Automatizability and Simple Stochastic Games

Lei Huang* and Toniann Pitassi*

University of Toronto
{leih,toni}@cs.toronto.edu

Abstract. The complexity of simple stochastic games (SSGs) has been open since they were defined by Condon in 1992. Despite intensive effort, the complexity of this problem is still unresolved. In this paper, building on the results of [4], we establish a connection between the complexity of SSGs and the complexity of an important problem in proof complexity–the proof search problem for low depth Frege systems. We prove that if depth-3 Frege systems are weakly automatizable, then SSGs are solvable in polynomial-time. Moreover we identify a natural combinatorial principle, which is a version of the well-known Graph Ordering Principle (GOP), that we call the integer-valued GOP (IGOP). We prove that if depth-2 Frege plus IGOP is weakly automatizable, then SSG is in P.

1 Introduction

In a groundbreaking paper from 1992, Condon defined the family of *simple stochastic games* (SSGs) [13]. A simple stochastic game is a directed graph with four types of vertices: min nodes, max nodes, average nodes, and two sinks, a 0-sink and a 1-sink. The game is played by two players, Max and Min, from a given start node. The goal of Max is to reach the 1-sink, while the goal of Min is to reach the 0-sink, or to continue the game indefinitely. The value of the game is the probability that Max wins, assuming that both players play optimally.

A fascinating open question is determining the complexity of finding the value of a game, or equivalently, the complexity of finding an optimal strategy. From a practical point of view, stochastic games are used to model a variety of problems in software verification and controller optimization, and from a theoretical point of view, the SSG problem is fundamental since an efficient algorithm for it will imply algorithms for a host of other problems. Indeed, it is known to be polynomial-time equivalent to many other computational problems such as the generalized linear complementarity problem, and the minimum stable circuit problem. Many other game-theoretic problems, such as mean payoff games, discounted payoff games, and parity games, all reduce to SSGs [2]. Furthermore, SSG is also the complete problem for the class AUC-SPACE($\log n$) of logspace bounded alternating random turing machines [13].

Despite considerable effort over the last decades, it remains a longstanding open problem to determine the complexity of SSGs. Condon [13,14] proved that the SSG decision problem, "does the max player have a strategy ensuring at least $\frac{1}{2}$ probability of winning", is in both NP and coNP. This makes SSGs one of the few combinatorial problems known to be in $NP \cap coNP$ and suggests that it is very unlikely to be

* Supported by NSERC.

L. Aceto, M. Henzinger, and J. Sgall (Eds.): ICALP 2011, Part I, LNCS 6755, pp. 605–617, 2011.

NP-complete. Another natural avenue for obtaining a hardness result for SSGs are the complexity classes associated with total functions in NP, such as PLS and PPAD. In fact, recently the complexity of the Nash Equilibrium problem was resolved in a celebrated sequence of papers, culminating in a proof of the PPAD-completeness of Nash [11,15] Unfortunately, such a hardness result for SSGs is unlikely, as it was shown to lie in *both* PLS and PPAD [6].

The SSG problem has also been studied extensively from an algorithmic point of view [14,17,24,22,18,5,21]. Many restrictions such as allowing only two of three types of vertices, or to a constant number of random nodes, are known to have polynomial time algorithms [14,17]. However, the current best algorithm for the full game is a $O(2^{\sqrt{n}})$ time randomized algorithm [22,18], and a $O(n \cdot |V_r|!)$ time deterministic algorithm [17], where V_r is the set of average nodes.

A seemingly unrelated, but also longstanding and important problem is to determine whether proofs in standard proof systems can be found efficiently. A proof system $\mathcal{P}$ is *automatizable* if there exists an algorithm that takes as input a tautology f, and outputs a $\mathcal{P}$-proof of f, and such that the runtime is polynomial in the size of the shortest $\mathcal{P}$-proof of f. $\mathcal{P}$ is *weakly automatizable* if there is an algorithm that on input f and a number r in unary, can distinguish the case where f is not a tautology from the case where f has a $\mathcal{P}$-proof of size at most r. It is known that $\mathcal{P}$ is weakly-automatizable if and only if there is an automatizable proof system that simulates $\mathcal{P}$ [3].

The question of whether standard proof systems are automatizable is a fundamental question in logic and automated theorem proving [9]. Following [20], it was shown that Frege systems are not weakly automatizable under a widely believed cryptographic assumption – that the Diffie Hellman (DH) problem (and hence factoring) is hard to compute. However, despite considerable effort, the weak automatizability question for low-depth Frege systems is unresolved. [7] showed that AC_0^k-Frege is not weakly automatizable under the assumption that the DH problem can be solved in time $exp(n^{c/k})$, where $c > 2$. The best algorithm for DH runs in time $exp(n^{1/2})$, and moreover the number field sieve is conjectured to solve DH in time $exp(n^{1/3})$; thus for small k (k less than 5 or 6), the weak automatizability of depth-k Frege is unresolved. Even for depth-1 Frege (i.e. resolution), there is no clear evidence for or against weak automatizability, despite results in both directions [1,12].

Results and Related Work. In a recent paper, Atserias and Maneva made an important new link between automatizability and game theory by proving that solving mean payoff games (MPGs) is reducible to the weak automatizability of depth-2 Frege systems and to feasible interpolation of depth-3 Frege systems[4]. In this paper, we prove that if depth-3 Frege systems are weakly automatizable, then simple stochastic games are solvable in polynomial time, thus establishing a link between SSGs and an important open problem in proof complexity.

Mean payoff games can be viewed as a special case of simple stochastic games, but depth-2 Frege systems can also be viewed as a special case of depth-3 Frege thus the results cannot be directly compared. However, the increase in depth suggests an approach to pinpointing a difference between MPGs and SSGs, an interesting open problem in its own right. Moreover, the only part of our proof that is not contained in depth-2 is in the proof of a natural combinatorial property about graphs that we will call the Integer-Valued Graph Ordering Principle (IGOP). IGOP states informally that in any finite undirected graph where all nodes are labelled by integers, there exists a vertex whose value is at least as large as its neighbors. This principle is expressible as a CNF formula, and we actually prove that if depth-2 Frege, augmented with IGOP, is weakly automatizable, then SSGs are in P. This raises the very interesting question as to the exact proof theoretic strength required to prove IGOP: If it has a polynomial size depth-2 Frege proof, then our result subsumes that of [4]. On the other hand, if not, then we have found a natural CNF formula separating depth-2 from depth-3 Frege, and furthermore, expose an essential difference between mean payoff games and simple stochastic games.

Our proof builds on [4] in that we use and further develop very low depth circuits that they invent for performing addition of a constant number of integers. However, at a high level our proofs are very different. [4] goes through an intermediate transformation to the Max Atom Problem that appears to hold only for mean payoff games. In contrast our proof strategy is much more generalizable. The main idea relies on the unique fixed point property of SSGs - ie SSGs can be characterized as a fixed point computation where the unique fixed point indicates the winner. This property is shared by a variety of other problems, including the more general class of stochastic games defined by Shapley [23].

Proof Overview. The basic idea behind our proof is to study the complexity of *stopping* games, which are polynomial-time equivalent to general SSGs. In a stopping game, the optimal solution is also the only solution to a set of local optimality conditions. We formalize the formulas, $MinWin(G)$ (and $MaxWin(G)$) expressing that there is a locally optimal strategy with value less than or equal to one half (or greater than one half). Since the locally optimal solution is always unique for stopping games, the formula $F(G) = MinWin(G) \wedge MaxWin(G)$ is unsatisfiable. The bulk of our argument is thus to show that $F(G)$, has an efficient depth-3 Frege refutation. Then if depth-3 Frege has feasible interpolation, the interpolant for F will return whether $MinWin(G)$ or $MaxWin(G)$ is unsatisfiable, thus revealing the winner of the game. Since weak automatizability implies interpolation, it immediately follows that weak automatizability of depth-3 Frege implies that SSG is in P. The technical difficulty is to prove that for any stopping game G, the locally optimal solution is unique. This involves a very careful analysis of low depth circuits for performing arithmetic on sets of numbers, as well as very efficient reasoning about these arithmetic formulas.

2 Definitions

Simple Stochastic Games. In a *simple stochastic game* (SSG) two players take turns moving a pebble along a directed graph $G : (V, E)$ with two terminal positions, a 0-sink and a 1-sink. Conventionally the game also has a unique source node from which gameplay starts. All non terminal nodes of G are partitioned into max nodes, min nodes, and random nodes. From a max (min) nodes, Max (Min) chooses the out-edge the pebble takes next. At a random node, the successor node is chosen by chance. We will consider only binary SSGs where every node has out degree two and random nodes choose each out-edge with probability $\frac{1}{2}$. All SSGs have such a binary equivalent.

In a given play, the max player wins if the pebble reaches the 1-sink and the min player wins otherwise (either by reaching the 0-sink or continuing play indefinitely). Let σ be a strategy for Max, $\sigma : V^* \to V$ mapping paths in a play to a legal successor node, and τ be a strategy for Min. The *value* of a node under σ and τ, $v_{\sigma,\tau}(i)$, is defined to be the probability that the game ends at 1-sink if the players use strategies σ, τ and the pebble starts at node *i*. The *optimal value* $v_{opt}(i)$ of a node is the minimax over all strategies $v_{opt}(i) = \max_\sigma \min_\tau v_{\sigma,\tau}(i) = \min_\tau \max_\sigma v_{\sigma,\tau}(i)$ The value of the game is $v_{opt}(0)$ where node 0 is the start node.

Fact 1. *(Shapley, 53) For every SSG there exists pure positional strategies σ and τ that achieve $v_{opt}(i)$ at every node i.*

A strategy can be thought of as simply a mapping $V \to V$ and optimal strategies ergodic, they are optimal regardless of which vertex is the start vertex.

Fact 2. *(Condon, 92) The optimal values of any n node SSG can be written as $\frac{a}{b}$ where $a, b \in \mathbb{N}$, $b = O(2^n)$*

Definition 1. *For a simple stochastic game G we define the associated function $I_G : \{0,1\}^n \to \{0,1\}^n$ as follows.*

$$I_G(\vec{x})(i) = \begin{cases} \max\{\vec{x}(j), \vec{x}(k)\} & \textit{if i is a max vertex with successors } j,k \\ \min\{\vec{x}(j), \vec{x}(k)\} & \textit{if i is a min vertex with successors } j,k \\ \frac{1}{2}\vec{x}(j) + \frac{1}{2}\vec{x}(k) & \textit{if i is a average vertex with successors } j,k \\ 0/1 & \textit{if i is the zero/one sink} \end{cases}$$

The vector $\vec{x}$ is *stable* for G F if for all *i*, $I_G(\vec{x})(i) = \vec{x}(i)$. If $\vec{v}$ is the vector of optimal values, $\vec{v}(i) = v_{opt}(i)$ for all *i*, then $\vec{v}$ is *stable* for G. In general, there can be more than one stable vector. However, for every game G, there is a corresponding SSG, G' such that G' has a unique stable vector and moreover, G has optimal value greater than $1/2$ if and only if G' has optimal value greater than $1/2$, [13,23].

Definition 2. *Let G be an n node SSG. The m-*stopping game *G' corresponding to G is a simple stochastic game with $n + mn$ nodes constructed such that any play eventually halts at a terminal node. For each vertex i in the original graph of G, G' has a network of average nodes structured as in Fig. 1. Every in-edge to i in G is replaced by an in-edge to e_{mi} in G'.*

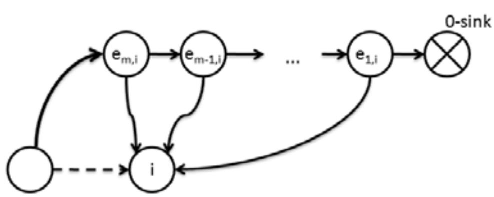

Fig. 1. Average node network added to every node *i* of original game *G*

Theorem 3. *(Condon '92, Shapley 53') If G is an m-*stopping game *(for any m), then G has a unique stable vector $\vec{x}$.*

Theorem 3 implies that finding the optimal values for a stopping game G is equivalent to finding a stable vector for I_G. This characterization of optimal values will be very useful to us since it allows us to express the statement "$\vec{v}$ is the vector of optimal values" in terms of very low level equalities and sums.

Theorem 4. *(Condon '92, Shapley 53') For every G there exists a constant c such that the cn*-stopping game *G' has value $> \frac{1}{2}$ if and only iff G has value $> \frac{1}{2}$.*

Formulas and Proof Systems. We will work with the propositional sequent calculus, PK. (See [19] for details.) The *size* of a PK proof is the sum of the sizes of all formulas occurring in the proof. The *depth* of a formula is the number of alternations of OR and ANDs. A formula is Σ_k^+ (Π_k^+) if it has depth $k+1$, where the top connective is OR (AND), and the bottom connective has constant size; a formula is Δ_k^+ if it can be written both as a Σ_k^+ and as a Π_k^+ formula. A depth-k PK proof is a PK proof where every formula in the proof has depth at most k. It is important to note that depth-k PK proofs are the same as depth-$k+1$ Frege proofs in other non-sequent style axiomatizations. We will present PK proofs where every formula in the proof is a Δ_k^+ formula, for some k. Such proofs translate into depth $k-1$ PK proofs, or equivalently, into depth k Frege proofs (in non sequent-style proof systems).

Definition 3. *A proof system $\mathcal{P}$ is automatizable if there exists an algorithm A such that for all unsatisfiable formulas f $A(f)$ returns a $\mathcal{P}$-proof of f, and the runtime of A on f is polynomial in the size of the smallest $\mathcal{P}$-proof of f. $\mathcal{P}$ is weakly automatizable if there exists a proof system that polynomially simulates $\mathcal{P}$ and that is automatizable.*

Definition 4. *Let $F = A(\vec{x},\vec{z}) \wedge B(\vec{y},\vec{z})$ be an unsatisfiable formula. An algorithm C is an interpolant for F if for all $\alpha \in \{0,1\}^{|\vec{z}|}$, $C(\alpha) = 1$ implies $A(\vec{x},\alpha)$ is unsatisfiable, and $C(\alpha) = 0$ implies $B(\vec{y},\alpha)$ is unsatisfiable. A proof system $\mathcal{P}$ admits feasible interpolation if for all unsatisfiable formulas $F = A(\vec{x},\vec{z}) \wedge B(\vec{y},\vec{z})$, there exists an algorithm C that is an interpolant for F, and the runtime of C is polynomial in the size of the shortest $\mathcal{P}$-proof of F. Note that, if $\mathcal{P}$ is weakly automatizable then $\mathcal{P}$ has feasible interpolation.*

Definition 5. *Let G be a n node graph with constant out-degree 2 where each node is labeled with an integer x_i, not all zero. The formula IGOP(G) intuitively says that there exists some i with a nonzero label at least as large as that of its children. Formally, if $\mathbf{x}(i)$ is a set of variables representing the value x_i in binary, IGOP(G) is the formula $\bigvee_i \mathbf{x}(i) \neq 0 \longrightarrow \bigvee_i [\mathbf{x}(i) > 0] \wedge [\mathbf{x}(i) \geq \mathbf{x}(c_1(i))] \wedge [\mathbf{x}(i) \geq \mathbf{x}(c_2(i))]$ where $c_1(i)$ and $c_2(i)$ are the children of i and $\geq$ denotes a standard Δ_2^+ formula for comparison. Note that $[\mathbf{x}(i) \geq \mathbf{x}(c_1(i))] \wedge [\mathbf{x}(i) \geq \mathbf{x}(c_2(i))]$ is NC^0 of Δ_2^+ which can be written in Δ_2^+ so that IGOP(G) is in Σ_2^+ formula.*

Definition 6. *The formula Max-Node$_n$ intuitively expresses that in any set of n integers, there exists a maximal element. Formally, let $\mathbf{x}(i)$ be the set of variables representing the value x_i. Max-Node$_n = \bigvee_i \bigwedge_{j \neq i} [\mathbf{x}(i) \geq \mathbf{x}(j)]$.*

For our purposes, we'd like to apply the IGOP principle to graphs where the nodes are labeled with differences of integers. This does not inherently add to the depth *IGOP*(G) since we will develop a Δ_2^+ circuit to compare difference (see Sect. 4).

Definition 7. *Let Σ_2^+-Frege+IGOP denote the Σ_2^+-Frege proof system augmented with the following axiom schema for IGOP. Let $\mathbf{d}(i)$ be shorthand for the difference $|\mathbf{x}(i) - \mathbf{y}(i)|$*

$$\bigvee_i [\mathbf{x}(i) \neq \mathbf{y}(i)] \longrightarrow \bigvee_i [\mathbf{d}(i) > 0] \wedge [\mathbf{d}(i)) \geq \mathbf{d}(c_1(i))] \wedge [\mathbf{d}(i) \geq \mathbf{d}(c_2(i))]$$

which states intuitively that if $\mathbf{x}$ *and* $\mathbf{y}$ *are not equal for every i then there exists an i where the difference* $\mathbf{d}(i)$ *is nonzero at least as the value of* $\mathbf{d}$ *at the children of i.*

We will show that *IGOP* has a Σ_3 proof (via Max-Node$_n$), Σ_2^+-Frege+*IGOP* is a special case of Σ_3-Frege.

3 Main Result

In this section we prove our main theorem, showing that if Σ_3-Frege admits feasible interpolation, then SSG can be solved in polynomial time.

Theorem 5. *If Σ_2^+-Frege+IGOP has feasible interpolation, then SSG is in P.*

Corollary 1. *If Σ_3-Frege has feasible interpolation, then SSG is in P.*

The main idea behind the reduction is as follows. Given an SSG, G, we first construct an m-stopping game G', where G' has value greater than $1/2$ if and only if G has value greater than $1/2$. This implies that the following statement is unsatisfiable.

$$F_{G'} = \left(I_{G'}(\vec{x}) = \vec{x} \wedge \vec{x}(0) > \frac{1}{2} \right) \bigwedge \left(I_{G'}(\vec{w}) = \vec{w} \wedge \vec{w}(0) \leq \frac{1}{2} \right),$$

where each $\vec{x}(i)$ and $\vec{w}(i)$ is an integer represented by a length N binary string where N is $O(n)$.

For every stopping game G' we will prove that $F_{G'}$ has a polynomial-sized Σ_3-Frege refutation. Thus if Σ_3-Frege has feasible interpolation, then the interpolant for F'_G solves the SSG-value problem for G. In order to provide a short Σ_3-Frege refutation of F'_G, it suffices to provide a polynomial-sized Σ_3-Frege proof of $Uniqueness(G) =$ "$I_G(\vec{x}) = \vec{x}, I_G(\vec{w}) = \vec{w} \longrightarrow \vec{x} = \vec{w}$".

To see this, note that if $Uniqueness(G)$ has a short Σ_3-Frege proof, then: (1) From $I_G(\vec{x}) = \vec{x}$ and $I_G(\vec{w}) = \vec{w}$, we can derive $\vec{x} = \vec{w}$; (2) Secondly, from $\vec{x} = \vec{w}$, we can derive $\vec{x}(0) = \vec{w}(0)$; and finally (3) From $\vec{x}(0) = \vec{w}(0)$, we can derive $\vec{x}(0) \leq \frac{1}{2}$, which contradicts F_G.

Theorem 6. *For any stopping simple stochastic game G, the statement $Uniqueness(G)$ can be proved in Σ_2^+-Frege+IGOP.*

We first present a proof of Theorem 3, the uniqueness theorem for SSGs. The remainder of our paper will focus on showing that this can be formalized in depth-3 Frege.

Proof. Theorem 3
Let G' be a m-stopping game. Suppose that $\vec{w}$ and $\vec{x}$ are stable for G' and $\vec{x} \neq \vec{w}$. Denote $\vec{d}$ the difference vector, $\vec{d} = |\vec{w} - \vec{x}|$. Let i be a node such that $\vec{d}(i) > 0$. The stopping game structure induces a discount on the original. If node i had children j and k in the original game, then in any stable solution $\vec{v}$ for G' i's children are nodes with the values $\left(1 - \frac{1}{2^m}\right)\vec{v}(j)$ and $\left(1 - \frac{1}{2^m}\right)\vec{v}(k)$. This implies that $\vec{d}(i) = \left(1 - \frac{1}{2^m}\right)(F(\vec{x}(j), \vec{w}(k)) - F(\vec{w}(j), \vec{w}(k)))$ where $F \in \{\min, \max, \text{ave}\}$, and thus at least one of $\vec{d}(j)$ or $\vec{d}(k)$ is strictly greater than $\vec{d}(i)$. If we choose i to be a node where $\vec{d}(i) \geq \vec{d}(j)$ and $\vec{d}(i) \geq \vec{d}(k)$ then we reach a contradiction. (Note that such an i exists since choosing i that maximizes $\vec{d}(i)$ satisfies these conditions.)

4 Arithmetic Formulas

We will represent integers using two's complement binary notation. In two's complement a set of variables $z_1 z_2 ... z_n$ has the value $z = -2^{n-1} \cdot z_1 + 2^{n-2} \cdot z_2 + ... + 2^1 \cdot z_{n-1} + 2^0 \cdot z_n$. To prevent overflow, we will also pad each integer by a constant k where k is the largest number of summands used in any operation so that $z_1 z_2 ... z_n \rightarrow z_1^k z_2 z_3 ... z_n$.

Let $\vec{x}$ be a vector that satisfies $\vec{x} = I_G(\vec{x})$ for some game G. Recall that in any SSG, the values can be written as $\frac{a}{b}$ where $b = O(2^n)$. After normalizing to a common denominator D and padding by k, each $x(i)$ will be expressed as a set of N binary variables $x_1 x_2 ... x_N$ so that $x(i) = \frac{1}{D}\left(-2^{N-1} \cdot x_1 + 2^{N-2} \cdot x_2 + ... + 2^1 \cdot x_{N-1} + 2^0\right)$

For the remainder of the paper, $[F(x)]$ will be used to represent the Δ_2^+ logical formula that is equivalent to the mathematical formula, $F(x)$. To simplify notation, only one set of brackets will be used when any nesting occurs so that $[[F(x)] \leftrightarrow [G(x)]]$ is just $[F(x) \leftrightarrow G(x)]$. Boldface variables $\mathbf{x}$ denote an array of $k \times N$ array of integers and $\mathbf{x}(i)$ is the i^{th} row. Normal font variables x represent a single integer expressed with N binary variables. We can represent the following formulas as Δ_2^+ formulas. See [19] for precise definitions.

- $[x \leftrightarrow y]$ is a formula that is true if and only if $\forall i\,(x_i \leftrightarrow y_i)$
- $[\mathbf{x}(1) + ... + \mathbf{x}(k) - \mathbf{y}(1)... - \mathbf{y}(l)]$ represents a set of N formulas, such that the i^{th} formula is true iff the i^{th} bit of $\mathbf{x}(1) + ... + \mathbf{x}(k) - \mathbf{y}(1)... - \mathbf{y}(k)$ is 1.
- $[\mathbf{x}(1) + ... + \mathbf{x}(k) - \mathbf{y}(1)... - \mathbf{y}(l)] \geq z]$ is true if and only if the sum on the left is $\geq z$. And similarly $[\mathbf{x}(1) + ... + \mathbf{x}(k) - \mathbf{y}(1)... - \mathbf{y}(l)] < z]$ is true iff the sum is less than z
- $[\max(x,y)]$ represents a set of formulas where the i^{th} formula is x_i iff $x = max(x,y)$ and $[\min(x,y)]$ represents the set of formulas where the i^{th} formula is x_i iff $x = min(x,y)$.

The technical difficulty lies in defining formulas, $F^k_{i \in 1,...,N}$ for calculating all of the bits in the sum of k $N-$bit integers for any constant k. Given $\{F_i^k\}$ the remainder of the formulas are fairly straightforward. For arbitrary k, [4] shows that there exists a Δ_2^+ formula for calculating whether the sum of k integers generates an overflow. We extend their formula so that it calculates all bits, F_i^k, of the sum. The following lemmas comprise the bulk of the work of our proof. (Proofs are in [19].)

Lemma 1 (Substitution)
For any Δ_2^+ formula $F(x_1, x_2, ...x_N, z...)$ there exists a short PK proof of the following sequent, where all formulas in the proof are Δ_2^+ formulas. $[x \leftrightarrow y] \longrightarrow [F(x_1, x_2, ...x_N, z...) \leftrightarrow F(y_1, y_2, ...y_N, z...)]$

Lemma 2. *For N bit integers x, y the following sequents have short PK proofs where all formulas are Δ_2^+: (1) $[x \geq y] \longrightarrow [\max(x, y) = x]$, and (2) $[x \geq y] \longrightarrow [\min(x, y) = y]$.*

Lemma 3 (Nested Addition). *For N bit integers a, b, c, and array of constant k N−bit integers* $\mathbf{x}$ *there exists short PK proofs of the following where all formulas are Δ_2^+:*

$$[a + b \leftrightarrow c] \longrightarrow [\mathbf{x}(1) + ... + \mathbf{x}(k) + a + b \leftrightarrow \mathbf{x}(1) + ... + \mathbf{x}(k) + c]$$
$$[a - b \leftrightarrow c] \longrightarrow [\mathbf{x}(1) + ... + \mathbf{x}(k) + a + \overline{b} + 1 \leftrightarrow \mathbf{x}(1) + ... + \mathbf{x}(k) + c]$$

Corollary 2. *Given $[x + a + b \leftrightarrow 1^N]$ we can derive $[\overline{x} \leftrightarrow a + b]$ for any integers x, a, b. In particular, $[x \leftrightarrow a - b] \longrightarrow [\overline{x} \leftrightarrow \overline{a} + b]$, and $[x \leftrightarrow a + b] \longrightarrow [\overline{x} \leftrightarrow \overline{a} - \overline{b}]$.*

Lemma 4. *Let x, y, x', y' be N-bit integers, and $m > 0$, and $\frac{1}{2^m}y$ represents $0^m y_1 y_2 ... y_{N-m}$. Then there exists a short PK proof of the following formula, where all formulas are Δ_2^+:*

$$[x > x'], [x \leftrightarrow y - \frac{1}{2^m}y], [x' \leftrightarrow y' - \frac{1}{2^m}y'] \longrightarrow [x - x' - y + y' < 0]$$

Lemma 5. *Fix an $a \times N$ array* $\mathbf{z}$ *and a $b \times N$ array* $\mathbf{x}$ *where each integer is padded by $k = a + b$ bits. There is a short proof of the following:*

$$[\mathbf{z}(1) + ...\mathbf{z}(a) < 0], [\mathbf{x}(1) + ... + \mathbf{x}(b) < 0] \longrightarrow [\mathbf{z}(1) + ...\mathbf{z}(a) + \mathbf{x}(1) + ...\mathbf{x}(b) < 0]$$
$$[\mathbf{z}(1) + ...\mathbf{z}(a) \geq 0], [\mathbf{x}(1) + ... + \mathbf{x}(b) \geq 0] \longrightarrow [\mathbf{z}(1) + ...\mathbf{z}(a) + \mathbf{x}(1) + ...\mathbf{x}(b) \geq 0]$$

5 Proof of Uniqueness

Equipped with the above lemmas, we can now prove uniqueness for SSGs in depth-3 Frege. Let $\mathbf{x}$ and $\mathbf{w}$ be arrays of integers as defined in the previous section such that each integer is padded by k bits and $\mathbf{x}(i)$ is the value associated with the i^{th} node of a $m-$stopping game G. We define a formula, Uniqueness(G) on the $N \cdot (n + mn)$ variables defined above which will state that if $\mathbf{x} = I_G(\mathbf{x})$ and $\mathbf{w} = I_G(\mathbf{w})$, then $\mathbf{x}(i) = \mathbf{w}(i)$ for every i. The premise $\mathbf{x} = I_G(\mathbf{x})$ can be stated as follows, using the arithmetic formulas defined in the previous section.

(1) For every max node i, with children e_{mj} and e_{mk}, $[\mathbf{x}(i) \leftrightarrow \max(\mathbf{x}(e_{mj}), \mathbf{x}(e_{mk}))]$
(2) For every min node i, with children e_{mj} and e_{mk}, $[\mathbf{x}(i) \leftrightarrow \min(\mathbf{x}(e_{mj}), \mathbf{x}(e_{mk}))]$
(3) For every average node i, with children e_{mj} and e_{mk}, $[2 \cdot \mathbf{x}(i) \leftrightarrow \mathbf{x}(e_{mj}) + \mathbf{x}(e_{mk})]$
(4) For every new average node of the form e_{ij}, $[\mathbf{x}(e_{ij}) \leftrightarrow \mathbf{x}(j) - \frac{1}{2^i}\mathbf{x}(j)]$.

For any i, we will prove that if $d(i)$ (the absolute value of $x(i) - w(i)$) is nonzero and at least as large as its neighbors, $d(j)$ and $d(k)$, then we can derive a contradiction. Formally, let $M(i)$ be the formula:

$$[\mathbf{x}(i) > \mathbf{w}(i)], \forall s \in \{j, k\} ([\mathbf{x}(i) - \mathbf{w}(i) - \mathbf{x}(s) + \mathbf{w}(s) \geq 0]),$$
$$\forall s \in \{j, k\} ([\mathbf{x}(i) - \mathbf{w}(i) - \mathbf{w}(s) + \mathbf{x}(s) \geq 0])$$

We will prove for every i: $M(i), [\mathbf{x} = I_G(\mathbf{x})], [\mathbf{w} = I_G(\mathbf{w})] \longrightarrow \bot$. Finally we can use the IGOP axiom schema to prove that there exists an i such that $M(i)$ holds which allows us to conclude: $[\mathbf{x} = I_G(\mathbf{x})], [\mathbf{w} = I_G(\mathbf{w})] \longrightarrow [\mathbf{x}(1) \leftrightarrow \mathbf{w}(1)], ..., [\mathbf{x}(n) \leftrightarrow \mathbf{w}(n)]$.

Case 1. i is a max node with children j and k (Proof for i is a min node is similar.) As in proof of Theorem 3, we can further divide Case 1 into four cases depending on the sign of $[\mathbf{x}(j) - \mathbf{x}(k)]$ and $[\mathbf{w}(j) - \mathbf{w}(k)]$. Because of the way we defined comparisons, $[a < b]$ is exactly $\neg[a \geq b]$ so it it's not hard to show that exactly one of four sub cases is true. We show a complete proof for when $[\mathbf{x}(e_{mj}) \geq \mathbf{x}(e_{mk})]$, $[\mathbf{w}(e_{mj}) \geq \mathbf{w}(e_{mk})]$, the remaining three subcases are in [19].

Let $H(\mathbf{x}, \mathbf{w})$ denote the subcase premise $[\mathbf{x}(e_{mj}) \geq \mathbf{x}(e_{mk})]$, $[\mathbf{w}(e_{mj}) \geq \mathbf{w}(e_{mk})]$. We prove for contradiction that $M(i), H(\mathbf{x}, \mathbf{w}), [\mathbf{x} = I_G(\mathbf{x})], [\mathbf{w} = I_G(\mathbf{w})] \longrightarrow \bot$. Using Lemma 2 we can prove $[\mathbf{x}(e_{mj}) \geq \mathbf{x}(e_{mk})] \longrightarrow [\max(\mathbf{x}(e_{mj}), \mathbf{x}(e_{mk})) \longrightarrow \mathbf{x}(e_{mj})]$. And by weakening the LHS we have $H(\mathbf{x}, \mathbf{w}) \longrightarrow [\max(\mathbf{x}(e_{mj}), \mathbf{x}(e_{mk})) \leftrightarrow \mathbf{x}(e_{mj})]$. From the statements forming $[\mathbf{x} = I_G(\mathbf{x})]$ above we can apply $[\mathbf{x}(i) \leftrightarrow \max(\mathbf{x}(e_{mj}), \mathbf{x}(e_{mk}))]$ and $[\mathbf{x}(e_{mj}) \leftrightarrow \mathbf{x}(j) - \frac{1}{2^m}\mathbf{x}(j)]$ to prove

$(1) : [\mathbf{x} = I_G(\mathbf{x})], H(\mathbf{x}, \mathbf{w}) \longrightarrow [\mathbf{x}(i) \leftrightarrow \mathbf{x}(j) - \frac{1}{2^m}\mathbf{x}(j)]$ and $(2) : [\mathbf{w} = I_G(\mathbf{w})], H(\mathbf{x}, \mathbf{w}) \longrightarrow [\mathbf{w}(i) \leftrightarrow \mathbf{w}(j) - \frac{1}{2^m}\mathbf{w}(j)]$. The next step is to prove the following sequent, which follows from Lemma 4:

$$(3) : [\mathbf{x}(i) > \mathbf{w}(i)], [\mathbf{x}(i) \leftrightarrow \mathbf{x}(j) - \frac{1}{2^m}\mathbf{x}(j)], [\mathbf{w}(i) \leftrightarrow \mathbf{w}(j) - \frac{1}{2^m}\mathbf{w}(j)] \longrightarrow [\mathbf{x}(i) - \mathbf{w}(i) - \mathbf{x}(j) + \mathbf{w}(j) < 0]$$

Finally to put things together, we apply cut on (1),(2) and (3) to obtain:

$$(4) : M(i), H(\mathbf{x}, \mathbf{w}), [\mathbf{x} = I_G(\mathbf{x})], [\mathbf{w} = I_G(\mathbf{w})] \longrightarrow [\mathbf{x}(i) - \mathbf{w}(i) - \mathbf{x}(j) + \mathbf{w}(j) < 0]$$

which is the same as $M(i), H(\mathbf{x}, \mathbf{w}), [\mathbf{x} = I_G(\mathbf{x})], [\mathbf{w} = I_G(\mathbf{w})] \longrightarrow \bot$.

Case 2. For an average node i, it is easy to show that $\mathbf{x}(j) - \mathbf{w}(j) > \mathbf{x}(e_{mj}) - \mathbf{w}(e_{mj})$ and $\mathbf{x}(k) - \mathbf{w}(k) > \mathbf{x}(e_{mk}) - \mathbf{w}(e_{mk})$ just as we did in the max node case using Lemma 4. By Lemma 5 we can combine the two negative sums above to the larger negative sum

$$[\mathbf{x}(e_{mj}) - \mathbf{w}(e_{mj}) - \mathbf{x}(j) + \mathbf{w}(j) + \mathbf{x}(e_{mk}) - \mathbf{w}(e_{mk}) - \mathbf{x}(k) + \mathbf{w}(k) < 0]$$

The goal is to make the substitutions that relate this inequality back to $\mathbf{x}(i) - \mathbf{w}(i)$. We can always add zero to any sum without changing its value, and due to Lemma 3 we can also add a pair of integers that sum to zero. Using this trick we will replace the terms containing e_{mj} and e_{mk} with terms containing i.

When i is an average node, $2\mathbf{x}(i) = [\mathbf{x}(e_{mj}) + \mathbf{x}(e_{mk})]$. Recall that $2x(i)$ does not represent a sum but a new set of binary variables $\mathbf{x}(i)_2 ... \mathbf{x}(i)_N 0$. This is very convenient as we can now freely use Lemma 3 to add $0 = 2\mathbf{x}(i) - 2\mathbf{x}(i)$ and $0 = 2\mathbf{w}(i) - 2\mathbf{w}(i)$ to the negative sum above to prove that

$$[2\mathbf{x}(i) - 2\mathbf{x}(i) + 2\mathbf{w}(i) - 2\mathbf{w}(i) + \mathbf{x}(e_{mj}) - \mathbf{w}(e_{mj}) - \mathbf{x}(j) + \mathbf{w}(j) + \mathbf{x}(e_{mk}) - \mathbf{w}(e_{mk}) - \mathbf{x}(k) + \mathbf{w}(k) < 0]$$

In order to cancel out the e_{mj} and e_{mk} terms, we need to make the following four substitutions:

- $2\mathbf{x}(i) \rightarrow [\mathbf{x}(i) + \mathbf{x}(i)]$
- $\overline{2\mathbf{x}(i)} \rightarrow [\overline{\mathbf{x}(e_{mj})} - \overline{\mathbf{x}(e_{mk})}]$
- $2\mathbf{w}(i) \rightarrow [\mathbf{w}(e_{mj}) + \mathbf{w}(e_{mk})]$
- $\overline{2\mathbf{w}(i)} \rightarrow [\overline{\mathbf{w}(i)} - \overline{\mathbf{w}(i)}]$

After removing the zero terms, we finally arrive at

$$[\mathbf{x} = I_G(\mathbf{x})], [\mathbf{w} = I_G(\mathbf{w})] \longrightarrow [\mathbf{x}(i) + \mathbf{x}(i) - \mathbf{w}(i) - \mathbf{w}(i) - \mathbf{x}(j) + \mathbf{w}(j) - \mathbf{x}(k) + \mathbf{w}(k) < 0]$$

On the other hand, if $M(i)$ is true then both $[\mathbf{x}(i) - \mathbf{w}(i) - \mathbf{x}(j) + \mathbf{w}(j) \geq 0]$ and $[\mathbf{x}(i) - \mathbf{w}(i) - \mathbf{x}(k) + \mathbf{w}(k) \geq 0]$ are true. Applying Lemma 5 to the two positive sums we can derive that the same sum must be positive.

$$M(i) \longrightarrow [\mathbf{x}(i) - \mathbf{w}(i) - \mathbf{x}(j) + \mathbf{w}(j) + \mathbf{x}(i) - \mathbf{w}(i) - \mathbf{x}(k) + \mathbf{w}(k) \geq 0]$$

As with the max/min nodes, we have shown that for all average nodes i from $M(i), [\mathbf{x} = I_G(\mathbf{x})], [\mathbf{w} = I_G(\mathbf{w})]$ we can derive a contradiction. Thus for any i, $M(i) \longrightarrow Uniqueness(G)$. Using the *IGOP* axiom schema, we can immediately derive $\bigvee_i M(i)$ from which we can conclude $Uniqueness(G)$.

5.1 Proving Max-Node$_n$

To achieve the Σ_3 result, it remains to prove Max-Node$_n$. Suppose that $x(i) - w(i)$ is maximal among the first $n-1$ nodes, and $x(n) \geq w(n)$ ($w(n) > x(n)$ follows by symmetry). Either (1) $[x(i) - w(i) \geq x(n) - w(n)]$ or (2) $[x(i) - w(i) < x(n) - w(n)]$. If (1) is true then $x(i) - w(i)$ remains maximal. If (2) is true then for each j, $[x(n) - w(n) - x(j) + w(j) < 0]$ would imply that $[x(i) - w(i) - x(j) + w(j)]$ as well (Lemma 5 followed by Lemma 3). Since we know the latter is not true, we can prove that for all j, $[x(n) - w(n) \geq x(j) - w(j)]$. In either case, we've proven Max-Node$_n$, by Σ_3^+ proofs.

5.2 Removing the Bottom Fan-In

So far, we have shown how to prove the uniqueness property with Σ_3^+ proofs, or with depth Σ_2^+ proofs plus the IGOP principle. With one additional idea, we can remove the small bottom fanin. Recall that the bottom fanin was only a constant – suppose the largest bottom fanin is c. We can replace our original formula, F (in our case the formula asserting uniqueness), by a new formula, F', as follows. First, we introduce (polynomially many) new variables, one associated with each conjunction of at most c literals. The new uniqueness formula F' asserts that G implies F, where G is a conjunction of new clauses (called "extension axioms") asserting that each new variable is equivalent to its associated conjunction. The new formula F' is still of polynomial size, and furthermore it is a tautology if and only if F is a tautology. It is a standard argument to show that if F has a PK proof where all formulas in the proof are Δ_k^+, then F' has a PK proof where all formulas in the proof are Δ_k. To complete the proof of Theorem 5, we replace our original uniqueness formula, F, by the new uniqueness formula, F'. Since F has efficient Σ_3^+-Frege proofs, F' has efficient Σ_3-Frege proofs.

6 Consequences and Open Problems

Tightness of our Result. In this paper, we have shown that if depth-2 Frege systems plus the Integer-Valued Graph Ordering Principle (IGOP) is automatizable, then SSGs are in P. This can be viewed as either an "easyness" result for SSGs or a hardness result for automatizability but more importantly this paper and [4] open up new ways of approaching both problems. Of course, any result in the converse direction would further strengthen this connection. An interesting question is whether there exists an efficient depth-2 proof of uniqueness for SSGs (this would follow immediately if IGOP could be proven in depth-2). However, we conjecture that IGOP does not have efficient depth-2 proofs. Our conjecture is based on the fact that this principle is simply an instance of the usual GOP but where the variables are replaced by Δ_2 formulas. GOP is a well-known propositional tautology as it is the classic (and only) example of a family of formulas that exhibits the tightness of size-width tradeoffs lower bounds for resolution in that it has efficient proofs but requires $O(\sqrt{n})$ clause width [8,16]. Our conjecture just says that this phenomena continues to hold when we scale GOP up to higher depth. In fact we make the stronger conjecture that our depth-3 proof of Uniqueness for SSGs is tight. Moreover, we conjecture that the depth-2 proofs of totality for mean payoff games [4] are also tight:

Conjecture 1. (1)Uniqueness for SSGs does not have polynomial-size depth-2 Frege proofs; (2)Totality of MPGs does not have polynomial-size depth-1 (resolution) proofs.

If Conjecture (1) is true, it would imply that SSGs cannot be efficiently reduced (in the type-2 setting) to mean payoff games. This follows from the main result of Atserias and Maneva [4], together with a theorem due to Morioka and Buresh-Oppenheim [10] (Theorem 10 in their paper).

More generally, it is interesting to study low-depth, efficient reductions between statements of totality of various game-theoretic problems. Just as we study the relative strength of various search problems, it is natural to consider the relative proof theoretic strength of the underlying principles, by studying efficient reductions between them in standard low-depth propositional proof systems. The study of total search problems in general, and their corresponding proofs of totality are widely studied in proof complexity. Indeed, the strength of weak systems of arithmetic are classified in terms of what kinds of search problems are provably total in the theory. (For every search problem there is a first order formula expressing that the search problem is total, and conversely for every statement of the form $\exists x A(x,y)$, there is a corresponding total search problem.) PLS figures prominently in this research, as it is precisely the class of total functions that are provably total in the theory TV^1. It would be interesting to expand this work to include the study of statements of totality for various game-theory problems, as well as to study low-depth reductions between such statements. In particular, consider the propositional statements expressing the following principles: (a) totality for SSGs, (b) totality of mean payoff games, (c) the iteration principle (underlying PLS), (d) the pigeonhole principle (underlying PPAD). Which ones can and cannot be reduced in low-depth to one another? Answers to these questions would likely yield an understanding of the relative strengths of the corresponding search classes: SSG, MPG, PLS, and PPAD.

References

1. Alekhnovich, M., Razborov, A.: Resolution is not automatizable unless w[p] is tractable. In: IEEE Symposium on Foundations of Computer Science (2001)
2. Andersson, D., Miltersen, P.B.: The complexity of solving stochastic games on graphs. In: Dong, Y., Du, D.-Z., Ibarra, O. (eds.) ISAAC 2009. LNCS, vol. 5878, pp. 112–121. Springer, Heidelberg (2009)
3. Atserias, A., Bonet, M.: On the automatizability of resolution and related propositional proof systems. Information and Computation 189(2), 182–201 (2004)
4. Atserias, A., Maneva, E.: Mean-payoff games and propositional proofs. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010. LNCS, vol. 6198, pp. 102–113. Springer, Heidelberg (2010)
5. Bjorklund, H., Vorobyov, S.: Combinatorial structure and randomized subexponential algorithms for infinite games. Theoretical Computer Science 349, 347–360 (2005)
6. Blum, M., Juba, B., Williams, R.: Non-monotone behaviors in min/max/avg circuits and their relationship to simple stochastic games
7. Bonet, M., Domingo, C., Gavalda, R., Maciel, A., Pitassi, T.: No feasible interpolation or automatization for AC^0-Frege proof systems (1998) (manuscript)
8. Bonet, M.L., Galesi, N.: optimality of size-width tradeoffs for resolution. Computational Complexity 10, 261–276 (2001)
9. Bonet, M.L., Pitassi, T., Raz, R.: On interpolation and automatization for frege systems. SIAM J. Comput. 29, 1939–1967 (2000)
10. Buresh-Oppenheim, J., Morioka, T.: Relativized np search problems and propositional proof systems. In: 19th IEEE Conference on Computational Complexity, CCC (2004)
11. Chen, X., Deng, X., Teng, S.H.: Settling the complexity of two-player nash equilibria. Journal of the ACM 56 (2009)
12. Clegg, M., Edmonds, J., Impagliazzo, R.: Using the Gröbner basis algorithm to find proofs of unsatisfiability. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, pp. 174–183 (May 1996)
13. Condon, A.: The complexity of stochastic games. Information and Computation 96, 203–224 (1992)
14. Condon, A.: On algorithms for simple stochastic games. In: Advances in Computational Complexity Theory. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 13, pp. 51–73 (1993)
15. Daskalakis, C., Goldberg, P., Papadimtriou, C.H.: The complexity of computing a nash equilibrium. SIAM Journal on Computing 39, 195–259 (2009)
16. Galesi, N., Lauria, M.: Optimality of size-degree tradeoffs for polynomial calculus. ACM Trans. Comput. Logic 12, 4:1–4:22 (2010)
17. Gimbert, H., Horn, F.: Simple stochastic games with few random vertices are easy to solve. In: Amadio, R. (ed.) FOSSACS 2008. LNCS, vol. 4962, pp. 5–19. Springer, Heidelberg (2008)
18. Halman, N.: Simple stochastic games, parity games, mean payoff games and discounted payoff games are all lp-type problems. Algorithmica 49(1), 37–50 (2007)
19. Huang, L., Pitassi, T.: Automatizability and simple stochastic games (2011), http://www.cs.toronto.edu/~leih/ssg_full.pdf
20. Krajicek, K., Pudlak, P.: Some consequences of cryptographic conjectures for S_2^1 and EF. In: Leivant, D. (ed.) LCC 1994. LNCS, vol. 960, pp. 210–220. Springer, Heidelberg (1995)
21. Kumar, V., Tripathi, R.: Algorithmic results in simple stochastic games. Technical Report 855, University of Rochester (2004)

22. Ludwig, W.: A subexponential randomized algorithm for the simple stochastic game problem. Inf. Comput. 117, 151–155 (1995)
23. Shapley, L.S.: Stochastic games. Proceedings of the National Academy of Sciences U.S.A. 39, 1095–1100 (1953)
24. Somla, R.: New algorithms for solving simple stochastic games. Electron. Notes Theor. Comput. Sci. 119(1), 51–65 (2005)