Finding equilibria in large sequential games of imperfect information

Andrew Gilpin and Tuomas Sandholm[1]

August 2005
CMU-CS-05-158

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Computing an equilibrium of an extensive form game of imperfect information is a fundamental problem in computational game theory, but current techniques do not scale to large games. To address this, we introduce the *ordered game isomorphism* and the related *ordered game isomorphic abstraction transformation*. For an n-player sequential game of imperfect information with observable actions and an ordered signal space, we prove that any Nash equilibrium in an abstracted smaller game, obtained by one or more applications of the transformation, can be easily converted into a Nash equilibrium in the original game. We present an efficient algorithm, *GameShrink*, which automatically and exhaustively abstracts the game. Using *GameShrink*, we find an equilibrium to a poker game that is over four orders of magnitude larger than the largest poker game solved previously. To address even larger games, we introduce approximation methods that do not preserve equilibrium, but nevertheless yield (*ex post*) provably close-to-optimal strategies.

[1]Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA
This material is based upon work supported by the National Science Foundation under ITR grants IIS-0121678 and IIS-0427858, and a Sloan Fellowship.

Keywords: game theory, sequential games of imperfect information, equilibrium computation, computer poker, automated abstraction

1 Introduction

In environments with multiple self-interested agents, an agent's outcome is generally affected by actions of the other agents. Consequently, the optimal action of one agent can depend on the actions of others. Game theory provides a normative framework for analyzing such strategic situations. In particular, it provides the notion of an *equilibrium*, a strategy profile in which no agent has incentive to deviate to a different strategy.

The question of how complex it is to construct a Nash equilibrium [29] in a 2-player game has been dubbed by Papadimitriou "a most fundamental computational problem whose complexity is wide open" and "together with factoring, [...] the most important concrete open question on the boundary of P today" [30]. The most prevalent algorithm for finding an equilibrium in a 2-player game is the *Lemke-Howson* algorithm [23], but it was recently shown that it takes exponentially many steps in the worst case [34]. (For a survey of equilibrium computation in 2-player games, see [40].) For more than two players, there have been many proposed algorithms, but these algorithms currently only scale to very small games [14, 27]. Recently, some progress has been made in developing efficient algorithms for computing Nash equilibria in certain restricted cases (*e.g.*, [31, 2, 24, 5]), as well as for computing market equilibria (*e.g.*, [9, 10, 15, 35]).

For sequential games with imperfect information, one could try to find an equilibrium using the normal (matrix) form, where every contingency plan of the agent is a pure strategy for the agent. Unfortunately (even if equivalent strategies are replaced by a single strategy [21]) this representation is generally exponential in the size of the game tree [39]. The *sequence form* is an alternative that results in a more compact representation [33, 17, 39]. For 2-player games, there is a polynomial-sized (in the size of the game tree) linear programming formulation (linear complementarity in the non-zero-sum case) based on the sequence form such that strategies for players 1 and 2 correspond to primal and dual variables. Thus, the equilibria of reasonable-sized 2-player games can be computed using this method [39, 18, 19].[1] However, this approach still yields enormous (unsolvable) optimization problems for many real-world games, such as poker.

In this paper, we take a different approach to tackling the difficult problem of equilibrium computation. Instead of developing an equilibrium-finding method *per se*, we instead develop a methodology for automatically abstracting games in such a way that any equilibrium in the smaller (abstracted) game corresponds directly to an equilibrium in the original game. Thus, by computing an equilibrium in the smaller game (using any available equilibrium-finding algorithm), we are able to construct an equilibrium in the original game. The motivation is that an equilibrium for the smaller game can be computed drastically faster than for the orignal game.

To this end, we introduce *games with ordered signals* (Section 2), a broad class of games that has enough structure which we can exploit for abstraction purposes. Instead of operating directly on the game tree (something we found to be technically challenging), we instead introduce the use of *information filters* (Section 2.1), which coarsen the information each player receives. They are used in our analysis and abstraction algorithm. By operating only in the space of filters, we are able to keep the strategic structure of the game intact, while abstracting out details of the game in a way that is lossless from the perspective of equilibrium finding. We introduce the *ordered game isomorphism* to describe strategically symmetric situations and the *ordered game isomorphic abstraction transformation* to take advantange of such symmetries (Section 3). As our main equilibrium result we have the following:

Theorem 2 *Let Γ be a game with ordered signals, and let F be an information filter for Γ. Let F' be an information filter constructed from F by one application of the ordered game isomorphic abstraction transformation, and let σ' be a Nash equilibrium strategy profile of*

[1] Recently this approach was extended to handle computing *sequential equilibria* [20] as well [28].

*the induced game $\Gamma_{F'}$ (*i.e.*, the game Γ using the filter F'). If σ is constructed by using the corresponding strategies of σ', then σ is a Nash equilibrium of Γ_F.*

The proof of the theorem uses an equivalent characterization of Nash equilibria: σ is a Nash equilibrium if and only if there exist beliefs μ (players' beliefs about unknown information) at all points of the game reachable by σ such that σ is sequentially rational (*i.e.*, a best response) given μ, where μ is updated using Bayes' rule. We can then use the fact that σ' is a Nash equilibrium to show that σ is a Nash equilibrium considering only local properties of the game.

In addition to the main equilibrium result, we also give a polynomial algorithm, *GameShrink*, for exhaustively abstracting the game (Section 4), several algorithmic and data structure related speed improvements (Section 4.1), and we demonstrate how a simple modification to our algorithm yields an approximation algorithm (Section 5).

1.1 Application to Rhode Island Hold'em poker

Poker is an enormously popular card game played around the world. The 2005 World Series of Poker is expected to have over \$100 million dollars in total prize money, including \$60 million for the main event. Increasingly, poker players compete in online casinos, and television stations regularly broadcast poker tournaments. Due to the uncertainty stemming from opponents' cards, opponents' future actions, and chance moves, poker has been identified as an important research area in CS [4]. Poker has been a popular subject in the game theory literature since the field's founding, but manual equilibrium analysis has been limited to extremely small games. Even with the use of computers, the largest poker games that have been solved have only about 140,000 nodes in the game tree [19]. Large-scale approximations have been developed [3], but those methods do not provide any guarantees about the performance of the computed strategies. Furthermore, the approximations were designed manually by a human expert. Our approach yields an automated abstraction mechanism along with theoretical guarantees on the strategies' performance.

Rhode Island Hold'em was invented as a testbed for computational game playing [36]. It was designed so that it was similar in style to Texas Hold'em, yet not so large that devising reasonably intelligent strategies would be impossible. (The rules of Rhode Island Hold'em are given in Appendix C. That appendix also shows how Rhode Island Hold'em can be modeled as a game with ordered signals.) We applied the techniques developed in this paper to exactly solve Rhode Island Hold'em, which has a game tree exceeding 3.1 billion nodes.

Applying the sequence form representation to Rhode Island Hold'em directly without abstractions yields a linear program with 91,224,226 rows, and the same number of columns. This is much too large for current linear programming algorithms to handle. We used *GameShrink* to reduce this, and it yielded a linear program with 1,237,238 rows and columns—with 50,428,638 non-zero coefficients. We then applied iterated elimination of dominated strategies, which further reduced this to 1,190,443 rows and 1,181,084 columns. (Applying iterated elimination of dominated strategies without *GameShrink* yielded 89,471,986 rows and 89,121,538 columns, which still would have been prohibitively large to solve.) *GameShrink* required less than one second to perform the shrinking (*i.e.*, to compute all of the ordered game isomorphic abstraction transformations). Using a 1.65GHz IBM eServer p5 570 with 64 gigabytes of RAM (we only needed 25 gigabytes), we solved it in 7 days and 17 hours using the barrier method of CPLEX version 9.1.2. We recently demonstrated our optimal Rhode Island Hold'em poker player at the AAAI-05 conference [11], and it is available for play on-line at `http://www.cs.cmu.edu/~gilpin/gsi.html`.

While others have worked on computer programs for playing Rhode Island Hold'em [36], no optimal strategy has been found before. This is the largest poker game solved to date by over four orders of magnitude.

2 Games with ordered signals

We find it convenient to work with a slightly restricted class of games, as compared to the full generality of the extensive form.[2] This class, which we call *games with ordered signals*, is highly structured, but still general enough to capture a wide range of strategic situations. Instances of this game family consist of a finite number of rounds in which players play a game on a directed tree. The only uncertainty players face stems from private signals the other players have received and in the unknown future signals. In each round, there are public signals (announced to all players) and private signals (confidentially communicated to individual players). Each player receives the same number of private signals at each round. The strongest assumption is that there is a partial ordering over sets of signals, and the payoffs are increasing (not necessarily strictly) in these signals. For example, in poker, this partial ordering corresponds exactly to the ranking of card hands.

Definition 1 *A* game with ordered signals *is a tuple* $\Gamma = \langle I, G, L, \Theta, \kappa, \gamma, p, \succeq, \omega, u \rangle$ *where:*

1. $I = \{1, \ldots, n\}$ *is a finite set of players.*
2. $G = \langle G^1, \ldots, G^r \rangle$, $G^j = \left(V^j, E^j\right)$, *is a finite collection of finite directed trees with vertices* V^j *and edges* E^j. *Let* Z^j *denote the leaf nodes of* G^j *and let* $N^j(v)$ *denote the outgoing neighbors of* $v \in V^j$. G^j *is the* stage game *for round* j.
3. $L = \langle L^1, \ldots, L^r \rangle$, $L^j : V^j \setminus Z^j \to I$ *indicates which player acts (chooses an outgoing edge) at each internal node in round* j.
4. Θ *is a finite set of* signals.
5. $\kappa = \langle \kappa^1, \ldots, \kappa^r \rangle$ *and* $\gamma = \langle \gamma^1, \ldots, \gamma^r \rangle$ *are vectors of nonnegative integers, where* κ^j *and* γ^j *denote the number of public and private signals (per player), respectively, revealed in round* j. *Each signal* $\theta \in \Theta$ *may only be revealed once, and in each round every player receives the same number of private signals, so we require* $\sum_{j=1}^{r} \kappa^j + n\gamma^j \leq |\Theta|$. *The public information revealed in round* j *is* $\alpha^j \in \Theta^{\kappa^j}$ *and the public information revealed in all rounds up through round* j *is* $\tilde{\alpha}^j = \left(\alpha^1, \ldots, \alpha^j\right)$. *The private information revealed to player* $i \in I$ *in round* j *is* $\beta_i^j \in \Theta^{\gamma^j}$ *and the private information revaled to player* $i \in I$ *in all rounds up through round* j *is* $\tilde{\beta}_i^j = \left(\beta_i^1, \ldots, \beta_i^j\right)$. *We also write* $\tilde{\beta}^j = \left(\tilde{\beta}_1^j, \ldots, \tilde{\beta}_n^j\right)$ *to represent all private information up through round* j, *and* $\left(\tilde{\beta'}_i^j, \tilde{\beta}_{-i}^j\right) = \left(\tilde{\beta}_1^j, \ldots, \tilde{\beta}_{i-1}^j, \tilde{\beta'}_i^j, \tilde{\beta}_{i+1}^j, \ldots, \tilde{\beta}_n^j\right)$ *is* $\tilde{\beta}^j$ *with* $\tilde{\beta}_i^j$ *replaced with* $\tilde{\beta'}_i^j$. *The total information revealed up through round* j, $\left(\tilde{\alpha}^j, \tilde{\beta}^j\right)$, *is said to be* legal *if no signals are repeated.*
6. p *is a probability distribution over* Θ, *with* $p(\theta) > 0$ *for all* $\theta \in \Theta$. *Signals are drawn from* Θ *according to* p *without replacement, so if* X *is the set of signals already revealed, then*

$$p(x \mid X) = \begin{cases} \frac{p(x)}{\sum_{y \notin X} p(y)} & \text{if} \quad x \notin X \\ 0 & \text{if} \quad x \in X. \end{cases}$$

7. $\succeq$ *is a partial ordering of subsets of* Θ *and is defined for at least those pairs required by* u.

[2] For readers unfamiliar with extensive form games, we provide a complete definition in Appendix A.

8. $\omega : \bigcup_{j=1}^{r} Z^j \to \{over, continue\}$ *is a mapping of terminal nodes within a stage game to one of two values:* over*, in which case the game ends, or* continue*, in which case the game continues to the next round. Clearly, we require $\omega(z) = over$ for all $z \in Z^r$. Note that ω is independent of the signals. Let $\omega^j_{over} = \left\{z \in Z^j \,|\, \omega(z) = over\right\}$ and $\omega^j_{cont} = \left\{z \in Z^j \,|\, \omega(z) = continue\right\}$.*

9. $u = (u^1, \ldots, u^r)$, $u^j : \bigtimes_{k=1}^{j-1} \omega^k_{cont} \times \omega^j_{over} \times \bigtimes_{k=1}^{j} \Theta^{\kappa^k} \times \bigtimes_{i=1}^{n} \bigtimes_{k=1}^{j} \Theta^{\gamma^k} \to \mathbb{R}^n$ *is a utility function such that for every j, $1 \le j \le r$, for every $i \in I$, and for every $\tilde{z} \in \bigtimes_{k=1}^{j-1} \omega^k_{cont} \times \omega^j_{over}$, at least one of the following two conditions holds:*

   (a) *Utility is signal independent: $u^j_i(\tilde{z}, \vartheta) = u^j_i(\tilde{z}, \vartheta')$ for all legal $\vartheta, \vartheta' \in \bigtimes_{k=1}^{j} \Theta^{\kappa^k} \times \bigtimes_{i=1}^{n} \bigtimes_{k=1}^{j} \Theta^{\gamma^k}$.*

   (b) *$\succeq$ is defined for all legal signals $\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right)$ and $\left(\tilde{\alpha}^j, \tilde{\beta}'^j_i\right)$ through round j and a player's utility is increasing in her private signals, everything else equal:*

$$\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right) \succeq \left(\tilde{\alpha}^j, \tilde{\beta}'^j_i\right) \longrightarrow u_i\left(\tilde{z}, \tilde{\alpha}^j, \left(\tilde{\beta}^j_i, \tilde{\beta}^j_{-i}\right)\right) \ge u_i\left(\tilde{z}, \tilde{\alpha}^j, \left(\tilde{\beta}'^j_i, \tilde{\beta}^j_{-i}\right)\right).$$

We will use the term game with ordered signals *and the term* ordered game *interchangeably.*

2.1 Information filters

In this subsection, we define an *information filter* for ordered games. Instead of completely revealing a signal (either public or private) to a player, the signal first passes through this filter, which outputs a *coarsened* signal to the player. By varying the filter applied to a game, we are able to obtain a wide variety of games while keeping the underlying action space of the game intact. We will use this when designing our abstraction techniques. Formally, an information filter is as follows.

Definition 2 *Let $\Gamma = \langle I, G, L, \Theta, \kappa, \gamma, p, \succeq, \omega, u\rangle$ be an ordered game. Let $S^j \subseteq \bigtimes_{k=1}^{j} \Theta^{\kappa^k} \times \bigtimes_{k=1}^{j} \Theta^{\gamma^k}$ be the set of legal signals (*i.e.*, no repeated signals) for one player through round j. An* information filter *for Γ is a collection $F = \langle F^1, \ldots, F^r\rangle$ where each F^j is a function $F^j : S^j \to 2^{S^j}$ such that each of the following conditions hold:*

1. *(Truthfulness) $\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right) \in F^j\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right)$ for all legal $\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right)$.*
2. *(Independence) The range of F^j is a partition of S^j.*
3. *(Information preservation) If two values of a signal are distinguishable in round k, then they are distinguishable in round $j > k$. Let $m^j = \sum_{l=1}^{j} \kappa^l + \gamma^l$. We require that for all legal $(\theta_1, \ldots, \theta_{m^k}, \ldots, \theta_{m^j}) \subseteq \Theta$ and $(\theta'_1, \ldots, \theta'_{m^k}, \ldots, \theta'_{m^j}) \subseteq \Theta$:*

$$(\theta'_1, \ldots, \theta'_{m^k}) \notin F^k(\theta_1, \ldots, \theta_{m^k}) \longrightarrow (\theta'_1, \ldots, \theta'_{m^k}, \ldots, \theta'_{m^j}) \notin F^j(\theta_1, \ldots, \theta_{m^k}, \ldots, \theta_{m^j}).$$

A game with ordered signals Γ and an information filter F for Γ defines a new game Γ_F. We refer to such games as *filtered ordered games*. We are left with the original game if we use the identity filter $F^j\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right) = \left\{\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right)\right\}$. We have the following simple (but important) result:

Proposition 1 *A filtered ordered game is an extensive form game satisfying perfect recall.*

A simple proof proceeds by constructing an extensive form game directly from the ordered game, and showing that it satisfies perfect recall. In determining the payoffs in a game with filtered signals, we take the average over all real signals in the filtered class, weighted by the probability of each real signal occurring.

2.2 Strategies and Nash equilibrium

We are now ready to define behavior strategies in the context of filtered ordered games.

Definition 3 *A* behavior strategy *for player i in round j of $\Gamma = \langle I, G, L, \Theta, \kappa, \gamma, p, \succeq, \omega, u \rangle$ with information filter F is a probability distribution over possible actions, defined for each player i, each round j, and each $v \in V^j \setminus Z^j$ where $L^j(v) = i$:*

$$\sigma^j_{i,v} : \bigtimes_{k=1}^{j-1} \omega^k_{cont} \times Range\left(F^j\right) \longrightarrow \Delta\left(\left\{w \in V^j \mid (v,w) \in E^j\right\}\right).$$

($\Delta(X)$ is the set of probability distributions over a finite set X.) A behavior strategy for player i in round j is $\sigma^j_i = \left(\sigma^j_{i,v_1}, \ldots, \sigma^j_{i,v_m}\right)$ for each $v_k \in V^j \setminus Z^j$ where $L^j(v_k) = i$. A behavior strategy for player i in Γ is $\sigma_i = \left(\sigma^1_i, \ldots, \sigma^r_i\right)$. A strategy profile *is $\sigma = (\sigma_1, \ldots, \sigma_n)$. A strategy profile with σ_i replaced by σ'_i is $(\sigma'_i, \sigma_{-i}) = (\sigma_1, \ldots, \sigma_{i-1}, \sigma'_i, \sigma_{i+1}, \ldots, \sigma_n)$.*

By an abuse of notation, we will say player i receives an expected payoff of $u_i(\sigma)$ when all players are playing the strategy profile σ. Strategy σ_i is said to be player i's *best response* to σ_{-i} if for all other strategies σ'_i for player i we have $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$. σ is a *Nash equilibrium* if, for every player i, σ_i is a best response for σ_{-i}. A Nash equilibrium always exists in finite extensive form games [29], and one exists in behavior strategies for games with perfect recall [22]. Using these observations, we have the following corollary to Proposition 1:

Corollary 1 *For any filtered ordered game, a Nash equilibrium exists in behavior strateges.*

3 Equilibrium-preserving abstractions

In this section, we present our main technique for reducing the size of games. We begin by defining a *filtered signal tree* which represents all of the chance moves in the game. The bold edges (*i.e.* the first two levels of the tree) in the game trees in Figure 1 correspond to the filtered signal trees in each game.

Definition 4 *Associated with every ordered game $\Gamma = \langle I, G, L, \Theta, \kappa, \gamma, p, \succeq, \omega, u \rangle$ and information filter F is a* filtered signal tree*, a directed tree in which each vertex corresponds to some revealed (filtered) signals and edges correspond to revealing specific (filtered) signals. The nodes in the filtered signal tree represent the set of all possible revealed filtered signals (public and private) at some point in time. The filtered public signals revealed in round j correspond to the vertices in the κ^j levels beginning at level $\sum_{k=1}^{j-1}\left(\kappa^k + n\gamma^k\right)$ and the private signals revealed in round j correspond to the vertices in the $n\gamma^j$ levels beginning at level $\sum_{k=1}^{j} \kappa^k + \sum_{k=1}^{j-1} n\gamma^k$. We denote children of a node x as $N(x)$. In addition, we associate weights with the edges corresponding to the probability of the particular edge being chosen given that its parent was reached.*

In many games, there are certain situations in the game that can be thought of as being strategically equivalent to other situations in the game. By melding these situations together, it is possible to arrive at a strategically equivalent smaller game. The next two definitions formalize this notion via the introduction of the *ordered game isomorphic* relation and the *ordered game isomorphic abstraction transformation*.

Definition 5 *Two subtrees beginning at internal nodes x and y of a filtered signal tree are* ordered game isomorphic *if x and y have the same parent and there is a bijection $f : N(x) \to N(y)$, such that for $w \in N(x)$ and $v \in N(y)$, $v = f(w)$ implies the weights on the edges (x, w) and (y, v) are the same and the subtrees beginning at w and v are ordered game isomorphic. Two leaves (corresponding to filtered signals ϑ and ϑ' up through round r) are ordered game isomorphic if for all $\tilde{z} \in \bigtimes_{j=1}^{r-1} \omega^j_{cont} \times \omega^r_{over}$, $u^r(\tilde{z}, \vartheta) = u^r(\tilde{z}, \vartheta')$.*

Definition 6 *Let $\Gamma = \langle I, G, L, \Theta, \kappa, \gamma, p, \succeq, \omega, u\rangle$ be an ordered game and let F be an information filter for Γ. Let ϑ and ϑ' be two information structures where the subtrees in the induced filtered signal tree corresponding to the nodes ϑ and ϑ' are ordered game isomorphic, and ϑ and ϑ' are at either level $\sum_{k=1}^{j-1}\left(\kappa^k + n\gamma^k\right)$ or $\sum_{k=1}^{j}\kappa^k + \sum_{k=1}^{j-1} n\gamma^k$ for some round j. The* ordered game isomorphic abstraction transformation *is given by creating a new information filter F':*

$$F'^j\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right) = \begin{cases} F^j\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right) & \text{if } \left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right) \notin \vartheta \cup \vartheta' \\ \vartheta \cup \vartheta' & \text{if } \left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right) \in \vartheta \cup \vartheta'. \end{cases}$$

Figure 1 shows the ordered game isomorphic abstraction transformation applied twice to a tiny poker game. Theorem 2, our main equilibrium result, shows how the ordered game isomorphic abstraction transformation can be used to compute equilibria faster.

Theorem 2 *Let $\Gamma = \langle I, G, L, \Theta, \kappa, \gamma, p, \succeq, \omega, u\rangle$ be an ordered game and F be an information filter for Γ. Let F' be an information filter constructed from F by one application of the isomorphic information structure abstraction transformation. Let σ' be a Nash equilibrium of the induced game $\Gamma_{F'}$. If we take $\sigma^j_{i,v}\left(\tilde{z}, F^j\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right)\right) = \sigma'^j_{i,v}\left(\tilde{z}, F'^j\left(\tilde{\alpha}^j, \tilde{\beta}^j_i\right)\right)$, σ is a Nash equilibrium of Γ_F.*

The main idea of the proof involves the use of an equivalent characterization of Nash equilibria using belief systems and the notion of sequential rationality. An outline of the proof, intended to give a flavor of the technique used, appears below. Three claims cover the necessary details to finish the proof; we prove these claims in Appendix B. The heart of the proof is as follows:

PROOF OF THEOREM 2. For an extensive form game, a *belief system* μ assigns a probability to every decision node x such that $\sum_{x\in h}\mu(x) = 1$ for every information set h. A strategy profile σ is *sequentially rational at h given belief system μ* if $u_i(\sigma_i, \sigma_{-i} \mid h, \mu) \geq u_i(\tau_i, \sigma_{-i} \mid h, \mu)$ for all other strategies τ_i, where i is the player who controls h. A basic result [26, Proposition 9.C.1] characterizing Nash equilibria dictates that σ is a Nash equilibrium if and only if there is a belief system μ such that for every information set h with $\Pr(h \mid \sigma) > 0$, the following two conditions hold: (C1) σ is sequentially rational at h given μ; and (C2) $\mu(x) = \frac{\Pr(x \mid \sigma)}{\Pr(h \mid \sigma)}$ for all $x \in h$. Since σ' is a Nash equilibrium of Γ', there exists such a belief system μ'. Using μ', we will construct a belief system μ for Γ and show that conditions C1 and C2 hold, thus supporting σ as a Nash equilibrium.

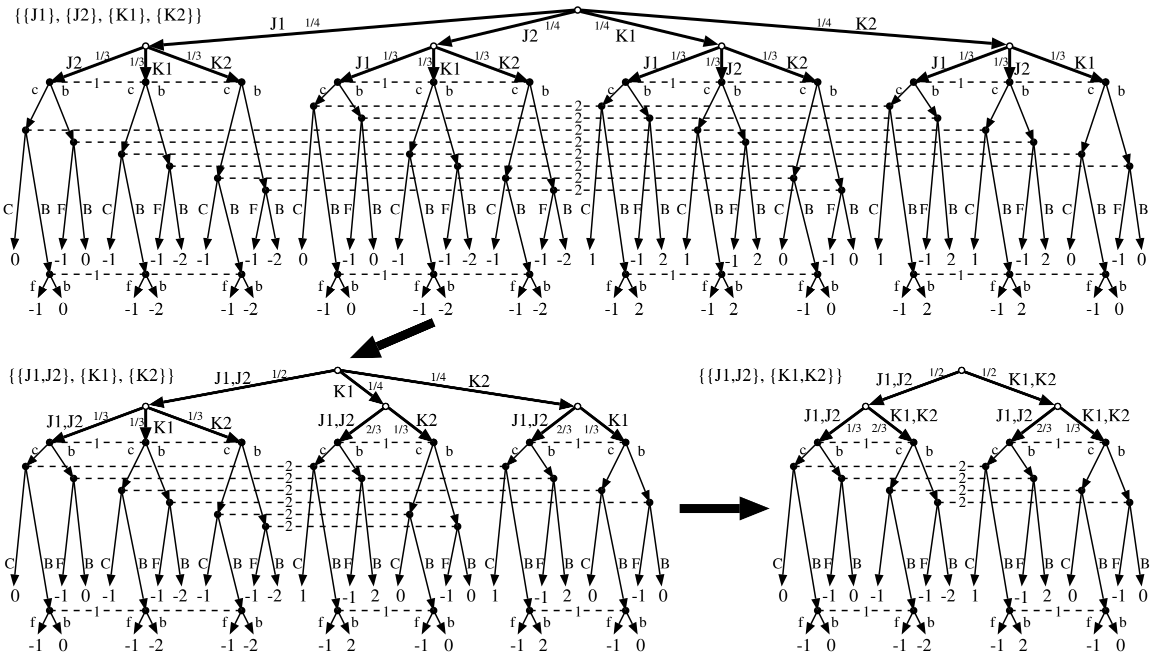

Figure 1: *GameShrink* applied to a tiny two-person four-card (two Jacks and two Kings) poker game. Next to each game tree is the range of the information filter F. Dotted lines denote information sets, which are labeled by the controlling player. Open circles are chance nodes with the indicated transition probabilities. The root node is the chance node for player 1's card, and the next level is for player 2's card. The payment from player 2 to player 1 is given below each leaf. In this example, the algorithm reduces the game tree from 53 nodes to 19 nodes.

Fix some player $i \in I$. Each of i's information sets in some round j corresponds to filtered signals $F^j\left(\tilde{\alpha}^{*j}, \tilde{\beta}_i^{*j}\right)$, history in the first $j-1$ rounds $(z_1, \ldots, z_{j-1}) \in \bigtimes_{k=1}^{j-1} \omega_{cont}^k$, and history so far in round j, $v \in V^j \setminus Z^j$. Let $\tilde{z} = (z_1, \ldots, z_{j-1}, v)$ represent all of the player actions leading to this information set. Thus, we can uniquely specify this information set using the information $\left(F^j\left(\tilde{\alpha}^{*j}, \tilde{\beta}_i^{*j}\right), \tilde{z}\right)$.

Each node in an information set corresponds to the possible private signals the other players have received. Denote by $\tilde{\beta}$ some legal member of

$$\left(F^j\left(\tilde{\alpha}^j, \tilde{\beta}_1^j\right), \ldots, F^j\left(\tilde{\alpha}^j, \tilde{\beta}_{i-1}^j\right), F^j\left(\tilde{\alpha}^j, \tilde{\beta}_{i+1}^j\right), \ldots, F^j\left(\tilde{\alpha}^j, \tilde{\beta}_n^j\right)\right).$$

In other words, there exists $\left(\tilde{\alpha}^j, \tilde{\beta}_1^j, \ldots, \tilde{\beta}_n^j\right)$ such that $\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) \in F^j\left(\tilde{\alpha}^{*j}, \tilde{\beta}_i^{*j}\right)$, $\left(\tilde{\alpha}^j, \tilde{\beta}_k^j\right) \in F^j\left(\tilde{\alpha}^j, \tilde{\beta}_k^j\right)$ for $k \neq i$, and no signals are repeated. Using such a set of signals $\left(\tilde{\alpha}^j, \tilde{\beta}_1^j, \ldots, \tilde{\beta}_n^j\right)$, let $\hat{\beta}'$ denote $\left(F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_1^j\right), \ldots, F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_{i-1}^j\right), F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_{i+1}^j\right), \ldots, F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_n^j\right)\right)$. (We will also abuse notation and write $F'^j_{-i}\left(\hat{\beta}\right) = \hat{\beta}'$.) We can now compute μ directly from μ':

$$\mu\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) = \begin{cases} \mu'\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) & \text{if } F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) \neq F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) \text{ or } \hat{\beta} = \hat{\beta}' \\ p^*\mu'\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) & \text{if } F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) = F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) \text{ and } \hat{\beta} \neq \hat{\beta}' \end{cases}$$

where $p^* = \frac{\Pr(\hat{\beta} \mid F^j(\tilde{\alpha}^j, \tilde{\beta}_i^j))}{\Pr(\hat{\beta}' \mid F'^j(\tilde{\alpha}^j, \tilde{\beta}_i^j))}$. The following claims show that μ as calculated above supports σ as a Nash equilibrium.

Claim 1 *μ is a valid belief system for Γ_F.*

Claim 2 *For all information sets h with $\Pr(h \mid \sigma) > 0$, $\mu(x) = \frac{\Pr(x \mid \sigma)}{\Pr(h \mid \sigma)}$ for all $x \in h$.*

Claim 3 *For all information sets h with $\Pr(h \mid \sigma) > 0$, σ is sequentially rational at h given μ.*

The proofs of Claims 1-3 are in Appendix B. By Claims 1 and 2, we know that condition C2 holds. By Claim 3, we know that condition C1 holds. Thus, σ is a Nash equilibrium. □

3.1 Non-trivial assumptions

Our model does not capture general sequential games of imperfect information because it is restricted in two ways: 1) all actions by the player (but not necessarily by nature) are observable,a nd 2) there is a common ordering of signals. In this subsection we show that removing either of these conditions can make our technique invalid.

First, we demonstrate a failure when removing the first assumption. Consider the game in Figure 2.[3] Nodes a and b are in the same information set, have the same parent (chance) node, have isomorphic subtrees with the same payoffs, and nodes c and d also have similar structural properties. By merging the subtrees beginning at a and b, we get the game on the right in Figure 2. In this game, player 1's only Nash equilibrium strategy is to play left. But in the original game, player 1 knows that node c will never be reached, and so should play right in that information set.

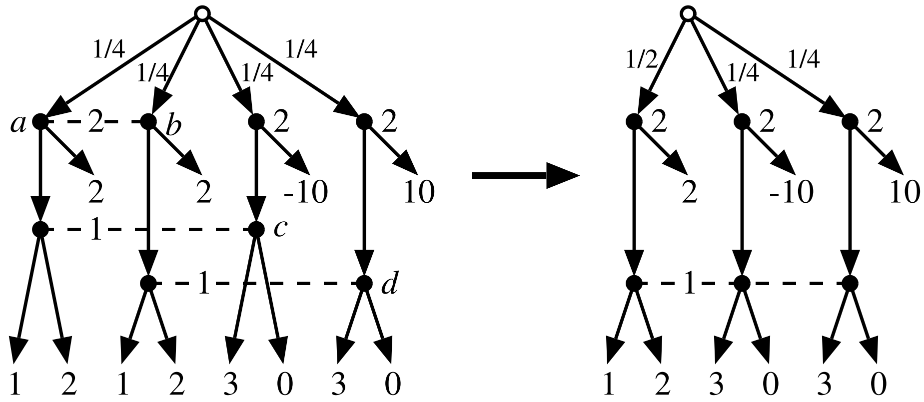

Figure 2: Example illustrating difficulty in developing a theory of equilibrium-preserving abstractions for general extensive form games with nonobservable actions.

Removing the second assumption (that the utility functions are based on a commond ordering of signals) can also cause failure. Consider a simple three-card game with a deck containing two Jacks ($J1$ and $J2$) and a King (K), where player 1's utility function is based on the ordering $K \succeq J1 \sim J2$ but player 2's

[3]We thank Albert Xin Jiang for providing this example.

utility function is based on the ordering $J2 \succeq K \succeq J1$. It is easy to check that in the abstracted game (where Player 1 treats $J1$ and $J2$ as being "equivalent") the Nash equilibrium does not correspond to a Nash equilibrium in the original game.[4]

4 An efficient algorithm for computing ordered game isomorphic abstraction transformations

We need the following subroutine for computing the ordered game ismorphic relation.

Algorithm 1 *OrderedGameIsomorphic?* $(\Gamma, F, \vartheta, \vartheta')$

1. *If ϑ and ϑ' are both leaves of the filtered (according to F) signal tree:*
   (a) *If $u^r(\vartheta \mid \tilde{z}) = u^r(\vartheta' \mid \tilde{z})$ for all $\tilde{z} \in \bigtimes_{j=1}^{r-1} \omega^j_{cont} \times \omega^r_{over}$, then return true.*
   (b) *Otherwise, return false.*
2. *Create a bipartite graph $G_{\vartheta,\vartheta'} = (V_1, V_2, E)$ with $V_1 = N(\vartheta)$ and $V_2 = N(\vartheta')$.*
3. *For each $v_1 \in V_1$ and $v_2 \in V_2$:*
   (a) *Create edge (v_1, v_2) if OrderedGameIsomorphic? (Γ, v_1, v_2).*
4. *Return true if $G_{\vartheta,\vartheta'}$ has a perfect matching; otherwise, return false.*

By evaluating this dynamic program from bottom to top, Algorithm 1 determines, in time polynomial in the size of the game tree, whether or not any pair of equal depth nodes x and y are ordered game isomorphic. Given this routine for determining ordered game isomorphisms in an ordered game, we are ready to present the main algorithm, *GameShrink*. Given as input a game Γ, it applies the shrinking ideas presented above as aggressively as possible. Once it finishes, there are no contractible nodes, and it outputs the corresponding information filter F. The correctness of *GameShrink* follows by a repeated application of Theorem 2.

Algorithm 2 *GameShrink* (Γ)

1. *Initialize F to be the identity filter for Γ.*
2. *For j from 1 to r:*
   For each pair of vertices ϑ, ϑ' with the same parent at either level $\sum_{k=1}^{j-1}\left(\kappa^k + n\gamma^k\right)$ or $\sum_{k=1}^{j} \kappa^k + \sum_{k=1}^{j-1} n\gamma^k$ in the filtered (according to F) signal tree:
   If $OrderedGameIsomorphic?(\vartheta, \vartheta')$, then $F^j(\vartheta) \leftarrow F^j(\vartheta') \leftarrow F^j(\vartheta) \cup F^j(\vartheta')$.
3. *Output F.*

4.1 Efficiency enhancements

We designed several speed enhancement techniques for *GameShrink*, and all of them are incorporated into our implementation. One technique is the use of the union-find data structure [8, Chapter 21] for storing the information filter F. This data structure uses time almost linear in the number of operations [38]. Initially each node in the signalling tree is its own set (this corresponds to the identity information filter); when two nodes are contracted they are joined into a new set. Upon termination, the filtered signals for the abstracted

[4]We thank an anonymous referee for providing this example.

game correspond exactly to the disjoint sets in the data structure. This is an efficient method of recording contractions within the game tree, and the memory requirements are only linear in the size of the signal tree.

Determining whether two nodes are ordered game isomorphic requires us to determine if a bipartite graph has a perfect matching. We can eliminate some of these computations by using easy-to-check necessary conditions for the ordered game isomorphic relation to hold. One such condition is to check that the nodes have the same number of chances as being ranked (according to $\succeq$) higher than, lower than, and the same as the opponents. We can precompute these frequencies for every game tree node. This substantially speeds up *GameShrink*, and we can leverage this database across multiple runs of the algorithm (for example, when trying different abstraction levels; see next section). The indices for this database depend on the private and public signals, but not the *order* in which they were revealed, and thus two nodes may have the same corresponding database entry. This makes the database significantly more compact. (For example in Texas Hold'em, the database is reduced by a factor $\binom{50}{3}\binom{47}{1}\binom{46}{1}/\binom{50}{5} = 20$.) We store the histograms in a 2-dimensional database. The first dimension is indexed by the private signals, the second by the public signals. The problem of computing the index in (either) one of the dimensions is exactly the problem of computing a bijection between all subsets of size r from a set of size n and integers in $\left[1, \ldots, \binom{n}{r}\right]$. We efficiently compute this using the subsets' *colexicographical ordering* [6].

5 Approximation methods

Some games are too large to compute an exact equilibrium, even after using the presented abstraction technique. In this section we discuss general techniques for computing approximately optimal strategy profiles. For a two-player game, we can always evaluate the worst-case performance of a strategy, thus providing some objective evaluation of the strength of the strategy. To illustrate this, suppose we know player 2's planned strategy for some game. We can then fix the probabilities of player 2's actions in the game tree as if they were chance moves. Then player 1 is faced with a single-agent decision problem, which can be solved bottom-up, maximizing expected payoff at every node. Thus, we can objectively determine the expected worst-case performance of player 2's strategy. This will be most useful when we want to evaluate how well a given strategy performs when we know that it is not an equilibrium strategy. (A variation of this technique may also be applied in n-person games where only one player's strategies are held fixed.) This technique provides *ex post* guarantees about the worst-case performance of a strategy, and can be used independently of the method that is used to compute the strategies in the first place.

5.1 State-space approximations

By slightly modifying the *GameShrink* algorithm we can obtain an algorithm that yields even smaller game trees, at the expense of losing the equilibrium guarantees of Theorem 2. Instead of requiring the payoffs at terminal nodes to match exactly, we can instead compute a penalty that increases as the difference in utility between two nodes increases.

There are many ways in which the penalty function could be defined and implemented. One possibility is to create edge weights in the bipartite graphs used in Algorithm 1, and then instead of requiring perfect matchings in the unweighted graph we would instead require perfect matchings with low cost (*i.e.*, only consider two nodes to be ordered game isomorphic if the corresponding bipartite graph has a perfect matching with cost below some threshold). Thus, with this threshold as a parameter, we have a knob to turn that in one extreme (threshold = 0) yields an optimal abstraction and in the other extreme (threshold = ∞) yields a highly abstracted game (this would in effect restrict players to ignoring all signals, but still observing actions). This knob also begets an *anytime* algorithm. One can solve increasingly less abstracted versions of the game, and evaluate the quality of the solution at every iteration using the *ex post* method discussed above.

5.2 Algorithmic approximations

In the case of two-player zero-sum games, the equilibrium computation can be modeled as a linear program (LP), which can in turn be solved using the simplex method. This approach has inherent features which we can leverage into desirable properties in the context of solving games.

In the LP, primal solutions correspond to strategies of player 2, and dual solutions correspond to strategies of player 1. The simplex method proceeds by simultaneously finding better and better primal and dual solutions (*i.e.*, better and better strategies for each player). Thus, the simplex method itself is an *anytime* algorithm (for a given abstraction). At any point in time, it can output the best strategies found so far.

Also, for any feasible solution to the LP, we can get bounds on the quality of the strategies by examining the primal and dual solutions. (When using the primal simplex method, dual solutions may be read off of the LP tableau.) Every feasible solution of the dual yields an upper bound on the optimal value of the primal, and vice versa [7, p. 57]. Thus, without requiring further computation, we get lower bounds on the expected utility of each agent's strategy against that agent's worst-case opponent. This is a method for finding ϵ-equilibria (*i.e.*, strategy profiles in which no agent can increase her expected utility more than ϵ by deviating), and can also be used as a termination criterion for an anytime algorithm.

6 Related research on abstraction

Abstraction techniques have been used in artificial intelligence research before. In contrast to our work, most (but not all) research involving abstraction has been for single-agent problems (*e.g.* [16, 25]). Furthermore, the use of abstraction typically leads to sub-optimal solutions, unlike the techniques presented in this paper, which yield optimal solutions. (A notable exception is the use of abstraction to compute optimal strategies in the (perfect information) game of Sprouts [1].)

One of the first pieces of research utilizing abstraction in multi-agent settings was the development of *partition search*, which is the algorithm behind GIB, the world's first expert-level computer bridge player [12, 13]. In contrast to other game tree search algorithms which store a particular game position at each node of the search tree, partition search stores *groups* of positions that it determines are equivalent. (Typically, the equivalence of two game positions is determined by ignoring irrelevant pieces of each game position and then checking whether the abstracted positions are consistent with each other.) Partition search can lead to substantial speed improvements over alpha-beta pruning and minimax search. However, it is not game-theoretically based (it does not consider information sets in the game tree), and thus does not allow one to solve for the equilibrium of a game of imperfect information, such as poker.[5]

There has been some research into the use of abstraction for games with imperfect information. Most notably, the paper by Billings *et al* [3] describes a manually constructed abstraction for the game of Texas Hold'em, and includes promising results against expert players. However, this approach has significant

[5]Bridge is also a game of imperfect information, and partition search does not find the equilibrium for that game either, although experimentally it plays quite well against human players. Instead, partition search is used in conjunction with statistical sampling to simulate the uncertainty in bridge. Other research that also uses perfect information search techniques in conjunction with statistical sampling and expert-defined abstraction has been developed for bridge [37]. Such (non-game-theoretic) techniques are unlikely to be helpful in poker games because of the greater importance on information hiding and bluffing.

drawbacks. First, it is highly specialized for Texas Hold'em. Second, a large amount of expert knowledge and effort was used in constructing the abstraction. Third, the abstraction does not preserve equilibrium: even if applied to a smaller game, it might not yield a game-theoretic equilibrium. Promising ideas for abstraction in the context of general extensive form games have been described in an extended abstract [32], but to our knowledge have not been fully developed.

7 Conclusions

We introduced the ordered game isomorphic abstraction transformation and gave an efficient algorithm, *GameShrink*, for automatically abstracting the game. We proved that in games with ordered signals, any Nash equilibrium in the smaller abstracted game maps directly to a Nash equilibrium in the original game. Using *GameShrink* we found the equilibrium to a poker game that is over four orders of magnitude larger than the largest poker game solved previously. We also introduced approximation methods for computing approximately optimal equilibria in general games, and described algorithmic techniques for devising bounds on suboptimality of the strategies. We described how all of these techniques can be converted into anytime algorithms.

References

[1] David Applegate, Guy Jacobson, and Daniel Sleator. Computer analysis of Sprouts. Technical Report CMU-CS-91-144, Carnegie Mellon University, Pittsburgh, PA, 1991.

[2] Nivan A. R. Bhat and Kevin Leyton-Brown. Computing Nash equilibria of action-graph games. In *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, Banff, Canada, 2004.

[3] 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)*, Acapulco, Mexico, 2003.

[4] Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. The challenge of poker. *Artificial Intelligence*, 134(1-2):201–240, 2002.

[5] Ben Blum, Christian R. Shelton, and Daphne Koller. A continuation method for Nash equilibria in structured games. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

[6] Béla Bollobás. *Combinatorics*. Cambridge University Press, 1986.

[7] Vasek Chvátal. *Linear Programming*. W. H. Freeman and Company, 1983.

[8] Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, second edition, 2001.

[9] Xiaotie Deng, Christos Papadimitriou, and Shmuel Safra. On the complexity of equilibria. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing*, pages 67–71, 2002.

[10] Nikhil R. Devanar, Christos H. Papadimitriou, Amin Saberi, and Vijay V. Vazirani. Market equilibrium via a primal-dual-type algorithm. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, pages 389–395, 2002.

[11] Andrew Gilpin and Tuomas Sandholm. Optimal Rhode Island Hold'em poker. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, PA, USA, 2005. Intelligent Systems Demonstration.

[12] Matthew L. Ginsberg. Partition search. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 228–233, Portland, OR, 1996.

[13] Matthew L. Ginsberg. GIB: Steps toward an expert-level bridge-playing program. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden, 1999.

[14] S. Govindan and R. Wilson. A global Newton method to compute Nash equilibria. *Journal of Economic Theory*, 110:65–86, 2003.

[15] Kamal Jain, M Mahdian, and Amin Saberi. Approximating market equilibria. In *Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2003.

[16] Craig A. Knoblock. Automatically generating abstractions for planning. *Artificial Intelligence*, 68(2):243–302, 1994.

[17] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast algorithms for finding randomized strategies in game trees. In *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC)*, pages 750–759, 1994.

[18] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2):247–259, 1996.

[19] Daphne Koller and Avi Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1):167–215, July 1997.

[20] David M. Kreps and Robert Wilson. Sequential equilibria. *Econometrica*, 50(4):863–894, 1982.

[21] H. Kuhn. Extensive games. *Proc. of the National Academy of Sciences*, 36:570–576, 1950.

[22] H. Kuhn. Extensive games and the problem of information. In H. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 2 of *Annals of Mathematics Studies, 28*, pages 193–216. Princeton University Press, 1953.

[23] Carlton Lemke and J. Howson. Equilibrium points of bimatrix games. *Journal of the Society of Industrial and Applied Mathematics*, 12:413–423, 1964.

[24] Kevin Leyton-Brown and Moshe Tennenholtz. Local-effect games. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

[25] Chao-Lin Liu and Michael Wellman. On state-space abstraction for anytime evaluation of Bayesian networks. *SIGART Bulletin*, 7(2):50–57, 1996. Special issue on Anytime Algorithms and Deliberation Scheduling.

[26] Andreu Mas-Colell, Michael Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[27] R. McKelvey and A. McLennan. Computation of equilibria in finite games. In R. H. Aumann, editor, *Handbook of Computational Economics*, volume 1. Elsevier, 1996.

[28] Peter Bro Miltersen and Troels Bjerre Sørensen. Computing sequential equilibria for two-player games, 2005. Manuscript.

[29] John Nash. Equilibrium points in n-person games. *Proc. of the National Academy of Sciences*, 36:48–49, 1950.

[30] Christos Papadimitriou. Algorithms, games and the Internet. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 749–753, 2001.

[31] Christos Papadimitriou and Tim Roughgarden. Computing equilibria in multi-player games. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 82–91, 2005.

[32] Avi Pfeffer, Daphne Koller, and Ken Takusagawa. State-space approximations for extensive form games, July 2000. Talk given at the First International Congress of the Game Theory Society, Bilbao, Spain.

[33] I. Romanovskii. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3:678–681, 1962.

[34] Rahul Savani and Bernhard von Stengel. Exponentially many steps for finding a Nash equilibrium in a bimatrix game. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, 2004.

[35] H. E. Scarf. The approximation of fixed points of a continuous mapping. *SIAM Journal of Applied Mathematics*, 15:1328–1343, 1967.

[36] Jiefu Shi and Michael Littman. Abstraction methods for game theoretic poker. In *Computers and Games*, pages 333–345. Springer-Verlag, 2001.

[37] Stephen J. J. Smith, Dana S. Nau, and Thomas Throop. Computer bridge: A big win for AI planning. *AI Magazine*, 19(2):93–105, 1998.

[38] Robert E. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225, 1975.

[39] Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

[40] Bernhard von Stengel. Computing equilibria for two-person games. In Robert Aumann and Sergiu Hart, editors, *Handbook of game theory*, volume 3. North Holland, Amsterdam, 2002.

A Extensive form games

Our model of an extensive form game is defined as usual.

Definition 7 *An n-*person game in extensive form *is a tuple $\Gamma = (I, V, E, P, H, A, u, p)$ satisfying the following conditions:*

1. *$I = \{0, 1, \ldots, n\}$ is a finite set of players. By convention, player 0 is the* chance *player.*
2. *The pair (V, E) is a finite directed tree with nodes V and edges E. Z denotes the leaves of the tree, called* terminal nodes. *$V \setminus Z$ are* decision nodes. *$N(x)$ denotes x's children and $N^*(x)$ denotes x's descendants.*
3. *$P : V \setminus Z \to I$ determines which player moves at each decision node. P induces a partition of $V \setminus Z$ and we define $P_i = \{x \in V \setminus Z \mid P(x) = i\}$.*
4. *$H = \{H_0, \ldots, H_n\}$ where each H_i is a partition of P_i. For each of player i's* information sets *$h \in H_i$ and for $x, y \in h$, we have $|N(x)| = |N(y)|$. We denote the information set of a node x as $h(x)$ and the player who controls h is $i(h)$.*
5. *$A = \{A_0, \ldots, A_n\}$, $A_i : H_i \to 2^E$ where for each $h \in H_i$, $A_i(h)$ is a partition of the set of edges $\{(x, y) \in E \mid x \in h\}$ leaving the information set h such that the cardinalities of the sets in $A_i(h)$ are the same and the edges are disjoint. Each $a \in A_i(h)$ is called an* action *at h.*
6. *$u : Z \to \mathbb{R}^N$ is the* payoff *function. For $x \in Z$, $u_i(x)$ is the payoff to player i in the event that the game ends at node x.*
7. *$p : H_0 \times \{a \in A_0(h) \mid h \in H_0\} \to [0, 1]$ where $\sum_{a \in A_0(h)} p(h, a) = 1$ for all $h \in H_0$ is the transition probability for chance nodes.*

In this paper we restrict our attention to games with *perfect recall* (formally defined in [22]), which means that players never forget information.

Definition 8 *An n-person game in extensive form satisfies* perfect recall *if the following two constraints hold:*

1. *Every path in (V, E) intersects h at most once.*
2. *If v and w are nodes in the same information set and there is a node u that preceeds v and $P(u) = P(v)$, then there must be some node x that is in the same information set as u and preceeds v and the paths taken from u to v is the same as from x to w.*

A straightforward representation for strategies in extensive form games is the *behavior strategy* representation. This is without loss of generality since Kuhn's theorem [22] states that for any mixed strategy there is a payoff-equivalent behavioral strategy in games with perfect recall. For each information set $h \in H_i$, a behavior strategy is $\sigma_i(h) \in \Delta(A_i(h))$ where $\Delta(A_i(h))$ is the set of all probability distributions over actions available at information set h. A group of strategies $\sigma = (\sigma_1, \ldots, \sigma_n)$ consisting of strategies for each player is a *strategy profile*. We sometimes write $\sigma_{-i} = (\sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n)$ and $(\sigma'_i, \sigma_{-i}) = (\sigma_1, \ldots, \sigma_{i-1}, \sigma'_i, \sigma_{i+1}, \ldots, \sigma_n)$.

B Proofs

Claim 1 *μ is a valid belief system for Γ_F.*

PROOF OF CLAIM 1. Let h be player i's information set after some history $\left(F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right)$. Clearly $\mu\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) \geq 0$ for all $\hat{\beta} \in h$. We need to show $\sum_{\hat{\beta} \in h} \mu\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) = 1$.

CASE 1. $F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) \neq F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)$. From the construction of F', $F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)$ is ordered game iso-

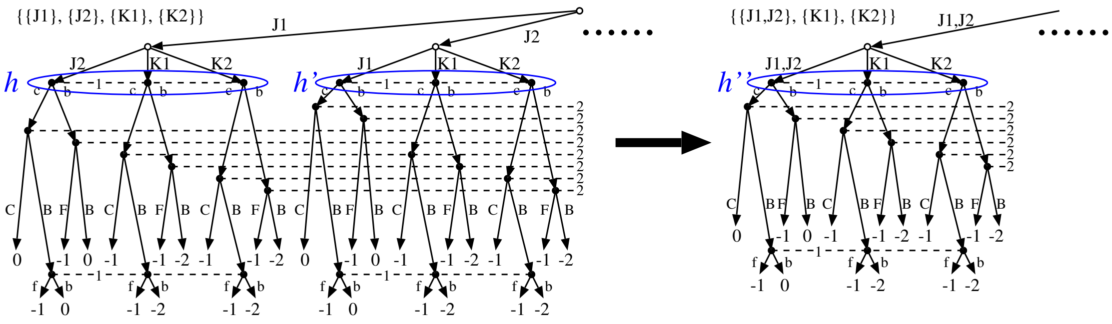

Figure 3: Illustration of Case 1 of Claim 1.

morphic to some $F^j\left(\tilde{\alpha}'^j, \tilde{\beta}_i'^j\right) \neq F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)$. Let h' be player i's information set corresponding to the history $\left(F^j\left(\tilde{\alpha}'^j, \tilde{\beta}_i'^j\right), \tilde{z}\right)$. By the definition of the ordered game isomorphism, there exists a perfect matching between the nodes in the information set h and h', where each matched pair of vertices corresponds to a pair of ordered game isomorphic information structures. Since $F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) = F'^j\left(\tilde{\alpha}'^j, \tilde{\beta}_i'^j\right)$, each edge in the matching corresponds to a node in the information set given by the history $\left(F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right)$ in $\Gamma_{F'}$; denote this information set by h''. (See Figure 3.) Thus, there is a bijection between h and h'' defined by the perfect matching. Using this matching:

$$\begin{aligned}
\sum_{\hat{\beta} \in h} \mu\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) &= \sum_{\hat{\beta} \in h} \mu'\left(F_{-i}'^j\left(\hat{\beta}\right) \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) \\
&= \sum_{\hat{\beta}' \in h''} \mu'\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) \\
&= 1.
\end{aligned}$$

CASE 2. $F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) = F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)$. We need to treat members of h differently depending on if they map to the same set of signals in $\Gamma_{F'}$ or not. Let $h_1 = \left\{\hat{\beta} \in h \mid \hat{\beta} = F_{-i}'^j\left(\hat{\beta}\right)\right\}$ and let $h_2 = \left\{\hat{\beta} \in h \mid \hat{\beta} \subset F_{-i}'^j\left(\hat{\beta}\right)\right\}$. Clearly (h_1, h_2) is a partition of h. Let h' be player i's information set corresponding to the history $\left(F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right)$ in $\Gamma_{F'}$. We can create a partition of h' by letting $h_3 = \left\{F_{-i}'^j\left(\hat{\beta}\right) \mid \hat{\beta} \in h_1\right\}$ and $h_4 = \left\{F_{-i}'^j\left(\hat{\beta}\right) \mid \hat{\beta} \in h_2\right\}$. Cleary (h_3, h_4) partitions h'. (See Figure 4.) The rest of the proof for this case proceeds in three steps.

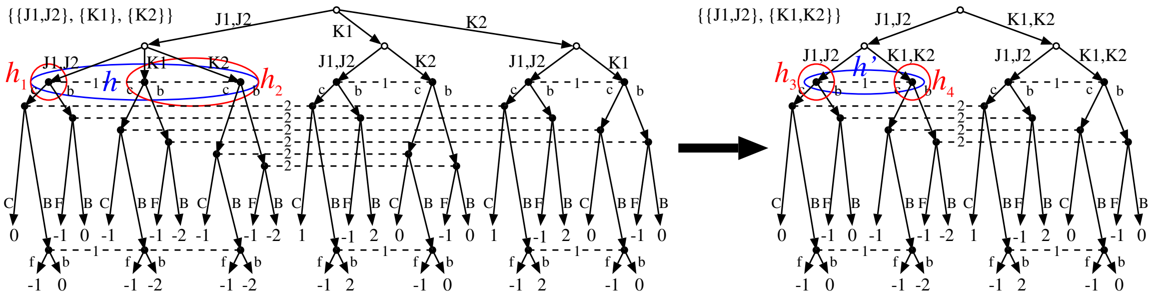

Figure 4: Illustration of Case 2 of Claim 1.

STEP 1. In this step we show the following relationship between h_1 and h_3:

$$\begin{aligned}\sum_{\hat\beta\in h_1}\mu\left(\hat\beta \mid F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right) &= \sum_{\hat\beta\in h_1}\mu'\left(F'^j_{-i}\left(\hat\beta\right)\mid F'^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right)\\ &= \sum_{\hat\beta'\in h_3}\mu'\left(\hat\beta'\mid F'^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right) \qquad (1)\end{aligned}$$

STEP 2. In this step we want to show a similar relationship between h_2 and h_4. In doing so, we use the following fact: $\hat\beta\subset\hat\beta'\rightarrow F'^j_{-i}\left(\hat\beta\right)=\hat\beta'$. With this in mind, we can write:

$$\sum_{\hat\beta\in h_2}\mu\left(\hat\beta|F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right)=\sum_{\hat\beta\in h_2}\frac{\Pr\left(\hat\beta|F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right)\right)}{\Pr\left(F'^j_{-i}\left(\hat\beta\right)|F'^j\left(\tilde\alpha^j,\tilde\beta_i^j\right)\right)}\mu'\left(F'^j_{-i}\left(\hat\beta\right)|F'^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right)$$

$$\begin{aligned}&= \sum_{\hat\beta'\in h_4}\sum_{\substack{\hat\beta\in h_2\\ \hat\beta\subset\hat\beta'}}\frac{\Pr\left(\hat\beta|F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right)\right)}{\Pr\left(F'^j_{-i}\left(\hat\beta\right)|F'^j\left(\tilde\alpha^j,\tilde\beta_i^j\right)\right)}\mu'\left(F'^j_{-i}\left(\hat\beta\right)|F'^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right)\\ &= \sum_{\hat\beta'\in h_4}\sum_{\substack{\hat\beta\in h_2\\ \hat\beta\subset\hat\beta'}}\frac{\Pr\left(\hat\beta|F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right)\right)}{\Pr\left(\hat\beta'|F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right)\right)}\mu'\left(\hat\beta'|F'^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right)\\ &= \sum_{\hat\beta'\in h_4}\mu'\left(\hat\beta'|F'^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right)\sum_{\substack{\hat\beta\in h_2\\ \hat\beta\subset\hat\beta'}}\frac{\Pr\left(\hat\beta|F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right)\right)}{\Pr\left(\hat\beta'|F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right)\right)}\\ &= \sum_{\hat\beta'\in h_4}\mu'\left(\hat\beta'|F'^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right) \qquad (2)\end{aligned}$$

STEP 3. Using (1) and (2):

$$\sum_{\hat\beta\in h}\mu\left(\hat\beta \mid F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right) = \sum_{\hat\beta\in h_1}\mu\left(\hat\beta \mid F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right)+\sum_{\hat\beta\in h_2}\mu\left(\hat\beta \mid F^j\left(\tilde\alpha^j,\tilde\beta_i^j\right),\tilde z\right)$$

$$
\begin{aligned}
&= \sum_{\hat{\beta}' \in h_3} \mu'\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) + \sum_{\hat{\beta}' \in h_4} \mu'\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) \\
&= \sum_{\hat{\beta}' \in h'} \mu'\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) \\
&= 1
\end{aligned}
$$

In both cases we have shown $\sum_{\hat{\beta} \in h} \mu\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) = 1$. □

Claim 2 *For all information sets h with $\Pr(h \mid \sigma) > 0$, $\mu(x) = \frac{\Pr(x \mid \sigma)}{\Pr(h \mid \sigma)}$ for all $x \in h$.*

PROOF OF CLAIM 2. Let h be player i's information set after some history $\left(F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right)$, and fix some $\hat{\beta} \in h$. Let $\hat{\beta}' = F'^j_{-i}\left(\hat{\beta}\right)$. We need to show that $\mu\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) = \frac{\Pr(\hat{\beta} \mid \sigma)}{\Pr(h \mid \sigma)}$. Let h' be player i's information set after history $\left(F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right)$.

CASE 1. $F^j(\tilde{\alpha}^j, \tilde{\beta}_i^j) \neq F'^j(\tilde{\alpha}^j, \tilde{\beta}_i^j)$.

$$
\begin{aligned}
\mu\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) &= \mu'\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) \\
&= \frac{\Pr\left(\hat{\beta}' \mid \sigma'\right)}{\Pr\left(h' \mid \sigma'\right)} \\
&= \frac{\frac{\Pr\left(\hat{\beta}, F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)}{\Pr\left(\hat{\beta}', F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)} \Pr\left(\hat{\beta}' \mid \sigma'\right)}{\frac{\Pr\left(\hat{\beta}, F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)}{\Pr\left(\hat{\beta}', F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)} \Pr\left(h' \mid \sigma'\right)} \\
&= \frac{\Pr\left(\hat{\beta} \mid \sigma\right)}{\Pr\left(h \mid \sigma\right)}
\end{aligned}
$$

CASE 2. $F^j(\tilde{\alpha}^j, \tilde{\beta}_i^j) = F'^j(\tilde{\alpha}^j, \tilde{\beta}_i^j)$ and $\hat{\beta} \neq \hat{\beta}'$.

$$
\begin{aligned}
\mu\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) &= \frac{\Pr\left(\tilde{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)}{\Pr\left(\tilde{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)} \mu'\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) \\
&= \frac{\Pr\left(\tilde{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)}{\Pr\left(\tilde{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)} \frac{\Pr\left(\hat{\beta}' \mid \sigma'\right)}{\Pr\left(h' \mid \sigma'\right)} \\
&= \frac{\Pr\left(\tilde{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)}{\Pr\left(\tilde{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)} \frac{\frac{\Pr\left(\tilde{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)}{\Pr\left(\tilde{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)} \Pr\left(\hat{\beta} \mid \sigma\right)}{\Pr\left(h \mid \sigma\right)} \\
&= \frac{\Pr\left(\hat{\beta} \mid \sigma\right)}{\Pr\left(h \mid \sigma\right)}
\end{aligned}
$$

CASE 3. $F^j(\tilde{\alpha}^j, \tilde{\beta}_i^j) = F'^j(\tilde{\alpha}^j, \tilde{\beta}_i^j)$ and $\hat{\beta} = \hat{\beta}'$.

$$\begin{aligned}
\mu\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) &= \mu'\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) \\
&= \frac{\Pr\left(\hat{\beta}' \mid \sigma'\right)}{\Pr\left(h' \mid \sigma'\right)} \\
&= \frac{\Pr\left(\hat{\beta} \mid \sigma\right)}{\Pr\left(h \mid \sigma\right)}
\end{aligned}$$

Thus we have $\mu(x) = \frac{\Pr(x \mid \sigma)}{\Pr(h \mid \sigma)}$ for all information sets h with $\Pr(h \mid \sigma) > 0$. □

Claim 3 *For all information sets h with $\Pr(h \mid \sigma) > 0$, σ is sequentially rational at h given μ.*

PROOF OF CLAIM 3. Suppose σ is not sequentially rational given μ. Then, there exists a strategy τ_i such that, for some $\left(F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right)$,

$$u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu\right) > u_i^j\left(\sigma_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu\right). \tag{3}$$

We will construct a strategy τ_i' for player i in $\Gamma_{F'}$ such that

$$u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu'\right) > u_i^j\left(\sigma_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu'\right),$$

thus contradicting the fact that σ' is a Nash equilibrium.

STEP 1. We first construct τ_i' from τ_i. For a given $F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)$, let

$$\Upsilon = \left\{F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right) \subseteq F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right\} \tag{4}$$

and let

$$\tau_{i,v}'^j\left(F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right) = \sum_{\vartheta \in \Upsilon} \Pr\left(\vartheta \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right) \tau_{i,v}^j\left(\vartheta, \tilde{z}\right).$$

In other words, the strategy τ_i' is the same as τ_i except in situations where only the filtered signal history is different, in which case τ_i' is a weighted average over the strategies at the corresponding information sets in Γ_F.

STEP 2. We need to show that $u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu'\right) = u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu\right)$ for all histories $\left(F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right)$. Fix $\left(F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}\right)$, and assume, w.l.o.g., the equality holds for all information sets coming after this one in Γ.

CASE 1. $F^j(\tilde{\alpha}^j, \tilde{\beta}_i^j) \neq F'^j(\tilde{\alpha}^j, \tilde{\beta}_i^j)$. Let z^j denote the current vertex of G^j and let Υ as in (4).

$u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu'\right)$

$$
\begin{aligned}
&= \sum_{\hat{\beta}' \in h'} \mu'\left(\hat{\beta}'\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}'\right) \\
&= \sum_{\hat{\beta} \in h} \mu'\left(F_{-i}'^j\left(\hat{\beta}\right)\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, F_{-i}'^j\left(\hat{\beta}\right)\right) \\
&= \sum_{\hat{\beta} \in h} \mu\left(\hat{\beta}\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, F_{-i}'^j\left(\hat{\beta}\right)\right) \\
&= \sum_{\hat{\beta} \in h} \mu\left(\hat{\beta}\right) \sum_{v \in N^j(z^j)} \tau_{i,v}'^j\left(\tilde{z}, F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), (\tilde{z}, v), F_{-i}'^j\left(\hat{\beta}\right)\right) \\
&= \sum_{\hat{\beta} \in h} \mu\left(\hat{\beta}\right) \sum_{v \in N^j(z^j)} \sum_{\vartheta \in \Upsilon} \Pr\left(\vartheta \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right) \tau_{i,v}^j\left(\tilde{z}, \vartheta\right) \cdot \\
&\qquad \left[u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), (\tilde{z}, v), F_{-i}'^j\left(\hat{\beta}\right)\right)\right] \\
&= \sum_{\hat{\beta} \in h} \mu\left(\hat{\beta}\right) \sum_{v \in N^j(z^j)} \sum_{\vartheta \in \Upsilon} \Pr\left(\vartheta \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right) \tau_{i,v}^j\left(\tilde{z}, \vartheta\right) \cdot \\
&\qquad \left[u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), (\tilde{z}, v), \hat{\beta}\right)\right] \\
&= \sum_{\hat{\beta} \in h} \mu\left(\hat{\beta}\right) \sum_{v \in N^j(z^j)} u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), (\tilde{z}, v), \hat{\beta}\right) \cdot \\
&\qquad \left[\sum_{\vartheta \in \Upsilon} \Pr\left(\vartheta \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right) \tau_{i,v}^j\left(\tilde{z}, \vartheta\right)\right] \\
&= \sum_{\hat{\beta} \in h} \mu\left(\hat{\beta}\right) \sum_{v \in N^j(z^j)} \tau_{i,v}^j\left(\tilde{z}, F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right) u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), (\tilde{z}, v), \hat{\beta}\right) \\
&= \sum_{\hat{\beta} \in h} \mu\left(\hat{\beta}\right) u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}\right) \\
&= u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu\right)
\end{aligned}
$$

CASE 2. $F^j(\tilde{\alpha}^j, \tilde{\beta}_i^j) = F'^j(\tilde{\alpha}^j, \tilde{\beta}_i^j)$. Let h_1, h_2, h_3, and h_4 as in the proof of Case 2 of Claim 1. We can show

$$
\sum_{\hat{\beta}' \in h_3} \mu'\left(\hat{\beta}'\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}'\right) = \sum_{\hat{\beta} \in h_1} \mu\left(\hat{\beta}\right) u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}\right) \tag{5}
$$

using a procedure similar to that in Case 1. We can show the following relationship between h_2 and h_4:

$$
\sum_{\hat{\beta}' \in h_4} \mu'\left(\hat{\beta}'\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}'\right)
$$

$$
\begin{aligned}
&= \sum_{\hat{\beta}' \in h_4} \sum_{\substack{\hat{\beta} \in h_2 \\ \hat{\beta} \subset \hat{\beta}'}} \frac{\Pr\left(\hat{\beta} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)}{\Pr\left(\hat{\beta}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right)} \mu'\left(\hat{\beta}'\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}'\right) \\
&= \sum_{\hat{\beta}' \in h_4} \sum_{\substack{\hat{\beta} \in h_2 \\ \hat{\beta} \subset \hat{\beta}'}} \mu\left(\hat{\beta}\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}'\right) \\
&= \sum_{\hat{\beta}' \in h_4} \sum_{\substack{\hat{\beta} \in h_2 \\ \hat{\beta} \subset \hat{\beta}'}} \mu\left(\hat{\beta}\right) \sum_{v \in N^j(z^j)} \tau_{i,v}'^j\left(\tilde{z}, F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), (\tilde{z}, v), \hat{\beta}'\right) \\
&= \sum_{\hat{\beta}' \in h_4} \sum_{\substack{\hat{\beta} \in h_2 \\ \hat{\beta} \subset \hat{\beta}'}} \mu\left(\hat{\beta}\right) \sum_{v \in N^j(z^j)} \tau_{i,v}^j\left(\tilde{z}, F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right)\right) u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), (\tilde{z}, v), \hat{\beta}\right) \\
&= \sum_{\hat{\beta}' \in h_4} \sum_{\substack{\hat{\beta} \in h_2 \\ \hat{\beta} \subset \hat{\beta}'}} \mu\left(\hat{\beta}\right) u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}\right) \\
&= \sum_{\hat{\beta} \in h_2} \mu\left(\hat{\beta}\right) u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}\right) \qquad (6)
\end{aligned}
$$

Using (5) and (6):

$$
\begin{aligned}
u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu'\right) &= \sum_{\tilde{\beta}' \in h'} \mu'\left(\tilde{\beta}'\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \tilde{\beta}'\right) \\
&= \sum_{\hat{\beta}' \in h_3} \mu'\left(\hat{\beta}'\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}'\right) \\
&\quad + \sum_{\hat{\beta}' \in h_4} \mu'\left(\hat{\beta}'\right) u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}'\right) \\
&= \sum_{\hat{\beta} \in h_1} \mu\left(\hat{\beta}\right) u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}\right) \\
&\quad + \sum_{\hat{\beta} \in h_2} \mu\left(\hat{\beta}\right) u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}\right) \\
&= \sum_{\hat{\beta} \in h} \mu\left(\hat{\beta}\right) u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \hat{\beta}\right) \\
&= u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu\right)
\end{aligned}
$$

In both cases we have shown:

$$
u_i^j\left(\tau_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu'\right) = u_i^j\left(\tau_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu\right). \qquad (7)
$$

STEP 3. We can show that

$$
u_i^j\left(\sigma_i, \sigma_{-i} \mid F^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu\right) = u_i^j\left(\sigma_i', \sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j, \tilde{\beta}_i^j\right), \tilde{z}, \mu'\right). \qquad (8)
$$

using a procedure similar to the previous step.

STEP 4. Combining (3), (7), and (8), we have:

$$u_i^j\left(\tau_i',\sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j,\tilde{\beta}_i^j\right),\tilde{z},\mu'\right)=u_i^j\left(\tau_i,\sigma_{-i} \mid F^j\left(\tilde{\alpha}^j,\tilde{\beta}_i^j\right),\tilde{z},\mu\right)>$$
$$u_i^j\left(\sigma_i,\sigma_{-i} \mid F^j\left(\tilde{\alpha}^j,\tilde{\beta}_i^j\right),\tilde{z},\mu\right)=u_i^j\left(\sigma_i',\sigma_{-i}' \mid F'^j\left(\tilde{\alpha}^j,\tilde{\beta}_i^j\right),\tilde{z},\mu'\right).$$

Thus, σ' is not a Nash equilibrium. Therefore, by contradiction, σ is sequentially rational at all information sets h with $\Pr(h \mid \sigma) > 0$. □

C Rhode Island Hold'em rules and modeling as an ordered game

Rhode Island Hold'em is a poker game which in this case is played by 2 players. The game was invented as a testbed for computer game-playing research [36], and it was designed so that it was similar in style to Texas Hold'em, yet not so large that devising reasonably intelligent strategies would be impossible. The game play proceeds as follows.

1. Each player pays an *ante* of 5 chips which is added to the *pot*. Both players initially receive a single card, face down; these are known as the *hole cards*.
2. After receiving the hole cards, the players participate in one betting round. Each player may *check* (not placing any money in the pot and passing) or *bet* (placing 10 chips into the pot) if no bets have been placed. If a bet has been placed, then the player may *fold* (thus forfeiting the game along with any money they have put into the pot), *call* (adding chips to the pot equal to the last player's bet), or *raise* (calling the current bet and making an additional bet). In Rhode Island Hold'em, the players are limited to three bets each per betting round. (A raise equals two bets.) In the first betting round, the bets are equal to 10 chips.
3. After the first betting round, a community card is dealt face up. This is called the *flop* card. Another betting round take places at this point, with bets equal to 20 chips.
4. Following the second betting round, another community card is dealt face up. This is called the *turn* card. A final betting round takes place at this point, with bets again equal to 20 chips.
5. If neither player folds, then the *showdown* takes place. Both players turn over their cards. The player who has the best 3-card poker hand takes the pot. In the event of a draw, the pot is split evenly.

Hands in 3-card poker games are ranked slightly differently than 5-card poker hands. The main differences are that the order of flushes and straights are reversed, and a three of a kind is better than straights or flushes. Table 1 describes the rankings. Within ranks, ties are broken by by ordering hands according to the rank of cards that make up the hand. If players are still tied after applying this criterion, *kickers* are used to determine the winner. A kicker is a card that is not used to make up the hand. For example, if player 1 has a pair of eights and a five, and player 2 has a pair of eights and a six, player 2 wins.

An ordered game is given by the tuple $\Gamma = \langle I, G, L, \Theta, \kappa, \gamma, p, \succeq, \omega, u\rangle$. Here we define each of these components for Rhode Island Hold'em. There are two players so $I = \{1, 2\}$. There are three rounds, and the stage game is the same in each round so we have $G = \langle G_{RI}, G_{RI}, G_{RI}\rangle$ where G_{RI} is given in Figure 5, which also specifies the player label L. Θ is the standard deck of 52 cards. The community cards are dealt in the second and third rounds, so $\kappa = \langle 0, 1, 1\rangle$. Each player receives a since face down card in the first round only, so $\gamma = \langle 1, 0, 0\rangle$. p is the uniform distribution over Θ. $\succeq$ is defined for three card hands and is defined using the ranking given in Table 1. The game-ending nodes ω are denoted in Figure 5 by ω. u is defined as in the above description; it is easy to verify that it satisfies the necessary conditions.

Rank	Hand	Prob.	Description	Example
1	Straight flush	0.00217	3 cards w/ consecutive rank and same suit	K♠, Q♠, J♠
2	Three of a kind	0.00235	3 cards of the same rank	Q♠, Q♡, Q♣
3	Straight	0.03258	3 cards w/ consecutive rank	3♣, 4♠, 5♡
4	Flush	0.04959	3 cards of the same suit	2♢, 5♢, 8♢
5	Pair	0.16941	2 cards of the same rank	2♢, 2♠, 3♡
6	High card	0.74389	None of the above	J♣, 9♡, 2♠

Table 1: Rankings of three-card poker hands.

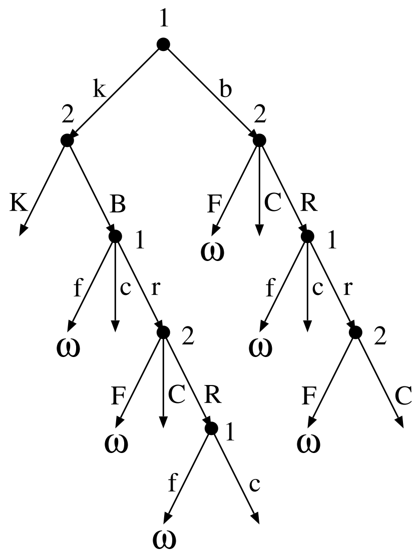

Figure 5: Stage game G_{RI}, player label L, and game-ending nodes ω for Rhode Island Hold'em.