have positive domination value, non-dominated strategies may also have positive domination value. For example, in the zero-sum normal-form game in Figure 2.1, one can check that no pure strategy for either player is even weakly dominated and that the only Nash equilibrium is the pure strategy profile (B, b) with $u_1(B, b) = 0$. However, the pure strategy for the row player that always plays A, for example, has a domination value of 1 since the column player earns 1 utility when playing b against A. Thus, domination value may be somewhat of a misnomer as not all non-dominated strategies have zero domination value as, it seems, was originally intended.

3.5.2 Monte Carlo CFR

In Section 2.2.3, we defined CS, ES, and OS, three MCCFR algorithms that differ in choice of blocks $\mathcal{Q}$. Recently, Johanson *et al.* [43] proposed a new MCCFR algorithm called **Public Chance Sampling (PCS)**. PCS is similar to CS, except that two histories are in the same block if and only if no two public chance actions differ. This means that in poker, one iteration of PCS will consider all possible private cards for the players, whereas CS considers just one sampled private hand per player. Johanson *et al.* show that in two-player 2-1 hold'em and in large enough card abstractions of two-player limit Texas hold'em, strategies generated with PCS are less exploitable than those generated by CS after a fixed amount of time. A major reason for this success comes from computing $O(k^2)$ terminal node evaluations in $O(k)$ time, where k is the number of possible private hands. This is achieved by exploiting a total ordering on the ranks of the private hands. PCS is currently the most efficient algorithm known for two-player limit hold'em. In Chapter 5, we will show that PCS may not be the best choice for no-limit and three-player poker games.

3.5.3 Imperfect Recall

To our knowledge, prior to our work later in Chapter 6, no theoretical results are known for regret minimization in imperfect recall games. On the other hand, Waugh *et al.* [73] explore the empirical advantages of using imperfect recall abstractions over perfect recall counterparts. For example, they show that a CFR solution to a 14s perfect recall abstraction of two-player limit hold'em is outperformed by a CFR solution to an imperfect recall abstraction containing roughly the same number of information sets as the 14s abstraction. More recently, Johanson *et al.* [45] similarly compared a perfect recall 10s abstraction to imperfect recall abstractions of the same size in two-player limit hold'em. They found that regret minimization in an imperfect recall abstraction led to a strategy that was significantly less exploitable than the least exploitable strategy representable in the perfect recall abstraction. Unfortunately, our theoretical analysis in Chapter 6 cannot directly explain these results. Nonetheless, Chapter 6 does present the first known class of imperfect recall games for which CFR is guaranteed to minimize regret.

3.5.4 Strategy Stitching

The earliest example of applying tree decomposition in poker is the PsOpti family of programs from 2003 [5]. The programs play two-player limit hold'em and were built using the process depicted in Figure 2.9. First, a base strategy called the *pre-flop model* was computed in a very coarse abstraction and was only used to play the pre-flop round during on-line play. Secondly, seven experts or *post-flop models* were created, one for each of the possible pre-flop betting sequences leading to the flop. The experts are not grafts; instead, each expert focused solely on its individual subtree and ignored the other parts of the game. In addition, the experts were provided with seeded probabilities of pre-flop play for both players according to the pre-flop model, much like the base strategy determines probabilities in grafting. While the PsOpti bots were strong for their time, they are no match for today's top programs. Due to resource and technology limitations back then, the abstractions used to build pre-flop and post-flop models were very limited. Nonetheless, we generalize this approach in Chapter 7 and provide further evidence that the PsOpti approach is indeed credible.

Using a similar approach, Abou Risk and Szafron [3] apply *heads-up experts* to three-player limit hold'em. They reason that because the subtrees immediately following a fold action in hold'em are dramatically smaller relative to the entire game tree, much finer abstractions can be used on these subtrees when the rest of the game is omitted. Thus, they create experts for six such subtrees in a 5s abstraction to go along with a base strategy contained in a much more coarse abstraction. The experts were similar to the PsOpti experts and ignored the parts of the game outside of the given subtree. Their strongest experts were seeded by a sensible strategy based on advice from poker professionals as opposed to using the base strategy. However, their results were mixed. In one case, the agent using experts performed better than just the base strategy alone, but in another case the experts agent was worse. In hindsight, the negative result was due to the 5s abstraction for the experts being no more effective than the imperfect recall abstraction they used for the base strategy. In Chapter 7, we show that heads-up experts can in fact significantly increase performance relative to a single base strategy.

Waugh *et al.* [71] apply strategy grafting to create grafted strategies for both two-player Leduc and two-player limit hold'em. Each Leduc grafted strategy consists of a base strategy in a simple abstraction and three grafts. Each grafting partition for player i splits the game tree according to either player i's private card or the flop card, and the resulting subtrees simply use the null abstraction. For hold'em, they construct a single grafted strategy consisting of a base strategy and twenty grafts. The base strategy plays the pre-flop while the grafts play from the flop onwards. Each possible combination of flop cards are classified as one of twenty types, and this classification is used to define the grafting partition. They found that the grafted strategies provided a significant performance boost relative to the base strategies alone and were competitive with other agents. However, they did not experiment with grafting partitions based on the players' actions.

Furthermore, Gilpin and Sandholm [30] create a poker strategy for two-player limit hold'em that is quite different from those discussed thus far. As with PsOpti, their strategy construction is performed in two phases. The first phase computes a strategy in an abstraction of the game with action probabilities fixed in the river round. This first phase strategy is only used to play the pre-flop and flop rounds. A second phase strategy that plays the turn and river rounds is computed on-line during an actual poker game. One drawback of this approach is that the on-line computations must be quick enough to play in real time. Despite fixing the flop cards, this constraint forced the authors to still employ a very coarse abstraction during the second phase.

3.5.5 Other Related Work

We end this chapter by briefly highlighting some other research related to our objectives. Firstly, Gilpin *et al.* [31] use an automated abstraction building tool to dynamically bucket hands for a two-player limit hold'em agent. While we are concerned with employing abstractions with high granularity, our goals do not focus on the actual abstraction building process itself. In general, our research directions are actually orthogonal to abstraction improvements and could be used in conjunction with more sophisticated abstraction techniques.

Secondly, Ganzfried and Sandholm [19, 20] developed algorithms for computing ϵ-Nash equilibria in non-zero-sum games and applied it to a small three-player no-limit poker game. The rules of the game allowed each player only one of two actions: either *jam* by going all-in, or fold the hand. Unfortunately, it is unclear how to extend these approaches to much larger games. Finally, the authors later developed a new algorithm for finding equilibria by constructing an infinite approximation of the original game and translating a solution from the infinite game back to the finite game [21]. While they suggest that their approach could provide strong strategies in large non-zero-sum games, their only experiments are with a small game similar to three-player Kuhn Poker. As mentioned previously, although Nash equilibria do not contain strictly dominated strategies, this dissertation is not concerned with finding equilibria in non-zero-sum games.

Chapter 4

Regret Minimization and Domination

CFR, described in Section 2.2.2, is a state-of-the-art approach for approximating Nash equilibria of large zero-sum extensive-form games. Recall that during computation, CFR stores both a current profile and an average profile. The average profile approaches equilibrium and is generally used in practice, while the current profile is discarded. CFR can still be applied in non-zero-sum games; however, the resulting average profile is not guaranteed to be in equilibrium [3, Table 2]. Despite this, CFR has been used to generate more aggressive, or *tilted*, poker strategies from two-player non-zero-sum games capable of defeating top poker professionals [44], as well as winning three-player Texas hold'em poker strategies [3] in the Annual Computer Poker Competition (ACPC) [4]. Previous work makes no attempt to explain why the average profile might perform well outside of two-player zero-sum games.

In this chapter, we provide the first theoretical groundings for regret minimization algorithms applied to non-zero-sum games. This is achieved by establishing elimination of iteratively dominated errors. As described in Section 2.1.2, these are mistakes where there exists an alternative that is guaranteed to do better, assuming the opponents do not make such errors themselves. Strategies avoiding such errors belong to a superset of Nash equilibria, generalizing previous zero-sum results to games with many players. Firstly, we prove that in normal-form games, common regret minimization techniques eliminate (play with probability zero) iteratively strictly dominated strategies. Secondly, we formally define a *dominated action* and prove that under certain conditions, both the current and average CFR profiles eliminate iteratively strictly dominated actions. Thirdly, for two-player non-zero-sum games, we bound the average profile's exploitability and measure this value empirically with a number of tilted poker strategies. Our theoretical results lead us to a simple modification of CFR for games with more than two players that just uses the current profile and does not average. We demonstrate that with this change, CFR generates strategies that perform just as well as those generated without the change, but now require less time and less memory to compute. Furthermore, for large games requiring abstraction, this reduction in memory allows finer-grained abstractions to

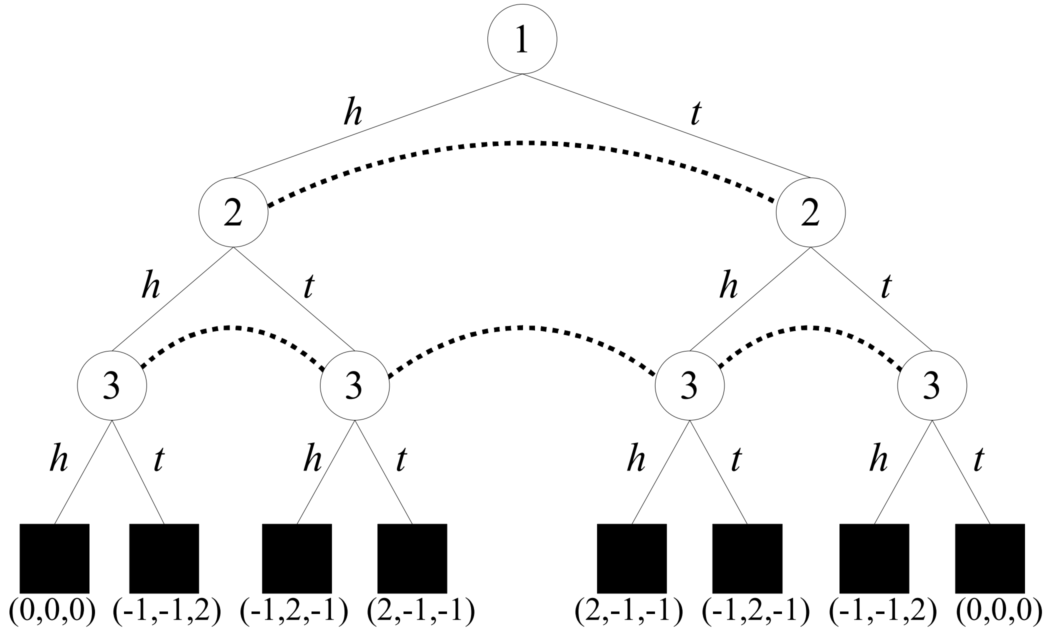


Figure 4.1: The mismatching pennies game. All actions are private until the game ends at a terminal history. The utilities for reaching each terminal history are written from left to right for players 1, 2, and 3 respectively below each terminal node.

be used by CFR, leading to even stronger strategies than previously possible. Finally, we demonstrate that with our modification of CFR, a similar strategy to our 2012 ACPC three-player hold'em entry can be computed in 25% of the time using only half the RAM. Throughout this chapter, we assume perfect recall in extensive-form games.

4.1 Equilibria in Non-Zero-Sum Games

As discussed in Section 2.1.2, a Nash equilibrium is a powerful solution concept in zero-sum games. Outside of zero-sum games, however, Nash equilibria are less useful. Consider the 3-player game of *mismatching pennies* shown in Figure 4.1. In this game, each player has exactly one information set ($|\mathcal{I}_i| = 1$ for $i = 1, 2, 3$) and privately chooses either *heads* (h) or *tails* (t). If all players choose the same action, they each receive zero utility. Otherwise, the player that chooses the unique action receives 2 utility while the others receive -1. The strategy profile σ^{htt}, where player 1 always picks h and players 2 and 3 always pick t, is a Nash equilibrium since none of the three players can increase their utility by unilaterally changing their strategy. Here, $u_1(\sigma^{htt}) = 2$, but player 1 is not guaranteed to earn this much by playing σ_1^{htt}. If instead of σ_2^{htt}, player 2 plays $\hat{\sigma}_2$ that always plays action h, then player 2 still earns $u_2(\sigma^{htt}) = -1 = u_2(\hat{\sigma}_2, \sigma_{-2}^{htt})$. However, player 1's utility is decreased to $u_1(\hat{\sigma}_2, \sigma_{-2}^{htt}) = -1$. This example shows that player 2 can arbitrarily push the utilities towards player 1 or player 3 while player 2's own utility stays the same. Since player 1 cannot control the strategies of the other players, it is not clear whether σ_1^{htt} is even a good strategy to play.

Another Nash equilibrium is the strategy profile σ^{RAND} where all players pick h or t with equal probability. For this profile, $u_i(\sigma^{\text{RAND}}) = 0$ for all $i \in N$. While players 2 and 3 have a higher expected utility under σ^{RAND} than σ^{htt}, player 1's utility is lower. Even if we decided that we do want to play part of a Nash profile, following an arbitrary Nash equilibrium may not yield as much utility as other Nash profiles. More importantly, no payoff guarantees even hold for equilibrium strategies when more than one player deviates from the equilibrium profile.

As mentioned above, the average strategy profile computed by CFR is not guaranteed to converge to an equilibrium in non-zero-sum games. However, if we assign probability $1/T$ to each of the profiles $\{\sigma^1, ..., \sigma^T\}$ generated by CFR or any other regret minimizer, then by equation (2.1) and minimization of regret, this distribution over profiles converges to a coarse correlated equilibrium as defined in Section 2.1.2. Though previous work omits this fact, it is unclear how this could be useful, let alone why the average strategy from CFR might be valuable. As we demonstrated earlier with Figure 2.5, a coarse correlated equilibrium might still recommend a strictly dominated strategy, and so this property alone is not enough to rule out domination.

4.2 Dominated Actions

Our contributions in this chapter begin with a formal definition of *dominated actions* that are specific to extensive-form games, and we relate such actions to dominated strategies. Our definition of a dominated action in an extensive-form game is not to be confused with the definition of a dominated pure strategy in a normal-form game, as the latter is covered by Definition 2.5. We say an action a at $I \in \mathcal{I}_i$ is a strictly dominated action if there exists a strategy σ_i' that guarantees higher counterfactual value at I to any other strategy σ_i that always plays a at I, regardless of what the opponents play but assuming they reach I with positive probability. The formal definition is below.

Definition 4.1. *An action $a \in A(I)$ of an extensive-form game is a* ***strictly dominated action*** *if there exists a strategy $\sigma_i' \in \Sigma_i$ such that for all profiles $\sigma \in \Sigma$ satisfying $\sum_{h \in I} \pi_{-i}^{\sigma}(h) > 0$, we have $v_i(I, \sigma_{(I \to a)}) < v_i(I, (\sigma_i', \sigma_{-i}))$.*

We use the counterfactual value v_i instead of u_i in Definition 4.1 because we are only concerned with the utility to player i from I onwards rather than over the entire game. Similar to iteratively dominated strategies, we also define an **iteratively strictly dominated action** as one that is either strictly dominated or becomes strictly dominated after successively removing strictly dominated actions from the players' action sets. Analogous to strategic dominance in Definition 2.5, **weak** and **very weak** action dominance allow equality rather than strict inequality for all but one profile σ and for all profiles respectively. In addition, weak and very weak action dominance do not require the condition that $\sum_{h \in I} \pi_{-i}^{\sigma}(h) > 0$.

While we are unaware of any other publications that consider dominated actions in extensive-form games, the Gambit Software Tools package [56] contains algorithms for removing similar notions of action dominance. However, Gambit considers an action a at I to be strictly dominated if there exists another action b at I such that for every history $h \in I$ and every pure strategy profile s, taking b leads to greater utility than taking a; *i.e.*, $\sum_{hb \sqsubseteq z} \pi^s(hb, z)u_i(z) > \sum_{ha \sqsubseteq z} \pi^s(ha, z)u_i(z)$ for all $h \in I$ and $s \in \mathcal{S}$[1]. Gambit's definition is more restrictive than Definition 4.1 for two reasons: Gambit requires domination by a pure strategy s_i rather than by a behavioral strategy σ_i', and Gambit requires that all strategies s_i lead to greater utility rather than just a single strategy σ_i'. For these reasons, we only consider Definition 4.1 as this potentially leads to more actions being strictly dominated.

Consider again Kuhn Poker as defined in Section 3.2. When either player is faced with a bet from the opponent, calling the bet when holding the Jack is a strictly dominated action. This is because the Jack is the worst card and thus never wins regardless of the opponent's private card. Similarly, folding with the King is a strictly dominated action. Note that a strategy that plays either of these actions with positive probability is not necessarily a strictly dominated strategy (but is a weakly dominated strategy, as Hoehn *et al.* [39] conclude) because there exist opponent strategies that never bet. In addition, once these two actions are removed, one can check that player 1's action of betting with the Queen is iteratively strictly dominated. Since player 2 now only folds with the Jack and only calls with the King, it is strictly better for player 1 to always check with the Queen and then call a player 2 bet with probability $2/3$. Thus, iteratively strictly dominated actions can identify errors that iteratively strictly dominated strategies cannot. Note that betting with the Queen for player 1 could lead to positive utility if player 2 holds the Jack, and checking with the Queen could lead to negative utility if player 2 holds the King. As such, Gambit does not label betting with the Queen for player 1 as even iteratively weakly dominated.

Proposition 4.1 below states a fundamental relationship between dominated actions and strategies. Any strategy that plays to reach information set I ($\pi_i^\sigma(I) > 0$) and plays a weakly dominated action a at I ($\sigma_i(I, a) > 0$) is a weakly dominated strategy. Since strictly dominated actions are also weakly dominated, it follows from Proposition 4.1 that any strategy that plays a strictly dominated action is a weakly dominated strategy. Note that perfect recall is required in Proposition 4.1 for $\pi_i^\sigma(I)$ to be well-defined as described in Section 2.1.1. We provide a proof sketch of the proposition below, while full proofs for this chapter can be found in Appendix B.

Proposition 4.1. *In an extensive-form game with perfect recall, if a is a weakly dominated action at $I \in \mathcal{I}_i$ and $\sigma_i \in \Sigma_i$ satisfies $\pi_i^\sigma(I)\sigma_i(I, a) > 0$, then σ_i is a weakly dominated strategy.*

Proof sketch. By definition of action dominance, there exists a strategy $\sigma_i' \in \Sigma_i$ such that $v_i(I, \sigma_{(I \to a)}) \leq v_i(I, (\sigma_i', \sigma_{-i})$ for all opponent profiles $\sigma_{-i} \in \Sigma_{-i}$. One can then construct a strategy σ_i'' that follows σ_i everywhere except within the subtree rooted at I, where instead we follow a mixture of σ_i and σ_i'. The weight in this mixture assigned to σ_i' is $(1 - \sigma_i(I, a)) > 0$. The

[1] Gambit also has an option for computing action dominance by taking utilities from the root rather than from information set I, but this is generally less useful.

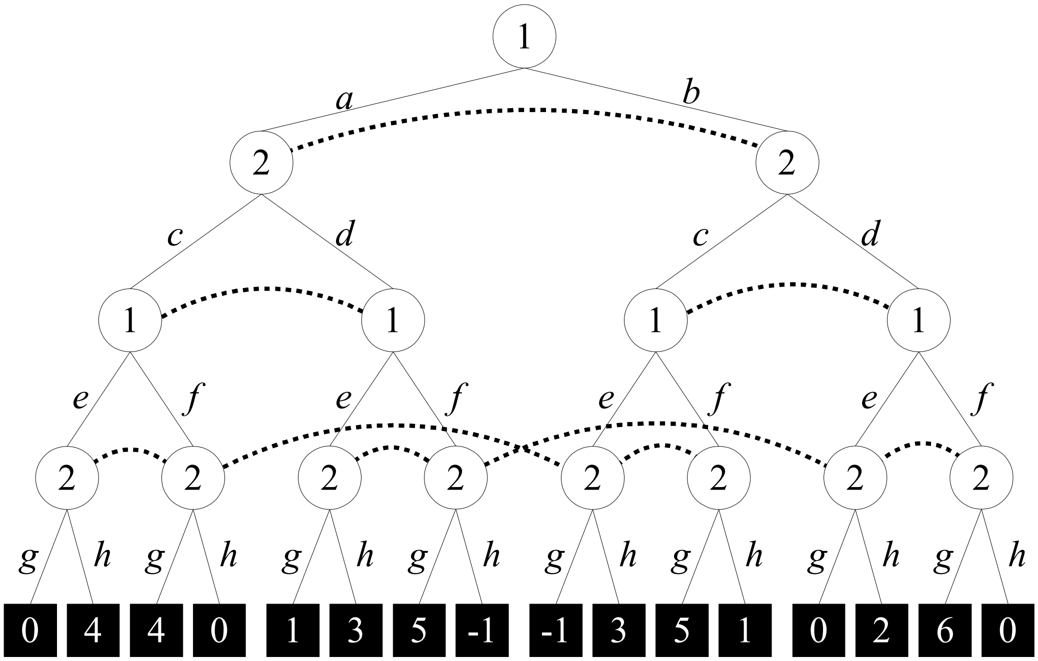


Figure 4.2: A zero-sum extensive-form game with strictly dominated strategies, but no strictly or weakly dominated actions. Nodes connected by a dashed line are in the same information set. Terminal values indicate utilities for player 1.

strategy σ_i is then weakly dominated by σ_i''. ■

It is possible, however, for a dominated strategy to not play any dominated actions. For example, consider the zero-sum extensive-form game in Figure 4.2 where both players take two private actions. The pure strategy for player 1 of playing b and then e is strictly dominated by the pure strategy that plays a and then e because the latter strategy guarantees exactly 1 more utility than the former, regardless of how player 2 plays. Similarly, the pure strategy that plays a and then f is strictly dominated by the pure strategy that plays b and then f. However, no action is even weakly dominated. For instance, after playing a (or b), the utility player 1 receives for playing e can be greater, equal to, or less than the utility for playing f depending on how player 2 plays.

4.3 Theoretical Results

Clearly, one should never play a strictly dominated action or strategy as there always exists a better alternative. Furthermore, if we make the common assumption that our opponents are rational and do not play strictly dominated actions or strategies themselves, then we should never play iteratively strictly dominated actions or strategies. In zero-sum games, CFR's average strategy profile converges to a Nash equilibrium, and so the average profile is guaranteed to eliminate strictly dominated strategies. For non-zero-sum games, however, Abou Risk and Szafron [3] demonstrated that CFR may not converge to a Nash equilibrium. In this section, we provide proof that under certain conditions, CFR does eliminate (*i.e.*, play with probability zero) strictly dominated actions and strategies.

We begin by showing that in normal-form games, all regret minimization algorithms that assign zero probability to actions with negative regret, which includes regret matching, remove iteratively strictly dominated strategies. This is a simple result that, to our knowledge, was previously unknown. Recall that the support of a strategy σ_i, supp(σ_i), is the set of actions assigned positive probability by σ_i.

Theorem 4.1. *Let $\sigma^1, \sigma^2, ...$ be a sequence of strategy profiles in a normal-form game where all players' strategies are computed by regret minimization algorithms where for all $i \in N$, $a \in A_i$, $T \geq 0$, if $R_i^T(a) < 0$ and $R_i^T(a) < \max_{b \in A_i} R_i^T(b)$, then $\sigma_i^{T+1}(a) = 0$. If σ_i is an iteratively strictly dominated strategy, then there exists an integer T_0 such that for all $T \geq T_0$,* supp$(\sigma_i) \nsubseteq$ supp(σ_i^T).

Proof sketch. For the non-iterative dominance case, by strict domination of σ_i, there exists another strategy $\sigma_i' \in \Sigma_i$ such that

$$\epsilon = \min_{a_{-i} \in A_{-i}} u_i(\sigma_i', a_{-i}) - u_i(\sigma_i, a_{-i}) > 0.$$

One can then show that there exists an action $a \in$ supp(σ_i) such that

$$R_i^T(a) \leq -\epsilon T + \max_{b \in A_i} R_i^T(b) \leq -\epsilon T + R_i^{T,+}.$$

Since $R_i^{T,+}/T \to 0$ as $T \to \infty$, it follows that $R_i^T(a) < 0$ after some finite number of iterations T_0. By our assumption, this implies $a \notin$ supp(σ_i^T) for all $T \geq T_0$ as desired. Using the fact that new iterative dominances only arise from removing actions and never from removing mixed strategies [15], iterative dominance is proven by induction on the finite number of iteratively dominated pure strategies (actions) that must first be removed to exhibit domination of σ_i. ■

Note that regret matching is a regret minimization algorithm that satisfies the conditions required by Theorem 4.1, as long as when the denominator of equation (2.3) is zero, we choose $\sigma_i^{T+1}(a) = 0$ when $R_i^T(a) < \max_{b \in A_i} R_i^T(b)$. Also, if a pure strategy $s_i(a) = 1$ is iteratively strictly dominated, then Theorem 4.1 implies that σ_i^T never plays action a after a finite number of iterations. However, we cannot guarantee that all iteratively strictly dominated strategies will be eliminated after a finite number of iterations. This is because a strategy could be strictly dominated by only an infinitesimal amount and could require a near infinite number of iterations to remove. Regardless, if one wanted to simply find the strategies in a normal-form game that avoid iterative strict domination, one can repeatedly solve the linear program (2.7) to do so in polynomial time.

Contrary to Theorem 4.1, it is not true that regret matching will always eliminate weakly dominated strategies. For example, consider the game in Figure 4.3 where the row player's (denoted player 1) pure strategies S and W are strictly and weakly dominated by A respectively,

$$\begin{array}{cc} & \begin{matrix} B & I \end{matrix} \\ \begin{matrix} A \\ W \\ S \end{matrix} & \begin{pmatrix} 2,1 & 1,0 \\ 2,0 & 0,1 \\ 0,1 & 0,0 \end{pmatrix} \end{array}$$

Figure 4.3: A two-player non-zero-sum normal-form game, where regret matching plays the weakly dominated row player strategy W with positive probability.

and the column player's (denoted player 2) pure strategy I is iteratively weakly dominated by B. Suppose that when the denominator of equation (2.3) is zero, we assign $\sigma_i^{T+1}(a) = 1/|X_i^T|$ if $a \in X_i^T = \{a \in A_i \mid R_i^T(a) = \max_{b \in A_i} R_i^T(b)\}$, and $\sigma_i^{T+1}(a) = 0$ otherwise. Starting with all regrets being zero, the initial profile σ^1 is the uniform random profile. After one iteration, we have regrets $R_1^1(A) = 2/3$, $R_1^1(W) = 1/6$, $R_1^1(S) = -5/6$, $R_2^1(B) = 1/6$ and $R_2^1(I) = -1/6$, yielding the next strategy profile $\sigma^2 = \{(A, 4/5), (W, 1/5), (S, 0), (B, 1), (I, 0)\}$. One can check that successive iterations after the first only add negative regret to S and I and that $\sigma^T = \sigma^2$ for all $T \geq 2$. In particular, the weakly dominated strategy W continues to be played indefinitely with probability $1/5$.

We now turn our attention to extensive-form games, which are our primary concern. Here, the linear program (2.7) cannot feasibly be applied to find non-iteratively strictly dominated strategies in even moderately-sized extensive-form games as the programs would require a number of constraints exponential in the size of the game. On the other hand, we can apply CFR.

First, we consider the removal of iteratively strictly dominated actions. Our results rely on two conditions. Let x^T be the number of iterations t where $\sum_{a \in A(I)} R_i^{t,+}(I, a) = 0$ for some $i \in N$ and $I \in \mathcal{I}_i$, $1 \leq t \leq T$. The first condition we require is that x^T be sublinear in T. Intuitively, this is necessary because otherwise, the denominator of equation (2.4) is zero too often, and so regret matching too often yields an arbitrary strategy at some $I \in \mathcal{I}_i$ that potentially plays a dominated action. While we cannot prove that this condition always holds, we show empirically that x^T/T decreases over time in the next section. Next, for $I \in \mathcal{I}_i$ and $\delta \geq 0$, define $\Sigma_\delta(I) = \{\sigma \in \Sigma \mid \sum_{h \in I} \pi_{-i}^\sigma(h) \geq \delta\}$ to be the set of profiles where the probability that the opponents play to reach I, $\sum_{h \in I} \pi_{-i}^\sigma(h)$, is at least δ. The second condition we require is that the opponents reach each information set I containing a dominated action *often enough*, meaning that there exist real numbers $\delta, \gamma > 0$ and an integer T' such that for all $T \geq T'$, $|\Sigma_\delta(I) \cap \{\sigma^t \mid T' \leq t \leq T\}| \geq \gamma T$. This condition appears necessary because the magnitude of the counterfactual regret $|r_i^t(I, a)| = |v_i(I, \sigma^t_{(I \to a)}) - v_i(\sigma^t)| \leq \Delta_i \sum_{h \in I} \pi_{-i}^{\sigma^t}(h)$ is weighted by the probability of the opponents reaching I. Thus, if the opponents reach I with probability zero, then we will stop *learning* how to adjust our strategy. Since it could take several iterations to eliminate an iteratively strictly dominated action, we may end up playing such an action when I is not reached by the opponents often enough.

Theorem 4.2. *Let $\sigma^1, \sigma^2, \ldots$ be strategy profiles generated by CFR in an extensive-form game with* *perfect recall, let $I \in \mathcal{I}_i$, and let a be an iteratively strictly dominated action at I, where removal in sequence of the iteratively strictly dominated actions $a_1, ..., a_k$ at $I_1, ..., I_k$ respectively yields iterative dominance of $a_{k+1} = a$. If for $1 \leq \ell \leq k+1$, there exist real numbers $\delta_\ell, \gamma_\ell > 0$ and an integer T_ℓ such that for all $T \geq T_\ell$, $|\Sigma_{\delta_\ell}(I_\ell) \cap \{\sigma^t \mid T_\ell \leq t \leq T\}| \geq \gamma_\ell T$, then*

(i) *there exists an integer T_0 such that for all $T \geq T_0$, $R_i^T(I, a) < 0$,*

(ii) *if $\lim_{T\to\infty} x^T/T = 0$, then $\lim_{T\to\infty} y^T(I, a)/T = 0$, where $y^T(I, a)$ is the number of iterations $1 \leq t \leq T$ satisfying $\sigma^t(I, a) > 0$, and*

(iii) *if $\lim_{T\to\infty} x^T/T = 0$, then $\lim_{T\to\infty} \pi_i^{\bar{\sigma}^T}(I)\bar{\sigma}_i^T(I, a) = 0$.*

Proof sketch. Similar to the proof of Theorem 4.1, there exists an $\epsilon > 0$ and a term F such that

$$R_i^T(I, a) \leq -\epsilon\gamma T + F$$

where in a game with perfect recall, $\lim_{T\to\infty} F/T = 0$. Again, this implies that there exists an integer T_0 such that for all $T \geq T_0$, $R_i^T(I, a) < 0$, establishing part (i). Since CFR applies regret matching at I, part (i) and equation (2.4) imply that for all $T \geq T_0$, either $\sum_{b\in A(I)} R_i^{T,+}(I, b) = 0$ or $\sigma_i^{T+1}(I, a) = 0$. From this, we have

$$\lim_{T\to\infty} \frac{y^T(I, a)}{T} \leq \lim_{T\to\infty} \frac{y^{T_0}(I, a) + x^T}{T} = 0,$$

proving part (ii). Finally, part (iii) follows according to

$$\lim_{T\to\infty} \pi_i^{\bar{\sigma}^T}(I)\bar{\sigma}_i^T(I, a) = \lim_{T\to\infty} \frac{\sum_{t=1}^T \pi_i^{\sigma^t}(I)\sigma_i^t(I, a)}{T} \leq \lim_{T\to\infty} \frac{y^T(I, a)}{T} = 0,$$

where the first equality is by the definition of the average strategy and the inequality is by definition of $y^T(I, a)$. ■

Part (iii) of Theorem 4.2 says that an iteratively strictly dominated action is not reached or is removed from the average profile $\bar{\sigma}^T$ in the limit, whereas part (i) suggests that iteratively strictly dominated actions are removed from the current profile σ^T after just a finite number of iterations (except possibly when $\sum_{a\in A(I)} R_i^{T,+}(I, a) = 0$). Finally, part (ii) states that the number of current profiles that play an iteratively strictly dominated action a at I, $y^T(I, a)$, is sublinear in T.

Next, we show that the profiles generated by CFR eliminate all iteratively strictly dominated strategies, assuming again that $x^T/T \to 0$.

Theorem 4.3. *Let $\sigma^1, \sigma^2, ...$ be strategy profiles generated by CFR in an extensive-form game with perfect recall, and let σ_i be an iteratively strictly dominated strategy. Then,*

(i) *there exists an integer T_0 such that for all $T \geq T_0$, there exist $I \in \mathcal{I}_i$, $a \in A(I)$ such that $\pi_i^\sigma(I)\sigma_i(I, a) > 0$ and $R_i^T(I, a) < 0$, and*

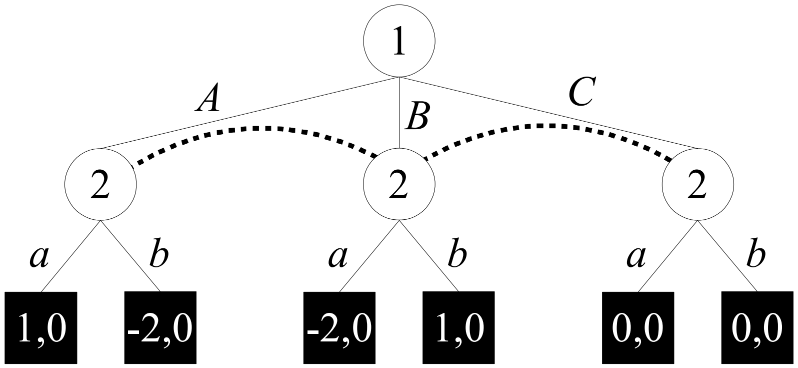


Figure 4.4: A two-player non-zero-sum extensive-form game where each player has a single information set.

(ii) *if* $\lim_{T\to\infty} x^T/T = 0$*, then* $\lim_{T\to\infty} y^T(\sigma_i)/T = 0$*, where* $y^T(\sigma_i)$ *is the number of iterations* $1 \leq t \leq T$ *satisfying supp*$(\sigma_i) \subseteq$ *supp*(σ_i^t).

Proof sketch. For $\sigma_i' \in \Sigma_i$, define

$$R^T_{i,\text{full}}(\sigma_i') = \sum_{t=1}^{T}(u_i(\sigma_i', \sigma_{-i}^t) - u_i(\sigma^t)).$$

Similar to the proof of Theorems 4.1 and 4.2, there exists an $\epsilon > 0$ and a term F' such that

$$R^T_{i,\text{full}}(\sigma_i) \leq -\epsilon T + F' \tag{4.1}$$

where in a game with perfect recall, $\lim_{T\to\infty} F'/T = 0$. Next, one can show that

$$R^T_{i,\text{full}}(\sigma_i) = \sum_{I\in\mathcal{I}_i} \pi_i^\sigma(I) \sum_{a\in A(I)} \sigma_i(I,a) R_i^T(I,a). \tag{4.2}$$

Since $\pi_i^\sigma(I), \sigma_i(I,a) \geq 0$, it follows by equations (4.1) and (4.2) that after a finite number of iterations T_0, there exist $I \in \mathcal{I}_i$, $a \in A(I)$ such that $\pi_i^\sigma(I)\sigma_i(I,a) > 0$ and $R_i^T(I,a) < 0$, establishing part (i). Part (ii) then follows as in the proof of part (ii) of Theorem 4.2. ■

Similar to part (i) of Theorem 4.2, part (i) of Theorem 4.3 says that after a finite number of iterations, there is always some information set I that the dominated strategy σ_i plays to reach and some action at I played by σ_i which σ_i^T does not play (except possibly when $\sum_{a\in A(I)} R_i^{T,+}(I,a) = 0$) and so $\sigma_i^T \neq \sigma_i$. Part (ii) similarly states that the number of profiles generated whose support contains supp(σ_i), $y^T(\sigma_i)$, is sublinear in T. Notice that Theorems 4.1 and 4.3 do not draw any conclusions upon the average profile $\bar{\sigma}^T$. Perhaps surprisingly, it is possible to have a sequence of profiles with no regret where the average profile converges to a strictly dominated strategy. Consider the two-player non-zero-sum game in Figure 4.4. The sequence of pure strategy profiles $(A,a), (B,b), (A,a), (B,b), ...$ has no positive regret for either player, and in the limit, the average profile for player 1, $\bar{\sigma}_1^T$, plays A and B each with probability 0.5. However, $\bar{\sigma}_1^T$ is strictly dominated by the pure strategy that always plays C. As with regret matching in normal-form games, CFR cannot guarantee elimination of weakly dominated actions or strategies by a similar counterexample to that presented in Figure 4.3.

Our final theoretical contribution of this chapter shows that in two-player non-zero-sum games, regret minimization yields a bound on the average strategy profile's exploitability.

Theorem 4.4. *Let $\epsilon, \delta \geq 0$ and let $\sigma^1, \sigma^2, ..., \sigma^T$ be strategy profiles in a two-player game with perfect recall. If $R_i^T/T \leq \epsilon$ for $i = 1, 2$, and $|u_1 + u_2| \leq \delta$, then $\bar{\sigma}^T$ is a $2(\epsilon+\delta)$-Nash equilibrium.*

Proof. We generalize the proof of Waugh [70, p. 11]. For $i = 1, 2$, by the definition of regret, we have

$$\begin{aligned}\epsilon &\geq \frac{1}{T} \max_{\sigma_i' \in \Sigma_i} \sum_{t=1}^{T} \big(u_i(\sigma_i', \sigma_{-i}^t) - u_i(\sigma^t)\big) \\ &= \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \bar{\sigma}_{-i}^T) - \frac{1}{T}\sum_{t=1}^{T} u_i(\sigma^t)\end{aligned}$$

by linearity of expectation and perfect recall. Summing the two inequalities for $i = 1, 2$ gives

$$\begin{aligned}2\epsilon &\geq \max_{\sigma_1' \in \Sigma_1} u_1(\sigma_1', \bar{\sigma}_2^T) + \max_{\sigma_2' \in \Sigma_2} u_2(\bar{\sigma}_1^T, \sigma_2') - \frac{1}{T}\sum_{t=1}^{T}\big(u_1(\sigma^t) + u_2(\sigma^t)\big) \\ &\geq \max_{\sigma_1' \in \Sigma_1} u_1(\sigma_1', \bar{\sigma}_2^T) + \max_{\sigma_2' \in \Sigma_2} \big(-u_1(\bar{\sigma}_1^T, \sigma_2') - \delta\big) - \delta \\ &= \max_{\sigma_1' \in \Sigma_1} u_1(\sigma_1', \bar{\sigma}_2^T) - \min_{\sigma_2' \in \Sigma_2} u_1(\bar{\sigma}_1^T, \sigma_2') - 2\delta \\ &\geq \max_{\sigma_1' \in \Sigma_1} u_1(\sigma_1', \bar{\sigma}_2^T) - u_1(\bar{\sigma}^T) - 2\delta,\end{aligned}$$

where the last line follows by setting $\sigma_2' = \bar{\sigma}_2^T$. Rearranging terms gives

$$\max_{\sigma_1' \in \Sigma_1} u_1(\sigma_1', \bar{\sigma}_2^T) \leq u_1(\bar{\sigma}^T) + 2(\epsilon + \delta).$$

Applying the same arguments but reversing the roles of the two players gives

$$\max_{\sigma_2' \in \Sigma_2} u_2(\bar{\sigma}_1^T, \sigma_2') \leq u_2(\bar{\sigma}^T) + 2(\epsilon + \delta),$$

and thus by definition $\bar{\sigma}^T$ is a $2(\epsilon + \delta)$-Nash equilibrium. ■

Theorem 4.4 is a generalization of Theorem 2.1. When $\delta = 0$, the game is zero-sum, and so the average profile converges to equilibrium as $\epsilon \to 0$. In addition, when the players' utilities sum to at most $\delta > 0$, then as $\epsilon \to 0$, the average profile converges to a 2δ-Nash equilibrium.

Remarks. Theorems 4.1, 4.2, and 4.3 provide evidence that regret minimization removes iterative strict domination. Of course, eliminating strict domination may not provide any useful insights in games where few strategies are iteratively strictly dominated. Despite this obvious limitation, Theorems 4.2 and 4.3 provide a better understanding of the strategies generated by CFR in non-zero-sum games than what coarse correlated equilibria provide. In the next section, we show that avoiding iteratively strictly dominated actions is enough to perform well in Kuhn Poker. However,

Table 4.1: Results of a six-agent mock tournament of Kuhn poker. Reported scores for the row strategy profile against the column profile are in expected milli-chips per game, averaged over both player orderings.

	Uni	ND	NID	Nash-0	Nash-0.5	Nash-1	**Overall**	Exploitability
Uni	-	-250	-187	-111	-139	-167	**-171**	458
ND	250	-	-21	-69	-56	-42	**13**	188
NID	187	21	-	-56	-42	-28	**17**	167
Nash-0	111	69	56	-	0	0	**47**	0
Nash-0.5	139	56	42	0	-	0	**47**	0
Nash-1	167	42	28	0	0	-	**47**	0

large games such as three-player Texas hold'em are too complex to analyze action and strategic dominance beyond obvious errors, such as folding the best hand. It remains open as to how well our theory explains the success of CFR in these large games.

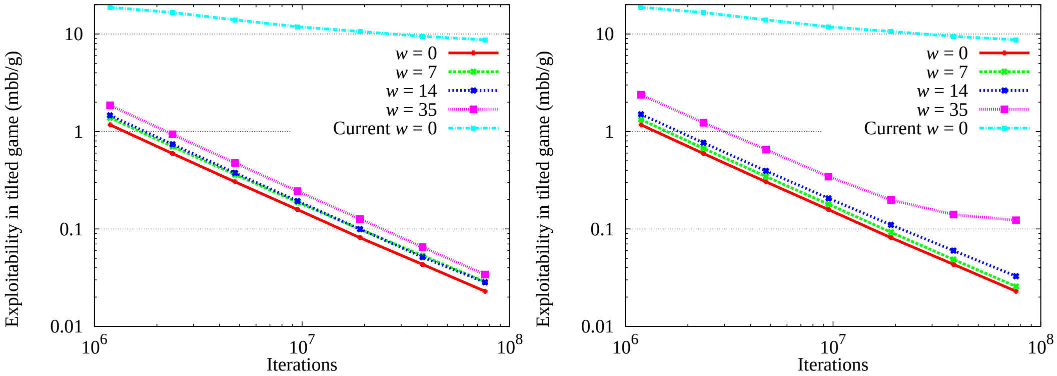
Perhaps more importantly, the theory developed here has guided us to a more efficient adaptation of CFR, in both time and memory, that we will use for games with more than two players. Given Theorems 4.2 and 4.3 and given we have only finite time, we suggest using the current profile in practice rather than the average. In fact, while Theorem 4.4 says that the average profile converges to a 2δ-Nash equilibrium in two-player games, there is no clear case for preferring the average over the current profile in three-or-more-player games. Furthermore, the average profile is not used in any computations during CFR, so when discarding the average, there is no reason to store the cumulative profile. This reduces the memory requirements of CFR by a factor of two, since then only one value per information set, action pair ($R_i^T(I,a)$) must be stored as opposed to two. Not only does this allow us to tackle larger games, the extra memory might be utilized to compute even stronger strategies than previously possible. Note that we are not the first to consider using the current profile. In CFR-BR, the algorithm described at the end of Section 2.4, the current profile converges to equilibrium with high probability in zero-sum games [42, Theorem 4]. The authors discuss similar benefits to discarding the cumulative profile in CFR-BR and just using the current strategy profile. Nonetheless, we are the first to suggest using the current profile in CFR and in games with more than two players. The next section explores these new insights.

4.4 Empirical Results

We now test our theory developed in the previous section across several poker domains.

4.4.1 Iteratively Strictly Dominated Actions and Performance in Kuhn Poker

To begin, we investigate the correlation between the presence of iteratively strictly dominated actions in one's strategy with the performance of the strategy in a mock ACPC-style tournament. In the ACPC, each game is evaluated according to two different scoring metrics. The total bankroll



(a) Orange tilt

(b) Green tilt

Figure 4.5: Log-log plots measuring exploitability of CFR strategies in w%-tilted 2-1 hold'em over iterations, measured in milli-big-blinds per game (mbb/g).

(TBR) metric simply ranks competitors according to their overall earnings in money per game averaged across all possible opponents. The instant runoff (IRO) metric, however, ranks competitors by iteratively eliminating the lowest scoring agent from consideration and reevaluating the overall scores by averaging only across the remaining agents. In a zero-sum game where players alternate positions, a Nash equilibrium strategy is optimal for winning IRO since it never loses in expectation to any opponent.

We ran a six-agent mock tournament of Kuhn Poker, which was introduced in Section 3.2. Kuhn Poker is a small enough game where we can easily identify all iteratively dominated actions and all Nash equilibrium strategies have already been classified [50]. Our agents consist of a uniform random strategy (**Uni**), a strategy that plays no strictly dominated actions (does not call with the Jack or fold with the King) but is otherwise uniform random (**ND**), a strategy that plays no iteratively strictly dominated actions (no strictly dominated actions and player 1 does not bet with the Queen) but is otherwise uniform random (**NID**), and three Nash equilibrium strategies (**Nash-γ**) for $\gamma = 0, 0.5, 1$, where γ is the probability of betting with the King. A cross table of the results for each pair of strategies is given in Table 4.1, along with each profile's exploitability. Not surprisingly, the Nash equilibrium strategies all tie for first place in IRO and happen to also tie for first in TBR. After the equilibrium strategies, NID is the next best agent in terms of IRO, TBR, and exploitability. Notice that despite winning less against Uni than ND wins, NID plays better against ND and the equilibrium strategies to still earn more than ND overall. Finally, we see that simply avoiding strictly dominated actions is enough to earn positive utility overall and be much closer to equilibrium compared to playing uniformly at random. This mock tournament provides one example where good performance can be achieved by simply avoiding (iteratively) strictly dominated actions.

4.4.2 Exploitability in Two-Player Non-Zero-Sum Games

Our next experiment applies CFR to non-zero-sum *tilted* variants of two-player 2-1 hold'em, where the original version of 2-1 hold'em was defined in Section 3.4.3. Tilted games are constructed by rewarding or penalizing players depending on the outcome of the game. This can lead to more aggressive play when applied to the regular, non-tilted game and were used by the poker program *Polaris* that won the 2008 Man-vs-Machine competition [44]. Here, we use the *orange* tilt that gives the winning player an extra $w\%$ bonus, and the *green* tilt that both reduces the losing player's loss in a showdown (*i.e.*, when neither player folded) by $w\%$ and penalizes the winning player by $w\%$ when the losing player folded. In both of these games, we can bound $|u_1 + u_2| \leq \Delta_i w/100$, and so Theorem 4.4 states that CFR will converge to at least a $\Delta_i w/50$-Nash equilibrium. For $w \in \{0, 7, 14, 35\}$, we ran the Public Chance Sampling variant of CFR, described in Section 3.5.2, and measured exploitability of the average profile in the $w\%$-tilted game by calculating $\max_{\sigma_i \in \Sigma_i} u_i(\sigma_i, \bar{\sigma}^T_{-i})$ and averaging over both players $i = 1, 2$. In addition, we also measured the exploitability of the current strategy profile in the zero-sum case ($w = 0$). These results are shown in Figure 4.5. As expected, in the non-tilted game ($w = 0$), the average profile is approaching a Nash equilibrium. For the tilted games, we see that as w is increased, most of the profiles are further from equilibrium, coinciding with Theorem 4.4. However, the strategies are much closer to equilibrium than the distance guaranteed by Theorem 4.4 (note that $\Delta_i = 8$ big blinds) and only in the green tilt with $w = 35$ is it obvious that CFR is not converging to an exact equilibrium. Of course, Theorem 4.4 only provides an upper bound on the average profile's exploitability, and this bound appears to be very loose. These results warrant further investigation into regret minimization in two-player non-zero-sum games. Finally, it is clear that the current strategy profile with $w = 0$ is not converging to equilibrium. Thus, unlike CFR-BR, the average profile from CFR is often preferred to the current profile in two-player games as it gives a better worst-case guarantee.

4.4.3 Nonpositive Regret and Current Profile in Three-Player Hold'em

Next, we examine how often $\sum_{a \in A(I)} R_i^{T,+}(I, a) = 0$ as required by parts of Theorems 4.2 and 4.3. External Sampling MCCFR was applied to two different abstractions of three-player limit Texas hold'em. The first, labelled *1X*, consists of 169, 900, 100, and 25 buckets per betting round respectively. This abstraction size was used in our winning 2010 ACPC 3-player agents described later in Section 8.1 and contains about 262 million information sets. The second abstraction, labelled *2X*, uses 169, 1800, 200, and 50 buckets per betting round respectively, resulting in an abstract game approximately twice the size. All of our abstractions were built using k-means clustering on earth mover's distance or OCHS as described in Section 3.4.2. For each abstraction, we measured

$$\xi^T = \left| \left\{ (I, t) \mid \sum_{a \in A(I)} R_i^{t,+}(I, a) = 0, I \in \hat{\mathcal{I}}_i^t, 1 \leq t \leq T \right\} \right|,$$

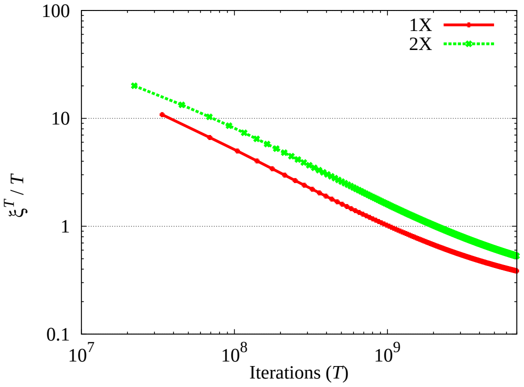
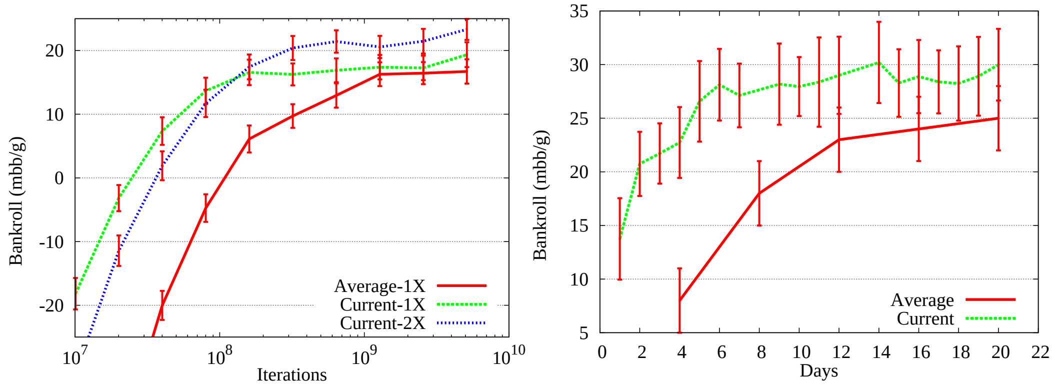


Figure 4.6: Log-log plot measuring the frequency at which an information set is visited where every action has nonpositive cumulative counterfactual regret during CFR in the 1X and 2X abstractions of three-player limit Texas hold'em.

the total number of times where External Sampling traversed an information set that had no positive regret at any action, where the set of information sets traversed on iteration t are denoted $\hat{\mathcal{I}}_i^t$. The average of ξ^T is plotted over iterations T in Figure 4.6. In both cases, we see that encountering an information set with no positive regret becomes less frequent over time, where we eventually encounter fewer than one such information set per iteration on average. While we cannot guarantee that $x^T/T \approx \xi^T/T \to 0$ as required by Theorems 4.2 and 4.3, we at least have evidence that having no positive regret becomes a rare event. By part (i) of Theorems 4.2 and 4.3, this means that iteratively strictly dominated actions and strategies will likely be avoided in the current strategy profile.

Using these same abstractions of three-player hold'em, we now show that the current profile can reach higher performance faster than the average profile, and that the extra savings in memory acquired by discarding the average profile can be utilized to generate even stronger strategies. In this experiment, we generated three different strategy profiles with CFR, saving the profiles at various iteration counts. For the 1X abstraction, we kept both the average and the current profile, while for the 2X abstraction, we kept just the current profile. Note that running CFR on the 2X abstract game without keeping the average profile requires no more RAM than running CFR on the 1X abstraction and keeping both profiles. For each of our saved profiles, we then played a four-agent round-robin competition (RRC) against the base strategy profiles[2] from the CPRG's 2009, 2010, and 2011 ACPC three-player entries for the IRO competitions. Figure 4.7a shows the amount won by each of our three strategies over iterations, averaged over 50 RRCs consisting of 10,000 games per match. Clearly, the 1X current profile reaches strong play much sooner than the average profile, which requires about ten times the number of iterations to peak at the same level of performance. Furthermore, while more iterations are needed in the 2X abstraction as expected by Theorem 2.5, we see that 2X eventually yields a current profile that outperforms both profiles in the 1X abstraction.

[2] The 2010 and 2011 agents employed special experts in two-player subtrees that were not used in this specific experiment.



(a)

(b)

Figure 4.7: (a) Performance over iterations (log scale) of three strategy profiles in a four-agent round-robin competition, measured in milli-big-blinds per game. *Current-2X* is the current profile generated by CFR in the 2X abstraction that is twice as large as the 1X abstraction used to generate *Average-1X* and *Current-1X*. Error bars indicate 95% confidence intervals over 50 competitions. (b) Performance over time (in days) of the average profile that won the three-player events of the 2012 ACPC, and of the current profile computed in the same abstraction. Error bars indicate 95% confidence intervals over 10 competitions versus the CPRG's 2009, 2010, and 2011 ACPC three-player agents for the IRO competitions.

Finally, as a final validation of our CFR modification, we show that the current strategy profile reaches peak performance faster than the average profile under the abstraction used for our 2012 ACPC three-player entry. The abstract game used contains approximately 2.5 billion information sets and is described later in Section 8.3. For the competition, the average profile was used and was computed over 20 days with External Sampling parallelized across 48 processors. We reran the same implementation of External Sampling with 48 processors on the same machine using the same abstraction, except now saving the current profile and discarding the average. For several checkpoints of the original average strategy and the new current strategy, we played 10 RRCs versus the CPRG's agents from the 2009, 2010, and 2011 competitions described in Chapter 8 and plotted the results in Figure 4.7b. While the average strategy takes 20 days before earning 25 mbb/g, the current strategy reaches better performance in just 5 days while requiring only half the memory to compute.

4.5 Conclusion

This chapter provides the first theoretical advancements for applying CFR to games that are not zero-sum. While previous work had demonstrated that CFR does not necessarily converge to a Nash equilibrium in such games, we have provided theoretical evidence that CFR eliminates iteratively strictly dominated actions and strategies. Thus, CFR provides a mechanism for removing iterative strict domination, which is infeasible with other techniques in large, non-zero-sum extensive-form

More details regarding these agents are provided in Chapter 8.

games. In addition, our theory is the first step to understanding why CFR generates well-performing strategies in non-zero-sum games. Though our experiments show that the current profile reaches a high level of performance faster than the average, it remains unclear whether this is due to faster removal of domination that our theory illustrates. Nonetheless, we have shown that just using the current profile gives a more time and memory efficient implementation of CFR for games with more than two players that can lead to increased performance.

Chapter 5

Generalized Sampling and Improved Monte Carlo CFR

Recall from Section 2.2.3 that MCCFR typically results in faster iterations than Vanilla CFR and can significantly reduce computation time to reach a given solution quality. Rather than compute the exact counterfactual values v_i as in Vanilla CFR, MCCFR computes sampled counterfactual values $\tilde{v}_i$ by traversing only a subset of the actions at each history. However, as we traverse fewer actions at a given node, the sampled counterfactual value is potentially less accurate. Figures 5.1a and 5.1b compare the values computed by Vanilla CFR and Outcome Sampling (OS) respectively to illustrate this point. For OS, an "informative" sampled counterfactual value for just a single action is obtained at each information set along the sampled block (history). All other actions are essentially assigned a sampled counterfactual value of zero by definition (equation (2.5)). While $\mathbf{E}_{\mathcal{Q}}[\tilde{v}_i] = v_i$, variance is introduced, affecting both the regret updates and the value recursed back to the parent. As we will see later in the chapter, OS is a poor choice of algorithm for many games.

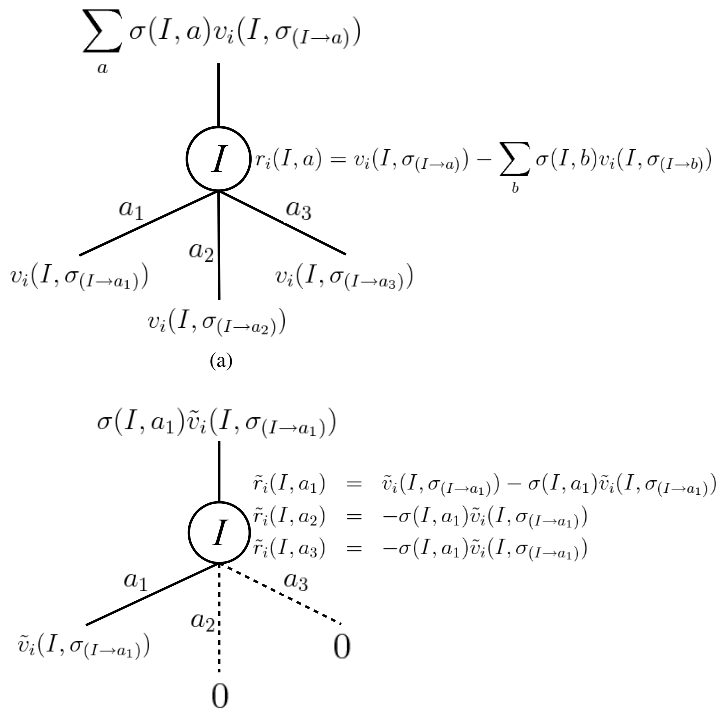
While Chance Sampling (CS) and External Sampling (ES) are typically better options than Vanilla CFR and OS, they may still not be suitable for games containing many player actions, such as no-limit poker. This is because during player i's traversal, both CS and ES traverse subtrees under *every* action at player i's reached information sets. For example, if player i can choose between one of k actions at an information set, both CS and ES will spend the time to traverse all k actions. When k is large, every iteration of CS and ES is computationally expensive.

In this chapter, we address these problems and more with the following contributions:

1. Tighter theoretical bounds on the number of iterations required by Vanilla CFR, CS, ES, and OS to reach a given solution quality.

2. A generalization of MCCFR in which we bound the average regret in terms of the variance of estimated counterfactual values, suggesting that estimates with lower variance are preferred.

3. A new *probing* CFR sampling algorithm that lives outside of the MCCFR family of algo-


$\sum_a \sigma(I,a)v_i(I,\sigma_{(I\to a)})$
I
$r_i(I,a) = v_i(I,\sigma_{(I\to a)}) - \sum_b \sigma(I,b)v_i(I,\sigma_{(I\to b)})$
a_1
a_2
a_3
$v_i(I,\sigma_{(I\to a_1)})$
$v_i(I,\sigma_{(I\to a_2)})$
$v_i(I,\sigma_{(I\to a_3)})$
$\sigma(I,a_1)\tilde{v}_i(I,\sigma_{(I\to a_1)})$
$\tilde{r}_i(I,a_1) = \tilde{v}_i(I,\sigma_{(I\to a_1)}) - \sigma(I,a_1)\tilde{v}_i(I,\sigma_{(I\to a_1)})$
$\tilde{r}_i(I,a_2) = -\sigma(I,a_1)\tilde{v}_i(I,\sigma_{(I\to a_1)})$
$\tilde{r}_i(I,a_3) = -\sigma(I,a_1)\tilde{v}_i(I,\sigma_{(I\to a_1)})$
I
a_1
a_2
a_3
$\tilde{v}_i(I,\sigma_{(I\to a_1)})$
0
0

(a)

(b)

Figure 5.1: **(a)** The computed values at information set I during Vanilla CFR. First, for each action, the counterfactual values are recursively computed. The counterfactual regrets are then computed before returning the counterfactual value at I to the parent. **(b)** The computed values at I during OS. Here, only action a_1 is sampled and its sampled counterfactual value is recursively computed. The remaining two actions are effectively assigned zero sampled counterfactual value. The sampled counterfactual regrets are then computed before returning the sampled counterfactual value at I to the parent.

rithms and demonstrates one way of reducing the variance in the updates to provide faster convergence to equilibrium in zero-sum games.

4. *Average Strategy Sampling (AS)*, a new MCCFR algorithm propelled from our tighter theoretical bounds that samples player actions and is more suitable for games involving many player choices.

5. *Pure CFR*, a second algorithm outside the MCCFR family that resembles the tree traversals of ES, but requires only half the computer memory.

We start by introducing the tighter theoretical bounds. Then, we discuss our MCCFR generalization, provide a regret bound, and demonstrate our *probing* technique that can reduce variance and converge to equilibrium faster than corresponding MCCFR algorithms. Next, we introduce AS and demonstrate its effectiveness in no-limit poker and a non-poker game called *Bluff*. Finally, we describe Pure CFR and conclude with a comparison of several of these algorithms across a number of different poker games.

5.1 New CFR Bounds

In this section, we revisit the original CFR analysis to uncover a tighter bound on each player's average regret. More importantly, this result provides insight into our new Average Strategy Sampling algorithm, which we present later in Section 5.4. The results in this section and in Section 5.4 are joint work with Marc Lanctot, Neil Burch, and Duane Szafron [24].

Throughout this chapter, we assume perfect recall as defined by Definition 2.3. In the original CFR paper, Zinkevich *et al.* [75] prove that a player's regret is bounded by the sum of the cumulative counterfactual regrets (Theorem 2.3). Through a more careful examination, we can actually *equate* a player's regret to a weighted sum of the cumulative counterfactual regrets. This result is our main improvement over the original analysis and is stated below in Theorem 5.1. For a strategy $\sigma_i \in \Sigma_i$ and an information set $I \in \mathcal{I}_i$, define $R_i^T(I, \sigma_i) = \sum_{a \in A(I)} \sigma_i(I,a) R_i^T(I,a)$. In addition, let $\sigma_i^* \in \Sigma_i$ be a player i strategy such that

$$\sigma_i^* = \operatorname*{argmax}_{\sigma_i' \in \Sigma_i} \sum_{t=1}^{T} u_i(\sigma_i', \sigma_{-i}^t). \tag{5.1}$$

Note that in a two-player game, $\sum_{t=1}^T u_i(\sigma_i^*, \sigma_{-i}^t) = T u_i(\sigma_i^*, \bar{\sigma}_{-i}^T)$, and thus σ_i^* is a best response to the opponent's average strategy after T iterations. We provide a sketch of the proof of Theorem 5.1 below, while all proofs in this chapter are provided in full in Appendix C.

Theorem 5.1. *In an extensive-form game with perfect recall,*

$$R_i^T = \sum_{I \in \mathcal{I}_i} \pi_i^{\sigma^*}(I) R_i^T(I, \sigma_i^*).$$

Proof sketch. The proof is by induction on the maximum number of information sets that player i can reach in a single game. The base case where player i has no information sets is trivial as $R_i^T = 0 = \sum_{I \in \emptyset} \pi_i^{\sigma^*}(I) R_i^T(I, \sigma_i^*)$. For the induction step, we can express part of the regret term $R_i^T(I, \sigma_i^*)$ as a weighted sum over future counterfactual values,

$$\sum_{I' \in Succ(I,a)} \sigma_i^*(I', a) v_i(I', \sigma^t) \tag{5.2}$$

for each $a \in A(I)$, where $Succ(I,a)$ is the set of all possible next information sets for player i after playing a at I. Next, we apply the induction hypothesis at each $I' \in Succ(I,a)$, equating the average regret in each subtree to the weighted sum of the cumulative counterfactual regrets within the subtree. Since $v_i(I', \sigma^t)$ appears in this equation, we can substitute for this term in equation (5.2). We then note that if the probability of player i reaching information set J under σ_i^* in a subtree is $\hat{\pi}_i^{\sigma^*}(J)$, then the probability of reaching J from I in the full game is $\pi_i^{\sigma^*}(I, J) = \sigma_i^*(I,a) \hat{\pi}_i^{\sigma^*}(J)$. Rearranging the remaining terms gives the result. ■

Theorem 5.1 leads to a tighter bound on the average regret when using CFR. To achieve this tighter bound, we extend the definition of the M-value presented in Section 2.2.2 as follows. For a strategy $\sigma_i \in \Sigma_i$ and for $\mathcal{B}_i$ as defined previously, define the **$\boldsymbol{M}$-value of $\boldsymbol{\sigma_i}$** to be $M_i(\sigma_i) = \sum_{B \in \mathcal{B}_i} \pi_i^\sigma(B)\sqrt{|B|}$, where $\pi_i^\sigma(B) = \max_{I \in B} \pi_i^\sigma(I)$. Clearly, $M_i(\sigma_i) \leq M_i$ for all $\sigma_i \in \Sigma_i$ since $\pi_i^\sigma(B) \leq 1$. For Vanilla CFR, we can simply replace M_i in Theorem 2.4 with $M_i(\sigma_i^*)$:

Theorem 5.2. *When using Vanilla CFR in a game with perfect recall, average regret is bounded by*

$$\frac{R_i^T}{T} \leq \frac{\Delta_i M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}}.$$

When $M_i(\sigma_i^*) < M_i$, Theorem 5.2 provides a tighter bound on player i's average regret at time T. This strict inequality occurs for player 1 in the game in Figure 2.6a, for example, whenever σ_1^* is not the strategy that always plays action r at every information set.

For MCCFR, we can show a similar improvement applied to Theorem 2.5 for both ES and OS. Our proof also includes a bound for CS that appears to have been omitted in previous work. Recall form Section 2.2.3 that these particular MCCFR algorithms allow us to decompose the probability of sampling a terminal history z according to $q(z) = \prod_{i \in N \cup \{c\}} q_i(z)$ so that $q_i(z)$ is the probability contributed to $q(z)$ by sampling player i's actions. In addition, a factor of $\sqrt{|A(\mathcal{I}_i)|}$ can be removed from the term introduced from sampling which appears to have been overlooked in the proof of Theorem 2.5. Details of this improvement are in Appendix C.

Theorem 5.3. *Let X be one of CS, ES, or OS (assuming OS samples chance and opponent actions according to σ_{-i}), let $p \in (0, 1]$, and let $\delta = \min_{z \in Z} q_i(z) > 0$ over all $1 \leq t \leq T$. When using X in a game with perfect recall, with probability $1 - p$, average regret is bounded by*

$$\frac{R_i^T}{T} \leq \left(M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|} + \frac{2\sqrt{|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}} \right) \left(\frac{1}{\delta}\right) \frac{\Delta_i}{\sqrt{T}}.$$

Recall that when using CS or ES, each of player i's actions are sampled with probability 1, and therefore $\delta = \min_{z \in Z} q_i(z) = 1$.

5.2 Generalized Sampling

Here, we present a new, generalized bound on the average regret to that of Theorems 2.5 and 5.3. While MCCFR provides an explicit form for the sampled counterfactual values $\tilde{v}_i(I, \sigma)$ given by equation (2.5), we let $\hat{v}_i(I, \sigma)$ denote *any* estimator of the true counterfactual value $v_i(I, \sigma)$. We then define the **estimated counterfactual regret** on iteration t for action a at I to be $\hat{r}_i^t(I, a) = \hat{v}_i(I, \sigma^t_{(I \to a)}) - \hat{v}_i(I, \sigma^t)$, and define the **cumulative estimated counterfactual regret** to be $\hat{R}_i^T(I, a) = \sum_{t=1}^T r_i^t(I, a)$. This generalization creates many possibilities not considered in MCCFR. For instance, instead of sampling a block Q of terminal histories, one can consider a sampled set of information sets and only update regrets at those sampled locations. Two more examples are provided later in Sections 5.3 and 5.5.

The results in this section and the following section are joint work with Marc Lanctot, Neil Burch, Duane Szafron, and Michael Bowling [25]. The following lemma probabilistically bounds the average regret in terms of the variance, covariance, and bias between the estimated and true counterfactual regrets:

Lemma 5.1. *Let $p \in (0,1]$ and suppose that there exists a bound $\hat{\Delta}_i$ on the difference between any two estimates, $\hat{v}_i(I, \sigma_{(I \to a)}) - \hat{v}_i(I, \sigma_{(I \to b)}) \le \hat{\Delta}_i$. If strategies are selected according to regret matching* (2.4) *on the cumulative estimated counterfactual regrets in a game with perfect recall, then with probability at least* $1 - p$*, the average regret is bounded by*

$$\frac{R_i^T}{T} \le |\mathcal{I}_i| \left(\frac{\hat{\Delta}_i \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} + \sqrt{\frac{\mathbf{Var}}{pT} + \frac{\mathbf{Cov}}{p} + \frac{\mathbf{E}^2}{p}} \right)$$

where

$$\mathbf{Var} = \max_{\substack{t \in \{1,\ldots,T\} \\ I \in \mathcal{I}_i \\ a \in A(I)}} \mathbf{Var}\left[r_i^t(I,a) - \hat{r}_i^t(I,a) \right],$$

$$\mathbf{Cov} = \max_{\substack{t,t' \in \{1,\ldots,T\} \\ t \neq t' \\ I \in \mathcal{I}_i \\ a \in A(I)}} \mathbf{Cov}\left[r_i^t(I,a) - \hat{r}_i^t(I,a), r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a) \right], \textit{ and}$$

$$\mathbf{E} = \max_{\substack{t \in \{1,\ldots,T\} \\ I \in \mathcal{I}_i \\ a \in A(I)}} \mathbf{E}[r_i^t(I,a) - \hat{r}_i^t(I,a)].$$

The proof is similar to that of Theorem 7 by Lanctot *et al.* [55] and of Theorem 5.3 and is provided in Appendix C. Lemma 5.1 implies that *unbiased* estimators of $v_i(I, \sigma)$ that are sampled independently on each iteration give a probabilistic guarantee of minimizing regret:

Theorem 5.4. *If in addition to the conditions of Lemma 5.1, for all $I \in \mathcal{I}_i$, $a \in A(I)$, $t \ge 1$, $\hat{v}_i(I, \sigma^t)$ and $\hat{v}_i(I, \sigma^t_{(I \to a)})$ are unbiased estimators of $v_i(I, \sigma^t)$ and $v_i(I, \sigma^t_{(I \to a)})$ respectively, and for all $t' \neq t$, $\hat{v}_i(I, \sigma^t)$ and $\hat{v}_i(I, \sigma^t_{(I \to a)})$ are sampled independently of $\hat{v}_i(I, \sigma^{t'})$ and $\hat{v}_i(I, \sigma^{t'}_{(I \to a)})$, then with probability at least* $1 - p$,

$$\frac{R_i^T}{T} \le \left(\hat{\Delta}_i \sqrt{|A(\mathcal{I}_i)|} + \frac{\sqrt{\mathbf{Var}}}{\sqrt{p}} \right) \frac{|\mathcal{I}_i|}{\sqrt{T}}. \tag{5.3}$$

Note that **Var** defined in Lemma 5.1 is bounded above by $\hat{\Delta}_i$. If we were to substitute this bound in for **Var** in equation (5.3), we would arrive at a bound similar to those provided by Theorems 2.5 and 5.3. However, by avoiding this simplification, Theorem 5.4 provides new insight into the role played by the variance of the estimator. Given two unbiased estimators $\hat{v}_i(I, \sigma)$ and $\hat{v}'_i(I, \sigma)$ with a common bound $\hat{\Delta}_i$ but differing variance, using the estimator with lower variance will yield a smaller bound on the average regret after T iterations. For a fixed $\epsilon > 0$, this suggests that in a zero-sum game, estimators with lower variance will require fewer iterations to reach an ϵ-Nash equilibrium. Furthermore, if some structure on the estimates $\hat{v}_i(I, \sigma)$ holds, we can produce a tighter bound than equation (5.3) by replacing $|\mathcal{I}_i|$ with $M_i(\sigma_i^*)$ introduced in the previous section. Details of this improvement can be found in Appendix C.

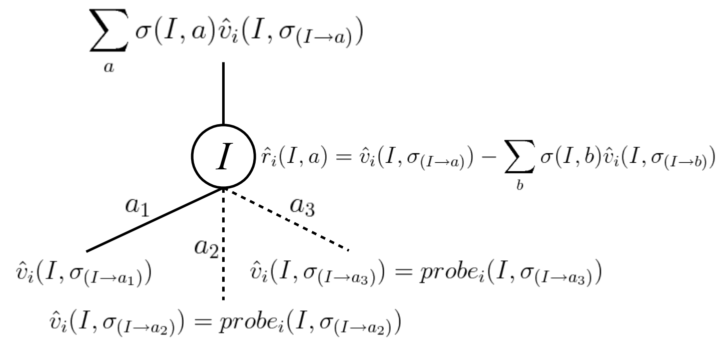

$\sum_a \sigma(I,a)\hat{v}_i(I,\sigma_{(I\to a)})$
I
$\hat{r}_i(I,a) = \hat{v}_i(I,\sigma_{(I\to a)}) - \sum_b \sigma(I,b)\hat{v}_i(I,\sigma_{(I\to b)})$
a_1
a_2
a_3
$\hat{v}_i(I,\sigma_{(I\to a_1)})$
$\hat{v}_i(I,\sigma_{(I\to a_3)}) = probe_i(I,\sigma_{(I\to a_3)})$
$\hat{v}_i(I,\sigma_{(I\to a_2)}) = probe_i(I,\sigma_{(I\to a_2)})$

Figure 5.2: An example of computed values at I during our new probing algorithm. In this example, again only a_1 is sampled and its estimated counterfactual value is recursively computed. The remaining two actions are *probed* to improve both the estimated counterfactual regrets and the returned estimated counterfactual value at I.

While unbiased estimators with lower variance may reduce the number of iterations required, we must define these estimators carefully. If the estimator is expensive to compute, the time per iteration will be costly and overall computation time may even increase. For example, the true counterfactual values $v_i(I,\sigma)$ have zero variance, but computing these values with Vanilla CFR is too time consuming in large games. In the next section, we present a new bounded, unbiased estimator that exhibits lower variance than $\tilde{v}_i(I,\sigma)$ and can be computed efficiently.

5.3 Probing

We now provide an example of how our new theoretical findings from Section 5.2 can be leveraged to produce better quality strategies after a fixed amount of computation time. Our example is an extension of MCCFR that attempts to reduce variance by replacing "zeroed-out" counterfactual values of player i's non-sampled actions, as in Figure 5.1b, with closer estimates of the true counterfactual values. Figure 5.2 illustrates this idea. The simplest instance of our new algorithm *probes* each non-sampled action a at I for its counterfactual value. A probe is a single Monte Carlo roll-out, starting with action a at I and selecting subsequent actions according to the current strategy profile σ^t until a terminal history z is reached. By rolling out actions according to the current profile, a probe is guaranteed to provide an unbiased estimate of the counterfactual value for a at I. In general, one can perform multiple probes per non-sampled action, probe only a subset of the non-sampled actions, probe off-policy, or factor in multiple terminal histories per probe. While Appendix C touches on this generalization, our presentation here sticks to the simple, inexpensive case of one on-policy, single trajectory probe for each non-sampled action.

We now formally define the estimated counterfactual value $\hat{v}_i(I,\sigma)$ obtained via probing, followed by a description of a new CFR sampling algorithm that updates regrets according to these estimates. Similar to MCCFR, let $\mathcal{Q}$ be a set of blocks spanning Z from which we sample a block

$Q \in \mathcal{Q}$ for player i on every iteration. To further simplify our discussion, we will assume for the remainder of this section that just as in External Sampling, each Q samples a single action at every history h not belonging to player i, sampled according to the known chance probabilities σ_c or the opponent's current strategy σ_{-i} respectively. Additionally, we assume that the set of actions sampled at $I \in \mathcal{I}_i$, denoted $Q(I)$, is nonempty and independent of every other set of actions sampled. While Appendix C shows that probing can be generalized to work for any choice of $\mathcal{Q}$, this simplification reduces the number of probabilities to compute in our algorithm and worked well in preliminary experiments. Once Q has been sampled, we form an additional set of terminal histories, or **probes**, $Y \subseteq Z \backslash Q$, generated as follows. For each non-terminal history $h \in H_i$ for player i reached and each action $a \in A(h)$ that Q does not sample ($a \notin Q(I(h))$), we generate exactly one terminal history $z = z_{ha} \in Y$, where $z \in Z \backslash Q$ is selected on-policy (*i.e.*, with probability $\pi^\sigma(ha, z)$). In other words, each non-sampled action is probed according to the current strategy profile σ and the known chance probabilities. Recall that Z_I is the set of terminal histories that have a prefix in the information set I. Given both Q and Y, when $Z_I \cap Q \neq \emptyset$, our estimated counterfactual value is defined to be

$$\hat{v}_i(I, \sigma) = \frac{1}{q_i(I)} \left[\sum_{z \in Z_I \cap Q} \pi_i^\sigma(z[I], z) u_i(z) + \sum_{z_{ha} \in Z_I \cap Y} \pi_i^\sigma(z_{ha}[I], ha) u_i(z_{ha}) \right],$$

where

$$q_i(I) = \prod_{(I', a') \in X_i(I)} \mathbf{Prob}[a' \in Q(I')]$$

is the probability contributed from sampling player i's actions that I is reached. Recall that in a game with perfect recall, $X_i(I)$ is the sequence of information set, action pairs for player i that lead to information set I, as defined in Section 2.1.1. When $Z_I \cap Q = \emptyset$, $\hat{v}_i(I, \sigma)$ is defined to be zero.

Proposition 5.1. *In a game with perfect recall, if $q_i(I) > 0$ for all $I \in \mathcal{I}_i$, then $\hat{v}_i(I, \sigma)$ is a bounded, unbiased estimate of $v_i(I, \sigma)$.*

The proof of Proposition 5.1 is provided in Appendix C. Since estimated counterfactual values are sampled independently between iterations, Proposition 5.1 and Theorem 5.4 provide a probabilistic guarantee that updating regret according to our estimated counterfactual values will minimize regret. Note that the differences $\tilde{v}_i(I, \sigma_{(I \to a)}) - \tilde{v}_i(I, \sigma_{(I \to b)})$ and $\hat{v}_i(I, \sigma_{(I \to a)}) - \hat{v}_i(I, \sigma_{(I \to b)})$ are both bounded above by $\hat{\Delta}_i = \Delta_i / \delta$, where $\delta = \min_{I \in \mathcal{I}_i} q_i(I)$. Thus, Theorem 5.4 suggests that variance reduction should lead to less regret after a fixed number of iterations. Probing specifically aims to achieve variance reduction through $\hat{v}_i(I, \sigma)$ when only a strict subset of player i's actions are sampled. Note that if we choose $\mathcal{Q}$ such that we always sample all of player i's actions, like in External Sampling, then we have $Y = \emptyset$ and $\hat{v}_i(I, \sigma) = \tilde{v}_i(I, \sigma)$.

5.3.1 Pseudocode

Algorithm 3 MCCFR + Probing (Two-player version)

```
 1: Require: $\forall\ I$, action set sampling distribution $\mathcal{Q}(I)$
 2: Initialize regret: $\forall I, \forall a \in A(I) : R(I, a) \leftarrow 0$
 3: Initialize cumulative profile: $\forall I, \forall a \in A(I) : s(I, a) \leftarrow 0$
 4:
 5: function Probe(history $h$, player $i$):
 6:     if $h \in Z$ then return $u_i(h)$ end if
 7:     if $P(h) = c$ then Sample action $a \sim \sigma_c(h, \cdot)$, return Probe($ha$, $i$) end if
 8:     $I \leftarrow I(h)$, $\sigma(I, \cdot) \leftarrow$RegretMatching($R(I, \cdot)$)
 9:     Sample action $a \sim \sigma(I, \cdot)$
10:     return Probe($ha$, $i$)
11:
12: function WalkTree(history $h$, player $i$, sample prob $q$):
13:     if $h \in Z$ then return $u_i(h)$ end if
14:     if $P(h) = c$ then Sample action $a \sim \sigma_c(h, \cdot)$, return WalkTree($ha$, $i$, $q$) end if
15:     $I \leftarrow I(h)$, $\sigma(I, \cdot) \leftarrow$RegretMatching($R(I, \cdot)$)
16:     if $P(h) \neq i$ then
17:         for $a \in A(I)$ do $s(I, a) \leftarrow s(I, a) + (\sigma(I, a)/q)$ end for
18:         Sample action $a \sim \sigma(I, \cdot)$, return WalkTree($ha$, $i$, $q$)
19:     end if
20:     Sample action set $Q(I) \sim \mathcal{Q}(I)$
21:     for $a \in A(I)$ do
22:         if $a \in Q(I)$ then
23:             $q' \leftarrow q \cdot \mathbf{P}_{\mathcal{Q}(I)}[a \in Q(I)]$
24:             $\hat{v}(a) \leftarrow$ WalkTree($ha$, $i$, $q'$)
25:         else
26:             $\hat{v}(a) \leftarrow$ Probe($ha$, $i$)
27:         end if
28:     end for
29:     for $a \in A(I)$ do $R(I, a) \leftarrow R(I, a) + (1/q)\left(\hat{v}(a) - \sum_{b \in A(I)} \sigma(I, b)\hat{v}(b)\right)$ end for
30:     return $\sum_{a \in A(I)} \sigma(I, a)\hat{v}(a)$
31:
32: function Solve(iterations $T$):
33:     for $t \in \{1, 2, ..., T\}$ do
34:         WalkTree($\emptyset$, 1, 1)
35:         WalkTree($\emptyset$, 2, 1)
36:     end for
```

Algorithm 3 provides pseudocode for the two-player version of our new probing algorithm that updates regrets according to our estimated counterfactual values. For more than two-players, an extra tree walk is required to update the cumulative profile as in Algorithm 2 and is omitted here. The Probe function recurses down the tree from history h following a single trajectory according to the known chance probabilities (line 7) and the current strategy obtained through Regret Matching (line 9, see equation (2.4)) until the utility at a single terminal history is returned (line 6). The WalkTree function is the same as the corresponding function of Algorithm 2, except at histories belonging to player i (lines 20 to 30). After sampling a set of actions (line 20), the value of each action a, $\hat{v}(a)$, is obtained. For each sampled action, we obtain its value by recursing down that action (line 24) after updating the sample probability for future histories (line 23). While MCCFR assigns zero value to each non-sampled action, Algorithm 3 instead obtains these action values through the probe function (line 26). Note that $q = q_i(I)$ is the probability of reaching I contributed from sampling player i's actions, and that the estimated counterfactual value $\hat{v}_i(I, \sigma_{(I \to a)}) = \hat{v}(a)/q$.

Note that $\hat{v}_i(I, \sigma)$ obtained through probing is not guaranteed to have lower variance than $\tilde{v}_i(I, \sigma)$ obtained through regular MCCFR under the same block Q. Probing could provide a less accurate update if the true counterfactual value of a non-sampled action is zero, but samples of the value have high variance. This is because MCCFR sets non-sampled values to zero, which could coincidentally be accurate estimates. However, our intuition suggests that probing should reduce variance in the majority of cases. Next, we show that probing can lead to faster convergence to equilibrium in zero-sum games.

5.3.2 Experiments

Here, we compare basic MCCFR to Algorithm 3 in two-player limit hold'em. To further validate probing, we compare the two algorithms in two additional domains, which we describe below. Experiments on these additional domains were performed by Marc Lanctot as part of our collaboration on this work. We focus on two-player games here so that we can measure the exploitability of the computed profiles.

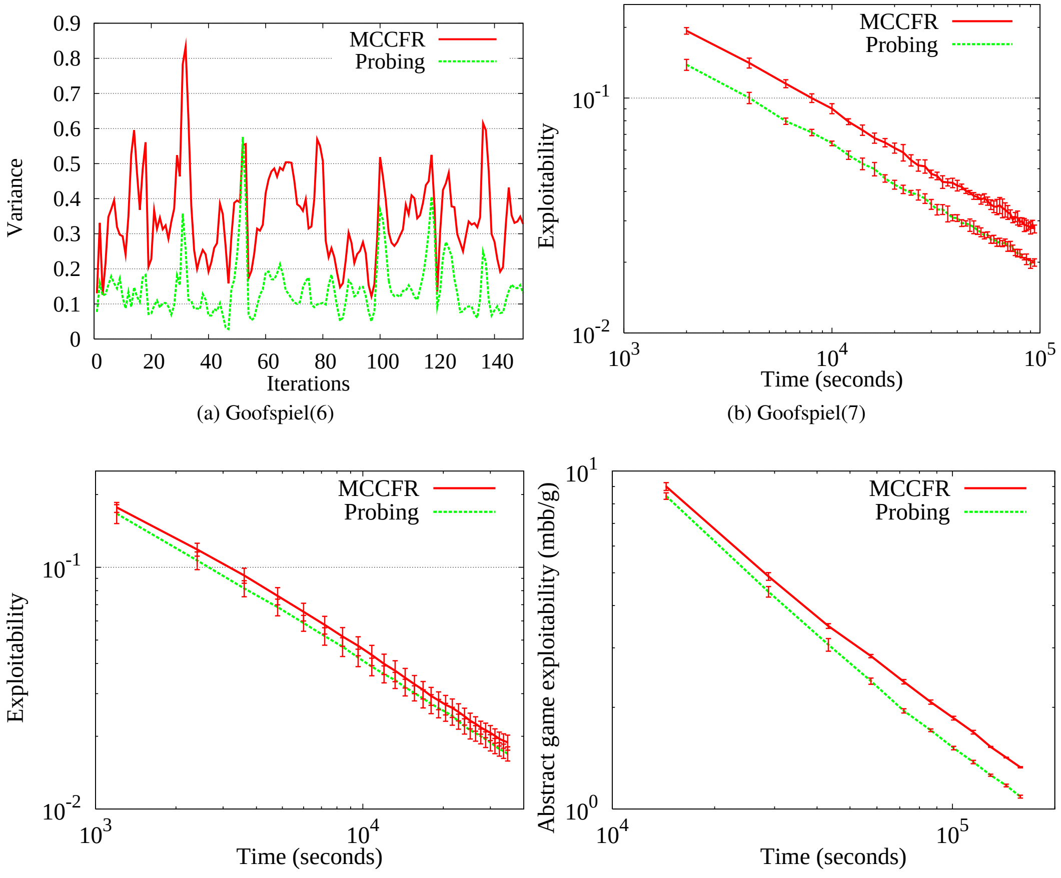
Goofspiel(n) is a two-player card-bidding game consisting of n rounds. Each player begins with a hand of bidding cards numbered 1 to n. In our version, on round k, players secretly and simultaneously play one bid from their remaining cards and the player with the highest bid receives $n - k + 1$ points; in the case of a tie, no points are awarded. The player with the highest score after n rounds receives a utility of $+1$ and the other player earns -1, and both receive 0 utility in a tie. Our version of Goofspiel is less informative than conventional Goofspiel as players know which of the previous bids were won or lost, but not which cards the opponent played.

Bluff(D_1, D_2) is a two-player dice-bidding game played over a number of rounds. Each player i starts with D_i six-sided dice. In each round, players roll their dice and look at the result without showing their opponent. Then, players alternate by bidding a quantity of a face value, q-f, of all dice in play until one player claims that the other is bluffing (*i.e.*, claims that the bid does not hold). To place a new bid, a player must increase q or f of the current bid. A face of 6 is considered *wild* and counts as any other face value. The player calling bluff wins the round if the opponent's last bid is incorrect, and loses otherwise. The losing player removes one of their dice from the game and a new round begins. Once a player has no more dice left, that player loses the game and receives a utility of -1, while the winning player earns $+1$ utility.

We use domain knowledge and our intuition to select the sampling schemes $\mathcal{Q}$. By our earlier assumption, we always sample a single action on-policy when $P(h) \neq i$, as is done in Algorithm 3. For the traversing player i, we focus on sampling actions leading to more "important" parts of the tree, while sampling other actions less frequently. Doing so updates the regret at the important information sets more frequently to quickly improve play at those locations. In Goofspiel, we always sample the lowest and highest bids, while sampling each of the remaining bids independently with probability 0.5. Strong play can be achieved by only ever playing the highest bid (giving the best chance at winning the bid) or the lowest bid (sacrificing the current bid, leaving higher cards for winning future bids), suggesting that these actions will often be taken in equilibrium. In Bluff(2,2), we always sample *bluff* and the bids 1-5, 2-5, 1-6, 2-6, and for each face f that we roll, k-f for all $1 \leq k \leq 4$. Bidding on the highest face is generally the best bluff since the opponent's next bid must increase the quantity, and bidding on one's own dice roll is more likely to be correct. Finally, in two-player limit hold'em, we always sample fold and raise actions, while sampling call with probability 0.5. Folds are cheap to sample (since the game ends) and raise actions increase the number of bets and consequently the magnitude of the utilities. In addition, we apply a 10s percentile hand strength squared abstraction that reduces the branching factor at each chance node down to ten, as described in Section 3.4.2. This abstract game contains roughly 57 million information sets.

Firstly, we performed a test run in Goofspiel(6) that measured the empirical variance of the samples $\tilde{v}_i(I, \sigma)$ provided by MCCFR and of $\hat{v}_i(I, \sigma)$ provided by Algorithm 3. During each iteration t of the test run, we performed 2000 traversals with no regret or strategy updates, where the first 1000 traversals computed $\tilde{v}_i(I, \sigma^t)$ and the second 1000 computed $\hat{v}_i(I, \sigma^t)$ at the root of the game. Both $\tilde{v}_i(I, \sigma^t)$ and $\hat{v}_i(I, \sigma^t)$ were computed under the same sampling scheme $\mathcal{Q}$ described above for Goofspiel. Once the empirical variance of each estimator was recorded from the samples at time t, a full Vanilla CFR traversal was then performed to update the regrets and acquire the next strategy σ^{t+1}. The empirical variances for $1 \leq t \leq 150$ are reported in Figure 5.3a. Since the estimators are unbiased, the variance here is also equal to the mean squared error of the estimates. Over 1000 test iterations, the average variances were 0.295 for MCCFR and 0.133 for Algorithm 3. This agrees with our earlier intuition that probing reduces variance and provides some validation for our choice of estimator.

Next, for each domain, we performed five runs for each of MCCFR and Algorithm 3, each under the same sampling schemes $\mathcal{Q}$ described above. For each domain, the average of the results are provided in Figures 5.3b to 5.3d. Our new algorithm converges faster than MCCFR in all three domains. In particular, at our final data points, Algorithm 3 shows a 31%, 10%, and 18% improvement over MCCFR in Goofspiel(7), Bluff(2,2), and two-player limit hold'em respectively. For both Goofspiel(7) and hold'em, the improvement was statistically significant. In Goofspiel(7), for example, the level of exploitability reached by MCCFR's last averaged data point is reached by Algorithm 3 in nearly half the time. To our knowledge, probing and our choice of $\mathcal{Q}$ is the fastest known algorithm for approximating equilibria in Goofspiel(7).



(a) Goofspiel(6)

(b) Goofspiel(7)

(c) Bluff(2,2)

(d) Two-player limit hold'em

Figure 5.3: **(a)** Empirical $\mathbf{Var}[\tilde{v}_i(I, \sigma^t)]$ and $\mathbf{Var}[\hat{v}_i(I, \sigma^t)]$ over iterations at the root of Goofspiel(6). **(b-d)** Log-log plots of exploitability over time of strategies computed by MCCFR and by Algorithm 3 using identical sampling schemes $\mathcal{Q}$, averaged over five runs. Error bars indicate 95% confidence intervals at each of the five averaged data points. In hold'em, exploitability is measured in terms of milli-big-blinds per game (mbb/g).

5.4 Average Strategy Sampling

While Algorithm 3 demonstrates a new technique outside of the MCCFR family, **Average Strategy Sampling (AS)** is a new MCCFR algorithm that is intended for games with many player actions. AS is inspired by Theorem 5.1 that states player i's regret is equal to the weighted sum of player i's cumulative counterfactual regrets at each $I \in \mathcal{I}_i$, where the weights are equal to player i's probability of reaching I under σ_i^* defined by equation (5.1). Since our goal is to minimize regret, this means that we only need to minimize the cumulative counterfactual regret at each $I \in \mathcal{I}_i$ that σ_i^* plays to reach. The more likely σ_i^* plays to reach I, the more important it is to minimize cumulative counterfactual regret at I. Therefore, AS intends to sample more often those information sets that σ_i^* plays to reach, and less often those information sets that σ_i^* avoids.

Unfortunately, we do not have the exact strategy σ_i^* on hand. Recall that in a two-player game,

σ_i^* is a best response to the opponent's average strategy, $\bar{\sigma}_{-i}^T$. In addition, for zero-sum games, the average profile $\bar{\sigma}^T$ converges to a Nash equilibrium. This means that player i's average strategy, $\bar{\sigma}_i^T$, converges to a best response of $\bar{\sigma}_{-i}^T$. While the average strategy is not an exact best response, we use it as a heuristic to guide sampling within AS. This is convenient since player i's average strategy can be directly obtained from the cumulative profile, $s(I, a)$, that CFR stores as in Algorithm 1.

AS selects actions for player i according to the cumulative profile and three predefined parameters. AS can be seen as a sampling scheme between OS and ES where a subset of player i's actions are sampled at each information set I, as opposed to sampling one action (OS) or sampling every action (ES). Given the cumulative profile $s_i^T(I, \cdot)$ on iteration T, an exploration parameter $\epsilon \in (0, 1]$, a threshold parameter $\tau \in [1, \infty)$, and a bonus parameter $\beta \in [0, \infty)$, each of player i's actions $a \in A(I)$ are sampled independently with probability

$$\rho(I, a) = \max\left\{\epsilon, \frac{\beta + \tau s_i^T(I, a)}{\beta + \sum_{b \in A(I)} s_i^T(I, b)}\right\}, \tag{5.4}$$

or with probability 1 if either $\rho(I, a) > 1$ or $\beta + \sum_{b \in A(I)} s_i^T(I, b) = 0$. As in ES, at opponent and chance nodes, a single action is sampled on-policy according to the current opponent profile σ_{-i}^T and the fixed chance probabilities σ_c respectively. Note that on the first iteration, $s_i^T(I, a) = 0$ and so all actions for player i are sampled with probability 1. Thus, the first iteration of AS is equivalent to ES. On later iterations, many actions may have a much lower probability of being sampled during AS.

The three parameters in AS have a variety of effects on the sampling probabilities. If $\tau = 1$ and $\beta = 0$, then $\rho(I, a)$ is equal to the probability that the average strategy $\bar{\sigma}_i^T = s_i^T(I, a) / \sum_{b \in A(I)} s_i^T(I, b)$ plays a at I, except that each action is sampled with probability at least ϵ. When $\tau > 1$, τ acts as a threshold so that any action taken with probability at least $1/\tau$ by the average strategy is always sampled by AS. In preliminary experiments, we found $\tau = 1$ to sample too few actions on each iteration, and that always sampling actions with significant probability by choosing $\tau > 1$ improved performance. Furthermore, β's purpose is to increase the rate of exploration during early AS iterations. When $\beta > 0$, we effectively add β as a bonus to the cumulative value $s_i^T(I, a)$ before normalizing. Since player i's average strategy $\bar{\sigma}_i^T$ is not a good approximation of σ_i^* for small T, we include β to avoid making ill-informed choices early-on. As the cumulative profile $s_i^T(I, \cdot)$ grows over time, β eventually becomes negligible. Alternatively, one can view β as a parameterization between "absolute" AS ($\beta = 0$) and ES ($\beta \to \infty$). Finally, when $\beta = 0$, ϵ is required to guarantee that sampled counterfactual values remain unbiased and ensures that the regret bound presented later in Theorem 5.5 is valid. For $\beta > 0$, ϵ maintains a minimum frequency at which actions are sampled, even after β becomes negligible. In Section 5.4.2, we present a set of values for ϵ, τ, and β that work well across all of the games we tested.

5.4.1 Pseudocode and Analysis

Algorithm 4 Average Strategy Sampling (Two-player version)

```
1:  Require: Parameters $\epsilon, \tau, \beta$
2:  Initialize regret: $\forall I, \forall a \in A(I) : R(I,a) \leftarrow 0$
3:  Initialize cumulative profile: $\forall I, \forall a \in A(I) : s(I,a) \leftarrow 0$
4:
5:  WalkTree(history $h$, player $i$, sample prob $q$):
6:      if $h \in Z$ then return $u_i(h)/q$ end if
7:      if $P(h) = c$ then Sample action $a \sim \sigma_c(h,\cdot)$, return WalkTree($ha$, $i$, $q$) end if
8:      $I \leftarrow I(h)$, $\sigma(I,\cdot) \leftarrow$ RegretMatching($R(I,\cdot)$)
9:      if $P(h) \neq i$ then
10:         for $a \in A(I)$ do $s(I,a) \leftarrow s(I,a) + (\sigma(I,a)/q)$ end for
11:         Sample action $a \sim \sigma(I,\cdot)$, return WalkTree($ha$, $i$, $q$)
12:     end if
13:     for $a \in A(I)$ do
14:         $\rho \leftarrow \max\left\{\epsilon, \frac{\beta + \tau s(I,a)}{\beta + \sum_{b \in A(I)} s(I,b)}\right\}$, $\tilde{v}(a) \leftarrow 0$
15:         if Random$(0,1) < \rho$ then $\tilde{v}(a) \leftarrow$ WalkTree($ha$, $i$, $q \cdot \min\{1, \rho\}$) end if
16:     end for
17:     for $a \in A(I)$ do $R(I,a) \leftarrow R(I,a) + \tilde{v}(a) - \sum_{b \in A(I)} \sigma(I,b)\tilde{v}(b)$ end for
18:     return $\sum_{a \in A(I)} \sigma(I,a)\tilde{v}(a)$
19:
20: Solve(iterations $T$):
21:     for $t \in \{1, 2, ..., T\}$ do
22:         WalkTree($\emptyset$, 1, 1)
23:         WalkTree($\emptyset$, 2, 1)
24:     end for
```

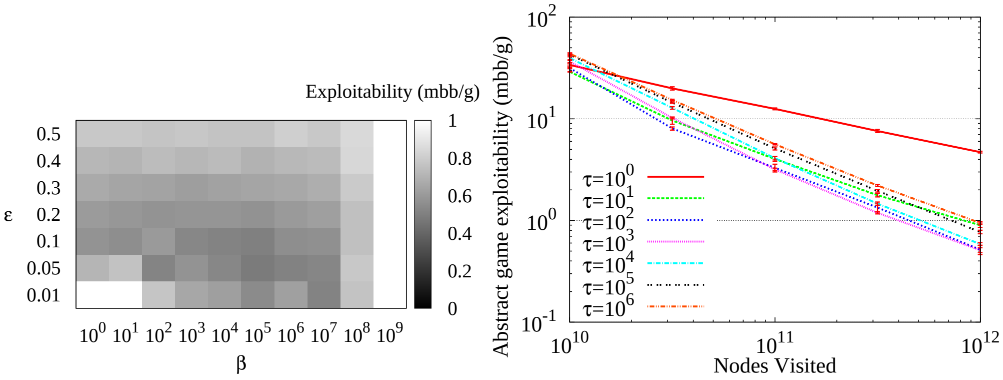
Pseudocode for a two-player version of AS is presented in Algorithm 4. Again, the WalkTree function is the same as in Algorithms 2 and 3, except when sampling actions for player i (lines 13 to 16). For each action a, we compute the probability ρ of sampling a and stochastically decide whether to sample a or not, where Random(0,1) returns a real number uniformly at random in $[0, 1)$. If we do sample a, then we recurse to obtain the sampled counterfactual value $\tilde{v}(a) = \tilde{v}_i(I, \sigma^t_{(I \to a)})$ (line 15). Otherwise, $\tilde{v}(a)$ is left as zero.

Running Solve(T) provides a probabilistic guarantee that all players' regret will be minimized. The bound is the same as that of Theorem 5.3 and is reported below for completeness.

Theorem 5.5. *Let $p \in (0, 1]$ and let $\delta = \min_{z \in Z} q_i(z) > 0$ over all $1 \leq t \leq T$. When using AS in a game with perfect recall, with probability $1 - p$, average regret is bounded by*

$$\frac{R_i^T}{T} \leq \left(M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|} + \frac{2\sqrt{|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}} \right) \left(\frac{1}{\delta} \right) \frac{\Delta_i}{\sqrt{T}}.$$

Note that δ in Theorem 5.5 is guaranteed to be positive for AS by the inclusion of ϵ in equation (5.4). However, for CS and ES, $\delta = 1$ since all of player i's actions are sampled, whereas $\delta \leq 1$ for OS and AS. While this suggests that fewer iterations of CS or ES are required to achieve the same regret bound compared to OS and AS, iterations for OS and AS are faster as they traverse less of the game tree. Just as CS, ES, and OS have been shown to benefit from this trade-off over Vanilla CFR, we will show that in practice, AS can likewise benefit over CS and ES and that AS is a better choice



(a) $\tau = 1000$

(b) $\epsilon = 0.05, \beta = 10^6$

Figure 5.4: **(a)** Abstract game exploitability of AS profiles for $\tau = 1000$ after 10^{12} nodes visited in 2-NL hold'em with $k = 30$ starting chips, where a darker colour represents lower exploitability and a lighter colour represents higher exploitability. **(b)** Log-log plot of abstract game exploitability over the number of nodes visited by AS with $\epsilon = 0.05$ and $\beta = 10^6$ in 2-NL hold'em, $k = 30$. For both figures, units are in milli-big-blinds per hand (mbb/g) and data points are averaged over five runs with different random seeds. Error bars in (b) indicate 95% confidence intervals.

than OS.

5.4.2 Experiments

We now compare the convergence rates of AS to those of CS, ES, and OS. While AS can be applied to any extensive-form game, the aim of AS is to provide faster convergence rates in games involving many player actions. Thus, we consider 2-NL hold'em, described in Section 3.4.3, and Bluff(D_1, D_2), described in Section 5.3.2, where we can easily scale the number of actions available to the players. Again, we focus on two-player games so that we can measure the exploitability of the computed profiles. The results for Bluff were again provided by Marc Lanctot in collaboration on this work. For the remainder of this section, let k be the starting stack sizes for both players in 2-NL hold'em. We also consider variations of Bluff where we use s-sided dice for various values of s rather than the usual six-sided dice. Note that 2-NL hold'em has up to k actions at an information set, whereas Bluff(D_1, D_2) has a maximum of $s(D_1 + D_2) + 1$ player actions.

Preliminary tests. Before comparing AS to CS, ES, and OS, we first run some preliminary experiments to find a good set of parameter values for ϵ, τ, and β to use with AS. All of our preliminary experiments are in two-player 2-NL hold'em with $k = 30$ chips. This time, we employ a 5s percentile hand strength squared abstraction that reduces the branching factor at each chance node down to five, as described in Section 3.4.2. Note that our preliminary experiments here are far from exhaustive and do not necessarily find the optimal set of parameters for AS. Nonetheless, our main results later in this section show that the set of parameters we do find work well enough to outperform CS, ES, and OS across multiple domains.

Firstly, we fix $\tau = 1000$ and test different values for ϵ and β. Recall that $\tau = 1000$ implies actions taken by the average strategy with probability at least 0.001 are always sampled by AS. Figure 5.4a shows the exploitability in the 5s abstract game, measured in milli-big-blinds per game (mbb/g), of the profile produced by AS after 10^{12} nodes visited. Each data point is averaged over five runs of AS. The $\epsilon = 0.05$ and $\beta = 10^5$ or 10^6 profiles are the least exploitable profiles (darkest colour) within statistical noise (not shown).

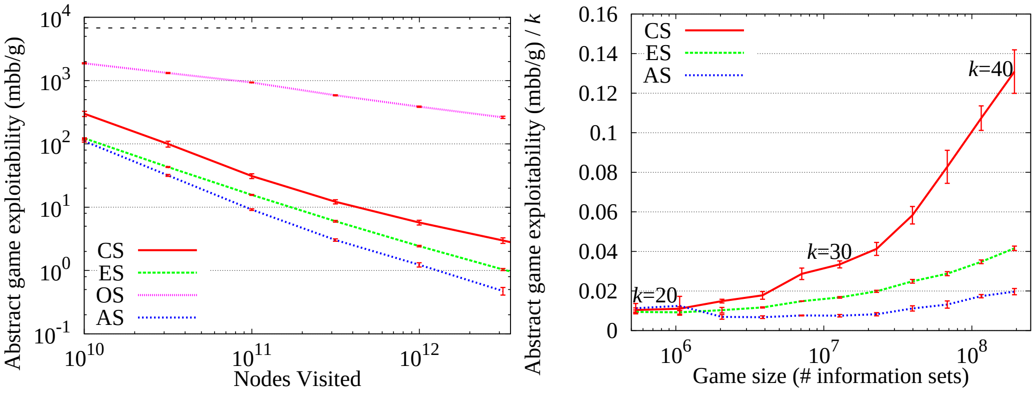
Next, we fix $\epsilon = 0.05$ and $\beta = 10^6$ and test different values for τ. Figure 5.4b shows the abstract game exploitability over the number of nodes visited by AS, where again each data point is averaged over five runs. Here, the least exploitable strategies after 10^{12} nodes visited are obtained with $\tau = 100$ and $\tau = 1000$ (again within statistical noise). Similar results to Figure 5.4b hold in 2-NL hold'em with $k = 40$ and are not shown.

Throughout the remainder of our experiments with AS, we use the fixed set of parameters $\epsilon = 0.05$, $\beta = 10^6$, and $\tau = 1000$. As can be seen in Figure 5.4, the performance of AS is not overly sensitive to the choice of parameters. Many other choices around this chosen set worked nearly as well in the preliminary experiments. Only for a few extreme choices of parameters ($\beta \geq 10^8$, $\epsilon = 0.01$ and $\beta \leq 10^1$, $\tau = 10^0$) did the performance of AS drastically worsen.

While we will not investigate parameter choices any further, we note here that there are some obvious cases where our choice of parameters may not be suitable. For example, for a game with up to millions of player actions at a single information set, choosing $\epsilon = 0.05$ would still result in at least tens of thousands of actions being sampled on average each iteration. It is likely that sampling this many actions will still be too costly and that better performance can be achieved by decreasing ϵ and τ. In addition, in games with large depth where each iteration is costly regardless of sampling, we may expect to run only a small number of iterations. For such games, $\beta = 10^6$ may never become negligible and result in AS behaving too much like ES. Here, a smaller value for β may be preferred.

Main results. We now compare AS to CS, ES, and OS in both 2-NL hold'em and Bluff(D_1, D_2). Similar to Lanctot *et al.* [54], our OS implementation is ϵ'-greedy so that the current player i samples a single action at random with probability $\epsilon' = 0.5$, and otherwise samples a single action according to the current strategy σ_i.

Firstly, we consider two-player 2-NL hold'em with starting stacks of $k = 20, 22, 24, ..., 38$, and 40 chips, for a total of eleven different 2-NL hold'em games. Again, we apply the same 5s abstraction as before to keep the games reasonably sized. For each game, we ran each of CS, ES, OS, and AS five times, measured the abstract game exploitability at a number of checkpoints, and averaged the results. Figure 5.5a displays the results for $k = 36$, a game with approximately 68 million information sets and 5 billion histories (nodes). Here, AS achieved an improvement of 54%



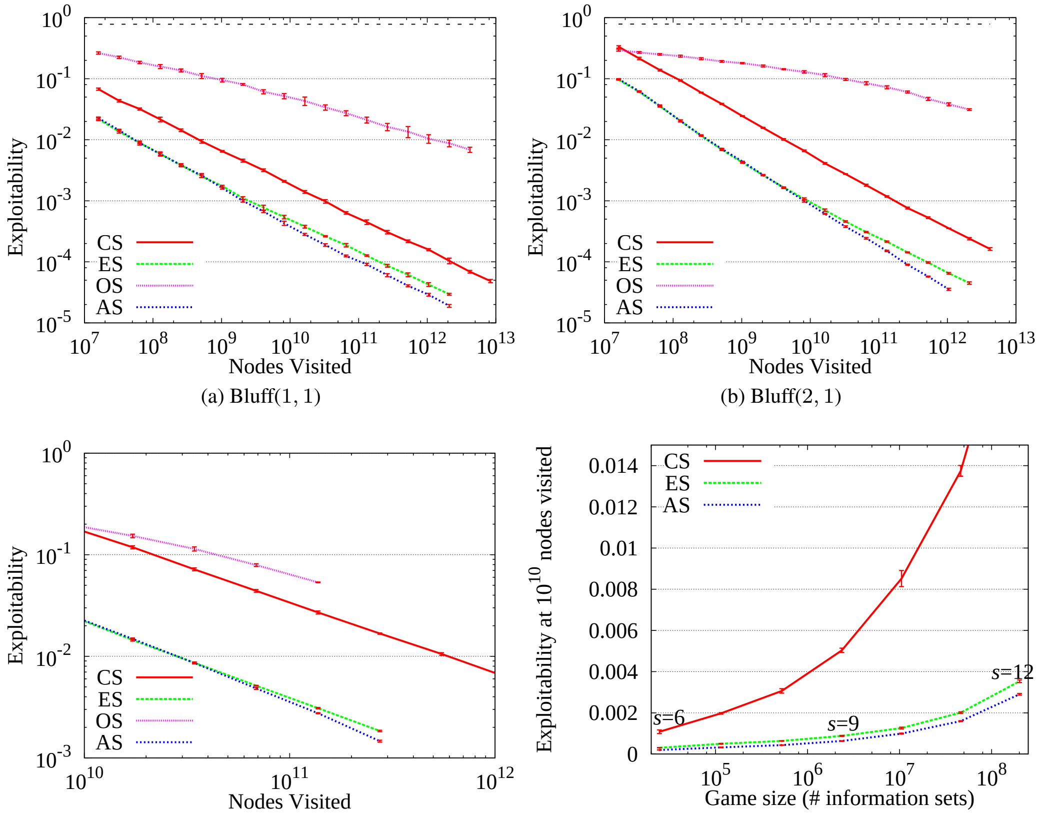
(a) 2-NL hold'em, $k = 36$

(b) 2-NL hold'em, $k \in \{20, 22, ..., 40\}$

Figure 5.5: **(a)** Log-log plot of abstract game exploitability over the number of nodes visited by CS, ES, OS, and AS in 2-NL hold'em with $k = 36$ starting chips. The initial uniform random profile is exploitable for 6793 mbb/g, as indicated by the black dashed line. **(b)** Abstract game exploitability after approximately 3.16×10^{12} nodes visited over the game size for 2-NL hold'em with even-sized starting stacks k between 20 and 40 chips. For both graphs, units are in milli-big-blinds per game (mbb/g) and data points are averaged over five runs with different random seeds. Error bars indicate 95% confidence intervals. For (b), units on the y-axis are normalized by dividing by the starting chip stacks.

over ES at the final data points. In addition, Figure 5.5b shows the average exploitability in each of the eleven games after approximately 3.16×10^{12} nodes visited by CS, ES, and AS. OS performed much worse and is not shown. Since one can lose more as the starting stacks are increased (*i.e.*, Δ_i becomes larger), we "normalized" exploitability across each game by dividing the units on the y-axis by k. While there is little difference between the algorithms for the smaller 20 and 22 chip games, we see a significant benefit to using AS over CS and ES for the larger games that contain many player actions. For the most part, the margins between AS, CS, and ES increase with the game size.

Figure 5.6 displays similar results for various Bluff games. Figures 5.6a to 5.6c consider standard Bluff games with $s = 6$-sided dice in Bluff(1,1), Bluff(2,1), and Bluff(2,2). Again, AS converged faster than CS, ES, and OS in all three Bluff games. Finally, Figure 5.6d shows the exploitability after approximately 10^{10} nodes visited by AS, CS, and ES for various choices of s in Bluff(1,1). OS was again too poor to be shown. Similar to the 2-NL hold'em(k) results, there is more benefit to using AS as we increase the number of player actions by increasing the number of die faces. Note that the same choices of parameters ($\epsilon = 0.05$, $\beta = 10^6$, $\tau = 1000$) that worked well in 2-NL hold'em with $k = 30$ continued to outperform CS, ES, and OS in other 2-NL hold'em games and in Bluff.



(a) Bluff(1, 1)

(b) Bluff(2, 1)

(c) Bluff(2, 2)

(d) Bluff(1, 1), $s \in \{6, 7, ..., 12\}$

Figure 5.6: **(a-c)** Log-log plots of exploitability over number of nodes visited by CS, ES, OS, and AS in Bluff(1, 1), Bluff(2, 1), and Bluff(2, 2) with $s = 6$-sided dice. The initial uniform random profile is exploitable for 0.780 and 0.784 in Bluff(1, 1) and Bluff(2, 1) respectively, as indicated by the black dashed lines. **(d)** Exploitability after approximately 10^{10} nodes visited over the game size of Bluff(1,1) with s-sided dice for $6 \leq s \leq 12$. For all graphs, data points are averaged over five runs with different random seeds and error bars indicate 95% confidence intervals.

5.5 Pure CFR

Our third and final new sampling algorithm, Pure CFR, comes from Oskari Tammelin [67], a hobbyist poker programmer from Finland. Our contributions here connect Pure CFR with our theory derived in Section 5.2. We also provide an open-source C++ implementation of Pure CFR [23] that can be applied to a variety of poker games, including Kuhn Poker, Leduc, and Texas hold'em. Finally, in Section 5.6, we compare Pure CFR to other sampling algorithms in three different Texas hold'em games.

Unlike MCCFR algorithms and our probing algorithm from Section 5.3, **Pure CFR** does not sample blocks of terminal histories $Q \in \mathcal{Q}$. Instead, Pure CFR samples a *pure* strategy profile $\hat{s}^t$ from the current profile σ^t, and a *pure* chance distribution $\hat{s}^t_c$ from σ_c. Then, an iteration of Vanilla CFR is performed using $\hat{s}^t$ and $\hat{s}^t_c$ in place of σ^t and σ_c respectively. When the utilities of the game are integers, as they are in poker, all computations in Pure CFR can be done with integer arithmetic.

Algorithm 5 Pure CFR (Two-player version)

```
Initialize regret: $\forall I, \forall a \in A(I) : R(I,a) \leftarrow 0$
Initialize cumulative profile: $\forall I, \forall a \in A(I) : s(I,a) \leftarrow 0$

WalkTree(history $h$, player $i$):
    **if** $h \in Z$ **then return** $u_i(h)$ **end if**
    **if** $P(h) = c$ **then** Sample action $a \sim \sigma_c(h, \cdot)$, **return** WalkTree($ha$, $i$) **end if**
    $I \leftarrow I(h)$, $\sigma(I, \cdot) \leftarrow$ RegretMatching($R(I, \cdot)$)
    Sample action $a \sim \sigma(I, \cdot)$
    **if** $P(h) \neq i$ **then** $s(I,a) \leftarrow s(I,a) + 1$, **return** WalkTree($ha$, $i$) **end if**
    **for** $b \in A(I)$ **do** $\hat{v}(b) \leftarrow$ WalkTree($hb$, $i$) **end for**
    **for** $b \in A(I)$ **do** $R(I,b) \leftarrow R(I,b) + \hat{v}(b) - \hat{v}(a)$ **end for**
    **return** $\hat{v}(a)$

Solve(iterations $T$):
    **for** $t \in \{1, 2, ..., T\}$ **do**
        WalkTree($\emptyset$, 1)
        WalkTree($\emptyset$, 2)
    **end for**
```

This provides two benefits. Firstly, integer arithmetic can be faster than floating-point arithmetic. Secondly, this also allows us to store both the cumulative regret and the cumulative profile as integers. For our implementation, storing values as integers rather than as floating-point numbers reduces our memory cost by 50%. This means that for the same cost in memory, abstractions with twice as many buckets can be employed by Pure CFR compared to Vanilla CFR and MCCFR algorithms. As we saw in three-player limit Texas hold'em from Figure 4.7a, increasing the number of buckets can lead to better performing strategies.

More formally, given our current strategy profile $\sigma = \sigma^t$, we sample $\hat{s}$ from σ by independently assigning a single action at each information set I, where

$$\textbf{Prob}[\hat{s}(I) = a] = \sigma(I,a) \text{ for all } a \in A(I),$$

and for each $h \in H_c$,

$$\textbf{Prob}[\hat{s}_c(h) = a] = \sigma_c(h,a) \text{ for all } a \in A(h).$$

We then define the estimated counterfactual value at information set I and strategy σ for Pure CFR to be

$$\hat{v}_i(I,\sigma) = \sum_{z \in Z_I} u_i(z)\pi_{-i}^{\hat{s}}(z[I])\pi^{\hat{s}}(z[I],z).$$

The estimated cumulative counterfactual regret $\hat{R}_i^T(I,a)$ is then defined as in Section 5.2.

5.5.1 Pseudocode and Analysis

Pseudocode for a two-player version of Pure CFR is provided in Algorithm 5. Notice the similarities between Pure CFR and ES listed in Algorithm 2. Both algorithms traverse all actions at player i's nodes (line 10 of Algorithm 5) and traverse a single, sampled action at all other nodes (lines 6 and 9). In Pure CFR, there is no need to traverse other opponent or chance actions a at history h because the estimated counterfactual values following a at h are weighted by $\pi_{-i}^{\hat{s}}(ha)$, the probability of the opponents and chance reaching ha under $\hat{s}$. This probability is zero by definition for all $a \neq \hat{s}(I(h))$. Similar cutoffs can be utilized in Chance Sampling when an opponent plays an action with zero probability under the current strategy profile.

Pure CFR only differs from ES in how the cumulative profile and the cumulative regrets are updated (lines 9 and 11 respectively), as well as the counterfactual value returned. In both of these updates, the current profile σ is replaced with the sampled pure profile $\hat{s}$. This results in a simple increment of the cumulative profile. For the regret update, the estimated counterfactual value at I, $\hat{v}(I, \sigma)$, is now equal to $\hat{v}(a) = \hat{v}_i(I, \sigma_{(I \to a)})$, the estimated value after taking action a at I. Intuitively, this estimate will often have higher variance than the estimate $\tilde{v}_i(I, \sigma) = \sum_{a \in A(I)} \sigma_i(I, a)\tilde{v}_i(I, \sigma_{(I \to a)})$ provided by ES that weights over all actions at I rather than just one. However, Pure CFR's estimates remain bounded and unbiased:

Proposition 5.2. *In Pure CFR, $\hat{v}_i(I, \sigma)$ is a bounded, unbiased estimate of $v_i(I, \sigma)$.*

Proof. To start, note that $\hat{v}_i(I, \sigma) \leq \pi_{-i}^{\hat{s}}(I)\Delta_i \leq \Delta_i$, and thus $\hat{v}_i(I, \sigma)$ is bounded. To prove $\hat{v}_i(I, \sigma)$ is unbiased, we have

$$\begin{aligned}
\mathbf{E}_{\hat{s}}[\hat{v}_i(I, \sigma)] &= \mathbf{E}\left[\sum_{z \in Z_I} u_i(z)\pi_{-i}^{\hat{s}}(z[I])\pi^{\hat{s}}(z[I], z)\right] \\
&= \sum_{z \in Z_I} u_i(z)\mathbf{E}\left[\pi_{-i}^{\hat{s}}(z[I])\pi^{\hat{s}}(z[I], z)\right] \\
&= \sum_{z \in Z_I} u_i(z)\mathbf{E}\left[\prod_{\substack{ha \sqsubseteq z[I] \\ P(h) \neq i}} \mathbf{I}[\hat{s}(I(h)) = a] \prod_{z[I] \sqsubseteq ha \sqsubseteq z} \mathbf{I}[\hat{s}(I(h)) = a]\right] \\
&\quad \text{where } \mathbf{I}[\cdot] \text{ is the indicator function and } I(h) = h \text{ when } P(h) = c \\
&= \sum_{z \in Z_I} u_i(z) \prod_{\substack{ha \sqsubseteq z[I] \\ P(h) \neq i}} \mathbf{E}\left[\mathbf{I}[\hat{s}(I(h)) = a]\right] \prod_{z[I] \sqsubseteq ha \sqsubseteq z} \mathbf{E}\left[\mathbf{I}[\hat{s}(I(h)) = a]\right] \\
&\quad \text{by independence of action sampling for } \hat{s} \\
&= \sum_{z \in Z_I} u_i(z) \prod_{\substack{ha \sqsubseteq z[I] \\ P(h) \neq i}} \sigma(I(h), a) \prod_{z[I] \sqsubseteq ha \sqsubseteq z} \sigma(I(h), a) \\
&= \sum_{z \in Z_I} u_i(z)\pi_{-i}^{\sigma}(z[I])\pi^{\sigma}(z[I], z) \\
&= v_i(I, \sigma). \blacksquare
\end{aligned}$$

Proposition 5.2 and Theorem 5.4 provide a probabilistic guarantee that Pure CFR minimizes regret. In fact, we can derive an identical bound on the average regret to that achieved by both CS and ES.

Theorem 5.6. *Let $p \in (0, 1]$. When using Pure CFR in a game with perfect recall, with probability $1 - p$, average regret is bounded by*

$$\frac{R_i^T}{T} \leq \left(M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|} + \frac{2\sqrt{|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}} \right) \frac{\Delta_i}{\sqrt{T}}.$$

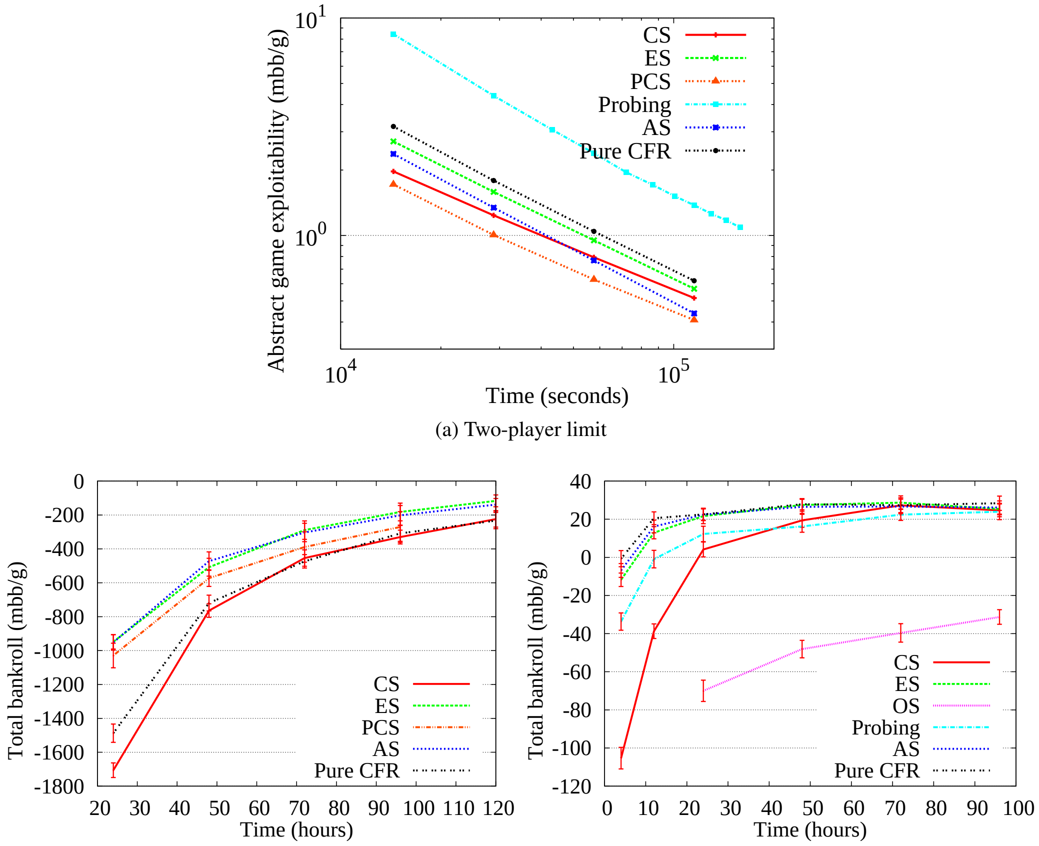
While the regret bounds presented here for Pure CFR and ES are the same, Pure CFR typically requires more iterations than ES in practice to reach the same solution quality. This increase in iterations is predicted by perhaps the more informative bound in Theorem 5.4 because we expect Pure CFR's estimates to have higher variance. Nonetheless, Pure CFR's lighter memory requirements make it an appealing choice for large games requiring abstraction, such as Texas hold'em.

5.6 Comparison of Algorithms in Texas Hold'em

In this chapter, we have presented a number of new CFR sampling algorithms and shown cases when our new algorithms have outperformed previous techniques. We now end this chapter by comparing the time efficiency of each of our new algorithms against previous MCCFR instances in an abstract game of two-player limit, of two-player no-limit, and of three-player limit Texas hold'em. In particular, we compare four previous MCCFR algorithms, CS, ES, OS, and PCS from Section 3.5.2, to our new algorithms AS, probing, and Pure CFR for a total of seven algorithms.

Firstly, we run each of the seven algorithms in two-player limit hold'em using the same 10s percentile hand strength squared abstraction from Section 5.3.2. Throughout this section, we use the same set of parameters for AS ($\epsilon = 0.05, \beta = 10^6, \tau = 1000$) that worked well in Section 5.4.2. Here and again later for three-player limit, the results for probing use the same sampling scheme derived from domain knowledge that was used in Section 5.3.2. These two-player limit results were generated from parallel implementations that used all 12 processors on machines with two six-core Intel Xeon X5650 2.66GHz processors and 48GB of RAM. The exploitability of the two-player limit hold'em strategies generated by each of the algorithms over time is shown in Figure 5.7a. OS was exploitable for 36.9 mbb/g after 32 hours of computation, which is too poor to be seen on the graph. While probing was shown to outperform MCCFR under a set of hand-chosen sampling blocks $\mathcal{Q}$ in Figure 5.3d, it appears that better sampling blocks $\mathcal{Q}$ for two-player limit hold'em are provided by the other algorithms. Recall that AS was designed specifically for games with many player actions, yet it surprisingly performs quite well here where there are a maximum of three player actions at any information set. However, PCS remains the most time-efficient algorithm for two-player limit hold'em, likely thanks to the $O(k^2)$ to $O(k)$ reduction described in Section 3.5.2.

Next, we move to two-player no-limit hold'em. For this experiment, we employ the abstraction used by the CPRG to win the no-limit instant run-off competition of the 2011 Annual Computer Poker Competition (ACPC). This abstraction limits the players' raise actions to a small number of choices relative to the current pot size. More specifically, players may raise a number of chips equal



(a) Two-player limit

(b) Two-player no-limit

(c) Three-player limit

Figure 5.7: **(a)** Log-log plot of exploitability over time achieved by CS, ES, PCS, probing, AS, and Pure CFR in a 10s abstraction of two-player limit Texas hold'em. **(b)** Performance over time for each of the sampling algorithms in an abstraction of two-player no-limit Texas hold'em. Error bars indicate 95% confidence intervals over 30 sets of duplicate matches in one-on-one play against the CPRG's no-limit program that won the 2011 instant run-off event of the ACPC. **(c)** Performance over time for each of the sampling algorithms in an abstraction of three-player limit Texas hold'em. Error bars indicate 95% confidence intervals over 10 sets of triplicate matches against each pair of opponents from the CPRG's 2009, 2010, and 2011 ACPC three-player instant runoff entries.

to the pot size, three times the pot size, eleven times the pot size, or may raise all-in. In addition, each player is allowed to make one raise equal to half the pot size and one raise equal to three-quarters of the pot size once per flop, turn, and river. Furthermore, an IR169/IR3700/IR3700/3700 card abstraction is employed using k-means clustering over earth mover's distance and OCHS (see Section 3.4.2).

The performances of the strategies generated over time by each of the sampling algorithms in this no-limit hold'em abstraction are shown in Figure 5.7b. We used the same parallel implementations across 12 processors and the same machines used in the two-player limit experiments. With imperfect recall abstractions and with larger betting spaces compared to two-player limit, we cannot easily measure abstract game exploitability here. Instead, performance is measured by playing 30 sets of 500,000 duplicate hand matches (each hand is played twice with agents switching positions) against the CPRG's 2011 two-player no-limit program that uses the same abstraction as our generated strategies. Again, OS performed too poorly to be seen in the graph, losing 7135 mbb/g after 120 hours of computation. Notice now that ES and AS outperform both CS and PCS as the latter two algorithms suffer from traversing all possible actions for both players on every iteration. While AS outperforms ES in 2-NL hold'em as seen in Figure 5.5, the betting abstraction employed here reduces the number of player actions to a maximum of eight per information set. This appears to be too few actions for AS to benefit over ES; however, AS and ES perform equally best in this game. Furthermore, Figures 5.7a and 5.7b show that despite performing similar tree walks to ES, Pure CFR is less time-efficient than ES as expected.

Finally, we perform a similar experiment in three-player limit hold'em. Here, we employ the same abstraction used for the experiments in Section 4.4 and for the CPRG's three-player entry in the 2012 ACPC. This abstraction is described in detail later in Section 8.3. We again measure performance by playing round robin competitions (RRCs) against the CPRG's competition entries from 2009, 2010, and 2011 as was done in Section 4.4. Performance over time for each of the generated strategies is shown in Figure 5.7c. These results were generated using 48 processors on machines with four Opteron AMD twelve-core processors at 2.2GHz and 256GB of RAM. Given our findings in Section 4.4, we use the current profiles here instead of the average profiles. Note that for AS, we must continue to store the cumulative profile to sample actions correctly, making AS less appealing in terms of memory efficiency. This time, PCS[1] performed too poorly to be seen, losing 169.8 mbb/g after 96 hours of computation. While ES, AS, and Pure CFR appear to be the most time-efficient algorithms here, the implementation of Pure CFR used only for the three-player experiments is actually a special, optimized routine. This Pure CFR implementation uses a number of technical tricks to reduce overhead that our current implementations of ES and AS still suffer from, making the comparison here somewhat unfair. These optimizations include storing a betting tree as opposed to applying function calls for faster tree walks and can be found in our Open Pure CFR implementation [23]. Nonetheless, Pure CFR reaches peak performance after about 48 hours of computation and still uses half the memory of ES (using integers instead of floating-point values) and a quarter the memory of AS (does not store the average profile).

5.7 Conclusion

In this chapter, we have established a number of improvements for computing strategies in extensive-form games through sampling with CFR, both theoretically and empirically. We have provided new, tighter bounds on the average regret when using Vanilla CFR or one of several different MCCFR sampling algorithms. In addition, we have provided a new theoretical framework that generalizes MCCFR and a new regret bound that depends on the variance of the estimates, suggesting estimates

[1] For PCS, we use an approximate terminal node evaluation function to perform $O(k^3)$ evaluations in $O(k)$ time. However, the effective k is likely smaller in the three-player case compared to the two-player case because player action sequences are longer on average and are therefore likely to provide private hand distributions with lower probabilities.

with lower variance are preferred. Furthermore, we have introduced three new sampling algorithms, AS, probing, and Pure CFR. Our experiments show that AS is often preferred in games with many player actions, that probing can reduce the variance of MCCFR estimates, and that Pure CFR can reduce memory requirements without a large penalty in time efficiency.

Chapter 6

CFR in Games with Imperfect Recall

So far, this dissertation has only considered extensive-form games with perfect recall, as defined in Definition 2.3. In games with perfect recall, players remember all information that was revealed to them throughout the game and the order in which that information was revealed. When perfect recall is assumed, CFR is guaranteed to minimize regret because a player's regret is bounded by the sum of the positive cumulative counterfactual regrets, as given by Theorem 2.3 and improved upon by Theorem 5.1. Unfortunately, these results are not guaranteed to hold in games with imperfect recall.

For example, consider again the extensive-form game shown in Figure 2.3a from Section 2.1.1. We will label player 2's information sets as $I = \{A, B\}$ and $J = \{Aa, Ab, Ba, Bb\}$. Let us consider the initial strategy profile $\sigma = \sigma^1$ where both player 1 and player 2 play actions uniformly at random at every information set. We can calculate player 2's counterfactual regret at J for action c according to

$$\begin{aligned}
r_2^1(J, c) &= v_2(J, \sigma_{(J\to c)}) - v_2(J, \sigma) \\
&= \sum_{z\in Z} \pi_1^\sigma(z[J])\pi^\sigma(z[J]c, z)u_2(z) - \sum_{z\in Z} \pi_1^\sigma(z[J])\pi^\sigma(z[J], z)u_2(z) \\
&= [0.5 \cdot 1 \cdot 4 + 0.5 \cdot 1 \cdot 0 + ...] - [0.5 \cdot 0.5 \cdot 4 + 0.5 \cdot 0.5 \cdot 4 + 0.5 \cdot 0.5 \cdot 0 + ...] \\
&= 2 - 2 \\
&= 0.
\end{aligned}$$

Similarly, $r_2^1(J, d) = r_2^1(I, a) = r_2^1(I, b) = 0$. Thus, player 2 has zero counterfactual regret at both I and J. However, player 2's regret, $R_2^1 = \max_{\sigma_2' \in \Sigma_2} u_2(\sigma_1^1, \sigma_2') - u_2(\sigma^1)$, is positive. This is because the player 2 strategy σ_2' where $\sigma_2'(I, a) = 1$ and $\sigma_2'(J, c) = 1$ has expected utility $u_2(\sigma_1^1, \sigma_2') = 2$, whereas following σ_2^1 only has expected utility $u_2(\sigma^1) = 1$. Therefore, $R_2^1 = 1 > 0 = \max_{a' \in A(I)} R_2^1(I, a') + \max_{a' \in A(J)} R_2^1(J, a')$, showing that Theorems 2.3 and 5.1 may not hold in games with imperfect recall.

Imperfect recall brings about a number of additional complications. In games with perfect recall, every mixed strategy has a utility-equivalent behavioral strategy [51], as discussed in Section

2.1.1. While certain lossless imperfect recall games share this property [46], Section 2.1.1 gave a counterexample showing that this is not true for imperfect recall games in general. In addition, the decision problem of determining if a player can assure themself a certain payoff in an imperfect recall game is NP-complete [48].

On the other hand, imperfect recall games are more versatile than perfect recall games for modelling large real-world problems. While perfect recall requires all past information to be remembered, imperfect recall allows irrelevant information to be forgotten so that the size of the game is smaller. As CFR's memory requirements are linear in the size of the game, more games become feasible through imperfect recall. Despite the complications above, CFR has been shown to work well in practice when applied to imperfect recall abstractions of Texas hold'em [73], but there is currently no theory to suggest why this is so.

This chapter presents theoretical groundings for applying CFR to games exhibiting imperfect recall. We define a general class of imperfect recall games and provide a bound on CFR's regret in such games. For a subset of this class, CFR minimizes regret in the extensive-form game. Moreover, our results also provide regret guarantees when applying CFR to an abstract game, provided the abstract game belongs to our general class. We test our theory in a new game called *die-roll poker*. To the best of our knowledge, this work demonstrates the first theoretical results for CFR applied to extensive-form games with imperfect recall.

In this chapter, we only consider games without *absentmindedness* [60] so that players cannot reach the same information set twice in a single game. In other words, we assume that for all $i \in N$ and $h, h' \in H_i$,

$$h \sqsubseteq h', h \neq h' \Rightarrow I(h) \neq I(h'). \tag{6.1}$$

Note that every perfect recall game satisfies equation (6.1), but not every imperfect recall game does. The work in this chapter is joint work with Marc Lanctot, Neil Burch, Martin Zinkevich, and Michael Bowling [53] and much of this work is also found in Lanctot's dissertation [52]. Our main contribution here is the formal analysis and theoretical proofs presented in Section 6.2 and Appendix D.

6.1 Die-Roll Poker

We now introduce a game that we will use as a running example throughout this chapter. **Die-roll poker (DRP)** is a simplified two-player poker game that uses dice rather than cards. To begin, each player antes one chip to the pot. There are two betting rounds, where at the beginning of each round, players roll a private six-sided die. The game has imperfect information due to the players not seeing the result of the opponent's die rolls. During a betting round, a player may fold, call, or raise by a fixed number of chips, with a maximum of two raises per round. In the first round, raises are worth two chips, whereas in the second round, raises are worth four chips. If both players have not folded by the end of the second round, a showdown occurs where the player with the largest sum of their two dice wins all of the chips in the pot.

DRP is naturally a game with perfect recall; players remember the exact sequence of bets made and the exact outcome of each die roll from both rounds. However, consider an imperfect recall abstraction of DRP, **DRP-IR**, where at the beginning of the second round, both players forget their first die roll and only know the sum of their two dice. In other words, any two histories are in the same abstract information set of DRP-IR if and only if the sum of the player's private dice is the same and the sequence of betting is the same. DRP-IR has imperfect recall since histories that were distinguishable in the first round (for example, a roll of 1 and a roll of 4) are no longer distinguishable in the second round (for example, a roll of 1 followed by a roll of 5, and a roll of 4 followed by a roll of 2).

6.2 CFR with Imperfect Recall

In this section, we investigate the application of CFR to games with imperfect recall. We begin by showing that CFR minimizes regret for a class of games that we call *well-formed games*. We then present a bound on the average regret for a more general class of imperfect recall games that we call *skew well-formed games*.

6.2.1 Well-formed Games

For extensive-form games $\Gamma = \langle N, H, P, \sigma_c, u, \mathcal{I} \rangle$ and $\breve{\Gamma} = \langle N, H, P, \sigma_c, u, \breve{\mathcal{I}} \rangle$, we say that $\breve{\Gamma}$ is a **perfect recall refinement of Γ** if $\breve{\Gamma}$ has perfect recall and Γ is an abstraction of $\breve{\Gamma}$. The information available to players in $\breve{\Gamma}$ is never forgotten, and is at least as informative as the information available to them in Γ. For example, DRP is a perfect recall refinement of DRP-IR. Every game has at least one perfect recall refinement by simply making $\breve{\Gamma}$ a perfect information game by choosing $\breve{I} = \{h\}$ for all $\breve{I} \in \breve{\mathcal{I}}_i$. Furthermore, a perfect recall game is a perfect recall refinement of itself. For $I \in \mathcal{I}_i$, we define

$$\breve{\mathcal{P}}(I) = \{\breve{I} \mid \breve{I} \in \breve{\mathcal{I}}_i, \breve{I} \subseteq I\}$$

to be the set of all information sets in $\breve{\mathcal{I}}_i$ that are subsets of I. Note that our notion of refinement is similar to the one described by Kaneko and Kline [46]. Our definition differs in that we consider any possible refinement, whereas Kaneko and Kline consider only the coarsest such refinement.

We now define a well-formed game. Intuitively, a game is well-formed if for each information set $I \in \mathcal{I}_i$, the structures around each $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$ of some perfect recall refinement are isomorphic across four conditions. Recall that $X(h)$ is the sequence of information set, action pairs leading to history h, as defined in Section 2.1.1. In addition, recall that Z_I is the set of terminal histories containing a prefix in the information set I, and that $z[I]$ is that prefix. Note that $z[I]$ is well-defined by equation (6.1).

Definition 6.1. *For a game Γ and a perfect recall refinement $\breve{\Gamma}$, we say that Γ is a* ***well-formed game with respect to*** $\breve{\mathbf{\Gamma}}$ *if for all $i \in N$, $I \in \mathcal{I}_i$, $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$, there exists a bijection $\phi : Z_{\breve{I}} \to Z_{\breve{I}'}$ and constants $k_{\breve{I},\breve{I}'}, \ell_{\breve{I},\breve{I}'} \in [0, \infty)$ such that for all $z \in Z_{\breve{I}}$:*

(i) $u_i(z) = k_{\breve{I},\breve{I}'} u_i(\phi(z))$,

(ii) $\pi_c(z) = \ell_{\breve{I},\breve{I}'} \pi_c(\phi(z))$,

(iii) *In Γ, $X_{-i}(z) = X_{-i}(\phi(z))$, and*

(iv) *In Γ, $X_i(z[\breve{I}], z) = X_i(\phi(z)[\breve{I}'], \phi(z))$.*

We say that Γ is a ***well-formed game*** *if it is well-formed with respect to some perfect recall refinement.*

Conditions (i) and (ii) state that the corresponding utilities and chance frequencies at each terminal history are proportional. Condition (iii) asserts that the opponents can never distinguish the corresponding histories at any point in Γ. Finally, condition (iv) states that player i cannot distinguish between corresponding histories from $\breve{I}$ and $\breve{I}'$ until the end of the game.

Consider again DRP as a perfect recall refinement of DRP-IR. In DRP, the available actions are independent of dice outcomes, and the final utilities are only dependent on the final sum of the players' dice. Therefore, in DRP the utilities are equivalent between, for example, the terminal histories where player i rolled a 1 followed by a 5, and the terminal histories where player i rolled a 4 followed by a 2 (condition (i)). In addition, the chance probabilities of reaching each terminal history are equal (condition (ii)). Furthermore, the opponents can never distinguish between two isomorphic histories since player i's rolls are private (condition (iii)). Finally, in DRP-IR, player i never remembers the outcome of the first roll from the second round on (condition (iv)). Thus, DRP-IR is well-formed with respect to DRP, with constants $k_{\breve{I},\breve{I}'} = \ell_{\breve{I},\breve{I}'} = 1$.

Any perfect recall game is well-formed with respect to itself since $\breve{\mathcal{P}}(I) = \{I\}$, ϕ equal to the identity bijection, and $k_{\breve{I},\breve{I}'} = \ell_{\breve{I},\breve{I}'} = 1$ satisfies Definition 6.1. However, many imperfect recall games are also well-formed, with DRP-IR being one example.

We now show that CFR can be applied to any well-formed game to minimize regret. A sketch of the proof is described below, while a full proof is provided in Appendix D.

Theorem 6.1. *If Γ is well-formed with respect to $\breve{\Gamma}$, then the average regret in $\breve{\Gamma}$ for player i when using CFR in Γ is bounded by*

$$\frac{\breve{R}_i^T}{T} \leq \frac{\Delta_i K \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}},$$

where $K = \sum_{I \in \mathcal{I}_i} \max_{\breve{I},\breve{I}' \in \breve{\mathcal{P}}(I)} k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'}$.

Proof sketch. One can show that conditions (i) to (iv) of Definition 6.1 imply that the positive regrets are proportional between any two information sets in $\breve{\Gamma}$ that are merged in the well-formed game, Γ. In other words, for all $I \in \mathcal{I}_i$, $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$, and $a \in A(I)$,

$$R_i^{T,+}(\breve{I}, a) = k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'} R_i^{T,+}(\breve{I}', a).$$

Since regrets between Γ and $\breve{\Gamma}$ are additive, *i.e.*,

$$R_i^T(I, a) = \sum_{\breve{I} \in \breve{\mathcal{P}}(I)} R_i^T(\breve{I}, a) \text{ for all } I \in \mathcal{I}_i,$$

the proportionality implies that minimizing regret at each $I \in \mathcal{I}_i$ minimizes regret at each $\breve{I} \in \breve{\mathcal{I}}_i$. Because $\breve{\Gamma}$ has perfect recall, applying Theorem 2.3 gives the result. ■

Since the strategy space is more expressive in $\breve{\Gamma}$ than in Γ ($\Sigma \subseteq \breve{\Sigma}$), $R_i^T \leq \breve{R}_i^T$ and thus it immediately follows that the average regret in Γ is minimized. In the case when Γ has perfect recall, because Γ is well-formed with respect to itself, Theorem 6.1 with $K = |\mathcal{I}_i|$ is a direct generalization of the original CFR bound [75, Theorem 4]. Note that unlike Theorem 2.4, we cannot incorporate the M-value of the game Γ into the bound of Theorem 6.1 since the M-value may not be well-defined in an imperfect recall game. Nonetheless, Theorem 6.1 not only guarantees regret minimization for perfect recall games, but also for well-formed imperfect recall games.

6.2.2 Skew Well-formed Games

We now present a generalization of well-formed games to which a regret bound can still be derived.

Definition 6.2. *For a game Γ and a perfect recall refinement $\breve{\Gamma}$, we say that Γ is a* ***skew well-formed game with respect to $\breve{\Gamma}$*** *if for all $i \in N$, $I \in \mathcal{I}_i$, $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$, there exists a bijection $\phi : Z_{\breve{I}} \to Z_{\breve{I}'}$ and constants $k_{\breve{I},\breve{I}'}, \delta_{\breve{I},\breve{I}'}, \ell_{\breve{I},\breve{I}'} \in [0, \infty)$ such that for all $z \in Z_{\breve{I}}$:*

(i) $\left| u_i(z) - k_{\breve{I},\breve{I}'} u_i(\phi(z)) \right| \leq \delta_{\breve{I},\breve{I}'}$,

(ii) $\pi_c(z) = \ell_{\breve{I},\breve{I}'} \pi_c(\phi(z))$,

(iii) *In Γ, $X_{-i}(z) = X_{-i}(\phi(z))$, and*

(iv) *In Γ, $X_i(z[\breve{I}], z) = X_i(\phi(z)[\breve{I}'], \phi(z))$.*

We say that Γ is a ***skew well-formed game*** *if it is skew well-formed with respect to some perfect recall refinement.*

The only difference between Definitions 6.1 and 6.2 is in condition (i). While utilities must be exactly proportional in a well-formed game, in a skew well-formed game they must only be proportional up to a constant $\delta_{\breve{I},\breve{I}'}$. Note that any well-formed game is skew well-formed by setting $\delta_{\breve{I},\breve{I}'} = 0$.

For example, consider a new version of DRP called **Skew-DRP($\boldsymbol{\delta}$)** with slightly modified payouts at the end of the game. Whenever the game reaches a showdown, player 1 receives a bonus

Table 6.1: DRP game sizes, where $\mathcal{A} = \{(I,a) : i \in N, I \in \mathcal{I}_i, a \in A(I)\}$ is the set of all information set, action pairs in the game.

Game	$\lvert\mathcal{A}\rvert$	Savings
Skew-DRP(δ)	2610	—
Skew-DRP-IR(δ)	860	67.05%

δ times the number of chips in the pot from player 2 if player 1's second die roll was even; otherwise, no bonus is awarded. The pot is then awarded to the player with the highest dice sum as usual. Analogously, define **Skew-DRP-IR($\boldsymbol{\delta}$)** to be the imperfect recall abstraction of Skew-DRP(δ) where in the second round, players only remember the sum of their two dice. Now, Skew-DRP-IR(δ) is not well-formed with respect to Skew-DRP(δ). To see this, note that the utilities resulting from the rolls 1,5 and the rolls 4,2 and the same sequence of betting are not exactly proportional because the second roll 5 is odd but 2 is even (utilities are off by δ times the pot size). However, Skew-DRP-IR(δ) is skew well-formed with respect to Skew-DRP(δ) with $\delta_{\breve{I},\breve{I}'} = \delta$ times the maximum pot size attainable from I.

Unfortunately, there is no guarantee that regret will be minimized by CFR in a skew well-formed game. However, we can still bound regret in a predictable manner according to the degree in which the utilities are skewed:

Theorem 6.2. *If Γ is skew well-formed with respect to $\breve{\Gamma}$, then the average regret in $\breve{\Gamma}$ for player i when using CFR in Γ is bounded by*

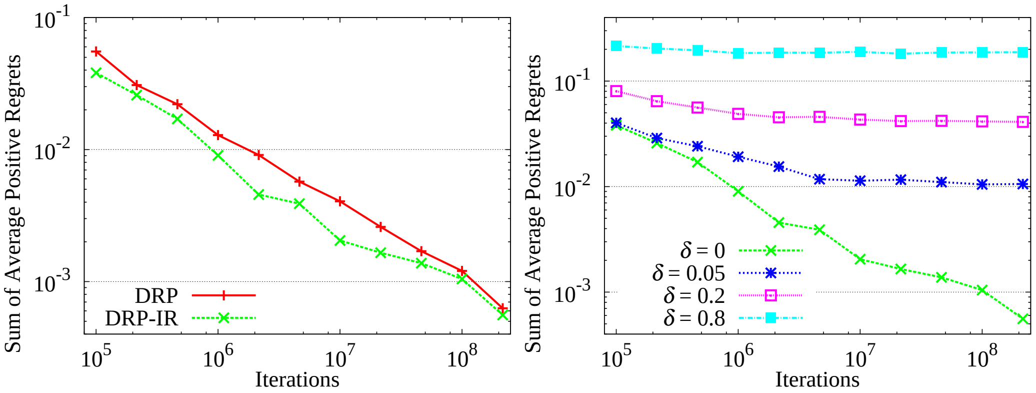
$$\frac{\breve{R}_i^T}{T} \le \frac{\Delta_i K \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} + \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| \delta_I,$$

where $K = \sum_{I \in \mathcal{I}_i} \max_{\breve{I},\breve{I}' \in \breve{\mathcal{P}}(I)} k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'}$ and $\delta_I = \max_{\breve{I},\breve{I}' \in \breve{\mathcal{P}}(I)} \delta_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'}$.

The proof is similar to that of Theorem 6.1 and can be found in Appendix D. Theorem 6.2 shows that as T approaches infinity, the bound on our regret approaches $\sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| \delta_I$. Our experiments in Section 6.3 demonstrate that as the skew δ grows, so does our regret in Skew-DRP-IR(δ) after a fixed number of iterations.

6.2.3 Remarks

Theorems 6.1 and 6.2 are, to our knowledge, the first results to provide theoretical guarantees in imperfect recall settings. However, these results are also relevant with regards to regret in the full game when CFR is applied to an abstraction. Recall that if Γ has perfect recall, then Γ is a perfect recall refinement of any (skew) well-formed abstract game. Thus, if we choose an abstraction that yields a (skew) well-formed game, then applying CFR to the abstract game achieves a bound on the average regret *in the full game*, Γ. This is true regardless of whether the abstraction exhibits perfect recall or imperfect recall. Previous counterexamples show that abstraction in general provides no



(a) DRP vs. DRP-IR

(b) Skew-DRP-IR(δ)

Figure 6.1: Sum of average positive regrets for both players, $(\breve{R}_1^{T,+} + \breve{R}_2^{T,+})/T$, as iterations increase for DRP, DRP-IR, and Skew-DRP-IR(δ). Both figures are log-log plots.

guarantees in the full game [72]. In contrast, our results show that applying CFR to an abstract game leads to bounded regret in the full game, provided we restrict ourselves to (skew) well-formed abstractions. If such an abstract game is much smaller than the full game, a significant amount of memory is saved when running CFR. This suggests that when using abstraction, we should construct an abstract game that is (skew) well-formed whenever possible to guarantee regret minimization in the original, larger game.

Unfortunately, the poker abstractions for Texas hold'em described in Section 3.4.2 and used throughout this dissertation are neither well-formed nor skew well-formed with respect to the original game. Much of the complication arises from cards being dealt without replacement. This allows players to infer a small amount of information about the opponents' hands from their own private cards. Many of our card abstractions are of the form *IR169/X/Y/Z* where information about the players' starting hands may be forgotten on later rounds. This information could be (mildly) retained by an opponent, breaking condition (iii) of Definitions 6.1 and 6.2. Furthermore, some of this information could be re-remembered on future rounds depending on well or how poorly the additional board cards play with the private cards, breaking condition (iv). We leave further analysis of the imperfect recall abstractions that we use for poker as future work.

6.3 Experiments

To complement our theoretical results, we run CFR in both DRP and Skew-DRP-IR(δ) for various skews δ and measure the sum of the average positive regrets for both players in the perfect recall refinements DRP and Skew-DRP(δ) respectively. As shown in Table 6.1, Skew-DRP-IR(δ) abstracts Skew-DRP(δ) to roughly 33% of the size of Skew-DRP(δ). Note that Skew-DRP(δ) is the same size as DRP (Skew-DRP(0)) regardless of the skew, and recall that CFR requires space linear in the size of the game.

For ease of implementation, our experiments use Chance Sampling MCCFR (CS) described in Section 2.2.3. Firstly, Figure 6.1a compares the rate of regret minimization for CS applied to DRP to that applied to DRP-IR. In addition to the savings in memory provided by DRP-IR over DRP, regret is also less in DRP-IR than in DRP at any given time. This is reflected in Theorem 6.1. Since DRP-IR contains fewer information sets than DRP, the constant K in Theorem 6.1 is smaller for DRP-IR, suggesting fewer iterations are required in DRP-IR to reach a given solution quality. Secondly, we measure the sum of the average positive regrets attained in Skew-DRP-IR(δ) for $\delta \in \{0, 0.05, 0.2, 0.8\}$. These regrets are reported in Figure 6.1b. We see that as δ increases, so does the regret as predicted by Theorem 6.2, though $\sum_{I \in \mathcal{I}_i} \left| \breve{\mathcal{P}}(I) \right| \delta_I$ appears to be a very loose bound on the final regret.

6.4 Discussion

Well-formed games are described by four conditions provided in Definition 6.1. Recall that Koller and Megiddo [48] prove that determining a player's guaranteed payoff in an imperfect recall game is NP-complete. However, Koller and Megiddo's NP-hardness reduction creates an imperfect recall game that breaks conditions (i), (iii), and (iv) of Definition 6.1. In this section, we discuss the following question: for minimizing regret, how important is it to satisfy each individual condition of Definition 6.1?

Skew well-formed games and Theorem 6.2 show that one can relax condition (i) of Definition 6.1 and still derive a bound on the average regret. In addition, Waugh *et al.* [73] use imperfect recall abstractions of Texas hold'em that do not satisfy condition (iii), but CFR still produces reliable solutions. This suggests that it may be possible to relax condition (iii) in a similar manner to the relaxation of condition (i) introduced by skew well-formed games. While we leave this question open, we now demonstrate that breaking condition (iii) can lead CFR to a dead-lock situation where one player has constant average regret.

Let us walk through the process of applying CFR to the game in Figure 6.2a. Note that this game satisfies all of the conditions of Definition 6.1, except for condition (iii) as player 2 can distinguish information forgotten by player 1. To begin, the current strategy profile σ^1 is set to be uniform random at every information set. Under this profile, when player 1 is at I_3, each of the four histories are equally likely. Thus, $v_i(I_3, \sigma^1_{(I_3 \to l)}) = v_i(I_3, \sigma^1_{(I_3 \to r)}) = v_i(I_3, \sigma^1) = 0$, and so $r^1_1(I_3, l) = r^1_1(I_3, r) = 0$. Similarly, under σ^1, the counterfactual value of the pass (p) and continue (c) actions at both I_1 and I_2 are zero, and thus the counterfactual regrets at I_1 and I_2 on iteration 1 are also zero. Player 2, however, has positive counterfactual regret for passing at histories ac and ec (to always receive ξ utility) and for continuing at bc and de (to always avoid receiving $-\xi$ utility), and has negative counterfactual regret for continuing at ac and ec and for passing at bc and de. Therefore, the next profile σ^2 still has player 1 playing uniformly random everywhere, but player 2 now always

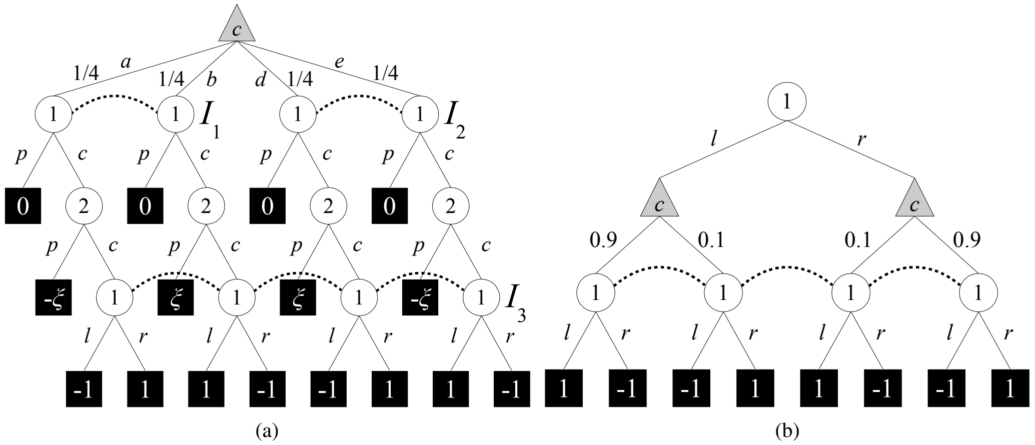


Figure 6.2: **(a)** A zero-sum game with imperfect recall where CFR does not minimize regret. The utilities for player 1 are given at the terminal histories, where $\xi \in (0, 1)$. **(b)** A single player game with imperfect recall where CFR does not minimize regret. In both games, nodes connected by a bold, dashed curve are in the same information set for player 1.

passes at ac and ec, and always continues at bc and dc. On the second iteration of CFR, the positive regrets for player 1 at I_3 remain the same because the histories bcc and dcc are equally likely. Also, player 2's positive regrets remain the same at all four information sets in $\mathcal{I}_2$. However, player 1's expected utility for continuing at I_1 or I_2 is now negative since player 2 now passes at ac and ec, and player 1 gains positive regret for passing at both I_1 and I_2. This leads us to the next profile $\sigma^3 = \{(I_1, p) = 1, (I_2, p) = 1, (ac, p) = 1, (bc, p) = 0, (dc, p) = 0, (ec, p) = 1, (I_3, l) = 0.5\}$. One can check that running CFR for more iterations yields $\sigma^t = \sigma^3$ for all $t \geq 3$. The average regret for playing this way will be constant and hence does not approach zero because player 1 would rather play $\sigma_1' = \{(I_1, p) = 1, (I_2, p) = 0, (I_3, l) = 0\}$ and get $u_1(\sigma_1', \sigma_2^3) = (1 - \xi)/4 > u_1(\sigma^3)$ for $\xi \in (0, 1)$. A similar example where condition (iii) holds, but chance's probabilities are not proportional (breaking condition (ii)) is given in Figure 6.2b. Here, CFR never plays anything but uniform random at both information sets. However, this strategy is strictly worse than an optimal strategy that always takes action l at both information sets.

Despite the problems of breaking conditions (ii) or (iii), condition (iv) of Definition 6.1 can be relaxed. Rather than enforcing player i's future information to be the same across the bijection ϕ, we only require that the corresponding subtrees be isomorphic, allowing player i to re-remember information that was previously forgotten. The details for this relaxation are in Appendix D. It is not clear, however, that this relaxation is possible in skew well-formed games, nor does it seem to provide any practical advantage.

6.5 Conclusion

This chapter has provided the first set of theoretical guarantees for CFR in imperfect recall games. We defined well-formed and skew well-formed games and provided bounds on the average regret that results from applying CFR to such games. In addition, our theory shows that we can achieve low average regret in a full, perfect recall game when employing CFR on an abstract version of the game, provided the abstract game is skew well-formed (with or without imperfect recall). Our experiments in DRP, DRP-IR, and Skew-DRP-IR(δ) confirm these theoretical results. Unfortunately, our theory developed here does not directly inform us about the quality of solutions generated by CFR when employing the abstractions for Texas hold'em described in Section 3.4.2.

Chapter 7

Strategy Stitching

As described in Section 2.4 and used throughout this dissertation, abstraction is often employed to tackle large extensive-form games that would otherwise be infeasible for CFR or other strategy computation techniques. Typically, and as evidenced by Figure 4.7a, strategies for the abstract game perform better in the real game as the granularity of abstraction is increased. For very large games, however, these abstractions need to be quite coarse, leaving many different information sets indistinguishable. In hold'em, this becomes problematic as the number of players increases; even just three-player limit hold'em has over 1000 times more information sets than two-player limit hold'em. Under current hardware with fixed memory limitations, card abstractions must resort to much fewer buckets in three-player hold'em than in the two-player game. On the other hand, if we partition the full game into smaller subtrees, strategies for the subtrees can be computed in much finer abstractions. Such *expert* strategies can then be pieced together, typically connecting to a *base strategy* computed in the full coarsely-abstracted game. As described in Chapter 1, we refer to this procedure as *strategy stitching*.

Strategy grafting [71], described in Section 2.4.1, is an example of strategy stitching. In strategy grafting, we convert a subset of one player's information sets into chance nodes and assign chance probabilities according to a precomputed base strategy. Other examples include the approaches used to construct PsOpti [5] and heads-up experts [3] in hold'em outlined in Section 3.5.4. Here, some of not just one, but of every player's information sets were converted to chance nodes to reduce the size of the games. As opposed to strategy grafting, a potential disadvantage of these two approaches is that the experts make assumptions about the other agents' strategies. In addition, for all of these approaches, computing the base strategy and the experts separately could suffer from loss of *cohesion* among the different components. In other words, if strong play requires coordination among the base strategy and the experts, then computing the different strategies in isolation could be detrimental.

In this chapter, we investigate stitched strategies in extensive-form games, focusing on the trade-offs between the sizes of the abstractions versus the assumptions made and the cohesion among the computed strategies. We define two strategy stitching techniques: (i) *static experts* that are computed in very fine abstractions with varying degrees of assumptions and little cohesion, and (ii) *dynamic experts* that are contained in abstractions with lower granularity, but make fewer assumptions and have perfect cohesion. This chapter generalizes previous strategy stitching efforts [3, 5, 71] under the more general static expert framework. We demonstrate in hold'em that, despite recent mixed results, static experts can create much stronger overall agents than the base strategy alone. Furthermore, we show that under a fixed memory limitation, a specific class of static experts are preferred to several alternatives because of the increase in granularity of abstraction allowed by the static approach. The techniques described in this chapter have been used to construct winning three-player limit Texas hold'em agents at the Annual Computer Poker Competition and is joint work with Duane Szafron [27].

7.1 Static Experts

As we just discussed, a natural approach to achieve abstractions with finer granularity is to break the game up into subtrees, abstract each of the subtrees independently, and compute a strategy for each abstract subtree. We now introduce a formalism for doing so that generalizes strategy grafting from Section 2.4.1 and the two poker-specific methods described in Section 3.5.4. First, select a subset $S \subseteq N$ of players. Secondly, for each $i \in S$, compute a base strategy σ_i for playing the full game using any strategy computation technique, such as CFR. Next, for each $i \in S$, choose a grafting partition G_i as given by Definition 2.7 so that each partition has an equal number of parts p. Then, compute a strategy, or *static expert*, for each subtree. Again, this can be done using CFR or any other appropriate method. Finally, since the subtrees are disjoint, create a *static expert strategy* by combining the static experts without any overlap to the base strategy in the undivided game. This process is outlined in the definition below:

Definition 7.1. *Let $S \subseteq N$ be a nonempty subset of players. For each $i \in S$, let σ_i be a strategy for player i and $G_i = \{G_{i,0}, G_{i,1}, ..., G_{i,p}\}$ be a grafting partition for player i. For $j \in \{1, 2, ..., p\}$, let Γ^j be the extensive-form game derived from the original game Γ where, for all $i \in S$ and $h \in H_i \backslash G_{i,j}$, we set $P(h) = c$ and $\sigma_c(h, a) = \sigma_i(h, a)$. That is, each player $i \in S$ only controls actions for histories in $G_{i,j}$ and is forced to play according to σ_i elsewhere. Let the* ***static expert*** *of $\{G_{i,j} \mid i \in S\}$, σ^j, be a strategy profile of the game Γ^j. Finally, define the* ***static expert strategy for player i****, σ_i^S, as*

$$\sigma_i^S(I(h), a) = \begin{cases} \sigma_i(I(h), a) & \text{if } h \in G_{i,0} \\ \sigma_i^j(I(h), a) & \text{if } h \in G_{i,j}. \end{cases}$$

We call $\{\sigma_i \mid i \in S\}$ the ***base strategies*** *and $\{G_i \mid i \in S\}$ the* ***grafting profile*** *for the static expert strategy σ_i^S.*

An illustrative summary of this process has already been provided in Section 2.4.1 by Figure 2.9.

Figure 7.1 shows an example of a game Γ^j from Definition 7.1, where $S = N$ and $j \in \{1, 2, ..., p\}$. This may be the only subtree for which static experts are computed ($j = p = 1$),

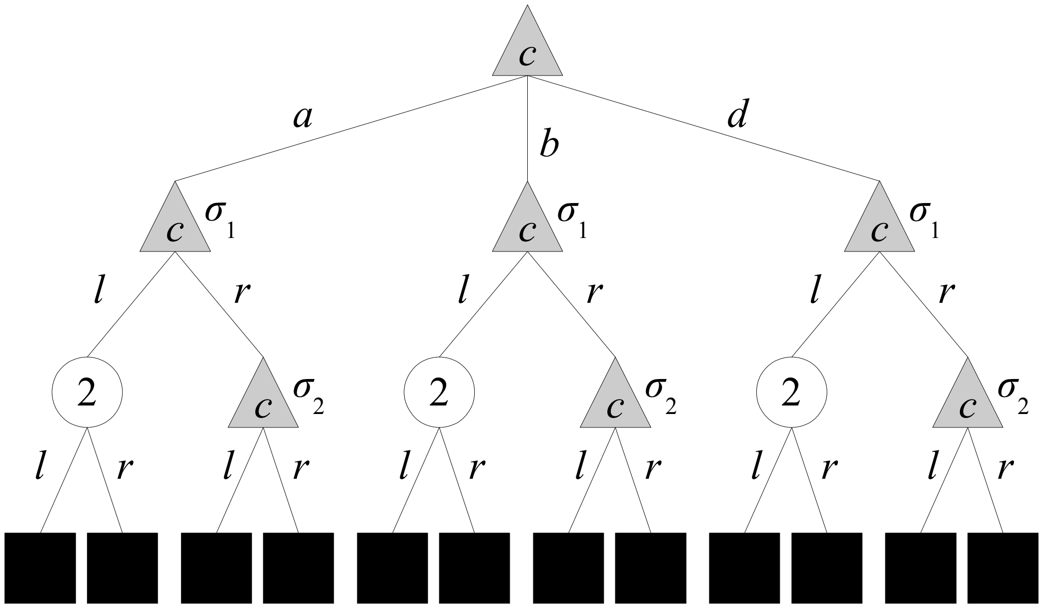


Figure 7.1: An example of a game Γ^j for a static expert derived from the game Γ in Figure 2.2. Here, $S = \{1, 2\}$, $G_{1,j} = \emptyset$, and $G_{2,j} = \{al, bl, dl\}$. All of player 1's actions are decided by the base strategy σ_1, computed beforehand. If player 1 takes action r, the base strategy σ_2 controls player 2's actions.

or there could be more subtrees contained in the grafting partitions ($p > 1$). Under a fixed memory limitation, we can employ finer abstractions for the subtrees Γ^j than we can in the full game Γ. This is because Γ^j removes some of the information sets belonging to players in S, freeing up memory for computing strategies on the subtrees.

Figure 7.2 provides another example of an extensive-form game Γ^j for some $j \in \{1, 2, ..., p\}$, this time derived from three-player Kuhn Poker shown in Figure 3.1. Here, $S = \{1, 3\}$ and we have chosen $G_{1,j} = \{h \in H_1 \mid akb \sqsubseteq h \text{ for some } a \in A(\emptyset)\}$ and $G_{3,j} = \{h \in H_3 \mid akb \sqsubseteq h \text{ for some } a \in A(\emptyset)\}$, where $A(\emptyset)$ is the set of all of chance's actions at the root of the game. The action probabilities for players 1 and 3 are fixed to the corresponding base strategies outside of the subtrees following a check (k) from player 1 and a bet (b) from player 2. Player 2, however, is unrestricted throughout the entire game. We call this a *kb static expert* game for three-player Kuhn Poker with $S = \{1, 3\}$.

When $|S| = 1$, the static expert approach is identical to strategy grafting given by Definition 2.8, with the exception that each static expert need not be an approximate Nash equilibrium. We relax the definition for static experts because Nash equilibria are difficult to compute in games with more than two players, and may not be the best solution concept outside of zero-sum games anyways. In addition, setting $S = N$ captures the expert construction processes from Section 3.5.4 of both Billings *et al.* in the PsOpti agents [5], and of Abou Risk and Szafron in three-player limit hold'em [3]. Following the naming convention for the three-player Kuhn expert game above, the seven post-

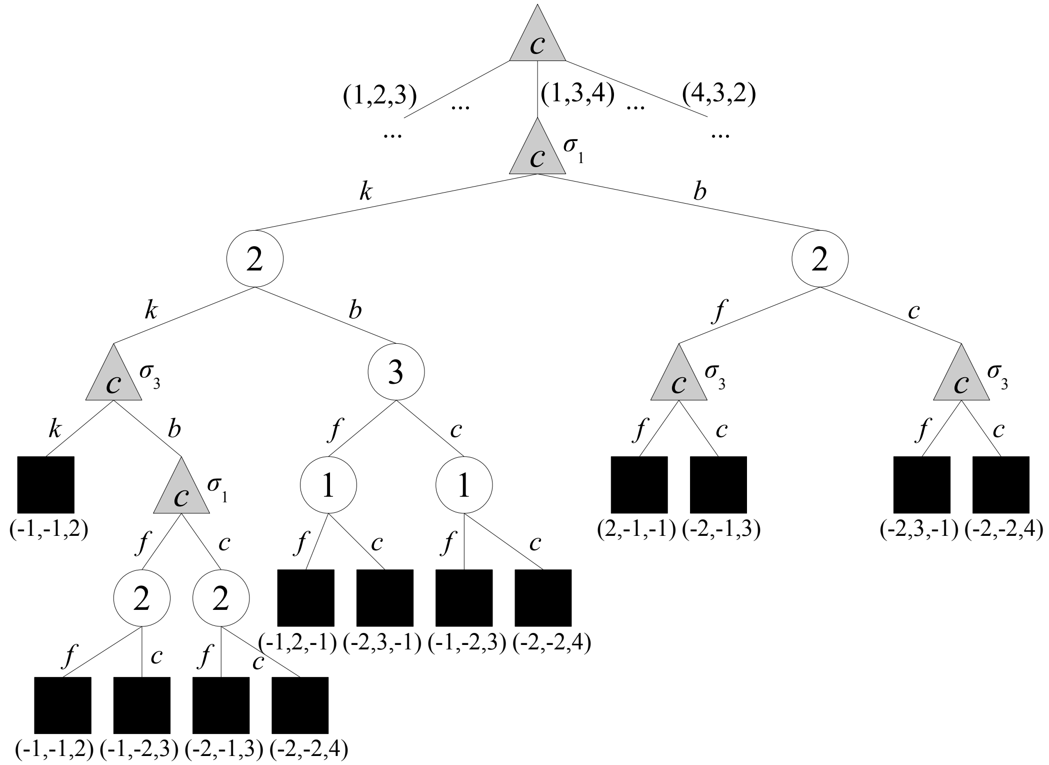


Figure 7.2: The kb static expert game for three-player Kuhn Poker with $S = \{1, 3\}$ derived from Figure 3.1.

flop models for PsOpti are ck, crc, $crrc$, $crrrc$, rc, rrc, and $rrrc$ static experts for two-player limit hold'em with $S = N$, where c denotes a call and r denotes a raise. The pre-flop model of PsOpti is used as a base strategy for the experts. As for the six heads-up experts in three-player limit hold'em, they are f, cf, rf, crf, rcf, and rrf static experts with $S = N$, where f denotes a fold. However, Abou Risk and Szafron did not define the new chance probabilities $\sigma_c(h, a)$ according to the base strategy used to play the rest of the game. Instead, they tested alternatives including a uniform random policy and distributions suggested by poker professionals.

Choosing $|S| > 1$ is potentially dangerous because we fix opponent probabilities and assume that our opponents are *static* at certain locations. For example, in Figure 7.2, it may not be wise for player 3 to assume that player 1 must follow σ_1 at the start of the round. Doing so can dramatically skew player 3's beliefs about the card held by player 1 and hurt the static expert's performance against opponents that do not follow σ_1. As we will see in Section 7.3, $|S| > 1$ can result in a more exploitable static expert strategy compared to $|S| = 1$.

On the other hand, by removing information sets for multiple players, the static expert approach results in fewer remaining information sets than strategy grafting does under the same set of grafting partitions. This can be seen by comparing the static expert game in Figure 7.1 with $S = N$ that contains three information sets, versus the static expert game in Figure 2.10 with $S = \{2\}$ that contains five information sets. As a result, in larger games, we can employ even finer abstractions within the subtrees. Section 7.3 later shows that despite the risks, the abstraction gains often lead to static experts with $S = N$ being preferred.

Regardless of the choice of S, the base strategy lacks cohesion with the static experts since the base strategy is computed prior to the existence of any experts. The base strategy may want to play towards the expert subtrees more often to increase utility. This observation motivates our next type of expert.

7.2 Dynamic Experts

While static experts are computed separately from a base strategy, *dynamic experts* are computed concurrently with a base. The full extensive-form game is divided into subtrees and each subtree is supplied its own abstraction. We then compute a strategy profile across all abstract subtrees, once again using any strategy computation technique desired. Our definition below is somewhat redundant to abstraction given in Definition 2.6 as we simply define a new abstract game derived from multiple, different abstractions. Nonetheless, we supply the definition below to provide the terms in bold that we will use throughout the remainder of this dissertation.

Definition 7.2. *Let $\alpha^0, \alpha^1, ..., \alpha^p$ be abstractions for the game Γ and for each $i \in N$, let $G_i = \{G_{i,0}, G_{i,1}, ..., G_{i,p}\}$ be a grafting partition for player i in Γ such that each abstract information set is contained entirely in some part of the grafting partition. In other words, for all $j \in \{0, 1, ..., p\}$ and $I \in \alpha_i^{j,\mathcal{I}}$, either $I \cap G_{i,j} = \emptyset$ or $I \subseteq G_{i,j}$. Let Γ' be the* ***dynamic expert abstract game*** *obtained from Γ by replacing $\mathcal{I}_i$ with $\bigcup_{j=0}^{p}\{I \in \alpha_i^{j,\mathcal{I}} \mid I \subseteq G_{i,j}\}$ and $A(h)$ with $\alpha_i^{j,A}(h)$ when $P(h) = i$ and $h \in G_{i,j}$, for all $i \in N$. Let the* ***dynamic expert strategy for player i****, σ_i', be a strategy for player i of the game Γ'. Finally define the* ***dynamic expert*** *of $G_{i,j}$, σ_i^j, to be σ_i' restricted to the histories in $G_{i,j}$, $\sigma_i'|_{G_{i,j}}$. The abstraction α^0 is denoted as the* ***base abstraction*** *and the dynamic expert σ_i^0 is denoted as the* ***base strategy****.*

An illustrative summary of this process is provided in Figure 7.3. The only difference between this process and the common procedure for computing a strategy profile outlined in Figure 2.7 is that we use multiple abstractions, rather than just one, to create our abstract game.

Figure 7.4 contains a dynamic expert abstract game tree Γ'. Using a similar naming convention used for static experts above, we call this an *l dynamic expert* because a finer-grained abstraction (the null abstraction) is used after player 1 takes the l action. We can view a dynamic expert strategy as a strategy computed in an abstraction with differing granularity dependent on the history of actions taken.

Under memory constraints, a dynamic expert strategy can sacrifice abstraction granularity in the base strategy to achieve finer granularity in the experts. We hope doing so achieves better performance at parts of the game that we believe may be more important. For instance, importance

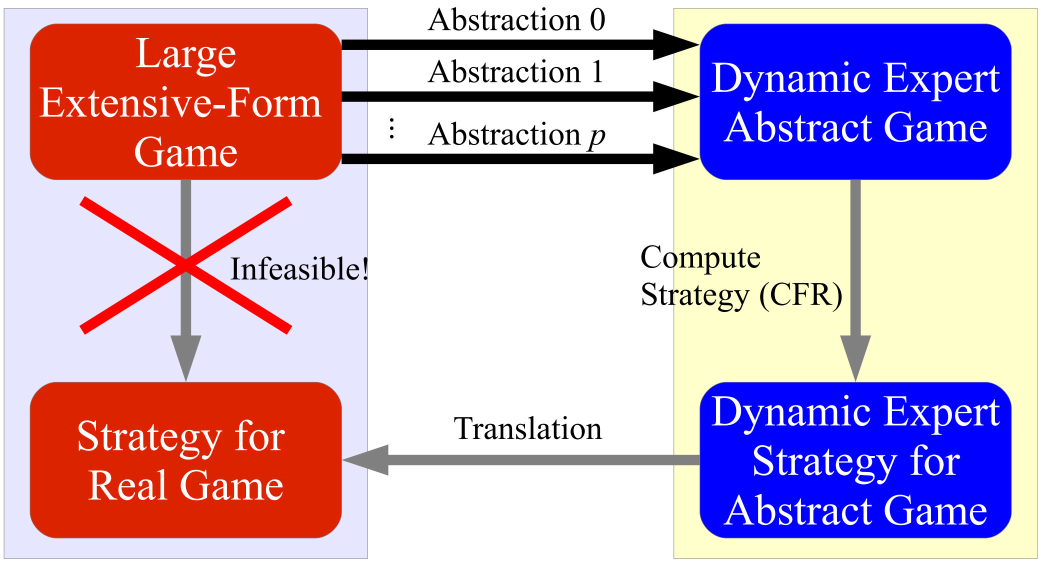


Figure 7.3: An overview of the process for creating a dynamic expert strategy for playing a large extensive-form game. We use different abstractions for different histories according to a partition of the game tree containing exactly $p + 1$ parts (not shown).

could depend on the predicted relative frequencies of reaching different subtrees, or on the predicted relative utilities of the subtrees. Compared to static experts, the base strategy of a dynamic expert profile has reduced abstraction granularity to guarantee perfect cohesion between the base and the experts; the base strategy knows about the experts and can calculate its probabilities dynamically during strategy computation based on the feedback from the experts. In Section 7.3, we contrast static and dynamic experts to compare this trade-off between abstraction size and strategy cohesion.

7.3 Experiments

In this section, we create several stitched strategies in both Leduc and hold'em, using Chance Sampling MCCFR (CS) for strategy computation. While previous experiments in Chapter 5 were concerned with the time-efficiency of different algorithms, here we only restrict our resources in terms of memory. Recall that Leduc, defined in Section 3.3, is a small poker game played with a six card deck over two betting rounds. Even though Leduc is small enough to not necessitate strategy stitching, we conduct experiments in Leduc to further evaluate our hypothesis that static experts with $S = N$ can improve play, and to compare the performances of static, grafted, and dynamic experts. For the remainder of this chapter, grafted experts refer to our static experts with $S = \{i\}$; in other words, our three-player grafted experts are simply computed using CS and are not necessarily approximate Nash equilibria of the corresponding expert games.

To be consistent with post-flop models [5] and heads-up experts [3], our grafting partitions are defined only in terms of the players' actions. To formalize the naming convention used in the previous sections, for each history $h \in H$, let $d = d(h)$ be the subsequence of h obtained by

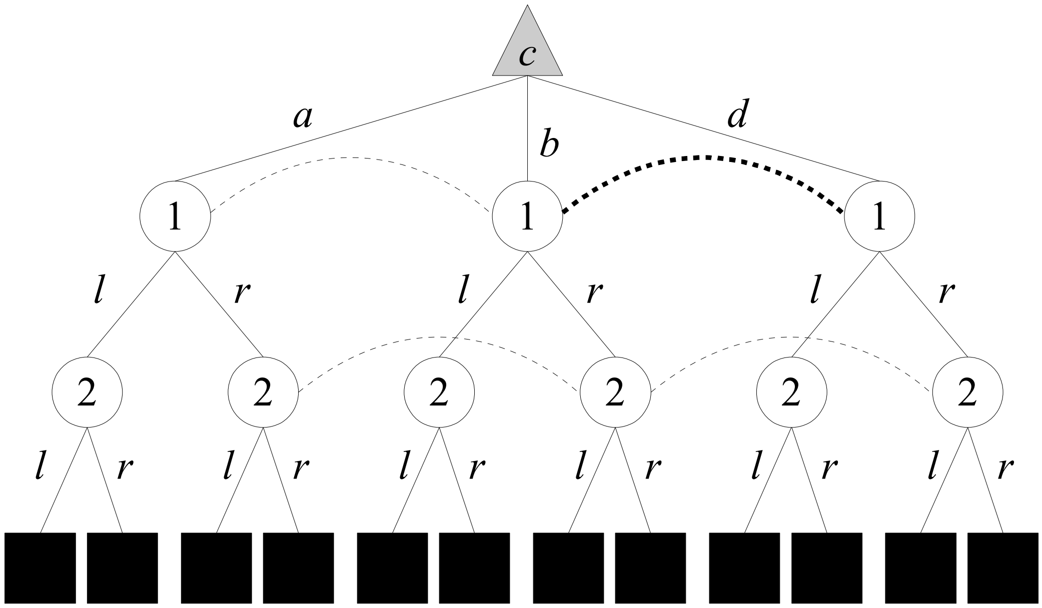


Figure 7.4: An example of a dynamic expert abstract game Γ' derived from the game Γ in Figure 2.2 and an alternative to the abstraction in Figure 2.8. Here, $p = 1$, α^1 is the null abstraction, and $G_{2,1} = \{al, bl, dl\}$. The base abstraction merges players' information sets into one abstract information set each.

removing all actions generated by chance. For a fixed sequence of actions d, we refer to a ***d*** **expert for player** ***i*** as an expert constructed for the subtree $G_i(d) = \{h \in H_i \mid d \sqsubseteq d(h)\}$ containing all histories where the players initially follow d.

We test two types of abstraction, *symmetric* and *asymmetric*, to employ within each expert j. Neither type employs any action abstraction. For a base abstraction α^0, a **symmetric abstraction for expert** ***j***, α^j, is such that for every player i, α_i^j has higher granularity than α_i^0 within the expert subtree. In particular, our symmetric abstractions use the same card bucketing scheme for all players. On the other hand, an **asymmetric abstraction for expert** ***j***, α^j, only increases granularity within the subtree for a single player i' so that $\alpha_i^{j,\mathcal{I}} = \alpha_i^{0,\mathcal{I}}$ for all $i \neq i'$. By only increasing the abstraction size for the player whose strategy we are after, our strategy can be more diverse than with symmetric abstractions under a fixed memory limitation. This is at the cost of being computed against opponent(s) in a coarse abstraction. For both symmetric and asymmetric abstractions, we leave the rest of the game that is outside of the expert subtree in the base abstraction α^0.

7.3.1 Leduc

Our Leduc experiments use three different perfect recall base abstractions for three corresponding base strategies. The first is simply the null abstraction where all cards can be distinguished from one another in both rounds. The second and third abstractions are the *JQ-K* and *J-QK* abstractions that, on the pre-flop, cannot distinguish between whether the private card is a Jack or Queen, or whether the private card is a Queen or King respectively. In addition, these two abstractions can only distinguish between whether the flop card pairs with the private card or not rather than knowing the identity of the flop card. Because Leduc is such a small game, we do not consider a fixed memory restriction and instead just compare the techniques within the same base abstraction. This also helps us better compare how much the static experts lose through lack of cohesion compared to the dynamic experts.

For each position in both two-player and three-player Leduc and for each of the three base abstractions, we build a base strategy, two dynamic expert strategies, two grafted strategies, and four static expert strategies with $S = N$. For two-player Leduc, we consider four expert subtrees: b, kb, kk/b, and kk/kb, where again k denotes the check action, b denotes the bet action, and / indicates a new betting round. We consider two types of static expert strategies. The *Pre-flop* static expert strategies only use the b and kb experts, whereas the *All* strategies use all four experts. Each of the dynamic and grafted strategies also employ all four experts. For three-player Leduc, we consider b, kb, kkb, kkk/b, kkk/kb, and kkk/kkb expert subtrees. The dynamic, grafted, and *All* static strategies use all six experts, whereas the *Pre-flop* static strategies use just the b, kb, and kkb experts. The null abstraction is employed on every expert subtree, either symmetrically or asymmetrically. All strategies are built from 100 million iterations of CS, which for full, unabstracted two-player Leduc converges to an ϵ-Nash equilibrium with ϵ less than one milli-ante per game.

Each strategy is evaluated against all combinations and orderings of opponent strategies, where the set of opponents is the set of all strategies that use a different base abstraction. The scores are then averaged across all of these opponents. For example, for each of our two-player strategy profiles σ in the JQ-K base abstraction, we compute $1/2(u_1(\sigma_1, \sigma'_2) + u_2(\sigma'_1, \sigma_2))$, averaged over all profiles σ' that use either the null or J-QK base abstraction. Leduc is a small enough game that the expected utilities can be computed exactly. These scores, along with two-player real-game exploitability values as defined in Section 2.1, are reported in Tables 7.1 and 7.2.

Firstly, the null base abstraction results demonstrate the value of cohesion between the base and expert strategies. In two-player, the grafted strategy suffers no more exploitability than the base strategy, which is guaranteed to converge to zero exploitability. Also, the Static.Preflop strategy performs just as well as the base strategy and only takes a minor hit in exploitability, while the Static.All strategy suffers slightly more in both measures. In three-player, the grafted strategy in the null abstraction surprisingly earns at least 20 milli-antes per game more than the corresponding base and static strategies. Overall, losing cohesion only hurts the grafted and static strategies by at most a small amount.

Secondly, by increasing abstraction granularity, nearly all of the expert strategies in the JQ-K and J-QK base abstractions earn more than the corresponding base strategy with no experts. Interestingly, of the JQ-K and J-QK base abstractions, the JQ-K strategies are generally more effective in two-player and the J-QK strategies are more effective in three-player. Within these two preferred

Table 7.1: The size, earnings, and exploitability in the real game of the two-player Leduc strategies. The sizes are measured in terms of the maximum number of information sets present within a single run of CS. The reported earnings, as described in the text, and exploitability are in milli-antes per game (ma/g).

Strategy	Abstraction Type	Size	Earnings (ma/g)	Exploitability (ma/g)
Null Base Abstraction				
Base / Dynamic	-	468	48	1
Grafted	-	468	46	1
Static.All	-	468	43	38
Static.Preflop	-	468	48	9
JQ-K Base Abstraction				
Base	-	132	39	496
Dynamic	symmetric	444	54	160
Grafted	symmetric	226	47	168
Static.All	symmetric	186	51	433
Static.Pre-flop	symmetric	186	48	214
Dynamic	asymmetric	288	-32	517
Grafted	asymmetric	159	38	587
Static.All	asymmetric	132	43	812
Static.Pre-flop	asymmetric	132	51	643
J-QK Base Abstraction				
Base	-	132	-166	740
Dynamic	symmetric	444	-8	124
Grafted	symmetric	226	-35	332
Static.All	symmetric	186	-89	609
Static.Pre-flop	symmetric	186	-48	335
Dynamic	asymmetric	288	-47	631
Grafted	asymmetric	159	-28	531
Static.All	asymmetric	132	-112	1119
Static.Pre-flop	asymmetric	132	-65	954

abstractions, static expert strategies earn more overall than grafted strategies, despite the two-player static strategies being more exploitable. The increased exploitability of the static strategies here is not surprising, though, and is likely due to the static opponent assumptions with $S = N$ described earlier in Section 7.1. Despite requiring much less memory to compute, the J-QK static strategies surprisingly earn more than the respective dynamic strategies in three-player Leduc. However, in the less effective base abstractions, the static strategies do worse. This is likely because the base strategy itself is poor and provides bad advice regarding the static opponent assumptions. Finally, we see that only using the two pre-flop static experts as opposed to all four reduces the number of dangerous assumptions to provide a less exploitable two-player strategy, and sometimes even a stronger overall strategy. However, as expected, the dynamic and grafted strategies are safer in terms of this worst case guarantee.

Table 7.2: The size and earnings of the three-player Leduc strategies. The sizes are measured in terms of the maximum number of information sets present within a single run of CS. The reported earnings, as described in the text, are in milli-antes per game (ma/g).

Strategy	Abstraction Type	Size	Earnings (ma/g)
Null Base Abstraction			
Base / Dynamic	-	6939	71
Grafted	-	6939	91
Static.All	-	6939	69
Static.Pre-flop	-	6939	68
J-QK Base Abstraction			
Base	-	1890	-185
Dynamic	symmetric	6903	-100
Grafted	symmetric	3017	-126
Static.All	symmetric	2145	-158
Static.Pre-flop	symmetric	2145	-155
Dynamic	asymmetric	3561	-134
Grafted	asymmetric	1977	-145
Static.All	asymmetric	1890	-163
Static.Pre-flop	asymmetric	1890	-158
J-QK Base Abstraction			
Base	-	1890	-67
Dynamic	symmetric	6903	105
Grafted	symmetric	3017	100
Static.All	symmetric	2145	107
Static.Pre-flop	symmetric	2145	110
Dynamic	asymmetric	3561	72
Grafted	asymmetric	1977	60
Static.All	asymmetric	1890	78
Static.Pre-flop	asymmetric	1890	81

7.3.2 Hold'em

Our hold'em experiments enforce a fixed memory restriction per run of CS, which we artificially set to 24 million information sets for two-player and 162 million information sets for three-player. Throughout this section, we use the definitions and notations for hold'em abstractions outlined in Section 3.4.2. We compute stitched strategies of each type using as many percentile $\mathbf{E}[\mathrm{HS}^2]$ buckets as possible within the restriction. Our two-player abstractions use perfect recall and distribute the buckets as close to uniformly as possible across the betting rounds. On the other hand, our three-player abstractions, both for the base strategy and the experts, all use imperfect recall with 169 pre-flop buckets that are forgotten on later rounds. Our three-player base abstractions additionally apply imperfect recall of buckets on all of the remaining rounds. As with the Leduc experiments, we again consider both symmetric and asymmetric abstractions for each type of expert strategy. Note that again, all of our strategies labelled *Static* are referring to static experts with $S = N$.

For two-player, our base strategy is contained in an 8s abstraction and is used to seed both our static and grafted strategies. Our static expert *Pre-flop* strategy employs r and cr experts, where c

Table 7.3: The abstraction sizes for the hold'em strategies. All abstractions use percentile $\mathbf{E}$[HS^2] bucketing. Each two-player abstraction contains no more than 24 million information sets, while each three-player abstraction contains no more than 162 million information sets.

Strategy	**Expert(s)**	**Symmetric Abstraction Size**	**Asymmetric Abstraction Size**
Two-Player			
Base	-	8/8/8/8	-
Dynamic	base	6/6/6/6	-
Dynamic	r	10/9/9/9	11/11/11/10
Grafted	r, cr	9/9/9/9	10/10/10/9
Grafted	ck/b, ck/kb	15/15/14/14	16/16/16/15
Static	r, cr	10/10/10/9	11/11/11/11
Static	ck/b, ck/kb	16/16/16/15	19/19/19/18
Three-Player			
Base	-	IR169/IR16/IR16/16	-
Dynamic	base	IR169/IR8/IR8/8	-
Dynamic	f, rf, rrf, rcf	IR169/90/9/9	IR169/121/11/11
Grafted	f	IR169/100/10/10	IR169/144/12/11
Grafted	rf	IR169/156/12/12	IR169/225/15/14
Grafted	rrf	IR169/196/13/13	IR169/272/16/16
Grafted	rcf	IR169/276/16/16	IR169/380/19/19
Static	f	IR169/169/13/12	IR169/240/15/15
Static	rf	IR169/256/16/16	IR169/361/19/19
Static	rrf	IR169/324/18/17	IR169/441/21/21
Static	rcf	IR169/441/21/21	IR169/625/25/25

denotes the call action and r denotes the raise action. Our grafted and static expert *All* strategies, on the other hand, use r, cr, ck/b, and ck/kb experts, and our dynamic strategy has just an r expert. These choices were based on preliminary experiments to make the most effective use of the limited memory available for each stitching technique. Similar to Abou Risk and Szafron, our three-player experts act only in subtrees after one player has folded. In particular, our static, grafted, and dynamic strategies all have an expert for each of the sequences f, rf, rrf, and rcf, where f denotes the fold action, as these appear to be the most commonly reached two-player scenarios [3, Table 4].

Our abstractions range quite dramatically in terms of number of buckets. For example, in three-player, our dynamic strategy's base abstraction has just 8 river buckets with 7290 river buckets for each expert in the symmetric abstraction, whereas our static and grafted strategies have 16 river buckets in the base abstraction with up to 390,625 river buckets for the static rcf expert in the asymmetric abstraction. The size of each percentile $\mathbf{E}$[HS^2] abstraction is provided in Table 7.3. All of the two-player base strategies and experts are built from 720 million iterations of CS, while we run CS for 100 million and 5 billion iterations for the three-player base strategies and experts respectively.

We evaluate our two-player strategy profiles by playing 500,000 duplicate hands (players play both sides of the dealt cards) of poker between each pair of profiles. In addition to our base and

Table 7.4: Earnings and 95% confidence intervals over 500,000 duplicate hands of two-player hold'em per pairing. The real game exploitability of the strategies is also provided. All values are in milli-big-blinds per game (mbb/g).

Strategy	**Abstraction Type**	**Earnings (mbb/g)**	**Exploitability (mbb/g)**
Base	-	-8 ± 2	310
Base.797M	-	33 ± 2	135
Dynamic	symmetric	-1 ± 2	308
Grafted	symmetric	-8 ± 2	301
Static.All	symmetric	-2 ± 2	288
Static.Pre-flop	symmetric	0 ± 2	289
Dynamic	asymmetric	0 ± 2	339
Grafted	asymmetric	-9 ± 2	316
Static.All	asymmetric	0 ± 2	314
Static.Pre-flop	asymmetric	-1 ± 2	312

Table 7.5: Earnings and 95% confidence intervals over 500,000 triplicate hands of three-player hold'em per combination. All values are in milli-big-blinds per game.

Strategy	**Abstraction Type**	**Earnings (mbb/g)**
Base	-	-9 ± 2
Hyperborean3p.IRO.2009	-	-16 ± 2
Hyperborean3p.IRO.2010	symmetric	27 ± 1
Dynamic	symmetric	-2 ± 2
Grafted	symmetric	-2 ± 2
Static	symmetric	5 ± 2
Dynamic	asymmetric	-3 ± 2
Grafted	asymmetric	-3 ± 3
Static	asymmetric	4 ± 3

stitched strategies, we also included a base strategy profile called *Base.797M* in an abstraction with over 797 million information sets that we expected to beat all of the strategies we were evaluating. Furthermore, using a specialized best response computation tool [44], we computed the exploitability of our two-player profiles. For three-player, we play 500,000 triplicate hands (each set of dealt cards played six times, one for each of the player orderings) between each combination of three strategy profiles. We also added to the pool of agents Hyperborean3p.IRO.2009 and Hyperborean3p.IRO.2010, the winners of the 2009 and 2010 Annual Computer Poker Competition instant run-off events. Hyperborean3p.IRO.2009 is a base strategy in an IR16/IR16/IR16/16 abstraction, whereas Hyperborean3p.IRO.2010 uses larger abstractions, static experts with $S = N$, and is described in detail later in Chapter 8. The results are provided in Tables 7.4 and 7.5.

Firstly, we see that the two-player static and dynamic strategies outperform the grafted strategies considerably, agreeing with the majority of the Leduc results. In fact, the grafted strategies fail to even improve upon the base strategy, which is somewhat surprising. For three-player, the static strategies are noticeably ahead of the dynamic and grafted strategies as the static strategies are the only competitors, aside from Hyperborean3p.iro.2010, to win money. By forcing one player to fold, the static experts essentially reduce the size of the game tree from a three-player to a two-player game, allowing many more buckets to be used. This result indicates that at least for poker, the gains in abstraction bucketing outweigh the risks of static opponent action assumptions and lack of cohesion between the base strategy and the experts. Further support for the use of static experts with $S = N$ comes from the two-player exploitability results, where the static strategies are slightly less exploitable in the real game than the grafted and dynamic strategies. However, the strategies computed here are not necessarily the least exploitable strategies for the real game that can be represented in the base and expert abstractions. Finding the least exploitable strategies is possible with CFR-BR [42] described in Section 2.4, but is not pursued here. Notice that in hold'em, the choice of symmetric versus asymmetric abstractions appears to matter little in overall performance. However, in two-player, all of the stitched strategies employing symmetric abstractions are less exploitable than the base and the stitched strategies employing asymmetric abstractions. In summary, the symmetric static expert strategies with $S = N$ are most preferred in the experiments we ran.

7.4 Conclusion

In this chapter, we formalized two strategy stitching techniques for extensive-form games. The static expert approach generalizes strategy grafting and some previous techniques used in poker, while dynamic experts offer a new way of interpreting special abstractions that vary in quality across a grafting profile. Despite the accompanying potential dangers and lack of cohesion, we have shown static experts with $S = N$ outperform the dynamic and grafted experts that we considered, especially when memory limitations are present. However, additional static experts with several forced actions can lead to a more exploitable strategy. Note that static and dynamic experts can also be combined by, for example, using a dynamic expert strategy as a base for a set of static experts. In fact, both static and dynamic experts have been used to construct winning three-player agents in the Annual Computer Poker Competition. These agents are described in detail in the following chapter.

Chapter 8

Results from Annual Computer Poker Competitions

Throughout this dissertation, much of our empirical analysis has involved computing strategy profiles for abstractions of Texas hold'em and playing these strategies in mock poker matches. To further validate our work, we have led the development of the CPRG's three-player Texas hold'em agents, nicknamed *Hyperborean3p*, that were entered in the 2010, 2011, 2012, and 2013 Annual Computer Poker Competitions (ACPC) [4].

The ACPC is an international event that attracts entrants from top universities and hobbyists from around the world. The competition aims to promote research and encourage new ideas and algorithms for problems arising in poker, such as strategy computation and abstraction. There are currently three games played in the ACPC: heads-up (two-player) limit, heads-up no-limit, and three-player limit Texas hold'em. As described in Section 4.4.1, there are currently two winner determination rules per game. Recall that the total bankroll (TBR) metric ranks agents by their total average winnings across all opponents, whereas the instant runoff (IRO) metric determines the winner by iteratively removing agents with the least earnings among the remaining competitors. Each team may enter one unique agent per game and per winner determination rule.

While the ACPC was first held in 2006, the three-player hold'em events were not included until 2009. In this first year of the three-player competitions, the CPRG entered two different agents. For the IRO event, the program entered was a strategy profile computed with Chance Sampling MCCFR (CS). Recalling our abstraction notation from Section 3.4.2, an IR16/IR16/IR16/16 percentile hand strength squared abstraction was used. For TBR, a static expert profile with $S = N$, as described in Section 7.1, was used that employed a 5s-sized abstraction for the experts and the IRO profile for the base [3]. While both entries won their respective divisions, these two agents are no match for the new agents that we present here.

In this chapter, we provide specific details of each three-player agent that we entered into the ACPC for the 2010, 2011, 2012, and 2013 competitions. These agents were developed jointly with the CPRG and were designed and generated primarily by the author of this dissertation. A summary of the results from the ACPC for each of these four years is also provided. Finally, we present a mock five-agent tournament consisting of the CPRG's top three-player entrants from each of the past five years of the ACPC. The results of this mock tournament demonstrate yearly improvement in our champion three-player poker agents.

8.1 2010

For the 2010 competition, similar to the TBR agent from 2009, we constructed two static expert strategy profiles as defined in Section 7.1, one for our IRO agent and one for our TBR agent. Both agents used the same base strategy profile, which itself was a dynamic expert profile as defined in Section 7.2. For this base strategy, the nonterminal histories were partitioned into two parts according to the number of players, zero or one, that had folded. For each part, an IR169/IR900/IR100/25-sized abstraction was employed. Flop and turn bucketing, as described in Section 3.4.2, was performed using k-means clustering over earth mover's distance, while river bucketing used percentile hand strength. Note that up to suit isomorphisms, there are exactly 169 different pre-flop deals of two private cards, and so essentially no abstraction is employed during the pre-flop. The only difference between the abstractions for the two parts was that the "zero-players-folded" part defined hand strength for earth mover's distance and percentile bucketing as the probability that the given hand wins against *two* random opponent hands in a showdown. The "one-player-folded" part used hand strength as the probability of winning against a single random hand as we defined it to be in Section 3.4.2. In hindsight, this special *three-player hand strength* that we defined over the zero-players-folded part had no significant benefit in practice and was abandoned in later competitions. This base profile was computed from 70 million iterations of CS. We note here that all of our competition strategies and experts from 2010, 2011, and 2012 use the average strategy computed by the MCCFR algorithms rather than the current strategy. Only our 2013 agent uses the current strategy. Our positive results regarding the current strategy in Chapter 4 were not discovered until after the 2012 competition.

Each of our static expert profiles consisted of our base strategy profile and four static experts with $S = N$, one for each of the betting sequences f, rf, rrf, and rcf, where f denotes fold, c denotes call, and r denotes raise. All of these experts employed the same IR169/IR60,000/IR180,000/26,160-sized abstraction. Here, flop, turn, and river bucketing was performed by first partitioning all possible card deals into 20 parts for the flop and 60 parts for the turn and river according to a similarity metric on just the public community cards as described by Waugh *et al.* [73]. Then, the hands in each part were bucketed into 3000 buckets on the flop and turn using k-means clustering on earth mover's distance, and 436 buckets on the river using OCHS as defined in Section 3.4.2. The only difference between the experts for our IRO agent and the experts for our TBR agent was that the TBR experts were computed in a tilted game. We previously used tilted games in Section 4.4 to create two-player non-zero-sum games. Here, during computation, we employed the *orange* tilt that awarded each TBR expert a fictitious 7% bonus for winning a hand

Table 8.1: Results of the 2010 ACPC three-player limit hold'em events. Earnings are in milli-big-blinds per game (mbb/g) and errors indicate 95% confidence intervals.

Total Bankroll

Agent	Total Earnings (mbb/g)
Hyperborean3p.TBR.2010	248 ± 45
LittleRock	116 ± 50
Bender	-40 ± 18
Arnold3	-104 ± 48
dcu3pl.tbr	-221 ± 92

Instant Runoff

Agent	Round 1	Round 2	Round 3
Hyperborean3p.IRO.2010	144 ± 32	105 ± 5	75 ± 6
dcu3pl.iro	98 ± 30	49 ± 6	-18 ± 7
LittleRock	65 ± 35	-20 ± 7	-58 ± 7
Arnold3	-135 ± 39	-135 ± 7	Eliminated
Bender	-172 ± 16	Eliminated	Eliminated

in an attempt to force the experts to play more aggressively. All experts were computed from 10 billion iterations of CS, except for the rf and rcf IRO experts that were generated from 8 billion iterations of CS. The use of four experts here was a reduction from six experts that were used in the 2009 TBR event. The cf and crf experts were not employed since these sequences appear to happen infrequently in practice. Omitting these experts also kept our disk usage below the allowed 30GB for the competition.

The abstract games for the base strategy profile and the experts contained approximately 262 million information sets and up to 266 million information sets respectively. These games were significantly larger than the abstract games employed in 2009 that only contained up to 155.4 million information sets. We were able to employ finer abstractions in 2010 thanks to a more memory efficient implementation of CS that did not require new hardware.

The results of the 2010 IRO and TBR events are presented in Table 8.1. Our agents finished first place out of the five entrants in both competitions. The victories were statistically significant.

8.2 2011

Our 2011 IRO agent was again a static expert profile. Compared to our 2010 base profile that only required about 10GB of RAM to compute with our implementation of CS, our IRO base strategy profile employed an abstraction with many more buckets compared to the previous year. This was possible because we gained access to supercomputers maintained by WestGrid and Compute Canada containing compute nodes with 256GB of RAM and 24 cores. We used this to compute our base profile from 180 million iterations of CS in an IR169/IR10,000/IR5450/500-sized abstraction. Bucketing on the flop and turn rounds again used k-means clustering on earth mover's distance, while river bucketing used OCHS. This abstract game contains over 5.9 billion information sets. For

Table 8.2: Results of the 2011 ACPC three-player limit hold'em events. Earnings are in milli-big-blinds per game (mbb/g) and errors indicate 95% confidence intervals.

Total Bankroll (mbb/g)

Agent	Total Earnings
Sartre3p	266 ± 24
Hyperborean3p.TBR.2011	171 ± 23
AAIMontybot	130 ± 45
LittleRock	122 ± 22
OwnBot	16 ± 35
Bnold3	−84 ± 28
Entropy	−99 ± 43
player.zeta.3p	−521 ± 40

Instant Runoff

Agent	Round 1	Round 2	Round 3	Round 4
Hyperborean3p.IRO.2011	204 ± 20	136 ± 21	116 ± 23	96 ± 27
Sartre3p	243 ± 20	161 ± 22	102 ± 23	77 ± 26
LittleRock	113 ± 19	56 ± 20	47 ± 21	31 ± 22
dcubot3plr	77 ± 19	38 ± 19	34 ± 20	20 ± 24
Bnold3	−91 ± 22	−125 ± 24	−63 ± 21	−75 ± 23
AAIMontybot	96 ± 44	17 ± 48	−113 ± 51	−148 ± 52
OwnBot	−4 ± 30	−101 ± 31	−122 ± 35	Eliminated
Entropy	−108 ± 36	−182 ± 37	Eliminated	Eliminated
player.zeta.3p	−530 ± 33	Eliminated	Eliminated	Eliminated

Agent	Round 5	Round 6	Round 7
Hyperborean3p.IRO.2011	75 ± 7	51 ± 8	24 ± 10
Sartre3p	40 ± 7	21 ± 7	−5 ± 9
LittleRock	6 ± 6	−14 ± 7	−29 ± 9
dcubot3plr	−17 ± 6	−58 ± 7	Eliminated
Bnold4	−103 ± 6	Eliminated	Eliminated

our experts, we again used four static experts with $S = N$ for the f, rf, rrf, and rcf sequences. Each of these experts used an IR169/IR180,000/IR540,000/78,480-sized abstraction using the same bucketing techniques that were used for the experts in 2010. These abstract games contained up to 797.8 million information sets and could be computed with the hardware used in 2010 and without the special cluster used for the base profile. In order to fit our base profile and our four experts on disk within the 30GB limitation, we represented each action probability by a single byte by rounding floating-point values to the nearest $1/256$.

For the TBR event, our 2011 agent was somewhat experimental. Given the presence of weaker agents like dcu3pl.tbr in the 2010 TBR competition, we attempted to build an agent that could earn more money from weaker opponents. To this end, we used abstractions similar to the asymmetric abstractions described in Section 7.3. For each player i, we computed a dynamic expert strategy by partitioning the nonterminal histories into two parts, H_i and H_{-i}, depending on whether player i acted at the history or not. For the H_i part, we used the IR169/IR900/IR100/25 abstraction used for the 2010 base strategy profile, whereas for the H_{-i} part, we used the IR16/IR16/IR16/16 abstraction used in 2009. In preliminary experiments, we found that this profile was moderately successful

Table 8.3: Results of the 2012 ACPC three-player limit hold'em events. Earnings are in milli-big-blinds per game (mbb/g) and errors indicate 95% confidence intervals.

Total Bankroll

Agent	Total Earnings
Hyperborean3p.2012	28 ± 5
little.rock	-4 ± 7
neo.poker.lab	-11 ± 5
sartre	-12 ± 7

Instant Runoff

Agent	Round 1	Round 2	Round 3
Hyperborean3p.2012	37 ± 5	28 ± 5	23 ± 8
little.rock	13 ± 6	-4 ± 7	-9 ± 9
neo.poker.lab	7 ± 5	-11 ± 5	-14 ± 6
sartre	5 ± 7	-12 ± 7	Eliminated
dcubot	-62 ± 8	Eliminated	Eliminated

against weaker opponents computed using very coarse abstractions; however, it was less successful against better opponents. So, our 2011 TBR agent was essentially a "meta-agent" that dynamically switched between this dynamic expert strategy and our 2010 IRO agent depending on which of the two had a higher estimate of winning. Estimates were computed using importance sampling as described by Bowling *et al.* [10]. We note here that we were also in the process of computing static experts to attach to our new dynamic expert strategy, but these computations were not completed in time for the competition due to a power outage.

The results of the 2011 IRO and TBR events are presented in Table 8.2. Our agents finished first and second place out of the nine and eight entrants in the respective competitions. Our victory in the IRO event was again statistically significant.

8.3 2012

For the 2012 competitions, we constructed just a single agent that played in both the IRO and TBR events. This agent was a dynamic expert strategy profile with a grafting profile that again partitioned the nonterminal histories into two parts, an *important* part and an *unimportant* part. The important histories were defined as follows. First, we scanned all of the 2011 ACPC match logs that our 2011 IRO agent played in and for each betting sequence, we calculated the frequency at which we were faced with a decision at that sequence. For example, the frequency we were faced with a decision at the empty betting sequence was $1/3$ since we were in the dealer position and first to act once in every three hands. Next, we multiplied each of these frequencies by the pot size at that betting sequence. For instance, we multiplied the $1/3$ frequency for the empty betting sequence by 15 since the game is played with a small blind of 5 chips and a big blind of 10 chips, creating an initial pot of 15 chips. For each history, if this value for the history's betting sequence was greater than $1/100$, then the history was labeled as important. In addition, any prefix of an important history was

Table 8.4: Results of the 2013 ACPC three-player instant-runoff limit hold'em event. Earnings are in milli-big-blinds per game (mbb/g) and errors in the final round indicate 95% confidence intervals (errors in other rounds are unavailable).

Agent	Round 1	Round 2	Round 3	Round 4
Hyperborean3p.IRO.2013	221	185	114	49.3 ± 14.4
littlerock	162	118	60	-8.6 ± 8.7
neo_poker_lab	123	150	52	-40.8 ± 18.5
kempfer	-166	-169	-226	Eliminated
HITSZ_CS_13	-74	-285	Eliminated	Eliminated
liacc	-266	Eliminated	Eliminated	Eliminated

also labeled as important, while the remaining histories were labeled as unimportant. Since betting actions are public information, this forms a valid grafting partition as defined by Definition 2.7 for each player. Only 0.023% of the nonterminal betting sequences in three-player hold'em belonged to the important part.

Using this grafting profile, our dynamic expert profile employed a very fine-grained abstraction on the important part and a coarse abstraction on the unimportant part. This way, our agent can distinguish between many more hands at the few sequences that historically were reached more frequently. Our coarse abstraction for the unimportant part was twice the size of our 2010 base abstraction, using IR169/IR1800/IR200/50 buckets per round from k-means clustering on earth mover's distance on the flop and turn and OCHS on the river. On the other hand, our fine-grained abstraction for the important part used IR169/IR180,000/IR765,000/840,000 buckets per round. Similar to the 2010 experts, these were constructed by first partitioning the flop, turn, and river hands into 9, 51, and 280 parts respectively according to hand-chosen public card textures. Then, for each part, the hands were independently bucketed into 20,000, 15,000, and 3000 buckets on the flop, turn, and river respectively. Again, we used k-means clustering on earth mover's distance for the flop and turn, and OCHS for the river. In total, this dynamic expert abstract game contained roughly 2.5 billion information sets.

The dynamic expert profile was computed using External Sampling MCCFR, run in parallel for 16 days using 48 2.2 GHz AMD processors with 256GB of total RAM. The 2012 competition results are presented in Table 8.3. Despite the field of competitors being generally better than the previous year, our agent won both the IRO and TBR events by significant margins.

8.4 2013

Finally, the IRO agent for the 2013 competition was constructed in a very similar manner to our 2012 agent, with a few differences that we note here. Firstly, the rules for the 2013 competition increased the maximum allowable disk space for an agent to 100GB. Thus, we designed our agent to use this maximum disk space entirely. Secondly, a new grafting profile was built,

Table 8.5: Results of a mock five-agent tournament between the CPRG's 2009 IRO agent and our 2010, 2011, 2012, and 2013 IRO agents. Earnings are in milli-big-blinds per game for the row player against the column players and errors indicate 95% confidence intervals.

Agent	**2009,2010**	**2009,2011**	**2009,2012**	**2009,2013**	**2010,2011**	**2010,2012**	**2010,2013**	**2011,2012**	**2011,2013**	**2012,2013**	**Overall**
2009	-	-	-	-	-21 ± 7	-26 ± 5	-28 ± 7	-31 ± 5	-35 ± 6	-35 ± 4	$\mathbf{-29 \pm 3}$
2010	-	0 ± 4	-5 ± 3	-10 ± 6	-	-	-	-23 ± 5	-27 ± 7	-27 ± 4	$\mathbf{-16 \pm 3}$
2011	21 ± 6	-	6 ± 5	3 ± 6	-	8 ± 5	-3 ± 8	-	-	-12 ± 4	$\mathbf{4 \pm 3}$
2012	31 ± 5	25 ± 5	-	11 ± 5	16 ± 4	-	3 ± 5	-	0 ± 5	-	$\mathbf{14 \pm 3}$
2013	38 ± 4	33 ± 5	25 ± 3	-	30 ± 6	24 ± 4	-	12 ± 3	-	-	$\mathbf{27 \pm 3}$

this time combining the logs from both the 2011 IRO and the 2012 competitions. This grafting profile was built in the same manner as the previous year, except now any history's betting sequence with a value greater than 0.0014 was labeled as important, of which about 0.13% of the nonterminal betting sequences achieved. The important part used a very fine-grained abstraction with IR169/IR1,348,620/IR1,530,000/2,800,000 buckets per round, while the unimportant part used IR169/IR180,000/IR18,630/875 buckets per round. This resulted in a dynamic expert abstract game containing over 38.8 billion information sets. The 2013 three-player hold'em TBR agent, on the other hand, was a new opponent modelling agent that was built by Nolan Bard, another member of the CPRG, and is not discussed here.

We again used 48 2.2 GHz AMD processors to compute this dynamic expert profile, this time with 512GB of total RAM. This was only possible because we used Pure CFR, run for approximately 303 billion iterations over 16 days, and only computed the current strategy. Note that our most efficient implementation of External Sampling, for example, would have required over 700GB of RAM and thus would have been infeasible for the machine that was readily available to us. The 2013 IRO competition results are reported in Table 8.4. Our agent won again by a comfortable margin.

8.5 Summary

In total, our agents placed first in six competitions and second in one competition out of the seven three-player events that we competed in from the 2010, 2011, 2012, and 2013 ACPC. To conclude this chapter, we compare the head-to-head performances of our strongest competition agents and demonstrate improvement in performance each year. We ran a mock five-agent tournament consisting of the CPRG's IRO agent from the 2009 ACPC described at the beginning of this chapter, and our four IRO agents from 2010, 2011, 2012, and 2013 presented above. Full results from this tournament are presented in Table 8.5. Looking at the overall totals, we see that our 2010, 2011, 2012, and 2013 IRO agents earn 13, 20, 10, and 13 milli-big-blinds per game more than the agent from the previous year respectively.

Chapter 9

Conclusions

This dissertation has studied regret minimization and strategy stitching in extensive-form games. The primary motivation of our work was to resolve a number of complications listed in Chapter 1 that arose when computing strategy profiles for three-player limit Texas hold'em and other large games with two or more players. We now summarize our efforts towards this goal.

Although strategy profiles computed with CFR have performed well in the three-player events of the ACPC, there was no theoretical explanation to support this. In Chapter 4, we formally defined dominated actions and gave theoretical evidence suggesting that CFR avoids iteratively strictly dominated actions and strategies. Supporting previous conclusions of Waugh using domination value [70], we showed that in two-player Kuhn Poker, simply avoiding iteratively strictly dominated actions led to good performance. In addition, domination can be avoided by using the current strategy from CFR without averaging, and doing so is more efficient in terms of both computation time and memory usage.

Between 2009 and 2011, the majority of the CPRG's ACPC entries were computed with Chance Sampling MCCFR (CS), which must traverse all player actions and resulted in long computation times being required. Chapter 5 explored other sampling techniques and proposed three new algorithms, Probing (Algorithm 3), Average Strategy Sampling (AS), and Pure CFR. Firstly, Probing can reduce the variance of traditional MCCFR algorithms and can result in faster convergence to equilibrium in zero-sum games. Secondly, AS reduces convergence time in games with many player actions, such as no-limit poker games with no action abstraction, while still being comparable to External Sampling in other domains. Thirdly, Pure CFR's memory costs are half of that of other sampling algorithms, allowing larger abstractions to be employed at only a minor cost in computation time. Today, CS is no longer used to compute any of the CPRG's competition agents.

Furthermore, the CPRG's strongest agents have employed imperfect recall abstractions of Texas hold'em, yet the original CFR analysis only guarantees regret minimization in perfect recall games. Our theoretical work in Chapter 6 provides the first regret bounds for CFR when applied to a class of games with imperfect recall. We defined well-formed and skew well-formed games, and proved that regret is minimized and bounded when using CFR in these games respectively. Using a variant of die-roll poker, we demonstrated that lower regret can be achieved in skew well-formed games than that guaranteed by our regret bound.

Finally, in large games like three-player limit hold'em, we must employ fairly coarse abstractions to feasibly compute strategies with CFR. In Chapter 7, we defined static experts and dynamic experts that attempt to alleviate this problem by partitioning a game into smaller parts and employing finer abstractions within a number of these parts. Our static expert framework generalized strategy grafting from Section 2.4.1 and two previous techniques used in poker. We showed that despite recent mixed results, static experts can improve play over a base strategy alone and for the types of partitions we considered, they can be preferred over other approaches. Our 2010, 2011, 2012, and 2013 three-player entries for the ACPC described in Chapter 8 won a total of six out of seven competitions using these techniques, further validating the use of static and dynamic experts.

9.1 Future Work

In spite of these contributions, our work in domination and imperfect recall games has only begun to enlighten our understanding of CFR outside of zero-sum perfect recall games. In addition, our generalized sampling techniques and strategy stitching methods leave us with some open questions. We end this thesis by listing a number of these questions and areas for future work:

1. Clearly, it is undesirable to play a dominated action or strategy due to the existence of an alternative that is guaranteed to do better, regardless of what the opponents do. Are there other properties of an action or strategy that are undesirable against a set of unknown opponents? More importantly, does CFR avoid such undesirable actions or strategies? Alternatively, what are the desirable properties of actions and strategies in non-zero-sum games and do CFR solutions exhibit these properties? As we argued at the beginning of Chapter 4, a Nash equilibrium is less meaningful outside of zero-sum games and we know CFR does not compute an equilibrium for these games anyways. Answering these questions could further improve our understanding of why CFR solutions can perform well in large, non-zero-sum games.

2. Theorem 5.4 demonstrates that estimated counterfactual values with low variance provide a better bound on a player's regret than estimates with high variance. Probing, presented in Algorithm 3, provides one example of how to reduce variance and converge to equilibrium faster in zero-sum games. We suspect that there are other efficiently-computable definitions of the estimated counterfactual values $\hat{v}_i(I, \sigma)$ that are bounded, unbiased, and exhibit lower variance than our probing example. Further improvements to convergence rates are likely possible through such alternative definitions. More importantly, however, we still lack a comprehensive theory for deciding which sampling algorithm to use when. As we saw in Section 5.6, the relative time efficiency of each algorithm we considered varied greatly between two-player limit, two-player no-limit, and three-player limit hold'em. A prominent area for future

work would be to build a set of rules that decide which sampling algorithm to use for a given game or how to perform sampling in CFR to minimize regret most efficiently.

3. From Definition 6.1, a well-formed game is required to satisfy four conditions. By introducing skew well-formed games in Definition 6.2, we showed that one of these conditions can be relaxed and a regret bound can still be established. However, the games presented in Figure 6.2 break conditions (ii) or (iii) of Definition 6.1 and CFR does not minimize regret in these games. Unfortunately, as described at the end of Section 6.2.3, these conditions are also broken by the imperfect recall abstract games used by the our three-player entries and the CPRG's other entries that performed well in the ACPC. We would like to expand on the set of imperfect recall games to which CFR gives regret guarantees. In particular, it may be possible to derive regret bounds for a new class of games where conditions (ii) and (iii) are relaxed to better justify the imperfect recall abstractions that we currently employ.

4. Static experts, as given by Definition 7.1, are constructed by holding the action probabilities for a nonempty subset S of players fixed off of the given subtree. Our experiments in Section 7.3 considered static experts for $S = \{i\}$ and $S = N$. However, in games with three or more players, other choices of S could produce interesting results. For example, in three-player hold'em we could compute an expert strategy for player i, say on a subtree where player j has folded, such that player j's probabilities are all fixed and the third player is left unrestricted. This would be achieved by setting $S = \{i, j\}$. These types of experts could be considered *hybrids* between strategy grafting and heads-up experts presented by Abou Risk and Szafron [3]. Furthermore, there are practically endless choices for grafting partitions. The profiles constructed in Section 7.3 and our ACPC agents described in Chapter 8 touched only a small number of these choices. Better grafting partitions could result in even greater performance for static and dynamic experts.

5. Finally, this work has only considered the problem of computing a stationary strategy profile for play in an extensive-form game. For zero-sum games, if our opponent is playing an equilibrium profile, then the best we can do is play an equilibrium profile ourselves. On the other hand, for repeated non-zero-sum games, we can likely do better than always following our stationary profile, even if the opponents are independently following different equilibrium profiles. While our stationary profiles continue to win three-player hold'em competitions, we believe that our agents could be greatly improved with opponent modelling capabilities. One approach to modelling would be to compute several profiles and dynamically switch to the best performing strategy on-line, similar to our 2011 TBR competition agent from Section 8.2. However, given disk space restrictions, only a small number of profiles can be stored and could severely limit performance. A more effective course of action may be to update our action probabilities on-line in response to the play of the opponents. This approach requires no

more disk space than a single, stationary profile. Unfortunately, no time-efficient and robust methods for this type of opponent modelling are currently known.

Bibliography

[1] The first man-machine poker competition. http://poker.cs.ualberta.ca/man-machine/2007, 2007. On-line; accessed 10-Apr-2013.

[2] The second man-machine poker competition. http://poker.cs.ualberta.ca/man-machine, 2008. On-line; accessed 10-Apr-2013.

[3] Nick Abou Risk and Duane Szafron. Using counterfactual regret minimization to create competitive multiplayer poker agents. In *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 159–166, 2010.

[4] Nolan Bard, John Hawkin, Jonathan Rubin, and Martin Zinkevich. The Annual Computer Poker Competition. *AI Magazine*, **34**(2):112–114, 2013.

[5] Darse Billings, Neil Burch, Aaron Davidson, Robert Holte, Jonathan Schaeffer, Terence Schauenberg, and Duane Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 661–668, 2003.

[6] Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. The challenge of poker. *Artificial Intelligence*, **134**:201–240, 2002.

[7] Darse Billings, Denis Papp, Jonathan Schaeffer, and Duane Szafron. Opponent modeling in poker. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI)*, pages 493–499, 1998.

[8] Darse Billings, Denis Papp, Jonathan Schaeffer, and Duane Szafron. Poker as a testbed for AI research. In R. Mercer and E. Neufeld, editors, *Advances in Artificial Intelligence*, pages 228–238. Springer Verlag, 1998.

[9] Avrim Blum and Yishay Mansour. Learning, regret minimization, and equilibria. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, pages 79–101. Cambridge University Press, 2007.

[10] Michael Bowling, Michael Johanson, Neil Burch, and Duane Szafron. Strategy evaluation in extensive games with importance sampling. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML)*, pages 72–79, 2008.

[11] Michael Buro. The Othello match of the year: Takeshi Murakami vs. Logistello. *International Computer Chess Association Journal*, **20**:189–193, 1997.

[12] Murray Campbell, A. Joseph Hoane Jr., and Feng-hsiung Hsu. Deep blue. *Artificial Intelligence*, **134**:57–83, 2002.

[13] Xi Chen and Xiaotie Deng. 3-Nash is PPAD-complete. *Electronic Colloquium on Computational Complexity (ECCC)*, **134**(TR05-134), 2005.

[14] Xi Chen and Xiaotie Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of the Forty-Seventh Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 261–272, 2006.

[15] Vincent Conitzer and Tuomas Sandholm. Complexity of (iterated) dominance. In *Proceedings of the Sixth ACM Conference on Electronic Commerce (EC)*, pages 88–97, 2005.

[16] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing (STOC)*, pages 71–78, 2006.

[17] Constantinos Daskalakis and Christos H. Papadimitriou. Three-player games are hard. *Electronic Colloquium on Computational Complexity (ECCC)*, **139**(TR05-139):81–87, 2005.

[18] Miroslav Dudík and Geoffrey J. Gordon. A sampling-based approach to computing equilibria in succinct extensive-form games. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 151–160, 2009.

[19] Sam Ganzfried and Tuomas Sandholm. Computing an approximate jam/fold equilibrium for 3-agent no-limit Texas hold'em tournaments. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 919–925, 2008.

[20] Sam Ganzfried and Tuomas Sandholm. Computing equilibria in multiplayer stochastic games of imperfect information. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 140–146, 2009.

[21] Sam Ganzfried and Tuomas Sandholm. Computing equilibria by incorporating qualitative models. In *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 183–190, 2010.

[22] Sylvain Gelly and David Silver. Achieving master level play in 9 × 9 Computer Go. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1537–1540, 2008.

[23] Richard Gibson. Open Pure CFR - An open implementation of Pure CFR applied to ACPC poker games, 2013. `https://github.com/rggibson/open-pure-cfr`.

[24] Richard Gibson, Neil Burch, Marc Lanctot, and Duane Szafron. Efficient Monte Carlo counterfactual regret minimization in games with many player actions. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1889–1897, 2012.

[25] Richard Gibson, Marc Lanctot, Neil Burch, Duane Szafron, and Michael Bowling. Generalized sampling and variance in counterfactual regret minimization. In *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI)*, pages 1355–1361, 2012.

[26] Richard Gibson and Duane Szafron. Regret minimization in multiplayer extensive games. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2802–2803, 2011.

[27] Richard G. Gibson and Duane Szafron. On strategy stitching in large extensive form multiplayer games. In *Advances in Neural Information Processing Systems (NIPS) 24*, pages 100–108, 2011.

[28] Itzhak Gilboa, Ehud Kalai, and Eitan Zemel. The complexity of eliminating dominated strategies. *Mathematics of Operations Research*, **18**(3):553–565, 1993.

[29] Andrew Gilpin and Tuomas Sandholm. Optimal Rhode Island hold'em poker. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, pages 1684–1685, 2005.

[30] Andrew Gilpin and Tuomas Sandholm. Better automated abstraction techniques for imperfect information games, with application to Texas hold'em poker. In *Proceedings of the Sixth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 192:1–192:8, 2007.

[31] Andrew Gilpin, Tuomas Sandholm, and Troels Bjerre Sørensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas hold'em poker. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 50–57, 2007.

[32] Andrew Gilpin, Tuomas Sandholm, and Troels Bjerre Sørensen. A heads-up no-limit Texas hold'em poker player: discretized betting models and automatically generated equilibrium-finding programs. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 911–918, 2008.

[33] Geoffrey J. Gordon. No-regret algorithms for online convex programs. In *Advances in Neural Information Processing Systems (NIPS) 19*, pages 489–496, 2007.

[34] Amy Greenwald, Zheng Li, and Casey Marks. Bounds for regret-matching algorithms. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2005.

[35] Joseph Y. Halpern and Rafael Pass. Iterated regret minimization: A new solution concept. *Games and Economic Behavior*, **74**(1):184–207, 2012.

[36] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI)*, pages 709–715, 2004.

[37] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, **68**(5):1127–1150, 2000.

[38] Samid Hoda, Andrew Gilpin, Javier Peña, and Tuomas Sandholm. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, **35**(2):494–512, 2010.

[39] Bret Hoehn, Finnegan Southey, Robert C. Holte, and Valeriy Bulitko. Effective short-term opponent exploitation in simplified poker. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, pages 783–788, 2005.

[40] Michael Johanson. Robust strategies and counter-strategies: Building a champion level computer poker player. Master's thesis, University of Alberta, 2007.

[41] Michael Johanson. Measuring the size of large no-limit poker games. Technical Report TR13-01, Department of Computing Science, University of Alberta, 2013.

[42] Michael Johanson, Nolan Bard, Neil Burch, and Michael Bowling. Finding optimal abstract strategies in extensive form games. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1371–1379, 2012.

[43] Michael Johanson, Nolan Bard, Marc Lanctot, Richard Gibson, and Michael Bowling. Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 837–846, 2012.

[44] Michael Johanson, Michael Bowling, Kevin Waugh, and Martin Zinkevich. Accelerating best response calculation in large extensive games. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 258–265, 2011.

[45] Michael Johanson, Neil Burch, Richard Valenzano, and Michael Bowling. Evaluating state-space abstractions in extensive-form games. In *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 271–278, 2013.

[46] Mamoru Kaneko and J. Jude Kline. Behavior strategies, mixed strategies and perfect recall. *International Journal of Game Theory*, **4**:127–145, 1995.

[47] Donald E. Knuth, Christos H. Papadimitriou, and John N. Tsitsiklis. A note on strategy elimination in bimatrix games. *Oper. Res. Lett.*, 7(3):103–107, June 1988.

[48] Daphne Koller and Nimrod Megiddo. The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior*, **4**:528–552, 1992.

[49] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast algorithms for finding randomized strategies in game trees. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing (STOC)*, pages 750–759, 1994.

[50] Harold W. Kuhn. Simplified two-person poker. In H.W. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games*, volume **1**, pages 97–103. Princeton University Press, 1950.

[51] Harold W. Kuhn. Extensive games and the problem of information. *Contributions to the Theory of Games*, **2**:193–216, 1953.

[52] Marc Lanctot. *Monte Carlo sampling and regret minimization for equilibrium computation and decision-making in large extensive form games*. PhD thesis, University of Alberta, 2013.

[53] Marc Lanctot, Richard Gibson, Neil Burch, and Michael Bowling. No-regret learning in extensive-form games with imperfect recall. In *Proceedings of the Twenty-Ninth International Conference on Machine Learning (ICML)*, pages 65–72, 2012.

[54] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte Carlo sampling for regret minimization in extensive games. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 1078–1086, 2009.

[55] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte Carlo sampling for regret minimization in extensive games. Technical Report TR09-15, University of Alberta, 2009.

[56] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory, version 13.1.0. `http://www.gambit-project.org/`, 2013. On-line; accessed 10-Sep-2013.

[57] John McLeod. Rules of card games: Poker hand rankings. `http://www.pagat.com/poker/rules/ranking.html`, 2012. On-line; accessed 2-Apr-2013.

[58] John Nash. Non-cooperative games. *The Annals of Mathematics*, **54**:286–295, 1951.

[59] Martin J. Osborne and Ariel Rubenstein. *A Course in Game Theory*. The MIT Press, Cambridge, Massachusetts, 1994.

[60] Michele Piccione and Ariel Rubinstein. On the interpretation of decision problems with imperfect recall. In *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 75–76, 1996.

[61] Ludovic Renou and Karl H. Schlag. Minimax regret and strategic uncertaintly. *Journal of Economic Theory*, **145**(1):264–286, 2009.

[62] Nick Abou Risk. Using counterfactual regret minimization to create a competitive multiplayer poker agent. Master's thesis, University of Alberta, 2009.

[63] Jonathan Schaeffer, Robert Lake, Paul Lu, and Martin Bryant. Chinook: The world Man-Machine Checkers Champion. *AI Magazine*, **17**:21–29, 1996.

[64] David Schnizlein, Michael Bowling, and Duane Szafron. Probabilistic state translation in extensive games with large action sets. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 276–284, 2009.

[65] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the Twenty-First Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 550–558, 2005.

[66] Duane Szafron, Richard Gibson, and Nathan Sturtevant. A parameterized family of equilibrium profiles for three-player Kuhn Poker. In *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 247–254, 2013.

[67] Oskari Tammelin. Personal communication, 2012.

[68] Ken Thompson. Retrograde analysis of certain endgames. *J. International Computer Chess Assoc.*, **9**(3):131–139, 1986.

[69] Bernhard von Stengel and Françoise Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, **33**(4):1002–1022, 2008.

[70] Kevin Waugh. Abstraction in large extensive games. Master's thesis, University of Alberta, 2009.

[71] Kevin Waugh, Michael Bowling, and Nolan Bard. Strategy grafting in extensive games. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 2026–2034, 2009.

[72] Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. Abstraction pathologies in extensive games. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 781–788, 2009.

[73] Kevin Waugh, Martin Zinkevich, Michael Johanson, Morgan Kan, David Schnizlein, and Michael Bowling. A practical use of imperfect recall. In *Proceedings of the Eighth Symposium on Abstraction, Reformulation and Approximation (SARA)*, pages 175–182, 2009.

[74] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. Technical Report TR07-14, University of Alberta, 2007.

[75] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems (NIPS) 20*, pages 1729–1736, 2008.

Appendix A

Supplementary Background Material

In this appendix, we provide a simpler proof of Theorem 2.2 to that of Gordon [33] and walk through CFR in a small example game.

Theorem 2.2. *If player i uses regret matching, then after T time steps,*

$$R_i^T \leq \Delta_i \sqrt{T|A_i|}.$$

Proof. We will prove by induction that

$$\sum_{a \in A_i} \left(R_i^{T,+}(a) \right)^2 \leq \Delta_i^2 T |A_i|. \tag{A.1}$$

Then, note that $R_i^T = \max_{a \in A_i} R_i^T(a)$ since mixing between two actions in σ_i' of equation (2.2) would require the actions to have the same value. Furthermore, we may assume $R_i^T > 0$, otherwise we are done. Then,

$$\begin{aligned}\left(R_i^T\right)^2 &= \left(\max_{a \in A_i} R_i^T(a)\right)^2 = \left(\max_{a \in A_i} R_i^{T,+}(a)\right)^2 = \max_{a \in A_i} \left(R_i^{T,+}(a)\right)^2 \leq \sum_{a \in A_i} \left(R_i^{T,+}(a)\right)^2 \\ &\leq \Delta_i^2 T |A_i|,\end{aligned}$$

and taking the square root of both sides gives the result.

To complete the proof, we prove equation (A.1) by induction on T. The base case $T = 0$ is trivial. For the induction step, we may assume $\sum_{a \in A_i} (R_i^{T-1,+}(a))^2 \leq \Delta_i^2 (T-1)|A_i|$. Firstly, one can easily verify that

$$\left((a+b)^+\right)^2 \leq (a^+ + b)^2 \text{ for all } a, b \in \mathbb{R} \tag{A.2}$$

by checking the cases where $a \leq 0$, $a > 0$ and $b \geq -a$, and $a > 0$ and $b < -a$. Then,

$$\begin{aligned}\left(R_i^{T,+}(a)\right)^2 &= \sum_{a \in A_i} \left(\left(R_i^{T-1}(a) + \left(u_i(a, \sigma_{-i}^T) - u_i(\sigma_i^T, \sigma_{-i}^T)\right)\right)^+ \right)^2 \\ &\leq \sum_{a \in A_i} \left(R_i^{T-1,+}(a) + \left(u_i(a, \sigma_{-i}^T) - u_i(\sigma_i^T, \sigma_{-i}^T)\right)\right)^2\end{aligned}$$

$$
\begin{aligned}
&= \sum_{a\in A_i}\left(R_i^{T-1,+}(a)\right)^2 + 2\sum_{a\in A_i} R_i^{T-1,+}(a)\left(u_i(a,\sigma_{-i}^T) - u_i(\sigma_i^T,\sigma_{-i}^T)\right)\\
&\quad + \sum_{a\in A_i}\left(u_i(a,\sigma_{-i}^T) - u_i(\sigma_i^T,\sigma_{-i}^T)\right)^2, \qquad \text{(A.3)}
\end{aligned}
$$

where the inequality follows by equation (A.2). We claim that

$$\sum_{a\in A_i} R_i^{T-1,+}(a)\left(u_i(a,\sigma_{-i}^T) - u_i(\sigma_i^T,\sigma_{-i}^T)\right) = 0.$$

Provided the claim is true, equation (A.3) gives us

$$
\begin{aligned}
\left(R_i^{T,+}(a)\right)^2 &\le \sum_{a\in A_i}\left(R_i^{T-1,+}(a)\right)^2 + \sum_{a\in A_i}\left(u_i(a,\sigma_{-i}^T) - u_i(\sigma_i^T,\sigma_{-i}^T)\right)^2\\
&\le \Delta_i^2(T-1)|A_i| + |A_i|\Delta_i^2 = \Delta_i^2 T|A_i|,
\end{aligned}
$$

completing the induction step.

Finally, it remains to prove the claim. To that end, we may assume $\sum_{a\in A_i} R_i^{T-1,+}(a) > 0$, otherwise we are done. Then,

$$
\begin{aligned}
&\sum_{a\in A_i} R_i^{T-1,+}(a)\left(u_i(a,\sigma_{-i}^T) - u_i(\sigma_i^T,\sigma_{-i}^T)\right)\\
&\quad= \sum_{a\in A_i} R_i^{T-1,+}(a)\left(u_i(a,\sigma_{-i}^t) - \sum_{b\in A_i}\sigma_i^T(b)u_i(b,\sigma_{-i}^T)\right)\\
&\quad= \sum_{a\in A_i} R_i^{T-1,+}(a)u_i(a,\sigma_{-i}^t) - \sum_{a\in A_i}\sum_{b\in A_i} R_i^{T-1,+}(a)\sigma_i^T(b)u_i(b,\sigma_{-i}^T)\\
&\quad= \sum_{a\in A_i} R_i^{T-1,+}(a)u_i(a,\sigma_{-i}^t) - \sum_{a\in A_i}\sum_{b\in A_i} \frac{R_i^{T-1,+}(a)R_i^{T-1,+}(b)u_i(b,\sigma_{-i}^T)}{\sum_{d\in A_i} R_i^{T-1,+}(d)} \text{ by (2.3)}\\
&\quad= \sum_{a\in A_i} R_i^{T-1,+}(a)u_i(a,\sigma_{-i}^t) - \sum_{b\in A_i} R_i^{T-1,+}(b)u_i(b,\sigma_{-i}^T)\frac{\sum_{a\in A_i} R_i^{T-1,+}(a)}{\sum_{d\in A_i} R_i^{T-1,+}(d)}\\
&\quad= \sum_{a\in A_i} R_i^{T-1,+}(a)u_i(a,\sigma_{-i}^t) - \sum_{b\in A_i} R_i^{T-1,+}(b)u_i(b,\sigma_{-i}^T)\\
&\quad= 0,
\end{aligned}
$$

completing the proof. ■

We will now walk through two iterations of CFR on the small game shown in Figure A.1. To begin, all the regret and cumulative profile values are initialized to zero. As stated on line 3 of Algorithm 1, our initial profile is the uniform random profile. We will also use uniform random when the denominator of equation 2.4 is zero.

Starting the first iteration for player 1, we have two information sets to visit, labeled I and I'. At I', we calculate the counterfactual values $v_1(I',\sigma^1_{I'\to f})$ and $v_1(I',\sigma^1_{I'\to c})$ according to

$$
\begin{aligned}
v_1(I',\sigma^1_{(I'\to f)}) &= \sum_{z\in Z_{I'}} \pi_{-1}^{\sigma^1}(z[I'])\pi^{\sigma^1}(z[I']f,z)u_1(z)\\
&= \pi_{-1}^{\sigma^1}(Jkb)\pi^{\sigma^1}(Jkbf,Jkbf)u_1(Jkbf) + \pi_{-1}^{\sigma^1}(Kkb)\pi^{\sigma^1}(Kkbf,Kkbf)u_1(Kkbf)
\end{aligned}
$$

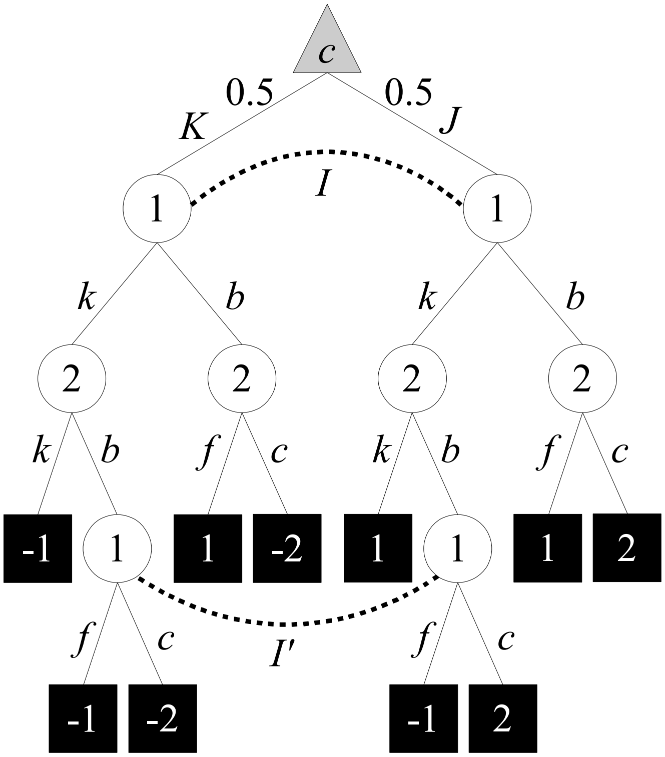


Figure A.1: A zero-sum extensive-form game where player 2 has perfect information. This game is Kuhn Poker, described in Chapter 3, where player 1 is always dealt the Queen.

$$
\begin{aligned}
&= 0.25 \cdot 1 \cdot (-1) + 0.25 \cdot 1 \cdot (-1) \\
&= -0.5
\end{aligned}
$$

and

$$
\begin{aligned}
v_1(I', \sigma^1_{(I' \to c)}) &= \pi^{\sigma^1}_{-1}(Jkb)\pi^{\sigma^1}(Jkbc, Jkbc)u_1(Jkbc) + \pi^{\sigma^1}_{-1}(Kkb)\pi^{\sigma^1}(Kkbc, Kkbc)u_1(Kkbc) \\
&= 0.25 \cdot 1 \cdot 2 + 0.25 \cdot 1 \cdot (-2) \\
&= 0.
\end{aligned}
$$

From these, the counterfactual value $v_1(I', \sigma^1)$ is straightforward to calculate from

$$
v_1(I', \sigma^1) = \sum_{a \in A(I')} \sigma^1(I', a) v_1(I', \sigma^1_{(I' \to a)}) = 0.5 \cdot (-0.5) + 0.5 \cdot 0 = -0.25.
$$

This gives us our regret updates

$$
R^1_1(I', f) = R^0_1(I', f) + v_1(I', \sigma^1_{(I' \to f)}) - v_1(I', \sigma^1) = 0 + (-0.5) - (-0.25) = -0.25
$$

and

$$
R^1_1(I', c) = R^0_1(I', c) + v_1(I', \sigma^1_{(I' \to c)}) - v_1(I', \sigma^1) = 0 + 0 - (-0.25) = 0.25.
$$

We also update the cumulative profile according to the current strategy, giving us

$$
s^1_1(I', f) = s^0_1(I', f) + \pi^{\sigma^1}_1(I')\sigma^1_1(I', f) = 0 + 0.5 \cdot 0.5 = 0.25
$$

and $s_1^1(I', c) = 0.25$. Similarly, at I, we can calculate the counterfactual values, summing over terminal nodes in Figure A.1 from left to right,

$$\begin{aligned}v_1(I, \sigma_{(I\to k)}) &= 0.5\cdot 0.5\cdot(-1) + 0.5\cdot 0.25\cdot(-1) + 0.5\cdot 0.25\cdot(-2)\\ &\quad + 0.5\cdot 0.5\cdot 1 + 0.5\cdot 0.25\cdot(-1) + 0.5\cdot 0.25\cdot 2\\ &= -0.25\end{aligned}$$

and

$$v_1(I, \sigma_{(I\to b)}) = 0.5\cdot 0.5\cdot 1 + 0.5\cdot 0.5\cdot(-2) + 0.5\cdot 0.5\cdot 1 + 0.5\cdot 0.5\cdot 2 = 0.5.$$

This gives us the counterfactual value

$$v_1(I, \sigma) = \sum_{a\in A(I)} \sigma^1(I, a) v_1(I, \sigma^1_{(I\to a)}) = 0.5\cdot(-0.25) + 0.5\cdot 0.5 = 0.125,$$

updated regret

$$R_1^1(I, k) = R_1^0(I, k) + v_1(I, \sigma_{(I\to k)}) - v_1(I, \sigma) = 0 + (-0.25) - 0.125 = -0.375$$

and

$$R_1^1(I, b) = R_1^0(I, b) + v_1(I, \sigma_{(I\to b)}) - v_1(I, \sigma) = 0 + 0.5 - 0.125 = 0.375,$$

and updated cumulative profile $s_1^1(I, k) = 0.5$ and $s_1^1(I, b) = 0.5$. Our new regret values give us player 1's current strategy for the second iteration according to equation (2.4), with

$$\sigma_1^2(I, k) = R_1^{1,+}(I, k)/\left(R_1^{1,+}(I, k) + R_1^{1,+}(I, b)\right) = 0/(0 + 0.375) = 0,$$

and similarly $\sigma_1^2(I, b) = 1$, $\sigma_1^2(I', f) = 0$, and $\sigma_1^2(I', c) = 1$. Player 2's updates follow the same procedures and we skip over these details.

On iteration 2, we repeat the same calculations, except this time using the updated current strategy profile σ^2. For example, at I', the counterfactual value

$$\begin{aligned}v_1(I', \sigma^2_{(I'\to c)}) &= \pi_{-1}^{\sigma^2}(Jkb)\pi^{\sigma^2}(Jkbc, Jkbc)u_1(Jkbc) + \pi_{-1}^{\sigma^2}(Kkb)\pi^{\sigma^2}(Kkbc, Kkbc)u_1(Kkbc)\\ &= 0.5\cdot 1\cdot(-2) + 0.5\cdot 1\cdot 2\\ &= 0\end{aligned}$$

and similarly $v_1(I', \sigma^2_{(I'\to f)}) = -1$, giving

$$v_1(I', \sigma^2) = \sum_{a\in A(I')} \sigma_1^2(I', a) v_1(I', \sigma^2_{(I'\to a)}) = 0\cdot(-1) + 1\cdot 0 = 0.$$

Our regret updates are then

$$R_1^2(I', f) = R_1^1(I', f) + v_1(I', \sigma^2_{(I'\to f)}) - v_1(I', \sigma^2) = -0.25 + (-1) - 0 = -1.25$$

and

$$R_1^2(I', c) = R_1^1(I', c) + v_1(I', \sigma^2_{(I' \to c)}) - v_1(I', \sigma^2) = 0.25 + 0 - 0 = 0.25,$$

and the cumulative profile is unchanged according to

$$s_1^2(I', c) = s_1^1(I', c) + \pi_1^{\sigma^2}(I')\sigma_1^2(I', c) = 0.25 + 0 \cdot 1 = 0.25$$

and $s_1^2(I', f) = 0.25$. The updates at I, however, result in a new cumulative profile value $s_1^2(I, b) = 1.5$ while $s_1^2(I, k) = 0.5$. If we were to terminate CFR after two iterations, the outputted average strategy for player 1 would be $\bar{\sigma}_1^2(I) = \{(k, 0.25), (b, 0.75)\}$ and $\bar{\sigma}_1^2(I') = \{(f, 0.5), (c, 0.5)\}$.

Appendix B

Proofs for Chapter 4: Regret Minimization and Domination

In this appendix, we prove Proposition 4.1 and Theorems 4.1, 4.2, and 4.3. To start, we need a few additional definitions. First, for $I \in \mathcal{I}_i$, define

$$D(I) = \{I' \in \mathcal{I}_i \mid \exists h \in I, h' \in I' \text{ such that } h \sqsubseteq h'\}$$

to be the set of information sets descending from I. Second, for $I, I' \in \mathcal{I}_i$, $h \in I$, $h' \in I'$, and $\sigma_i \in \Sigma_i$, define $\pi_i^\sigma(I, I') = \pi_i^\sigma(h, h')$. As in Chapter 4, perfect recall is assumed throughout this appendix, which ensures that $\pi_i^\sigma(I, I')$ is well-defined.

Proposition 4.1. *In an extensive-form game with perfect recall, if a is a weakly dominated action at $I \in \mathcal{I}_i$ and $\sigma_i \in \Sigma_i$ satisfies $\pi_i^\sigma(I)\sigma_i(I,a) > 0$, then σ_i is a weakly dominated strategy.*

Proof. Since a is weakly dominated, there exists a strategy $\sigma_i' \in \Sigma_i$ such that $v_i(I, \sigma_{(I\to a)}) \leq v_i(I, (\sigma_i', \sigma_{-i}))$ for all opponent profiles $\sigma_{-i} \in \Sigma_{-i}$, and there exists an opponent profile σ'_{-i} such that $v_i(I, (\sigma_{i(I\to a)}, \sigma'_{-i})) < v_i(I, (\sigma_i', \sigma'_{-i}))$. Let $\hat{\sigma}_i$ be the strategy σ_i except at I, where $\hat{\sigma}_i(I,a) = 0$ and $\hat{\sigma}_i(I,b) = \sigma_i(I,b)/(1-\sigma_i(I,a))$ for all $b \in A(I)$, $b \neq a$. Next, for all $J \in \mathcal{I}_i$ and $b \in A(J)$, define

$$\sigma_i''(J,b) = \begin{cases} \frac{\sigma_i(I,a)\pi_i^{\sigma'}(I,J)\sigma_i'(J,b)+(1-\sigma_i(I,a))\pi_i^{\hat{\sigma}}(I,J)\hat{\sigma}_i(J,b)}{\sigma_i(I,a)\pi_i^{\sigma'}(I,J)+(1-\sigma_i(I,a))\pi_i^{\hat{\sigma}}(I,J)} & \text{if } J \in D(I) \\ & \text{(and arbitrary when the} \\ & \text{ denominator is zero),} \\ \sigma_i(J,b) & \text{if } J \notin D(I). \end{cases}$$

This is well-defined due to perfect recall and one can verify that $\sigma_i'' \in \Sigma_i$ is a valid strategy for player i. Now, fix $\sigma_{-i} \in \Sigma_{-i}$. Then,

$$\begin{aligned} u_i(\sigma_i, \sigma_{-i}) &= \sum_{z \in Z_I} \pi^\sigma(z) u_i(z) + \sum_{z \notin Z_I} \pi^\sigma(z) u_i(z) \\ &= \pi_i^\sigma(I) \sum_{b \in A(I)} \sigma_i(I,b) v_i(I, \sigma_{(I\to b)}) + \sum_{z \notin Z_I} \pi^\sigma(z) u_i(z) \\ &\leq \pi_i^\sigma(I)\sigma_i(I,a) v_i(I, (\sigma_i', \sigma_{-i})) \end{aligned}$$

$$
\begin{aligned}
&+ \pi_i^\sigma(I)(1-\sigma_i(I,a)) \sum_{\substack{b\in A(I)\\ b\neq a}} \hat{\sigma}_i(I,b) v_i(I,(\hat{\sigma}_{i(I\to b)},\sigma_{-i}))\\
&+ \sum_{z\notin Z_I} \pi^\sigma(z) u_i(z)\\
&= \pi_i^\sigma(I) v_i(I,(\sigma_i'',\sigma_{-i})) + \sum_{z\notin Z_I} \pi^\sigma(z) u_i(z)\\
&= u_i(\sigma_i'',\sigma_{-i}).
\end{aligned}
$$

Thus, $u_i(\sigma_i,\sigma_{-i}) \le u_i(\sigma_i'',\sigma_{-i})$ for all $\sigma_{-i}\in\Sigma_{-i}$. A similar argument shows that $u_i(\sigma_i,\sigma'_{-i}) < u_i(\sigma_i'',\sigma'_{-i})$, proving that σ_i is weakly dominated by σ_i''. ■

Next, we prove Theorem 4.1, using the fact that new iterative dominances only arise from removing actions and never from removing mixed strategies [15]:

Theorem 4.1. *Let $\sigma^1,\sigma^2,...$ be a sequence of strategy profiles in a normal-form game where all players' strategies are computed by regret minimization algorithms where for all $i\in N$, $a\in A_i$, if $R_i^T(a) < 0$ and $R_i^T(a) < \max_{b\in A_i} R_i^T(b)$, then $\sigma_i^{T+1}(a) = 0$. If σ_i is an iteratively strictly dominated strategy, then there exists an integer T_0 such that for all $T\ge T_0$,* supp$(\sigma_i) \nsubseteq$ supp(σ_i^T).

Proof. Let $a^1,a^2,...,a^k$ be iteratively strictly dominated actions (pure strategies) for players $j_1,j_2,...,j_k$ respectively that once removed in sequence yields strict domination of σ_i. Let $B_{-i} = A_{-i}\backslash\{a^1,a^2,...,a^k\}$ be the set of opponent actions other than $a^1,a^2,...,a^k$. Next, by iterative strict domination of σ_i and because the game is finite, there exists another strategy $\sigma_i'\in\Sigma_i$ such that

$$\epsilon = \min_{a_{-i}\in B_{-i}} u_i(\sigma_i',a_{-i}) - u_i(\sigma_i,a_{-i}) > 0,$$

so that $u_i(\sigma_i,a_{-i}) \le u_i(\sigma_i',a_{-i}) - \epsilon$ for all $a_{-i}\in B_{-i}$. Then,

$$
\begin{aligned}
\sum_{a\in A_i} \sigma_i(a) R_i^T(a) &= \sum_{a\in A_i} \sigma_i(a) R_i^T(a) - \sum_{a\in A_i} \sigma_i'(a) R_i^T(a) + \sum_{a\in A_i} \sigma_i'(a) R_i^T(a)\\
&= \sum_{a\in A_i} (\sigma_i(a) - \sigma_i'(a)) \sum_{t=1}^{T} \big(u_i(a,\sigma_{-i}^t) - u_i(\sigma^t)\big)\\
&\quad + \sum_{a\in A_i} \sigma_i'(a) R_i^T(a)\\
&= \sum_{t=1}^{T} \big(u_i(\sigma_i,\sigma_{-i}^t) - u_i(\sigma_i',\sigma_{-i}^t)\big) + \sum_{a\in A_i} \sigma_i'(a) R_i^T(a)\\
&= \sum_{\substack{\text{supp}(\sigma_{-i}^t)\nsubseteq B_{-i}\\ 1\le t\le T}} \big(u_i(\sigma_i,\sigma_{-i}^t) - u_i(\sigma_i',\sigma_{-i}^t)\big)\\
&\quad + \sum_{\substack{\text{supp}(\sigma_{-i}^t)\subseteq B_{-i}\\ 1\le t\le T}} \big(u_i(\sigma_i,\sigma_{-i}^t) - u_i(\sigma_i',\sigma_{-i}^t)\big) + \sum_{a\in A_i} \sigma_i'(a) R_i^T(a)\\
&= \sum_{\substack{\text{supp}(\sigma_{-i}^t)\nsubseteq B_{-i}\\ 1\le t\le T}} \big(u_i(\sigma_i,\sigma_{-i}^t) - u_i(\sigma_i',\sigma_{-i}^t)\big)
\end{aligned}
$$

$$
\begin{aligned}
&+ \sum_{\substack{\text{supp}(\sigma_{-i}^t)\subseteq B_{-i} \\ 1\le t\le T}} \sum_{a_{-i}\in B_{-i}} \sigma_{-i}^t(a_{-i})\left(u_i(\sigma_i, a_{-i}) - u_i(\sigma_i', a_{-i})\right) \\
&+ \sum_{a\in A_i} \sigma_i'(a) R_i^T(a), \text{ where } \sigma_{-i}(a_{-i}) = \prod_{j\neq i} \sigma_j(a_j) \\
\le &\sum_{\substack{\text{supp}(\sigma_{-i}^t)\not\subseteq B_{-i} \\ 1\le t\le T}} \big(u_i(\sigma_i, \sigma_{-i}^t) - u_i(\sigma_i', \sigma_{-i}^t)\big) \\
&+ \sum_{\substack{\text{supp}(\sigma_{-i}^t)\subseteq B_{-i} \\ 1\le t\le T}} (-\epsilon) + \max_{a\in A_i} R_i^T(a). \qquad \text{(B.1)}
\end{aligned}
$$

We claim that there exists an integer T_0 such that for all $T \ge T_0$, there exists $a \in \text{supp}(\sigma_i)$ such that $R_i^T(a) < 0$ and $R_i^T(a) < \max_{b\in A_i} R_i^T(b)$. By our assumption, this implies that for all $T \ge T_0$, there exists an action $a \in \text{supp}(\sigma_i)$ such that $a \notin \text{supp}(\sigma_i^T)$, establishing the theorem.

To complete the proof, it remains to establish the claim, which we prove by strong induction on k, the number of actions removed to yield iterative strict dominance of σ_i. For the base case $k = 0$, we have $B_{-i} = A_{-i}$, and so by equation (B.1) we have

$$
\begin{aligned}
\min_{a\in\text{supp}(\sigma_i)} R_i^T(a) &\le \sum_{a\in A_i} \sigma_i(a) R_i^T(a) \\
&\le -\epsilon T + \max_{a\in A_i} R_i^T(a) \qquad \text{(B.2)} \\
&\le -\epsilon T + R_i^{T,+}.
\end{aligned}
$$

Dividing both sides by T and taking the limit superior gives

$$
\begin{aligned}
\limsup_{T\to\infty} \frac{1}{T} \min_{a\in\text{supp}(\sigma_i)} R_i^T(a) &\le -\epsilon + \limsup_{T\to\infty} \frac{R_i^{T,+}}{T} \\
&= -\epsilon \\
&< 0.
\end{aligned}
$$

Thus, there exists an integer T_0 such that for all $T \ge T_0$, $R_i^T(a^*) < 0$ where $a^* = \text{argmin}_{a\in\text{supp}(\sigma_i)} R_i^T(a)$. Also, by equation (B.2), $R_i^T(a^*) \le -\epsilon T + \max_{a\in A_i} R_i^T(a) < \max_{a\in A_i} R_i^T(a)$, completing the base case.

For the induction step, we may assume that there exist integers $T_1, ..., T_k$ such that for all $1 \le \ell \le k$, $T \ge T_\ell$, $R_{j_\ell}^T(a^\ell) < 0$ and $R_{j_\ell}^T(a^\ell) < \max_{b\in A_{j_\ell}} R_{j_\ell}^T(b)$. This means that for all $T \ge T_0' = \max\{T_1, ..., T_k\}$, $a^\ell \notin \text{supp}(\sigma_{j_\ell}^T)$ for all $1 \le \ell \le k$. Hence, $\text{supp}(\sigma_{-i}^T) \subseteq B_{-i}$ for all $T \ge T_0'$. Therefore, again setting $a^* = \text{argmin}_{a\in\text{supp}(\sigma_i)} R_i^T(a)$, by equation (B.1) we have

$$
\begin{aligned}
R_i^T(a^*) &\le \sum_{a\in A_i} \sigma_i(a) R_i^T(a) \\
&\le \sum_{\substack{\text{supp}(\sigma_{-i}^t)\not\subseteq B_{-i} \\ 1\le t\le T}} \big(u_i(\sigma_i, \sigma_{-i}^t) - u_i(\sigma_i', \sigma_{-i}^t)\big) \\
&\quad + \sum_{\substack{\text{supp}(\sigma_{-i}^t)\subseteq B_{-i} \\ 1\le t\le T}} (-\epsilon) + \max_{a\in A_i} R_i^T(a)
\end{aligned}
$$

$$\leq T_0'\Delta_i - \epsilon(T - T_0') + \max_{a\in A_i} R_i^T(a), \text{ where } \Delta_i = \max_{a,a'\in A} u_i(a) - u_i(a') \tag{B.3}$$
$$\leq T_0'\Delta_i - \epsilon(T - T_0') + R_i^{T,+}.$$

Dividing both sides by T and taking the limit superior gives

$$\begin{aligned}\limsup_{T\to\infty} \frac{R_i^T(a^*)}{T} &\leq \limsup_{T\to\infty}\left(\frac{T_0'\Delta_i}{T} - \frac{\epsilon(T-T_0')}{T} + \frac{R_i^{T,+}}{T}\right)\\ &= -\epsilon\\ &< 0.\end{aligned}$$

Thus, there exists an integer T_0 such that for all $T \geq T_0$, $T_0'\Delta_i < \epsilon(T - T_0')$ and $R_i^T(a^*) < 0$. By equation (B.3), this also means that for $T \geq T_0$, $R_i^T(a^*) < \max_{a\in A_i} R_i^T(a)$, completing the induction step. This establishes the claim and completes the proof. ■

Before proving Theorems 4.2 and 4.3, we need an additional lemma. For $\sigma_i \in \Sigma_i$ and $I \in \mathcal{I}_i$, define the **full counterfactual regret** for σ_i at I to be

$$R_{i,\text{full}}^T(I,\sigma_i) = \sum_{t=1}^T (v_i(I,(\sigma_i,\sigma_{-i}^t)) - v_i(I,\sigma^t)). \tag{B.4}$$

We begin by relating full counterfactual regret to a sum over cumulative counterfactual regrets. A similar step was part of the original CFR analysis [74], but we improve the analysis here by equating terms rather than simply bounding them. Perfect recall is again required to ensure $\pi_i^\sigma(I,I')$ is well-defined. This lemma will also be used to prove Theorem 5.1 in Appendix C.

Lemma B.1. *In an extensive-form game with perfect recall,*

$$R_{i,\text{full}}^T(I,\sigma_i) = \sum_{I'\in D(I)} \pi_i^\sigma(I,I') \sum_{a\in A(I')} \sigma_i(I',a) R_i^T(I',a).$$

Proof. We prove the lemma by strong induction on $|D(I)|$. For $I \in \mathcal{I}_i$ and $a \in A(I)$, define

$$\begin{aligned}S(I,a) = \{I' \in \mathcal{I}_i \mid &\exists h \in I, h' \in I' \text{ where } ha \sqsubseteq h'\\ &\text{and } \nexists h'' \in H_i \text{ where } ha \sqsubseteq h'' \sqsubset h'\}\end{aligned}$$

to be the set of all possible successor information sets for player i after taking action a at I. In addition, define $Z(I,a)$ to be the set of terminal histories where the last action taken by player i was a at I. To begin,

$$\begin{aligned}R_{i,\text{full}}^T(I,\sigma_i) &= \sum_{t=1}^T v_i(I,(\sigma_i,\sigma_{-i}^t)) - \sum_{t=1}^T v_i(I,\sigma^t)\\ &= \sum_{t=1}^T \sum_{a\in A(I)} \sigma_i(I,a) v_i(I,(\sigma_{i(I\to a)},\sigma_{-i}^t)) - \sum_{t=1}^T v_i(I,\sigma^t)\\ &= \sum_{a\in A(I)} \sigma_i(I,a) \sum_{t=1}^T \left(\sum_{z\in Z(I,a)} \pi_{-i}^{\sigma^t}(z) u_i(z)\right.\end{aligned}$$

$$+\sum_{I' \in S(I,a)} v_i(I', (\sigma_i, \sigma_{-i}^t))\Bigg) - \sum_{t=1}^{T} v_i(I, \sigma^t). \tag{B.5}$$

For the base case $D(I) = \{I\}$, we have $S(I,a) = \emptyset$ and $Z(I,a) = Z_I$, and so the right hand side of equation (B.5) reduces to $\sum_{a \in A(I)} \sigma_i(I,a) R_i^T(I,a)$ as desired. For the induction step, note that $|D(I')| < |D(I)|$ for all $I' \in S(I,a)$, and so we may apply the induction hypothesis to get, for all $I' \in S(I,a)$,

$$\begin{aligned}
\sum_{t=1}^{T} v_i(I', (\sigma_i, \sigma_{-i}^t)) &= R_{i,\text{full}}^T(I', \sigma_i) + \sum_{t=1}^{T} v_i(I', \sigma^t) \\
&= \sum_{I'' \in D(I')} \pi_i^\sigma(I', I'') \sum_{b \in A(I'')} \sigma(I'', b) R_i^T(I'', b) \\
&\quad + \sum_{t=1}^{T} v_i(I', \sigma^t).
\end{aligned}$$

Finally, substituting into equation (B.5), we have

$$\begin{aligned}
R_{i,\text{full}}^T(I, \sigma_i) &= \sum_{a \in A(I)} \sigma_i(I,a) \Bigg[\sum_{t=1}^{T} \sum_{z \in Z(I,a)} \pi_{-i}^{\sigma^t}(z) u_i(z) \\
&\quad + \sum_{I' \in S(I,a)} \Bigg(\sum_{I'' \in D(I')} \pi_i^\sigma(I', I'') \sum_{b \in A(I'')} \sigma_i(I'', b) R_i^T(I'', b) \\
&\quad + \sum_{t=1}^{T} v_i(I', \sigma^t) \Bigg) \Bigg] - \sum_{t=1}^{T} v_i(I, \sigma^t) \\
&= \sum_{a \in A(I)} \sigma_i(I,a) \sum_{t=1}^{T} v_i(I, \sigma_{(I \to a)}^t) - \sum_{t=1}^{T} v_i(I, \sigma^t) \\
&\quad + \sum_{a \in A(I)} \sigma_i(I,a) \sum_{I' \in S(I,a)} \Bigg(\sum_{I'' \in D(I')} \pi_i^\sigma(I', I'') \sum_{b \in A(I'')} \sigma_i(I'', b) R_i^T(I'', b) \Bigg) \\
&= \sum_{a \in A(I)} \sigma_i(I,a) R_i^T(I,a) \\
&\quad + \sum_{\substack{I' \in D(I) \\ I' \neq I}} \pi_i^\sigma(I, I') \sum_{b \in A(I')} \sigma_i(I', b) R_i^T(I', b) \\
&= \sum_{I' \in D(I)} \pi_i^\sigma(I, I') \sum_{a \in A(I')} \sigma_i(I', a) R_i^T(I', a),
\end{aligned}$$

completing the proof. ■

Corollary B.1. *In an extensive-form game with perfect recall,*

$$R_{i,\text{full}}^T(I, \sigma_i) \leq \Delta_i |D(I)| \sqrt{|A(\mathcal{I}_i)| T}.$$

Proof. By Lemma B.1,

$$R_{i,\text{full}}^T(I, \sigma_i) = \sum_{I' \in D(I)} \pi_i^\sigma(I, I') \sum_{a \in A(I')} \sigma_i(I', a) R_i^T(I', a)$$

$$\begin{aligned}&\le \sum_{I' \in D(I)} \max_{a \in A(I)} R_i^{T,+}(I', a) \\ &\le |D(I)| \Delta_i \sqrt{|A(\mathcal{I}_i)| T}\end{aligned}$$

by Theorem 2.2 since regret matching is used at each $I' \in D(I)$. ■

Theorem 4.2. *Let $\sigma^1, \sigma^2, ...$ be strategy profiles generated by CFR in an extensive-form game with perfect recall, let $I \in \mathcal{I}_i$, and let a be an iteratively strictly dominated action at I, where removal in sequence of the iteratively strictly dominated actions $a_1, ..., a_k$ at $I_1, ..., I_k$ respectively yields iterative dominance of $a_{k+1} = a$. If for $1 \le \ell \le k+1$, there exist real numbers $\delta_\ell, \gamma_\ell > 0$ and an integer T_ℓ such that for all $T \ge T_\ell$, $|\Sigma_{\delta_\ell}(I_\ell) \cap \{\sigma^t \mid T_\ell \le t \le T\}| \ge \gamma_\ell T$, then*

(i) *there exists an integer T_0 such that for all $T \ge T_0$, $R_i^T(I, a) < 0$,*

(ii) *if $\lim_{T\to\infty} x^T/T = 0$, then $\lim_{T\to\infty} y^T(I,a)/T = 0$, where $y^T(I,a)$ is the number of iterations $1 \le t \le T$ satisfying $\sigma^t(I,a) > 0$, and*

(iii) *if $\lim_{T\to\infty} x^T/T = 0$, then $\lim_{T\to\infty} \pi_i^{\bar{\sigma}^T}(I) \bar{\sigma}_i^T(I,a) = 0$.*

Proof. We will first prove parts (i) and (ii) by strong induction on k, followed by proving (iii) from (ii). For $\delta \ge 0$, let $\hat{\Sigma}_\delta(I) = \{\sigma \in \Sigma_\delta(I) \mid \sigma(I_\ell, a_\ell) = 0, 1 \le \ell \le k\}$ be the set of strategies in $\Sigma_\delta(I)$ that do not play $a_1, ..., a_k$. By iterative strict domination of a, there exists $\sigma_i' \in \Sigma_i$ such that $v_i(I, \sigma_{(I\to a)}) < v_i(I, (\sigma_i', \sigma_{-i}))$ for all $\sigma \in \hat{\Sigma}_0(I)$. Next, let $\delta = \delta_{k+1}$ and $\gamma = \gamma_{k+1}$. Then, since $\hat{\Sigma}_\delta(I)$ is a closed and bounded set and $v_i(I, \cdot)$ is continuous, by the Balzano-Weierstrass Theorem there exists an $\epsilon > 0$ such that $v_i(I, \sigma_{(I\to a)}) \le v_i(I, (\sigma_i', \sigma_{-i})) - \epsilon$ for all $\sigma \in \hat{\Sigma}_\delta(I)$. Then, with $T_0' = \max_{1\le\ell\le k+1} T_\ell$,

$$\begin{aligned}R_i^T(I,a) &= R_i^T(I,a) - R_{i,\text{full}}^T(I,\sigma_i') + R_{i,\text{full}}^T(I,\sigma_i') \\ &= \sum_{t=1}^{T} \Big(v_i(I, \sigma^t_{(I\to a)}) - v_i(I, (\sigma_i', \sigma_{-i}^t)) \Big) + R_{i,\text{full}}^T(I,\sigma_i') \\ &= \sum_{t=1}^{T_0'-1} \Big(v_i(I, \sigma^t_{(I\to a)}) - v_i(I, (\sigma_i', \sigma_{-i}^t)) \Big) \\ &\quad + \sum_{\substack{T_0' \le t \le T \\ \sigma^t \notin \hat{\Sigma}_0(I)}} \Big(v_i(I, \sigma^t_{(I\to a)}) - v_i(I, (\sigma_i', \sigma_{-i}^t)) \Big) \\ &\quad + \sum_{\substack{T_0' \le t \le T \\ \sigma^t \in \hat{\Sigma}_\delta(I)}} \Big(v_i(I, \sigma^t_{(I\to a)}) - v_i(I, (\sigma_i', \sigma_{-i}^t)) \Big) \\ &\quad + \sum_{\substack{T_0' \le t \le T \\ \sigma^t \in \hat{\Sigma}_0(I) \backslash \hat{\Sigma}_\delta(I)}} \Big(v_i(I, \sigma^t_{(I\to a)}) - v_i(I, (\sigma_i', \sigma_{-i}^t)) \Big) + R_{i,\text{full}}^T(I,\sigma_i'). \qquad \text{(B.6)}\end{aligned}$$

For the base case $k = 0$, we have $\hat{\Sigma}_0(I) = \Sigma$ and $\hat{\Sigma}_\delta(I) = \Sigma_\delta(I)$. Choose T_0 to be any integer greater than $\max\{T_0', \Delta_i^2|D(I)|^2|A(\mathcal{I}_i)|/\epsilon^2\gamma^2\}$ so that for all $T \geq T_0$,

$$\begin{aligned}
R_i^T(I,a) &= \sum_{t=1}^{T_0'-1}\Big(v_i(I,\sigma^t_{(I\to a)}) - v_i(I,(\sigma_i',\sigma_{-i}^t))\Big) \\
&\quad+ \sum_{\substack{T_0'\leq t\leq T\\ \sigma^t\in\Sigma_\delta(I)}}\Big(v_i(I,\sigma^t_{(I\to a)}) - v_i(I,(\sigma_i',\sigma_{-i}^t))\Big) \\
&\quad+ \sum_{\substack{T_0'\leq t\leq T\\ \sigma^t\notin\Sigma_\delta(I)}}\Big(v_i(I,\sigma^t_{(I\to a)}) - v_i(I,(\sigma_i',\sigma_{-i}^t))\Big) + R^T_{i,\text{full}}(I,\sigma_i') \\
&\leq -\epsilon|\Sigma_\delta(I)\cap\{\sigma^t \mid T_0\leq t\leq T\}| + R^T_{i,\text{full}}(I,\sigma_i') \\
&\leq -\epsilon\gamma T + \Delta_i|D(I)|\sqrt{|A(\mathcal{I}_i)|T} \text{ by Corollary B.1} \\
&< 0
\end{aligned}$$

by choice of T_0. This establishes part (i) of the base case. For part (ii), since CFR applies regret matching at I, by equation (2.4) it follows that for all $T \geq T_0$, either $\sum_{b\in A(I)} R_i^{T,+}(I,b) = 0$ or $\sigma_i^{T+1}(I,a) = 0$. Thus,

$$\begin{aligned}
\lim_{T\to\infty}\frac{y^T(I,a)}{T} &= \lim_{T\to\infty}\frac{y^{T_0}(I,a) + (y^T(I,a) - y^{T_0}(I,a))}{T} \\
&\leq \lim_{T\to\infty}\frac{y^{T_0}(I,a) + x^T}{T} \\
&= 0.
\end{aligned}$$

Thus, (ii) holds and we have established the base case of our induction.

For the induction step, we now assume that parts (i) and (ii) hold for all $a_1, ..., a_k$. We will show that there exists an integer T_0 such that for all $T \geq T_0$, $R_i^T(I,a) < 0$. This will establish part (i), and part (ii) will then follow as before to complete the induction step.

Firstly, note that

$$\sum_{\substack{T_0'\leq t\leq T\\ \sigma^t\in\hat\Sigma_0(I)\backslash\hat\Sigma_\delta(I)}}\Big(v_i(I,\sigma^t_{(I\to a)}) - v_i(I,(\sigma_i',\sigma_{-i}^t))\Big) \leq 0$$

by iterative domination of a. Secondly,

$$\begin{aligned}
&\sum_{\substack{T_0'\leq t\leq T\\ \sigma^t\in\hat\Sigma_\delta(I)}}\Big(v_i(I,\sigma^t_{(I\to a)}) - v_i(I,(\sigma_i',\sigma_{-i}^t))\Big) \\
&\qquad\leq -\epsilon|\hat\Sigma_\delta(I)\cap\{\sigma^t \mid T_0\leq t\leq T\}| \\
&\qquad= -\epsilon\Big(|\Sigma_\delta(I)\cap\{\sigma^t \mid T_0\leq t\leq T\}| - |(\Sigma_\delta(I)\backslash\hat\Sigma_\delta(I))\cap\{\sigma^t \mid T_0\leq t\leq T\}|\Big) \\
&\qquad\leq -\epsilon\gamma T + \epsilon\sum_{\ell=1}^k y^T(I_\ell,a_\ell).
\end{aligned}$$

Thirdly,

$$\sum_{\substack{T_0' \leq t \leq T \\ \sigma^t \notin \hat{\Sigma}_0(I)}} \left(v_i(I, \sigma^t_{(I \to a)}) - v_i(I, (\sigma_i', \sigma^t_{-i})) \right) \leq \Delta_i \sum_{\ell=1}^{k} y^T(I_\ell, a_\ell).$$

Thus, substituting these three inequalities and Corollary B.1 into equation (B.6) gives

$$\begin{aligned} R_i^T(I,a) \leq & \sum_{t=1}^{T_0'-1} \left(v_i(I, \sigma^t_{(I \to a)}) - v_i(I, (\sigma_i', \sigma^t_{-i})) \right) \\ & + \Delta_i \sum_{\ell=1}^{k} y^T(I_\ell, a_\ell) - \epsilon\gamma T + \epsilon \sum_{\ell=1}^{k} y^T(I_\ell, a_\ell) + \Delta_i |D(I)| \sqrt{|A(\mathcal{I}_i)| T}. \end{aligned}$$

Dividing both sides by T and taking the limit superior gives

$$\begin{aligned} \limsup_{T \to \infty} \frac{R_i^T(I,a)}{T} &\leq \sum_{t=1}^{T_0'-1} \left(v_i(I, \sigma^t_{(I \to a)}) - v_i(I, (\sigma_i', \sigma^t_{-i})) \right) \limsup_{T \to \infty} \frac{1}{T} \\ &\quad + (\Delta_i + \epsilon) \sum_{\ell=1}^{k} \limsup_{T \to \infty} \frac{y^T(I_\ell, a_\ell)}{T} - \epsilon\gamma + \Delta_i |D(I)| \sqrt{|A(\mathcal{I}_i)|} \limsup_{T \to \infty} \frac{1}{\sqrt{T}} \\ &= -\epsilon\gamma \\ &< 0 \end{aligned}$$

by applying part (ii) of the induction hypothesis. Therefore, there exists an integer T_0 such that for all $T \geq T_0$, $R_i^T(I,a)/T < 0$ and thus $R_i^T(I,a) < 0$, completing the induction step.

Parts (i) and (ii) are now proven. It remains to prove (iii). To that end,

$$\begin{aligned} \lim_{T \to \infty} \pi_i^{\bar{\sigma}^T}(I) \bar{\sigma}_i^T(I,a) &= \lim_{T \to \infty} \left(\frac{1}{T} \sum_{t=1}^{T} \pi_i^{\sigma^t}(I) \right) \frac{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I) \sigma_i^t(I,a)}{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)} \\ &= \lim_{T \to \infty} \frac{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I) \sigma_i^t(I,a)}{T} \\ &\leq \lim_{T \to \infty} \frac{y^T(I,a)}{T} \\ &= 0 \end{aligned}$$

by part (ii). Since $\pi_i^{\bar{\sigma}^T}(I) \bar{\sigma}_i^T(I,a)$ is nonnegative, it follows that $\lim_{T \to \infty} \pi_i^{\bar{\sigma}^T}(I) \bar{\sigma}_i^T(I,a) = 0$, completing the proof. ■

Theorem 4.3. *Let $\sigma^1, \sigma^2, \ldots$ be strategy profiles generated by CFR in an extensive-form game with perfect recall, and let σ_i be an iteratively strictly dominated strategy. Then,*

(i) *there exists an integer T_0 such that for all $T \geq T_0$, there exist $I \in \mathcal{I}_i$, $a \in A(I)$ such that $\pi_i^{\sigma}(I)\sigma_i(I,a) > 0$ and $R_i^T(I,a) < 0$, and*

(ii) *if $\lim_{T \to \infty} x^T/T = 0$, then $\lim_{T \to \infty} y^T(\sigma_i)/T = 0$, where $y^T(\sigma_i)$ is the number of iterations $1 \leq t \leq T$ satisfying supp$(\sigma_i) \subseteq$ supp(σ_i^t).*

Proof. Let $s^1_{j_1}, s^2_{j_2}, ..., s^k_{j_k}$ be iteratively strictly dominated pure strategies that once removed in sequence yields strict domination of σ_i. Let $S_{-i} = \mathcal{S}_{-i} \backslash \{s^1_{j_1}, s^2_{j_2}, ..., s^k_{j_k}\}$ be the set of opponent pure strategy profiles that do not play any of $s^1_{j_1}, s^2_{j_2}, ..., s^k_{j_k}$. Next, by iterative strict domination of σ_i and because the game is finite, there exists another strategy $\sigma_i' \in \Sigma_i$ such that

$$\epsilon = \min_{s_{-i} \in S_{-i}} u_i(\sigma_i', s_{-i}) - u_i(\sigma_i, s_{-i}) > 0,$$

so that $u_i(\sigma_i, s_{-i}) \leq u_i(\sigma_i', s_{-i}) - \epsilon$ for all $s_{-i} \in S_{-i}$.

For $\hat{\sigma}_i \in \Sigma_i$, define $R^T_{i,\text{full}}(\hat{\sigma}_i) = \sum_{t=1}^T \big(u_i(\hat{\sigma}_i, \sigma^t_{-i}) - u_i(\sigma^t)\big)$. Note that

$$R^T_{i,\text{full}}(\hat{\sigma}_i) = \sum_{I \in \mathcal{I}^0_i} R^T_{i,\text{full}}(I, \hat{\sigma}_i),$$

where $\mathcal{I}^0_i = \{I \in \mathcal{I}_i \mid \forall h \in I, h' \sqsubset h, P(h') \neq i\}$ is the set of all possible first information sets for player i reached. So, by Corollary B.1, $R^T_{i,\text{full}}(\hat{\sigma}_i) \leq \Delta_i |\mathcal{I}_i| \sqrt{|A(\mathcal{I}_i)| T}$ for all $\hat{\sigma}_i \in \Sigma_i$. Then by Lemma B.1, we have

$$\begin{aligned}
&\sum_{I \in \mathcal{I}_i} \pi^\sigma_i(I) \sum_{a \in A(I)} \sigma_i(I,a) R^T_i(I,a) \\
&\qquad = R^T_{i,\text{full}}(\sigma_i) - R^T_{i,\text{full}}(\sigma_i') + R^T_{i,\text{full}}(\sigma_i') \\
&\qquad = \sum_{t=1}^T \big(u_i(\sigma_i, \sigma^t_{-i}) - u_i(\sigma_i', \sigma^t_{-i})\big) + R^T_{i,\text{full}}(\sigma_i') \\
&\qquad = \sum_{\substack{\text{supp}(\sigma^t_{-i}) \subseteq S_{-i} \\ 1 \leq t \leq T}} \sum_{s_{-i} \in S_{-i}} \sigma^t_{-i}(s_{-i}) \left(u_i(\sigma_i, s_{-i}) - u_i(\sigma_i', s_{-i})\right) \\
&\qquad\quad + \sum_{\substack{\text{supp}(\sigma^t_{-i}) \not\subseteq S_{-i} \\ 1 \leq t \leq T}} \big(u_i(\sigma_i, \sigma^t_{-i}) - u_i(\sigma_i', \sigma^t_{-i})\big) + R^T_{i,\text{full}}(\sigma_i'), \\
&\qquad\quad \text{where } \sigma_{-i}(s_{-i}) = \prod_{\substack{j \neq i \\ I \in \mathcal{I}_j}} \sigma_j(I, s_j(I)) \\
&\qquad \leq -\epsilon \left(T - \sum_{\ell=1}^k y^T(s^\ell_{j_\ell})\right) + \Delta_i \sum_{\ell=1}^k y^T(s^\ell_{j_\ell}) + \Delta_i |\mathcal{I}_i| \sqrt{|A(\mathcal{I}_i)| T}.
\end{aligned} \tag{B.7}$$

We claim that

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{I \in \mathcal{I}_i} \pi^\sigma_i(I) \sum_{a \in A(I)} \sigma_i(I,a) R^T_i(I,a) < 0.$$

Assuming the claim holds, because $(1/T)$, $\pi^\sigma_i(I)$, and $\sigma_i(I,a)$ are nonnegative, it follows that there exists an integer T_0 such that for all $T \geq T_0$, there exist $I \in \mathcal{I}_i$, $a \in A(I)$ such that $\pi^\sigma_i(I)\sigma_i(I,a) > 0$ and $R^T_i(I,a) < 0$, establishing (i). For part (ii), note that part (i) and equation (2.4) imply that for all $T \geq T_0$, either $\sum_{b \in A(I)} R^{T,+}_i(I,b) = 0$ or supp$(\sigma_i) \not\subseteq$ supp(σ^T_i). Thus,

$$\begin{aligned}
\lim_{T \to \infty} \frac{y^T(\sigma_i)}{T} &= \lim_{T \to \infty} \frac{y^{T_0}(\sigma_i) + (y^T(\sigma_i) - y^{T_0}(\sigma_i))}{T} \\
&\leq \lim_{T \to \infty} \frac{y^{T_0}(\sigma_i) + x^T}{T}
\end{aligned}$$

$$= 0,$$

establishing part (ii).

To complete the proof, it remains to prove the claim, which we will prove by induction on k. For the base case $k = 0$, equation (B.7) gives

$$\begin{aligned}\limsup_{T\to\infty} \frac{1}{T} \sum_{I\in\mathcal{I}_i} \pi_i^\sigma(I) \sum_{a\in A(I)} \sigma_i(I,a) R_i^T(I,a) &\leq \limsup_{T\to\infty} -\epsilon + \frac{\Delta_i|\mathcal{I}_i|\sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} \\ &= -\epsilon \\ &< 0.\end{aligned}$$

For the induction step, we may assume that parts (i) and (ii) hold for all $s_{j_1}^1, s_{j_2}^2, ..., s_{j_k}^k$. Then equation (B.7) implies

$$\begin{aligned}\limsup_{T\to\infty} \frac{1}{T} \sum_{I\in\mathcal{I}_i} \pi_i^\sigma(I) \sum_{a\in A(I)} \sigma_i(I,a) R_i^T(I,a) \leq -\epsilon + (\epsilon + \Delta_i) \sum_{\ell=1}^{k} \limsup_{T\to\infty} \frac{y^T(s_{j_\ell}^\ell)}{T} \\ + \limsup_{T\to\infty} \frac{\Delta_i|\mathcal{I}_i|\sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} \\ = -\epsilon \\ < 0,\end{aligned}$$

proving the claim. ■

Appendix C

Proofs for Chapter 5: Generalized Sampling and Improved Monte Carlo CFR

In this appendix, we prove Theorems 5.1 through 5.6 and prove Proposition 5.1. In addition, we state and prove a tighter regret bound than that of Theorem 5.4 when some structure on the estimate $\hat{v}_i$ is assumed. We also present a more general version of our probing algorithm. As we did in Chapter 5, we assume perfect recall throughout this appendix.

C.1 Preliminaries

To begin, we state a number of results that we will use in our analysis throughout this appendix. All of these results are provided and proved by Lanctot *et al.* in their technical report [55], thus we do not repeat the proofs here.

Lemma C.1 (Lanctot *et al.* [55], Lemma 2). *For any random variable X,*

$$\textbf{Prob}\left[|X| \geq k\sqrt{\mathbf{E}[X^2]}\right] \leq \frac{1}{k^2}.$$

Lemma C.2 (Lanctot *et al.* [55], Lemma 5). *If $b_1, ..., b_k$ are nonnegative real numbers where $\sum_{i=1}^{k} b_i^2 = S$, then $\sum_{i=1}^{k} b_i \leq \sqrt{Sk}$.*

For the next lemma, recall the definition of $\mathcal{B}_i$ from Section 2.2.2 and the definition of $\pi_{-i}^{\sigma}(I) = \sum_{h \in I} \pi_{-i}^{\sigma}(h)$.

Lemma C.3 (Lanctot *et al.* [55], Lemma 16). *For any strategy profile σ and for any $B \in \mathcal{B}_i$, in a game with perfect recall,*

$$\sum_{I \in B} \pi_{-i}^{\sigma}(I) \leq 1.$$

The final lemma of this section also holds true in imperfect recall games and will be used in both this appendix and later in Appendix D.

Lemma C.4 (Lanctot *et al.* [55], Theorem 6). *Define* Δ_t *to be* $\max_{a,b\in A(I)}(\hat{v}_i(I,\sigma^t_{(I\to a)}) - \hat{v}_i(I,\sigma^t_{(I\to b)}))$. *When using regret matching at* I*,*

$$\sum_{a\in A(I)} \left(\hat{R}_i^{T,+}(I,a)\right)^2 \leq |A(I)| \sum_{t=1}^{T} \Delta_t^2.$$

The proof of Lemma C.4 is analogous to the proof of Theorem 2.2 given in Appendix A.

C.2 New CFR Bounds

In this section, we prove Theorems 5.1, 5.2, 5.3, and 5.5. To begin, Theorem 5.1 is simply a special case of Lemma B.1 established in the previous appendix. Again, perfect recall is required for $\pi_i^{\sigma^*}(I)$ to be well-defined.

Theorem 5.1. *In an extensive-form game with perfect recall,*

$$R_i^T = \sum_{I\in\mathcal{I}_i} \pi_i^{\sigma^*}(I) R_i^T(I,\sigma_i^*).$$

Proof. Let $\mathcal{I}_i^0 = \{I \in \mathcal{I}_i \mid \forall h \in I, h' \sqsubset h, P(h') \neq i\}$ be the set of all possible first information sets for player i reached. Then,

$$\begin{aligned}
R_i^T &= \max_{\sigma_i'\in\Sigma_i} \sum_{t=1}^{T} (u_i(\sigma_i',\sigma_{-i}^t) - u_i(\sigma_i^t,\sigma_{-i}^t)) \\
&= \sum_{I\in\mathcal{I}_i^0} R_{i,\text{full}}^T(I,\sigma_i^*) \text{ as defined by equation (B.4)} \\
&= \sum_{I\in\mathcal{I}_i^0} \sum_{I'\in D(I)} \pi_i^{\sigma^*}(I,I') \sum_{a\in A(I')} \sigma_i^*(I',a) R_i^T(I',a) \text{ by Lemma B.1} \\
&= \sum_{I\in\mathcal{I}_i} \pi_i^{\sigma^*}(I) R_i^T(I,\sigma_i^*),
\end{aligned}$$

where the last line follows since $\pi_i^\sigma(I) = 1$ for all $I \in \mathcal{I}_i^0$ and by definition of $R_i^T(I,\sigma_i)$. ■

Theorem 5.2. *When using Vanilla CFR in a game with perfect recall, average regret is bounded by*

$$\frac{R_i^T}{T} \leq \frac{\Delta_i M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}}.$$

Proof.

$$\begin{aligned}
R_i^T &= \sum_{I\in\mathcal{I}_i} \pi_i^{\sigma^*}(I) R_i^T(I,\sigma_i^*) \text{ by Theorem 5.1} \\
&= \sum_{I\in\mathcal{I}_i} \pi_i^{\sigma^*}(I) \sum_{a\in A(I)} \sigma_i^*(I,a) R_i^T(I,a) \\
&= \sum_{I\in\mathcal{I}_i} \pi_i^{\sigma^*}(I) \max_{a\in A(I)} R_i^T(I,a) \\
&\leq \sum_{I\in\mathcal{I}_i} \pi_i^{\sigma^*}(I) \sqrt{\sum_{a\in A(I)} (R_i^{T,+}(I,a))^2}
\end{aligned}$$

$$\leq \sum_{I \in \mathcal{I}_i} \pi_i^{\sigma^*}(I)\Delta_i \sqrt{|A(I)|}\sqrt{\sum_{t=1}^{T}(\pi_{-i}^{\sigma^t}(I))^2}$$

by Lemma C.4 with $\Delta_t = \Delta_i \pi_{-i}^{\sigma^t}(I)$

$$\leq \Delta_i \sqrt{|A(\mathcal{I}_i)|} \sum_{B \in \mathcal{B}_i} \pi_i^{\sigma^*}(B) \sum_{I \in B} \sqrt{\sum_{t=1}^{T}(\pi_{-i}^{\sigma^t}(I))^2}$$

$$\leq \Delta_i \sqrt{|A(\mathcal{I}_i)|} \sum_{B \in \mathcal{B}_i} \pi_i^{\sigma^*}(B) \sqrt{|B| \sum_{t=1}^{T} \sum_{I \in B} \pi_{-i}^{\sigma_t}(I)} \text{ by Lemma C.2}$$

$$\leq \Delta_i \sqrt{|A(\mathcal{I}_i)|} \sum_{B \in \mathcal{B}_i} \pi_i^{\sigma^*}(B) \sqrt{|B|T} \text{ by Lemma C.3}$$

$$= \Delta_i \sqrt{|A(\mathcal{I}_i)|T} M_i(\sigma_i^*).$$

Dividing both sides by T gives the result. ■

We now prove a general, probabilistic bound that can be applied to any generalized sampling algorithm. We then apply this bound to our MCCFR algorithms to prove Theorems 5.3 and 5.5. This lemma will also be used at the end of this appendix to prove Theorem 5.6. We again require perfect recall so that $\mathcal{B}_i$ and $M_i(\sigma_i^*)$ are well-defined and so that we can also apply Lemma C.3 and Theorem 5.1.

Lemma C.5. *Let* $p, \delta \in (0, 1]$. *When using any generalized sampling algorithm with unbiased estimated counterfactual values* $\hat{v}_i(I, \sigma)$*, sampled independently on each iteration, in a game with perfect recall, if there exists a bound* $\hat{\Delta}_i(I, \sigma)$ *on the difference between any two estimates* $\hat{v}_i(I, \sigma_{(I \to a)}) - \hat{v}_i(I, \sigma_{(I \to b)}) \leq \hat{\Delta}_i(I, \sigma)$*, and if*

$$\sum_{I \in B} (\hat{\Delta}_i(I, \sigma))^2 \leq \frac{(\Delta_i)^2}{\delta^2} \tag{C.1}$$

for all $B \in \mathcal{B}_i$*, then with probability at least* $1 - p$*, average regret is bounded by*

$$\frac{R_i^T}{T} \leq \left(M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|} + \frac{2\sqrt{|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}} \right) \left(\frac{1}{\delta} \right) \frac{\Delta_i}{\sqrt{T}}.$$

Proof. Our proof follows that of [55, Theorem 7]. To start, we may assume σ_i^* defined in equation (5.1) is pure as it is a best response to the average correlated play of the opponents up to time T. The proof will proceed as follows. First, we prove a bound on the weighted sum of the cumulative estimated counterfactual regrets $\sum_{I \in \mathcal{I}} \pi_i^{\sigma^*}(I)\hat{R}_i^T(I, \sigma_i^*)$. Secondly, we prove a probabilistic bound on the expected squared difference between $\sum_{I \in \mathcal{I}_i} \pi_i^{\sigma^*}(I)R_i^T(I, \sigma_i^*)$ and $\sum_{I \in \mathcal{I}} \pi_i^{\sigma^*}(I)\hat{R}_i^T(I, \sigma_i^*)$, showing that the true counterfactual regrets are not too far from the estimated counterfactual regrets. Finally, we apply Theorem 5.1 to obtain the bound on the average regret.

For the first step,

$$\sum_{I \in \mathcal{I}_i} \pi_i^{\sigma^*}(I)\hat{R}_i^T(I, \sigma_i^*) \leq \sum_{I \in \mathcal{I}_i} \pi_i^{\sigma^*}(I) \sqrt{\sum_{a \in A(I)} \left(\hat{R}_i^{T,+}(I, a)\right)^2}$$

$$\begin{aligned}
&\leq \sum_{I\in\mathcal{I}_i} \pi_i^{\sigma^*}(I)\sqrt{|A(I)|\sum_{t=1}^{T}(\hat{\Delta}_i(I,\sigma^t))^2} \text{ by Lemma C.4}\\
&\leq \sqrt{|A(\mathcal{I}_i)|}\sum_{B\in\mathcal{B}_i}\pi_i^{\sigma^*}(B)\sum_{I\in B}\sqrt{\sum_{t=1}^{T}(\hat{\Delta}_i(I,\sigma^t))^2}\\
&\leq \sqrt{|A(\mathcal{I}_i)|}\sum_{B\in\mathcal{B}_i}\pi_i^{\sigma^*}(B)\sqrt{|B|\sum_{t=1}^{T}\sum_{I\in B}(\hat{\Delta}_i(I,\sigma^t))^2}\\
&\quad \text{by Lemma C.2}\\
&\leq \sqrt{|A(\mathcal{I}_i)|}\sum_{B\in\mathcal{B}_i}\pi_i^{\sigma^*}(B)\sqrt{|B|T\frac{(\Delta_i)^2}{\delta^2}} \text{ by equation (C.1)}\\
&= \frac{\Delta_i M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|T}}{\delta}.
\end{aligned} \tag{C.2}$$

Secondly, for $I \in \mathcal{I}_i$ with $\sigma_i^*(I,a) = 1$,

$$\begin{aligned}
\left(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*)\right)^2 &= \left(\sum_{t=1}^{T}\left(r_i^t(I,a) - \hat{r}_i^t(I,a)\right)\right)^2\\
&= \sum_{t=1}^{T}\left(r_i^t(I,a) - \hat{r}_i^t(I,a)\right)^2\\
&\quad + 2\sum_{t=1}^{T}\sum_{t'=t+1}^{T}\left(r_i^t(I,a) - \hat{r}_i^t(I,a)\right)\left(r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a)\right).
\end{aligned} \tag{C.3}$$

We now multiply both sides by $(\pi_i^{\sigma^*}(I))^2$ and take the expectation of both sides. Note that

$$\begin{aligned}
&\mathbf{E}\left[\left(r_i^t(I,a) - \hat{r}_i^t(I,a)\right)\left(r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a)\right)\right]\\
&= \mathbf{E}\left[\mathbf{E}\left[\left(r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a)\right) \mid r_i^t(I,a), \hat{r}_i^t(I,a)\right]\left(r_i^t(I,a) - \hat{r}_i^t(I,a)\right)\right]
\end{aligned}$$

and that $\mathbf{E}\left[\left(r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a)\right) \mid r_i^t(I,a), \hat{r}_i^t(I,a)\right] = 0$ since for $t' > t$, $\hat{r}_i^{t'}$ is an unbiased estimate of $r_i^{t'}$ and is sampled independently of $r_i^t(I,a)$ and $\hat{r}_i^t(I,a)$. Thus from equation (C.3), we have

$$\begin{aligned}
&\mathbf{E}\left[(\pi_i^{\sigma^*}(I))^2\left(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*)\right)^2\right]\\
&\quad = \sum_{t=1}^{T}\mathbf{E}\left[(\pi_i^{\sigma^*}(I))^2\left(r_i^t(I,a) - \hat{r}_i^t(I,a)\right)^2\right]\\
&\quad \leq \sum_{t=1}^{T}\mathbf{E}\left[\left(r_i^t(I,a)\right)^2 - 2r_i^t(I,a)\hat{r}_i^t(I,a) + \left(\hat{r}_i^t(I,a)\right)^2\right]\\
&\quad \leq \sum_{t=1}^{T}\left[\left(\pi_{-i}^{\sigma^t}(I)\right)^2\Delta_i^2 + 2\pi_{-i}^{\sigma^t}(I)\max\{\Delta_i^2, \left(\hat{\Delta}_i(I,\sigma^t)\right)^2\} + \left(\hat{\Delta}_i(I,\sigma^t)\right)^2\right].
\end{aligned} \tag{C.4}$$

We can now bound the expected squared difference between $\sum_{I\in\mathcal{I}_i}\pi_i^{\sigma^*}(I)R_i^T(I,\sigma_i^*)$ and $\sum_{I\in\mathcal{I}_i}\pi_i^{\sigma^*}(I)\hat{R}_i^T(I,\sigma_i^*)$ by

$$
\begin{aligned}
&\mathbf{E}\left[\left(\sum_{I\in\mathcal{I}_i}\pi_i^{\sigma^*}(I)\left(R_i^T(I,\sigma_i^*)-\hat{R}_i^T(I,\sigma_i^*)\right)\right)^2\right]\\
&\quad\le \mathbf{E}\left[\left(\sum_{I\in\mathcal{I}_i}\left|\pi_i^{\sigma^*}(I)\left(R_i^T(I,\sigma_i^*)-\hat{R}_i^T(I,\sigma_i^*)\right)\right|\right)^2\right]\\
&\quad\le \mathbf{E}\left[\left(\sqrt{|\mathcal{I}_i|\sum_{I\in\mathcal{I}_i}\left|\pi_i^{\sigma^*}(I)\left(R_i^T(I,\sigma_i^*)-\hat{R}_i^T(I,\sigma_i^*)\right)\right|^2}\right)^2\right]\\
&\qquad\text{by Lemma C.2}\\
&\quad= |\mathcal{I}_i|\sum_{I\in\mathcal{I}_i}\mathbf{E}\left[\left(\pi_i^{\sigma^*}(I)\right)^2\left(R_i^T(I,\sigma_i^*)-\hat{R}_i^T(I,\sigma_i^*)\right)^2\right]\\
&\quad\le |\mathcal{I}_i|\sum_{I\in\mathcal{I}_i}\sum_{t=1}^{T}\left[\left(\pi_{-i}^{\sigma^t}(I)\right)^2\Delta_i^2+2\pi_{-i}^{\sigma^t}(I)\max\{\Delta_i^2,\left(\hat{\Delta}_i(I,\sigma^t)\right)^2\}+\left(\hat{\Delta}_i(I,\sigma^t)\right)^2\right]\\
&\qquad\text{by equation (C.4)}\\
&\quad\le |\mathcal{I}_i|\sum_{B\in\mathcal{B}_i}\sum_{t=1}^{T}\left[\sum_{I\in B}\left(\pi_{-i}^{\sigma^t}(I)\right)^2\Delta_i^2+\sum_{I\in B}2\pi_{-i}^{\sigma^t}(I)\max\{\Delta_i^2,\left(\hat{\Delta}_i(I,\sigma^t)\right)^2\}\right.\\
&\qquad\qquad\left.+\sum_{I\in B}\left(\hat{\Delta}_i(I,\sigma^t)\right)^2\right]\\
&\quad\le |\mathcal{I}_i|\sum_{B\in\mathcal{B}_i}\sum_{t=1}^{T}\left[\Delta_i^2+\frac{3\Delta_i^2}{\delta^2}\right]\text{ by Lemma C.3 and equation (C.1)}\\
&\quad\le \frac{4|\mathcal{I}_i||\mathcal{B}_i|T\Delta_i^2}{\delta^2}
\end{aligned}
\tag{C.5}
$$

Finally, with probability $1-p$, we can bound the regret by

$$
\begin{aligned}
R_i^T &= \sum_{I\in\mathcal{I}_i}\pi_i^{\sigma^*}(I)R_i^T(I,\sigma_i^*)\text{ by Theorem 5.1}\\
&= \sum_{I\in\mathcal{I}_i}\pi_i^{\sigma^*}(I)\left(R_i^T(I,\sigma_i^*)-\hat{R}_i^T(I,\sigma_i^*)+\hat{R}_i^T(I,\sigma_i^*)\right)\\
&\le \left|\sum_{I\in\mathcal{I}_i}\pi_i^{\sigma^*}(I)\left(R_i^T(I,\sigma_i^*)-\hat{R}_i^T(I,\sigma_i^*)\right)\right|+\sum_{I\in\mathcal{I}_i}\pi_i^{\sigma^*}(I)\hat{R}_i^T(I,\sigma_i^*)\\
&\le \frac{1}{\sqrt{p}}\sqrt{\mathbf{E}\left[\left(\sum_{I\in\mathcal{I}_i}\pi_i^{\sigma^*}(I)\left(R_i^T(I,\sigma_i^*)-\hat{R}_i^T(I,\sigma_i^*)\right)\right)^2\right]}+\frac{\Delta_i M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|T}}{\delta}\\
&\qquad\text{by Lemma C.1 and equation (C.2)}\\
&\le \left(\frac{2\sqrt{|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}}+M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|}\right)\left(\frac{1}{\delta}\right)\Delta_i\sqrt{T}
\end{aligned}
$$

by equation (C.5). Dividing both sides by T gives the result. ■

Theorems 5.3 and 5.5. *Let X be one of CS, ES, OS (assuming OS samples opponent actions according to σ_{-i}), or AS, let $p\in(0,1]$, and let $\delta=\min_{z\in Z}q_i(z)>0$ over all $1\le t\le T$. When* *using X in a game with perfect recall, with probability $1-p$, average regret is bounded by*

$$\frac{R_i^T}{T} \leq \left(M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|} + \frac{2\sqrt{|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}} \right) \left(\frac{1}{\delta} \right) \frac{\Delta_i}{\sqrt{T}}.$$

Proof. To start, recall that $\tilde{v}_i(I,\sigma)$ is an unbiased estimate of the true counterfactual value $v_i(I,\sigma)$ [54, Lemma 1]. Next, for $Q \in \mathcal{Q}$, define

$$\hat{\Delta}_i(I,\sigma) = \Delta_i \sum_{z \in Q \cap Z_I} \frac{\pi^\sigma(z[I],z)\pi^\sigma_{-i}(z[I])}{q(z)}$$

so that the difference between two sampled counterfactual values at information set I is bounded by

$$\tilde{v}_i(I,\sigma_{(I\to a)}) - \tilde{v}_i(I,\sigma_{(I\to b)}) \leq \hat{\Delta}_i(I,\sigma)$$

for all $a,b \in A(I)$. By Lemma C.5, it suffices to show that

$$Y = \frac{1}{\Delta_i^2}\sum_{I \in B}(\hat{\Delta}_i(I,\sigma))^2 = \sum_{I\in B}\left(\sum_{z \in Q\cap Z_I} \frac{\pi^{\sigma^t}(z[I],z)\pi^{\sigma^t}_{-i}(z[I])}{q(z)} \right)^2 \leq \frac{1}{\delta^2}$$

for all $B \in \mathcal{B}_i$, $Q \in \mathcal{Q}$, and $\sigma \in \Sigma$. To that end, fix $B \in \mathcal{B}_i$, $Q \in \mathcal{Q}$, and $\sigma \in \Sigma$. Since X samples a single action at each $h \in H_c$ according to σ_c, there exists a unique $a_h^* \in A(h)$ such that if $z \in Q$ and $h \sqsubseteq z$, then $ha_h^* \sqsubseteq z$. Consider the new chance probability distribution $\hat{\sigma}_c$ defined according to

$$\hat{\sigma}_c(h,a) = \begin{cases} 1 & \text{if } a = a_h^* \\ 0 & \text{if } a \neq a_h^* \end{cases}$$

for all $h \in H_c$, $a \in A(h)$. When $X \neq CS$, we also have a unique such action a_I^* for each $I \in \mathcal{I}_{-i}$ sampled according to σ_{-i}, so we can similarly define the new opponent profile $\hat{\sigma}_{-i}$ according to

$$\hat{\sigma}_{-i}(I,a) = \begin{cases} \sigma_{-i}(I,a) & \text{if } X = \text{ CS} \\ 1 & \text{if } X \neq \text{ CS and } a = a_I^* \\ 0 & \text{if } X \neq \text{ CS and } a \neq a_I^* \end{cases}$$

for all $I \in \mathcal{I}_{-i}$, $a \in A(I)$. Then

$$\begin{aligned} Y &= \sum_{I\in B}\left(\sum_{z\in Q\cap Z_I} \frac{\pi_i^\sigma(z[I],z)\pi^\sigma_{-i}(z)}{q(z)} \right)^2 \\ &= \sum_{I\in B}\left(\sum_{z\in Z_I} \frac{\pi_i^\sigma(z[I],z)\pi^{\hat{\sigma}}_{-i}(z)}{q_i(z)} \right)^2 \\ &\leq \frac{1}{\delta^2}\sum_{I\in B}\left(\sum_{z\in Z_I} \pi^{\hat{\sigma}}_{-i}(z) \right)^2 \\ &= \frac{1}{\delta^2}\sum_{I\in B}\left(\pi^{\hat{\sigma}}_{-i}(I)\right)^2 \\ &\leq \frac{1}{\delta^2}, \end{aligned}$$

where the last line follows by Lemma C.3. ■

C.3 Generalized Sampling

Below, we restate and prove Lemma 5.1 and Theorem 5.4. We then show how to obtain a tighter bound than that of Theorem 5.4 when some structure on the estimate $\hat{v}_i(I,\sigma)$ is assumed.

Lemma 5.1. *Let $p \in (0,1]$ and suppose that there exists a bound $\hat{\Delta}_i$ on the difference between any two estimates, $\hat{v}_i(I,\sigma_{(I\to a)}) - \hat{v}_i(I,\sigma_{(I\to b)}) \leq \hat{\Delta}_i$. If strategies are selected according to regret matching* (2.4) *on the cumulative estimated counterfactual regrets in a game with perfect recall, then with probability at least $1-p$, the average regret is bounded by*

$$\frac{R_i^T}{T} \leq |\mathcal{I}_i| \left(\frac{\hat{\Delta}_i \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} + \sqrt{\frac{\mathbf{Var}\left[r_i - \hat{r}_i\right]}{pT} + \frac{\mathbf{Cov}\left[r_i^t - \hat{r}_i^t, r_i^{t'} - \hat{r}_i^{t'}\right]}{p} + \frac{\mathbf{E}[r_i - \hat{r}_i]^2}{p}} \right),$$

where

$$\mathbf{Var}\left[r_i - \hat{r}_i\right] = \max_{\substack{t\in\{1,\ldots,T\}\\ I\in\mathcal{I}_i\\ a\in A(I)}} \mathbf{Var}\left[r_i^t(I,a) - \hat{r}_i^t(I,a)\right],$$

$$\mathbf{Cov}\left[r_i^t - \hat{r}_i^t, r_i^{t'} - \hat{r}_i^{t'}\right] = \max_{\substack{t,t'\in\{1,\ldots,T\}\\ t\neq t'\\ I\in\mathcal{I}_i\\ a\in A(I)}} \mathbf{Cov}\left[r_i^t(I,a) - \hat{r}_i^t(I,a), r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a)\right], \textit{ and}$$

$$\mathbf{E}[r_i - \hat{r}_i] = \max_{\substack{t\in\{1,\ldots,T\}\\ I\in\mathcal{I}_i\\ a\in A(I)}} \mathbf{E}[r_i^t(I,a) - \hat{r}_i^t(I,a)].$$

Proof. The proof is a generalized version of the proof for Lemma C.5. Let σ_i^* be a strategy defined by equation (5.1). Again, we may assume σ_i^* is pure. Firstly, for $I \in \mathcal{I}_i$,

$$\begin{aligned} \frac{\hat{R}_i^T(I,\sigma_i^*)}{T} &\leq \frac{1}{T}\sqrt{\sum_{a\in A(I)} \left(\hat{R}_i^{T,+}(I,a)\right)^2} \\ &\leq \frac{\hat{\Delta}_i\sqrt{|A(I)|}}{\sqrt{T}} \text{ by Lemma C.4.} \end{aligned} \tag{C.6}$$

Next, define $\mathcal{J}_i = \{I \in \mathcal{I}_i \mid R_i^T(I,\sigma_i^*) \geq 0\}$. Similar to the proof of Lemma C.5, we now prove a probabilistic bound on the expected squared difference between $\sum_{I\in\mathcal{J}_i} R_i^T(I,\sigma_i^*)$ and $\sum_{I\in\mathcal{J}_i} \hat{R}_i^T(I,\sigma_i^*)$. Given perfect recall, we then complete the proof by using Theorem 5.1.

Now, for $I \in \mathcal{I}_i$,

$$\begin{aligned} (R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*))^2 &= \sum_{t=1}^T \left(r_i^t(I,a) - \hat{r}_i^t(I,a)\right)^2 \\ &\quad +2\sum_{t=1}^T \sum_{t'=t+1}^T (r_i^t(I,a) - \hat{r}_i^t(I,a))(r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a)). \end{aligned}$$

Taking the expectation of both sides gives

$$\begin{aligned}
&\mathbf{E}\left[(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*))^2\right] \\
&\quad\leq \left[\sum_{t=1}^{T} \mathbf{E}\left[(r_i^t(I,a) - \hat{r}_i^t(I,a))^2\right]\right. \\
&\qquad \left. +2\sum_{t=1}^{T}\sum_{t'=t+1}^{T} \mathbf{E}\left[(r_i^t(I,a) - \hat{r}_i^t(I,a))(r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a))\right]\right] \\
&\quad= \sum_{t=1}^{T}\left(\mathbf{Var}\left[r_i^t(I,a) - \hat{r}_i^t(I,a)\right] + \mathbf{E}\left[r_i^t(I,a) - \hat{r}_i^t(I,a)\right]^2\right) \\
&\qquad +2\sum_{t=1}^{T}\sum_{t'=t+1}^{T}\Big(\mathbf{Cov}\left[(r_i^t(I,a) - \hat{r}_i^t(I,a)),(r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a))\right] \\
&\qquad +\mathbf{E}[r_i^t(I,a) - \hat{r}_i^t(I,a)]\mathbf{E}[r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a)]\ \Big) \\
&\quad= \sum_{t=1}^{T}\mathbf{Var}\left[r_i^t(I,a) - \hat{r}_i^t(I,a)\right] \\
&\qquad +2\sum_{t=1}^{T}\sum_{t'=t+1}^{T}\mathbf{Cov}\left[r_i^t(I,a) - \hat{r}_i^t(I,a), r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a)\right] \\
&\qquad +\left(\sum_{t=1}^{T}\mathbf{E}[r_i^t(I,a) - \hat{r}_i^t(I,a)]\right)^2. \qquad \text{(C.7)}
\end{aligned}$$

We can now bound the expected squared difference between $\sum_{I\in\mathcal{J}_i} R_i^T(I,\sigma_i^*)$ and $\sum_{I\in\mathcal{J}_i} \hat{R}_i^T(I,\sigma_i^*)$ as follows:

$$\begin{aligned}
&\mathbf{E}\left[\left(\sum_{I\in\mathcal{J}_i}(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*))\right)^2\right] \\
&\quad\leq \mathbf{E}\left[\left(\sum_{I\in\mathcal{J}_i}\left|R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*)\right|\right)^2\right] \\
&\quad\leq \mathbf{E}\left[\left(\sqrt{|\mathcal{I}_i|\sum_{I\in\mathcal{J}_i}\left|R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*)\right|^2}\right)^2\right] \quad \text{by Lemma C.2} \\
&\quad= |\mathcal{I}_i|\sum_{I\in\mathcal{J}_i}\mathbf{E}\left[(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*))^2\right] \\
&\quad\leq |\mathcal{I}_i|\sum_{I\in\mathcal{J}_i}\sum_{t=1}^{T}\mathbf{Var}\left[r_i^t(I,a) - \hat{r}_i^t(I,a)\right] \\
&\qquad +2|\mathcal{I}_i|\sum_{I\in\mathcal{J}_i}\sum_{t=1}^{T}\sum_{t'=t+1}^{T}\mathbf{Cov}\left[r_i^t(I,a) - \hat{r}_i^t(I,a), r_i^{t'}(I,a) - \hat{r}_i^{t'}(I,a)\right] \\
&\qquad +|\mathcal{I}_i|\sum_{I\in\mathcal{J}_i}\left(\sum_{t=1}^{T}\mathbf{E}[r_i^t(I,a) - \hat{r}_i^t(I,a)]\right)^2 \quad \text{by (C.7)} \\
&\quad\leq |\mathcal{I}_i|^2\left(T\mathbf{Var}\left[r_i - \hat{r}_i\right] + T^2\mathbf{Cov}_{t\neq t'}\left[r_i^t - \hat{r}_i^t, r_i^{t'} - \hat{r}_i^{t'}\right] + T^2\mathbf{E}[r_i - \hat{r}_i]^2\right). \qquad \text{(C.8)}
\end{aligned}$$

Finally, with probability at least $1 - p$,

$$\begin{aligned}
\frac{R_i^T}{T} &\leq \frac{1}{T}\sum_{I\in\mathcal{J}_i} R_i^T(I,\sigma_i^*) \text{ by Theorem 5.1} \\
&= \frac{1}{T}\sum_{I\in\mathcal{J}_i}\left(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*) + \hat{R}_i^T(I,\sigma_i^*)\right) \\
&\leq \frac{1}{T}\left|\sum_{I\in\mathcal{J}_i}\left(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*)\right)\right| + \sum_{I\in\mathcal{J}_i}\frac{\hat{R}_i^T(I,\sigma_i^*)}{T} \\
&\leq \frac{1}{T\sqrt{p}}\sqrt{\mathbf{E}\left[\left(\sum_{I\in\mathcal{J}_i}(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*))\right)^2\right]} + \sum_{I\in\mathcal{J}_i}\frac{\hat{\Delta}_i\sqrt{|A(I)|}}{\sqrt{T}} \\
&\qquad \text{by Lemma C.1 and (C.6)} \\
&\leq |\mathcal{I}_i|\left(\frac{\hat{\Delta}_i\sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} + \sqrt{\frac{\mathbf{Var}\left[r_i - \hat{r}_i\right]}{pT} + \frac{\mathbf{Cov}\left[r_i^t - \hat{r}_i^t, r_i^{t'} - \hat{r}_i^{t'}\right]}{p} + \frac{\mathbf{E}[r_i - \hat{r}_i]^2}{p}}\right) \\
&\qquad \text{by (C.8). } \blacksquare
\end{aligned}$$

Theorem 5.4. *If in addition to the conditions of Lemma 5.1, for all $I \in \mathcal{I}_i$, $a \in A(I)$, $t \geq 1$, $\hat{v}_i(I,\sigma^t)$ and $\hat{v}_i(I,\sigma^t_{(I\to a)})$ are unbiased estimators of $v_i(I,\sigma^t)$ and $v_i(I,\sigma^t_{(I\to a)})$ respectively, and for all $t' \neq t$, $\hat{v}_i(I,\sigma^t)$ and $\hat{v}_i(I,\sigma^t_{(I\to a)})$ are sampled independently of $\hat{v}_i(I,\sigma^{t'})$ and $\hat{v}_i(I,\sigma^{t'}_{(I\to a)})$, then with probability at least $1-p$,*

$$\frac{R_i^T}{T} \leq \left(\hat{\Delta}_i\sqrt{|A(\mathcal{I}_i)|} + \frac{\sqrt{\mathbf{Var}\left[r_i - \hat{r}_i\right]}}{\sqrt{p}}\right)\frac{|\mathcal{I}_i|}{\sqrt{T}}.$$

Proof. Since $\hat{v}_i(I,\sigma)$ is unbiased, it immediately follows that $\hat{r}_i^t(I,a) = \hat{v}_i(I,\sigma^t_{(I\to a)}) - \hat{v}_i(I,\sigma^t)$ is also unbiased. Thus,

$$\mathbf{E}[r_i - \hat{r}_i] = 0.$$

In addition,

$$\mathbf{Cov}_{t\neq t'}\left[r_i^t - \hat{r}_i^t, r_i^{t'} - \hat{r}_i^{t'}\right] = 0$$

because samples are chosen independently between iterations. The result follows by Lemma 5.1. $\blacksquare$

Now, if some structure on the estimates $\hat{v}_i(I,\sigma)$ holds, then using Theorem 5.1 and the value $M_i(\sigma_i^*)$, we can tighten the bound of Theorem 5.4. The structure we require is that the difference between any two estimates $\hat{v}_i(I,\sigma_{(I\to a)})$ and $\hat{v}_i(I,\sigma_{(I\to b)})$ can be bounded by

$$\hat{v}_i(I,\sigma_{(I\to a)}) - \hat{v}_i(I,\sigma_{(I\to b)}) \leq \hat{\Delta}_i \equiv \pi^\sigma_{-i}(I)\hat{\Delta}'_i$$

for some $\hat{\Delta}'_i$. Note that both

$$v_i(I,\sigma_{(I\to a)}) - v_i(I,\sigma_{(I\to b)}) \leq \pi^\sigma_{-i}(I)\Delta_i$$

and

$$\tilde{v}_i(I,\sigma_{(I\to a)}) - \tilde{v}_i(I,\sigma_{(I\to b)}) \leq \pi^\sigma_{-i}(I)\frac{\Delta_i}{\delta}$$

with $\delta = \min_{z \in Z} q(z)$ can be bounded in this way. We can similarly derive a bound on the difference between any two estimates $\hat{v}_i(I, \sigma_{(I \to a)})$ and $\hat{v}_i(I, \sigma_{(I \to b)})$ where $\hat{v}_i(I, \sigma)$ is as defined for probing (Algorithm 3). This bound is derived in Lemma C.6 of Section C.4.

Theorem C.1. *If in addition to the conditions of Lemma 5.1 and Theorem 5.4, there exists $\hat{\Delta}'_i$ such that we can bound the difference between any two estimates by $\hat{v}_i(I, \sigma_{(I \to a)}) - \hat{v}_i(I, \sigma_{(I \to b)}) \leq \pi^{\sigma}_{-i}(I)\hat{\Delta}'_i$, then with probability at least $1 - p$, the average regret is bounded by*

$$\frac{R_i^T}{T} \leq \frac{\hat{\Delta}'_i M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} + \frac{|\mathcal{I}_i|\sqrt{\mathbf{Var}\left[r_i - \hat{r}_i\right]}}{\sqrt{pT}},$$

where

$$\mathbf{Var}\left[r_i - \hat{r}_i\right] = \max_{\substack{t \in 1,\ldots,T \\ I \in \mathcal{I}_i \\ a \in A(I)}} \mathbf{Var}\left[(r_i^t(I,a) - \hat{r}_i^t(I,a))\right].$$

Proof. The proof follows the same general steps as the proof of Lemma 5.1, with a few minor changes as noted here. Firstly,

$$\begin{aligned}
\sum_{I \in \mathcal{J}_i} \pi_i^{\sigma^*}(I)\hat{R}_i^T(I, \sigma_i^*) &\leq \sum_{I \in \mathcal{I}_i} \pi_i^{\sigma^*}(I)\sqrt{\sum_{a \in A(I)} \left(\hat{R}_i^{T,+}(I,a)\right)^2} \\
&\leq \sum_{I \in \mathcal{I}_i} \pi_i^{\sigma^*}(I)\sqrt{|A(I)| \sum_{t=1}^{T} (\pi^{\sigma}_{-i}(I)\hat{\Delta}'_i)^2} \\
&\quad \text{by Lemma C.4} \\
&\leq \sqrt{|A(\mathcal{I}_i)|}\hat{\Delta}'_i \sum_{B \in \mathcal{B}_i} \pi_i^{\sigma^*}(B) \sum_{I \in B} \sqrt{\sum_{t=1}^{T} (\pi^{\sigma}_{-i}(I))^2} \\
&\leq \sqrt{|A(\mathcal{I}_i)|}\hat{\Delta}'_i \sum_{B \in \mathcal{B}_i} \pi_i^{\sigma^*}(B)\sqrt{|B| \sum_{t=1}^{T} \sum_{I \in B} (\pi^{\sigma}_{-i}(I))^2} \\
&\quad \text{by Lemma C.2} \\
&\leq \sqrt{|A(\mathcal{I}_i)|}\hat{\Delta}'_i \sum_{B \in \mathcal{B}_i} \pi_i^{\sigma^*}(B)\sqrt{|B|T} \text{ by Lemma C.3} \\
&= \hat{\Delta}'_i M_i(\sigma_i^*)\sqrt{|A(\mathcal{I}_i)|T}. \qquad \text{(C.9)}
\end{aligned}$$

Next, note that the covariance and expectation terms from equation (C.8) are zero by the arguments in the proof of Theorem 5.4 above. Also, since $\pi_i^{\sigma^*}(I) \in [0,1]$, we have

$$\mathbf{E}\left[\left(\pi_i^{\sigma^*}(I)(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*))\right)^2\right] \leq \mathbf{E}\left[(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*))^2\right]$$

and so following the same arguments used to reach equation (C.8), we have

$$\mathbf{E}\left[\left(\sum_{I \in \mathcal{J}_i} \pi_i^{\sigma^*}(I)(R_i^T(I,\sigma_i^*) - \hat{R}_i^T(I,\sigma_i^*))\right)^2\right] \leq |\mathcal{I}_i|^2 T \mathbf{Var}\left[r_i - \hat{r}_i\right].$$

Therefore, given perfect recall, by Theorem 5.1, we have

$$\begin{aligned}
\frac{R_i^T}{T} &\leq \frac{1}{T} \sum_{I \in \mathcal{J}_i} \pi_i^{\sigma^*}(I) R_i^T(I, \sigma_i^*) \\
&= \frac{1}{T} \sum_{I \in \mathcal{J}_i} \pi_i^{\sigma^*}(I) \left(R_i^T(I, \sigma_i^*) - \hat{R}_i^T(I, \sigma_i^*) + \hat{R}_i^T(I, \sigma_i^*) \right) \\
&\leq \frac{1}{T} \left| \sum_{I \in \mathcal{J}_i} \pi_i^{\sigma^*}(I) \left(R_i^T(I, \sigma_i^*) - \hat{R}_i^T(I, \sigma_i^*) \right) \right| + \sum_{I \in \mathcal{J}_i} \frac{\pi_i^{\sigma^*}(I) \hat{R}_i^T(I, \sigma_i^*)}{T} \\
&\leq \frac{1}{T\sqrt{p}} \sqrt{\mathbf{E}\left[\left(\sum_{I \in \mathcal{J}_i} \pi_i^{\sigma^*}(I)(R_i^T(I, \sigma_i^*) - \hat{R}_i^T(I, \sigma_i^*)) \right)^2 \right]} + \frac{\hat{\Delta}_i' M_i(\sigma_i^*) \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} \\
&\qquad \text{by Lemma C.1 and (C.9)} \\
&\leq \frac{\hat{\Delta}_i' M_i(\sigma_i^*) \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} + \frac{|\mathcal{I}_i| \sqrt{\mathbf{Var}\left[r_i - \hat{r}_i \right]}}{\sqrt{pT}}. \blacksquare
\end{aligned}$$

C.4 Probing

In this section, we formally define the estimated counterfactual value $\hat{v}_i(I, \sigma)$ obtained via probing in its most general form. We then prove that $\hat{v}_i(I, \sigma)$ is bounded and unbiased, which leads to the proof of Proposition 5.1. Fix the iteration t and let K be the number of probes to perform at each information set. Suppose that on iteration t, we sample a block $X \in \mathcal{X}$, $X \subseteq Z$, according to some probability distribution on $\mathcal{X}$ that spans Z (as with $\mathcal{Q}$ in MCCFR). The set X represents candidate terminal histories that may be visited on iteration t, though some terminal histories in X may not be visited. Now, in addition, sample a **sub-block** $Q \in \mathcal{Q}_X$, $Q \subseteq X$, according to some probability distribution on $\mathcal{Q}_X$, where the union of all sets in $\mathcal{Q}_X$ spans X. Finally, for each $k = 1, 2, ..., K$, we independently sample a set of probes $Y^k \in \mathcal{Y}_{X,Q}$, $Y^k \subseteq X \backslash Q$ according to some probability distribution on $\mathcal{Y}_{X,Q}$, where the union of all sets in $\mathcal{Y}_{X,Q}$ spans $X \backslash Q$. It is assumed that all X, Q, and Y are sampled with some positive probability. When $Z_I \cap Q \neq \emptyset$, we define our **estimated counterfactual regret under probing** to be

$$\begin{aligned}
\hat{v}_i(I, \sigma) \quad = \quad & \frac{1}{q(I|X)} \left[\sum_{z \in Z_I \cap Q} \frac{\pi_{-i}^{\sigma}(z[I]) \pi^{\sigma}(z[I], z) u_i(z)}{x(z)} \right. \\
& \left. + \frac{1}{K} \sum_{k=1}^{K} \sum_{z \in Z_I \cap Y^k} \frac{\pi_{-i}^{\sigma}(z[I]) \pi^{\sigma}(z[I], z) u_i(z)}{x(z) y(z|X, Q)} \right], \qquad \text{(C.10)}
\end{aligned}$$

where

$$\begin{aligned}
x(z) &= \mathbf{P}[z \in X] \text{ for all } z \in Z, \\
q(I|X) &= \mathbf{P}[Z_I \cap Q \neq \emptyset \mid X], \text{ and} \\
y(z|X, Q) &= \mathbf{P}[z \in Y \mid X, Q] \text{ for all } z \in X \backslash Q.
\end{aligned}$$

Otherwise, when $Z_I \cap Q = \emptyset$, we define $\hat{v}_i(I, \sigma) = 0$.

Lemma C.6. *Given perfect recall, the following two conditions hold:*

(i) *For $a, a' \in A(I)$, $\hat{v}_i(I, \sigma_{(I\to a)}) - \hat{v}_i(I, \sigma_{(I \to a')}) \leq \pi^\sigma_{-i}(I)\hat{\Delta}'_i$, where $\hat{\Delta}'_i = \Delta_i/\delta$ and $\delta = \min_{z,X,Q,I} x(z)q(I|X)y(z|X,Q)$, and*

(ii) *$\hat{v}_i(I, \sigma)$ is an unbiased estimate of $v_i(I, \sigma)$.*

Proof. (i) $\hat{v}_i(I, \sigma_{(I\to a)}) - \hat{v}_i(I, \sigma_{(I\to a')})$

$$\begin{aligned}
=&\sum_{z \in Z_I \cap Q} \frac{\pi^\sigma_{-i}(z[I])(\pi^\sigma(z[I]a, z)u_i(z) - \pi^\sigma(z[I]a', z)u_i(z))}{x(z)q(I|X)} \\
&+ \frac{1}{K}\sum_{k=1}^K \sum_{z \in Z_I \cap Y^k} \frac{\pi^\sigma_{-i}(z[I])(\pi^\sigma(z[I]a, z)u_i(z) - \pi^\sigma(z[I]a', z)u_i(z))}{x(z)q(I|X)y(z|X,Q)} \\
\leq& \frac{\Delta_i}{\delta}\frac{1}{K}\sum_{k=1}^K \sum_{z \in Z_I \cap (Q \cup Y^k)} \pi^\sigma_{-i}(z[I]) \\
\leq& \hat{\Delta}'_i \sum_{h \in I} \pi^\sigma_{-i}(h) \sum_{k=1}^K \frac{1}{K} \\
=& \pi^\sigma_{-i}(I)\hat{\Delta}'_i.
\end{aligned}$$

(ii) For brevity, define $U^\sigma_i(z) = \pi^\sigma_{-i}(z[I])\pi^\sigma(z[I], z)u_i(z)$. We will extend the definition of $y(z|X, Q)$ by defining

$$y'(z|X,Q) = \mathbf{P}[z \in Y \cup Q \mid X, Q] \text{ for all } z \in X.$$

Note that for $z \in X \backslash Q$, $y'(z|X,Q) = y(z|X,Q)$ and for $z \in Q$, $y'(z|X,Q) = 1$. So, we can rewrite $\hat{v}_i(I, \sigma)$ as

$$\hat{v}_i(I, \sigma) = \frac{\mathbf{I}[Z_I \cap Q \neq \emptyset]}{Kq(I|X)} \sum_{k=1}^K \sum_{z \in Z_I \cap (Q \cup Y^k)} \frac{U^\sigma_i(z)}{x(z)y'(z|X,Q)},$$

where $\mathbf{I}[\cdot]$ is the indicator function. Given this form for $\hat{v}_i(I, \sigma)$, we have

$$\begin{aligned}
\mathbf{E}\left[\hat{v}_i(I,\sigma)\right] =& \sum_{X \in \mathcal{X}} \sum_{Q \in \mathcal{Q}_X} \sum_{Y^1 \in \mathcal{Y}_{X,Q}} \cdots \sum_{Y^K \in \mathcal{Y}_{X,Q}} \mathbf{P}[X]\mathbf{P}[Q \mid X] \prod_{k=1}^K \mathbf{P}[Y^k \mid X, Q] \\
&\frac{\mathbf{I}[Z_I \cap Q \neq \emptyset]}{Kq(I|X)} \sum_{k=1}^K \sum_{z \in Z_I \cap (Q \cup Y^k)} U^\sigma_i(z)/(x(z)y'(z|X,Q)) \\
=& \sum_{X \in \mathcal{X}} \mathbf{P}[X] \sum_{z \in Z_I \cap X} \frac{U^\sigma_i(z)}{x(z)q(I|X)} \sum_{Q \in \mathcal{Q}_X} \frac{\mathbf{P}[Q \mid X]\, \mathbf{I}[Z_I \cap Q \neq \emptyset]}{y'(z|X,Q)} \\
&\cdot \frac{1}{K} \sum_{k=1}^K \sum_{Y^k \in \mathcal{Y}_{X,Q}} \mathbf{P}[Y^k \mid X, Q]\, \mathbf{I}[z \in Q \cup Y^k]
\end{aligned}$$

$$\sum_{Y^1 \in \mathcal{Y}_{X,Q}} \cdots \sum_{Y^{k-1} \in \mathcal{Y}_{X,Q}} \sum_{Y^{k+1} \in \mathcal{Y}_{X,Q}} \cdots \sum_{Y^K \in \mathcal{Y}_{X,Q}} \prod_{\substack{\ell=1 \\ \ell \neq k}}^{K} \mathbf{P}[Y^\ell \mid X, Q]. \tag{C.11}$$

Now, notice that

$$\sum_{Y^1 \in \mathcal{Y}_{X,Q}} \cdots \sum_{Y^{k-1} \in \mathcal{Y}_{X,Q}} \sum_{Y^{k+1} \in \mathcal{Y}_{X,Q}} \cdots \sum_{Y^K \in \mathcal{Y}_{X,Q}} \prod_{\substack{\ell=1 \\ \ell \neq k}}^{K} \mathbf{P}[Y^\ell \mid X, Q] = 1$$

and

$$\frac{1}{K} \sum_{k=1}^{K} \sum_{Y^k \in \mathcal{Y}_{X,Q}} \mathbf{P}[Y^k \mid X, Q] \, \mathbf{I}[z \in Q \cup Y^k] = y'(z|X, Q).$$

Thus,

$$\begin{aligned}
\mathbf{E}\left[\hat{v}_i(I, \sigma)\right] &= \sum_{X \in \mathcal{X}} \mathbf{P}[X] \sum_{z \in Z_I \cap X} \frac{U_i^\sigma(z)}{x(z) q(I|X)} \sum_{Q \in \mathcal{Q}_X} \mathbf{P}[Q \mid X] \, \mathbf{I}[Z_I \cap Q \neq \emptyset] \text{ by (C.11)} \\
&= \sum_{X \in \mathcal{X}} \mathbf{P}[X] \sum_{z \in Z_I \cap X} \frac{U_i^\sigma(z)}{x(z)}, \text{ since } \sum_{Q \in \mathcal{Q}_X} \mathbf{P}[Q \mid X] \, \mathbf{I}[Z_I \cap Q \neq \emptyset] = q(I|X) \\
&= \sum_{z \in Z_I} \frac{U_i^\sigma(z)}{x(z)} \sum_{X \in \mathcal{X}} \mathbf{P}[X] \, \mathbf{I}[z \in X] \\
&= \sum_{z \in Z_I} U_i(z), \text{ since } \sum_{X \in \mathcal{X}} \mathbf{P}[X] \, \mathbf{I}[z \in X] = x(z) \\
&= v_i(I, \sigma). \blacksquare
\end{aligned}$$

Proof of Proposition 5.1. Let $\mathcal{X}$ be the partition of Z such that two terminal histories are in different blocks if and only if some chance or opponent action differs, and sample $X \in \mathcal{X}$ according to the known chance probabilities and current profile[1] so that $x(z) = \pi_{-i}^\sigma(z)$. Set $K = 1$ and define the probes Y as in Section 5.3 so that $y(z_{ha}|X, Q) = \pi_i^\sigma(ha, z)$. Thus, when $Z_I \cap Q \neq \emptyset$, the estimated counterfactual regret simplifies to

$$\hat{v}_i(I, \sigma) = \frac{1}{q_i(I)} \left[\sum_{z \in Z_I \cap Q} \pi_i^\sigma(z[I], z) u_i(z) + \sum_{z_{ha} \in Z_I \cap Y} \pi_i^\sigma(z[I], ha) u_i(z) \right].$$

The result now follows by Lemma C.6. $\blacksquare$

C.5 Pure CFR

Lastly, we now prove Theorem 5.6 by once again applying Lemma C.5.

Theorem 5.6. *Let $p \in (0, 1]$. When using Pure CFR in a game with perfect recall, with probability $1 - p$, average regret is bounded by*

$$\frac{R_i^T}{T} \leq \left(M_i(\sigma_i^*) \sqrt{|A(\mathcal{I}_i)|} + \frac{2\sqrt{|\mathcal{I}_i||\mathcal{B}_i|}}{\sqrt{p}} \right) \frac{\Delta_i}{\sqrt{T}}.$$

[1] As mentioned in Section 2.2.3, when $\pi_{-i}^\sigma(z) = 0$, $\hat{v}_i(I, \sigma)$ is still an unbiased estimate if $x(z) = 0$ and we simply treat z's contribution to the sum in equation (C.10) as zero.

Proof. By Proposition 5.2, the estimated counterfactual value $\hat{v}_i(I, \sigma)$ from Pure CFR is an unbiased estimate of the true counterfactual value $v_i(I, \sigma)$. Define $\hat{\Delta}_i(I, \sigma) = \pi^{\hat{s}}_{-i}(I)\Delta_i$, where $\hat{s}$ is the pure strategy profile and pure chance distribution sampled from σ. The difference between any two counterfactual values is then bounded by

$$\hat{v}_i(I, \sigma_{(I \to a)}) - \hat{v}_i(I, \sigma_{(I \to b)}) \leq \hat{\Delta}_i(I, \sigma)$$

for all $a, b \in A(I)$. Then, given perfect recall, by Lemma C.3, we have

$$\sum_{I \in B} (\hat{\Delta}_i(I, \sigma))^2 = \sum_{I \in B} (\pi^{\hat{s}}_{-i}(I))^2 \Delta_i^2 \leq \Delta_i^2 \sum_{I \in B} \pi^{\hat{s}}_{-i}(I) \leq \Delta_i^2$$

for all $B \in \mathcal{B}_i$. The result now follows by Lemma C.5 with $\delta = 1$. ■

Appendix D

Proofs for Chapter 6: CFR in Games with Imperfect Recall

In this appendix, we prove Theorems 6.1 and 6.2. In addition, we consider an alternative extension of well-formed games called *nearly well-formed games* and prove that regret is also minimized in nearly well-formed games.

D.1 Proof of Theorems 6.1 and 6.2

To begin, note that by the definition of counterfactual value, the regrets between Γ and a perfect recall refinement $\breve{\Gamma}$ are additive; specifically, for $I \in \mathcal{I}_i$ in Γ,

$$R_i^T(I,a) = \sum_{\breve{I} \in \breve{\mathcal{P}}(I)} R_i^T(\breve{I},a). \tag{D.1}$$

First, we provide a lemma stating that if the immediate counterfactual regrets of each $\breve{I} \in \breve{\mathcal{P}}(I)$ are proportional up to some difference D, then the average regret can be bounded above:

Lemma D.1. *Let $\breve{\Gamma}$ be a perfect recall refinement of a game Γ. If for all $I \in \mathcal{I}_i$, $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$, and $a \in A(I)$, there exist constants $C_{\breve{I},\breve{I}',a}, D_{\breve{I},\breve{I}',a} \in [0,\infty)$ such that*

$$\frac{1}{T}\left|R_i^{T,+}(\breve{I},a) - C_{\breve{I},\breve{I}',a} R_i^{T,+}(\breve{I}',a)\right| \le D_{\breve{I},\breve{I}',a}, \tag{D.2}$$

then the average regret in $\breve{\Gamma}$ is bounded by

$$\frac{\breve{R}_i^T}{T} \le \frac{\Delta_i C \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} + \sum_{I \in \mathcal{I}} |\breve{\mathcal{P}}(I)| D_I,$$

where

$$C = \sum_{I \in \mathcal{I}_i} \max_{\substack{\breve{I},\breve{I}' \in \breve{\mathcal{P}}(I) \\ a \in A(I)}} C_{\breve{I},\breve{I}',a}$$

and

$$D_I = \max_{\substack{\breve{I},\breve{I}' \in \breve{\mathcal{P}}(I) \\ a \in A(I)}} D_{\breve{I},\breve{I}',a}.$$

Proof.

$$\begin{aligned}
\breve{R}_i^T &\leq \sum_{\breve{I} \in \breve{\mathcal{I}}_i} \max_{a \in A(I)} R_i^{T,+}(\breve{I}, a) \text{ by Theorem 2.3} \\
&= \sum_{I \in \mathcal{I}_i} \sum_{\breve{I} \in \breve{\mathcal{P}}(I)} \max_{a \in A(I)} R_i^{T,+}(\breve{I}, a) \text{ by definition of a perfect recall refinement} \\
&\leq \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| R_i^{T,+}(\breve{I}^*, a^*) \text{ where } \breve{I}^* = \operatorname*{argmax}_{\breve{I} \in \breve{\mathcal{P}}(I)} \max_{a \in A(I)} R_i^{T,+}(\breve{I}, a) \\
&\quad \text{and } a^* = \operatorname*{argmax}_{a \in A(I)} R_i^{T,+}(\breve{I}^*, a) \\
&\leq \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| \left(C_{\breve{I}^*, \breve{I}^{**}, a^*} R_i^{T,+}(\breve{I}^{**}, a^*) + T D_{\breve{I}^*, \breve{I}^{**}, a^*} \right) \text{ by (D.2),} \\
&\quad \text{where } \breve{I}^{**} = \operatorname*{argmin}_{\breve{I} \in \breve{\mathcal{P}}(I)} R_i^T(\breve{I}, a^*) \\
&\leq \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| C_{\breve{I}^*, \breve{I}^{**}, a^*} \left(\frac{1}{|\breve{\mathcal{P}}(I)|} \sum_{\breve{I} \in \breve{\mathcal{P}}(I)} R_i^T(\breve{I}, a^*) \right)^+ + T \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| D_I \\
&\quad \text{because the minimum is at most the average and } (\cdot)^+ \text{ is monotone increasing} \\
&= \sum_{I \in \mathcal{I}_i} C_{\breve{I}^*, \breve{I}^{**}, a^*} R_i^{T,+}(I, a^*) + T \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| D_I \text{ by (D.1)} \\
&\leq \sum_{I \in \mathcal{I}_i} C_{\breve{I}^*, \breve{I}^{**}, a^*} \sqrt{\sum_{a \in A(I)} \left(R_i^{T,+}(I, a) \right)^2} + T \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| D_I \\
&\leq \sum_{I \in \mathcal{I}_i} C_{\breve{I}^*, \breve{I}^{**}, a^*} \Delta_i \sqrt{|A(I)| T} + T \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| D_I \text{ by Lemma C.4 with } \Delta_t = \Delta_i \\
&\leq \Delta_i C \sqrt{|A(\mathcal{I}_i)| T} + T \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| D_I.
\end{aligned}$$

Dividing both sides by T establishes the lemma. ■

Note that if Γ has perfect recall, then the constants $C_{I,I,a} = 1$ and $D_{I,I,a} = 0$ for all $I \in \mathcal{I}_i$ and $a \in A(I)$ satisfies the condition of Lemma D.1. In this case, $C = |\mathcal{I}_i|$ and $D_I = 0$, and so $R_i^T/T \leq \Delta_i |\mathcal{I}_i| \sqrt{|A(\mathcal{I}_i)|}/\sqrt{T}$, recovering Theorem 2.4 in the case when $M_i = |\mathcal{I}_i|$.

We now use Lemma D.1 to prove Theorems 6.1 and 6.2:

Theorem 6.2. *If Γ is skew well-formed with respect to $\breve{\Gamma}$, then the average regret in $\breve{\Gamma}$ for player i when using CFR in Γ is bounded by*

$$\frac{\breve{R}_i^T}{T} \leq \frac{\Delta_i K \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} + \sum_{I \in \mathcal{I}_i} |\breve{\mathcal{P}}(I)| \delta_I,$$

where $K = \sum_{I \in \mathcal{I}_i} \max_{\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)} k_{\breve{I}, \breve{I}'} \ell_{\breve{I}, \breve{I}'}$ and $\delta_I = \max_{\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)} \delta_{\breve{I}, \breve{I}'} \ell_{\breve{I}, \breve{I}'}$.

Proof. We will show that for all $I \in \mathcal{I}_i$, $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$, and $a \in A(I)$,

$$\frac{1}{T} \left| R_i^{T,+}(\breve{I}, a) - k_{\breve{I}, \breve{I}'} \ell_{\breve{I}, \breve{I}'} R_i^{T,+}(\breve{I}', a) \right| \leq \delta_{\breve{I}, \breve{I}'} \ell_{\breve{I}, \breve{I}'}, \qquad \text{(D.3)}$$

which, by Lemma D.1, proves the theorem.

Fix $I \in \mathcal{I}_i$, $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$, and $a \in A(I)$. Firstly, for all $z \in Z_{\breve{I}}$ and $\sigma \in \Sigma$, by conditions (ii) and (iii) of Definition 6.2, we have

$$\begin{aligned}
\pi^{\sigma}_{-i}(z) &= \pi_c(z) \prod_{(I,a) \in X_{-i}(z)} \sigma(I,a) \\
&= \ell_{\breve{I},\breve{I}'} \pi_c(\phi(z)) \prod_{(I,a) \in X_{-i}(\phi(z))} \sigma(I,a) \\
&= \ell_{\breve{I},\breve{I}'} \pi^{\sigma}_{-i}(\phi(z))
\end{aligned} \tag{D.4}$$

and by condition (iv) of Definition 6.2, we similarly have

$$\pi^{\sigma}_i(z[\breve{I}], z) = \pi^{\sigma}_i(\phi(z)[\breve{I}'], \phi(z)) \tag{D.5}$$

and

$$\pi^{\sigma}_i(z[\breve{I}]a, z) = \pi^{\sigma}_i(\phi(z)[\breve{I}']a, \phi(z)). \tag{D.6}$$

We can then bound the positive part of the cumulative counterfactual regret $R^{T,+}_i(\breve{I}, a)$ above by

$$\begin{aligned}
R^{T,+}_i(\breve{I}, a) &= \left(\sum_{t=1}^{T} r^t_i(\breve{I}, a) \right)^{+} \\
&= \left(\sum_{t=1}^{T} \sum_{z \in Z_{\breve{I}}} \pi^{\sigma}_{-i}(z)(\pi^{\sigma}_i(z[\breve{I}]a, z) - \pi^{\sigma}_i(z[\breve{I}], z)) u_i(z) \right)^{+} \\
&\leq \Bigg(\sum_{t=1}^{T} \sum_{z \in Z_{\breve{I}}} \ell_{\breve{I},\breve{I}'} \pi^{\sigma}_{-i}(\phi(z))(\pi^{\sigma}_i(\phi(z)[\breve{I}']a, \phi(z)) \\
&\qquad\qquad - \pi^{\sigma}_i(\phi(z)[\breve{I}'], \phi(z)))(k_{\breve{I},\breve{I}'} u_i(\phi(z)) + \delta_{\breve{I},\breve{I}'}) \Bigg)^{+}
\end{aligned}$$

by equations (D.4), (D.5), (D.6), and condition (i) of Definition 6.2

$$\begin{aligned}
&= \Bigg(\sum_{t=1}^{T} \sum_{z \in Z_{\breve{I}'}} \ell_{\breve{I},\breve{I}'} \pi^{\sigma}_{-i}(z)(\pi^{\sigma}_i(z[\breve{I}']a, z) \\
&\qquad\qquad - \pi^{\sigma}_i(z[\breve{I}'], z))(k_{\breve{I},\breve{I}'} u_i(z) + \delta_{\breve{I},\breve{I}'}) \Bigg)^{+}
\end{aligned}$$

since ϕ is a bijection

$$\begin{aligned}
&\leq \left(\sum_{t=1}^{T} \sum_{z \in Z_{\breve{I}'}} k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'} \pi^{\sigma}_{-i}(z)(\pi^{\sigma}_i(z[\breve{I}]a, z) - \pi^{\sigma}_i(z[\breve{I}], z)) u_i(z) \right)^{+} \\
&\quad + \left(\sum_{t=1}^{T} \sum_{z \in Z_{\breve{I}'}} \delta_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'} \pi^{\sigma}_{-i}(z)(\pi^{\sigma}_i(z[\breve{I}]a, z) - \pi^{\sigma}_i(z[\breve{I}], z)) \right)^{+} \\
&\leq k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'} R^{T,+}_i(\breve{I}', a) + \sum_{t=1}^{T} \delta_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'} \pi^{\sigma}_{-i}(\breve{I}')
\end{aligned}$$

$$\leq k_{\breve{I},\breve{I}'}\ell_{\breve{I},\breve{I}'}R_i^{T,+}(\breve{I}',a) + T\delta_{\breve{I},\breve{I}'}\ell_{\breve{I},\breve{I}'}, \tag{D.7}$$

where the last line follows because $\pi^{\sigma}_{-i}(\breve{I}') = \sum_{z\in Z_{\breve{I}'}} \pi^{\sigma}_{-i}(z[\breve{I}']) \leq 1$ in a perfect recall game $\breve{\Gamma}$. Similarly,

$$R_i^{T,+}(\breve{I},a) \geq k_{\breve{I},\breve{I}'}\ell_{\breve{I},\breve{I}'}R_i^{T,+}(\breve{I}',a) - T\delta_{\breve{I},\breve{I}'}\ell_{\breve{I},\breve{I}'}, \tag{D.8}$$

which together with equation (D.7) and dividing by T establishes (D.3), completing the proof. ■
Note that Theorem 6.1 immediately follows from Theorem 6.2 since a well-formed game is skew well-formed with $\delta_{\breve{I},\breve{I}'} = 0$ for all $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$.

D.2 Nearly Well-Formed Games

In this section, we consider an alternative extension of well-formed games that relaxes condition (iv) of Definition 6.1. Similar to the definition of $D(I)$ in Appendix B, for a subset of histories $L \subseteq H_i$, define

$$D_i(L) = \{I \mid I \in \mathcal{I}_i, \exists h \in L, h' \in I \text{ such that } h \sqsubseteq h'\}$$

to be the set of all information sets descending from any history in L.

Definition D.1. *For a game Γ and a perfect recall refinement $\breve{\Gamma}$, we say that Γ is a* ***nearly well-formed game with respect to $\breve{\Gamma}$*** *if for all $i \in N$, $I \in \mathcal{I}_i$, $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$, $J \in D_i(\breve{I})$, there exist bijections $\phi : Z_{\breve{I}} \to Z_{\breve{I}'}$, $\psi : D_i(\breve{I}) \to D_i(\breve{I}')$, $\omega : A(J) \to A(\psi(J))$ and constants $k_{\breve{I},\breve{I}'}, \ell_{\breve{I},\breve{I}'} \in [0,\infty)$ such that for all $z \in Z_{\breve{I}}$:*

(i) $u_i(z) = k_{\breve{I},\breve{I}'}u_i(\phi(z))$,

(ii) $\pi_c(z) = \ell_{\breve{I},\breve{I}'}\pi_c(\phi(z))$,

(iii) *In Γ, $X_{-i}(z) = X_{-i}(\phi(z))$, and*

(iv) $X_i(z[\breve{I}], z) = (J_1, a_1), ..., (J_m, a_m)$ *if and only if*
$X_i(\phi(z)[\breve{I}'], \phi(z)) = (\psi(J_1), \omega(a_1)), ..., (\psi(J_m), \omega(a_m))$.

We say that Γ is a ***nearly well-formed game*** *if it is nearly well-formed with respect to some perfect recall refinement.*

In a nearly well-formed game, condition (iv) says that player i may now remember information that was once forgotten, provided the descendants from $\breve{I}$ and $\breve{I}'$ are isomorphic across ϕ. This relaxes the corresponding condition for a well-formed game where player i could never remember information once it was forgotten. Clearly, any well-formed game is nearly well-formed by choosing ψ and ω to be the identity bijections.

For example, consider a longer version of DRP, **DRP-3**, that consists of three betting rounds instead of two where a third die is rolled at the beginning of round 3. We then define **DRP-IR-3** to be the imperfect recall abstraction of DRP-3 where during round 2, players only know the sum of their two dice. In round 3, players once again know the outcome of each individual die roll, recovering information from the first round that was forgotten in the second. For instance, corresponding histories where player i's first two rolls were 1,5 and where the first two rolls were 4,2 will be in the same information set during round 2, but will be in different information sets in round 3. However, betting is independent of die rolls and utilities are only dependent on the final sum of the three dice. Therefore, the descendants from these histories are isomorphic across ϕ and thus DRP-IR-3 is nearly well-formed with respect to DRP-3.

CFR guarantees that average regret is also minimized in nearly well-formed games:

Theorem D.1. *If Γ is nearly well-formed with respect to $\breve{\Gamma}$, then the average regret in $\breve{\Gamma}$ for player i when using CFR in Γ is bounded by*

$$\frac{\breve{R}_i^T}{T} \leq \frac{\Delta_i K \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}},$$

where $K = \sum_{I \in \mathcal{I}_i} \max_{\breve{I},\breve{I}' \in \breve{\mathcal{P}}(I)} k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'}$.

Proof. Fix $I \in \mathcal{I}_i$, $\breve{I}, \breve{I}' \in \breve{\mathcal{P}}(I)$, and $a \in A(I)$. By conditions (ii) and (iii) of Definition D.1, equation (D.4) holds.

Claim: $R_i^T(J,b) = k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'} R_i^T(\psi(J), \omega(b))$ for all $J \in D_i(\breve{I})$, $b \in A(J)$, $T \geq 0$.

Provided the claim is true, and assuming we play uniformly at random when the denominator of equation (2.4) is zero, we have

$$\begin{aligned}
\sigma^{T+1}(J,b) &= \begin{cases} \frac{R_i^{T,+}(J,b)}{\sum_{d \in A(J)} R_i^{T,+}(J,d)} & \text{if } \sum_{d \in A(J)} R_i^{T,+}(J,d) > 0 \\ \frac{1}{|A(J)|} & \text{otherwise} \end{cases} \\
&= \begin{cases} \frac{k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'} R_i^{T,+}(\psi(J),\omega(b))}{\sum_{d \in A(J)} k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'} R_i^{T,+}(\psi(J),\omega(b))} \\ \qquad \text{if } \sum_{d \in A(J)} k_{\breve{I},\breve{I}'} \ell_{\breve{I},\breve{I}'} R_i^{T,+}(\psi(J),\omega(b)) > 0 \\ \frac{1}{|A(\psi(J))|} \qquad \text{otherwise} \end{cases} \\
&\quad \text{since } \omega \text{ is a bijection} \\
&= \sigma^{T+1}(\psi(J), \omega(b)) \qquad \text{(D.9)}
\end{aligned}$$

for all $J \in D_i(\breve{I})$, $b \in A(J)$, $T \geq 0$. Therefore, for $t \geq 1$,

$$\begin{aligned}
\pi_i^{\sigma^t}(z[\breve{I}], z) &= \prod_{(J,b) \in X_i(z[\breve{I}],z)} \sigma^t(J,b) \\
&= \prod_{(J,b) \in X_i(z[\breve{I}],z)} \sigma^t(\psi(J), \omega(b)) \\
&= \prod_{(J,b) \in X_i(\phi(z)[\breve{I}'],\phi(z))} \sigma^t(J,b) \text{ by condition (iv) of Definition D.1}
\end{aligned}$$

$$= \pi_i^{\sigma^t}(\phi(z)[\breve{I}'], \phi(z)),$$

and thus equation (D.5) and similarly equation (D.6) hold for $\sigma = \sigma^t$. By following the proof of Theorem 6.2, we then have that equations (D.7) and (D.8) with $\delta_{\breve{I},\breve{I}'} = 0$ hold, and hence equation (D.3) with $\delta_{\breve{I},\breve{I}'} = 0$ holds. This establishes the theorem by Lemma D.1.

To complete the proof, we are left to show that the claim holds. We will do so by induction on T. The base case $T = 0$ holds since $R_i^0(I, a) = 0$ for all $I \in \mathcal{I}_i$, $a \in A(I)$. For the inductive step, assume that $R_i^{T-1}(J, b) = k_{\breve{I},\breve{I}'}\ell_{\breve{I},\breve{I}'}R_i^{T-1}(\psi(J), \omega(b))$ for all $J \in D_i(\breve{I})$, $b \in A(J)$. We will show that $R_i^T(J, b) = k_{\breve{I},\breve{I}'}\ell_{\breve{I},\breve{I}'}R_i^T(\psi(J), \omega(b))$ for all $J \in D_i(\breve{I})$, $b \in A(J)$.

Fix $J \in D_i(\breve{I})$ and $b \in A(J)$. By the induction hypothesis and equation (D.9) with T set to $T - 1$, we have for all $z \in Z_J$,

$$\begin{aligned}
\pi_i^{\sigma^T}(z[J], z) &= \prod_{(J',b') \in X_i(z[J],z)} \sigma^T(J', b') \\
&= \prod_{(J',b') \in X_i(z[J],z)} \sigma^T(\psi(J'), \omega(b')) \text{ by equation (D.9)} \\
&= \prod_{(J',b') \in X_i(\phi(z)[\psi(J)],\phi(z))} \sigma^T(J', b') \\
&\quad \text{by condition (iv) of Definition D.1 since } X_i(z[J], z) \text{ is a subsequence} \\
&\quad \text{(more precisely, a suffix) of } X_i(z[\breve{I}], z) \\
&= \pi_i^{\sigma^T}(\phi(z)[\psi(J)], \phi(z)) \qquad \text{(D.10)}
\end{aligned}$$

and similarly

$$\pi_i^{\sigma^T}(z[J]b, z) = \pi_i^{\sigma^T}(\phi(z)[\psi(J)]\omega(b), \phi(z)). \qquad \text{(D.11)}$$

Now consider the counterfactual regret at time T,

$$\begin{aligned}
r_i^T(J, b) &= \sum_{z \in Z_J} \pi_{-i}^{\sigma^T}(z)(\pi_i^{\sigma^T}(z[J]b, z) - \pi_i^{\sigma^T}(z[J], z))u_i(z) \\
&= \sum_{z \in Z_J} \ell_{\breve{I},\breve{I}'}\pi_{-i}^{\sigma^T}(\phi(z))(\pi_i^{\sigma^T}(\phi(z)[\psi(J)]\omega(b), \phi(z)) \\
&\qquad - \pi_i^{\sigma^T}(\phi(z)[\psi(J)], \phi(z)))k_{\breve{I},\breve{I}'}u_i(\phi(z)) \\
&\quad \text{by equations (D.10), (D.11) and conditions (i), (ii), and (iii) of Definition D.1} \\
&= \ell_{\breve{I},\breve{I}'}k_{\breve{I},\breve{I}'}r_i^T(\psi(J), \omega(b)).
\end{aligned}$$

Finally,

$$\begin{aligned}
R_i^T(J, b) &= \sum_{t=1}^{T} r_i^t(J, b) \\
&= R_i^{T-1}(J, b) + r_i^T(J, b) \\
&= \ell_{\breve{I},\breve{I}'}k_{\breve{I},\breve{I}'}(R_i^{T-1}(\psi(J), \omega(b)) + r_i^T(\psi(J), \omega(b))) \\
&\quad \text{by the induction hypothesis and the above}
\end{aligned}$$

$$
\begin{aligned}
&= \ell_{\breve{I},\breve{I}'} k_{\breve{I},\breve{I}'} \sum_{t=1}^{T} r_i^t(\psi(J), \omega(b)) \\
&= \ell_{\breve{I},\breve{I}'} k_{\breve{I},\breve{I}'} R_i^T(\psi(J), \omega(b)),
\end{aligned}
$$

establishing the inductive step. This completes the proof. ■