Optimization Complexity of Linear Logic Proof Games

Patrick D. Lincoln* John C. Mitchell† Andre Scedrov‡

Abstract

A class of linear logic proof games is developed, each with a numeric score that depends on the number of preferred axioms used in a complete or partial proof tree. The complexity of these games is analyzed for the NP-complete multiplicative fragment (MLL) extended with additive constants and the PSPACE-complete multiplicative, additive fragment (MALL) of propositional linear logic. In each case, it is shown that it is as hard to compute an approximation of the best possible score as it is to determine the optimal strategy. Furthermore, it is shown that no efficient heuristics exist unless there is an unexpected collapse in the complexity hierarchy.

*`lincoln@csl.sri.com` SRI International Computer Science Laboratory, Menlo Park CA 94025 USA. Work supported under NSF Grant CCR-9224858 and ONR Grant N00014-95-C-0168.

†`mitchell@cs.stanford.edu` WWW: `http://theory.stanford.edu/people/jcm/home.html` Department of Computer Science, Stanford University, Stanford, CA 94305-9045 USA. Partially supported by NSF.

‡`scedrov@cis.upenn.edu` WWW: `http://www.cis.upenn.edu/~scedrov` Department of Mathematics, University of Pennsylvania, Philadelphia, PA 19104-6395 USA. Partially supported by NSF Grant CCR-94-00907, by ONR Grant N00014-92-J-1916, and by a Centennial Research Fellowship from the American Mathematical Society.

1 Introduction

Linear logic, introduced in [12], is a refinement of classical logic often described as being *resource sensitive* because of its intrinsic ability to reflect computational states, events, and resources [13, 32, 33, 23]. Several notions of game semantics for linear logic are investigated in [6, 1, 2, 15, 19, 17, 9].

Connections between linear logic and probabilistic games considered in complexity theory are investigated in [24, 27, 25]. In particular, linear logic proof search may also be seen as a game. This game, the *linear logic proof game*, is played on linear logic formulas, and its moves are instances of inference rules of linear logic. There are two players, called proponent and opponent, and a separate verifier. Proponent's goal is to play a sequence of moves that constitute a formal proof of an input formula, consisting of axioms and matching inference rules. Opponent tries to force the direction of proponent's evidence in a way that makes it impossible for proponent to obtain a formal proof. Several versions of this game are discussed in [25, 27], each with a numeric score that reflects the number of certain preferred axioms used in a complete or partial formal proof. The capabilities of the players may differ. While proponent is always omnipotent, in some versions of the game opponent's decisions are based only on a fair coin toss.

Two fragments of propositional linear logic are considered here: the multiplicative additive fragment, MALL, and the multiplicative fragment extended with additive constants, MLL$\top$. MALL is PSPACE-complete [20]. It follows from the NP-completeness of the pure multiplicative fragment, MLL [18, 22], that MLL$\top$ is NP-complete. These are *global hardness* properties in that they provide lower bounds on proponent's optimal strategy.

Games from complexity-theoretic literature [5, 14, 28, 34, 11, 8, 7, 16, 30] may be represented in the linear logic proof game, with the new complexity results obtained as corollaries of the complexity properties of games from the literature just mentioned. A representative case is studied here in detail in Section 7. The reader is referred to [25] for an outline of other cases and for a brief overview of the relevant notions and results from complexity theory. The game representations considered in Section 7 are defined in a move-by-move fashion; that is, they preserve proponent's moves, opponent's moves, proponent's strategies, as well as proponent's optimal strategies (that is, optimal with respect to the score).

In this way, one transfers to the linear logic proof game the complexity lower bounds for the approximation of the expected score when proponent plays optimally. In the case of the PSPACE-complete multiplicative-additive fragment of propositional linear logic [20], it is shown in Section 3 that it is as hard to compute an approximation of the optimal score as it is to determine proponent's optimal strategy.

One way to explain this intuitively and informally is that provability in linear logic is not only globally hard, but also *locally hard*. Indeed, in chess and in many other intricate games choosing the best next move often seems just as hard as developing a complete winning strategy. In other words, these games are locally hard. This property is studied in Section 8 for the linear logic proof game. Let us say that an ϵ-heuristic, where $0 < \epsilon < 1$, is a function from formulas to instances of inference rules (that is, proponent's strategy) such that the optimum score arising from the use of this inference rule instance is close (within multiplicative ratio ϵ) of the optimal score. It is shown that unless P $=$ NP, there is no polynomial-time ϵ-heuristic for MLL$\top$. It is also shown that computing any ϵ-heuristic H for MALL would allow us to decide membership in any language in PSPACE, using time and space at most a polynomial greater than the time and space needed to compute H.

2 Linear logic proof games

Let p be a propositional atom, let A, B be MALL formulas, let $\Gamma, \Delta, \Theta, \Xi$ be finite multisets of MALL formulas, and let Σ be a finite multiset of literals or constants $1, 0$. We write $\Delta \uplus \Theta$ for the (disjoint) multiset union of Δ and Θ. As usual, we write Γ, A for the multiset obtained by adding an instance of A to Γ. An expression of the form $\vdash \Gamma$ is called a *sequent*. An expression of the form $\vdash \Sigma$ is called a *primitive sequent*.

The English names for MALL inference rules are: *identity*, *cut*, *par*, *tensor*, *bottom*, *one*, *plus*, *with*, and *top*. $\otimes$ and $⅋$ are *multiplicative* connectives; 1 and $\perp$ are *multiplicative* propositional constants. $\oplus$ and $\&$ are *additive* connectives; 0 and $\top$ are *additive* propositional constants. There is no rule for 0. *Linear negation* $^\perp$, mentioned in the identity and cut rules, is defined by recursion on the structure of formulas: $(p^\perp)^\perp$ is p, $(A \otimes B)^\perp$ is $A^\perp ⅋ B^\perp$, $(A ⅋ B)^\perp$ is $A^\perp \otimes B^\perp$, $1^\perp$ is $\perp$, $\perp^\perp$ is 1, $(A \& B)^\perp$ is $A^\perp \oplus B^\perp$, $(A \oplus B)^\perp$ is $A^\perp \& B^\perp$, $\top^\perp$ is 0, and $0^\perp$ is $\top$.

MALL proof rules are

$$
\begin{array}{cccc}
\mathbf{I} & \dfrac{}{\vdash p, p^\perp} & \dfrac{\vdash A, \Gamma \qquad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta} & \mathbf{Cut} \\[2ex]
⅋ & \dfrac{\vdash A, B, \Gamma}{\vdash (A ⅋ B), \Gamma} & \dfrac{\vdash A, \Gamma \qquad \vdash B, \Delta}{\vdash (A \otimes B), \Gamma, \Delta} & \otimes \\[2ex]
\perp & \dfrac{\vdash \Gamma}{\vdash \perp, \Gamma} & \dfrac{}{\vdash 1} & 1 \\[2ex]
\oplus\mathbf{1} & \dfrac{\vdash A, \Gamma}{\vdash (A \oplus B), \Gamma} & \dfrac{\vdash A, \Gamma \qquad \vdash B, \Gamma}{\vdash (A \& B), \Gamma} & \& \\[2ex]
\oplus\mathbf{2} & \dfrac{\vdash B, \Gamma}{\vdash (A \oplus B), \Gamma} & \dfrac{}{\vdash \top, \Gamma} & \top
\end{array}
$$

MALL enjoys the *cut-elimination property* and the *subformula property* [12, 20]. In particular, if a MALL formula is provable, then it is provable without the use of the cut rule, and the required proof rules involve only subformulas of the given formula. The fragment MLL$\top$ consists of MALL formulas that do not involve $\&, \oplus$. The inference rules of MLL$\top$ are the rules of MALL except the rules for $\&, \oplus$. The cut-elimination and subformula properties again hold for MLL$\top$.

Let us describe several variations of the proof game discussed in [24, 27, 25], all involving the same moves. There are two players, called proponent and opponent, and a separate, *polynomial-time* verifier. Proponent's goal is to play a number of moves demonstrating or giving evidence for a sequent. In order to do this, proponent plays proof rule instances. Opponent tries to force the direction of proponent's evidence in a way that makes it impossible for proponent to win. Opponent plays special markers that may block one side of proponent's $\&$ moves. If proponent plays a $\otimes$ move, then opponent does not block either of the premises. Note that opponent is absent in the case of MLL$\top$, that is, the game on MLL$\top$ sequents is a kind of solitaire game.

Polynomial-time verifier scores completed plays of the game. Various forms of the game differ in the way they are scored. The main objective of proponent is to never allow opponent to succeed in forcing an unprovable primitive sequent. However, in some forms of the game proponent will be more ambitious, that is, in addition to the main requirement, proponent will try to achieve the best score possible.

Let us first consider a simple version of the game against a randomized opponent, which can be described as an avg/max game played on MALL sequents. The game may also be presented as a board game with tiles, where each tile is marked by a linear logic inference rule [24, 26]. Proponent chooses the inference rule to be applied. In the case $\otimes$, proponent chooses a partition and requires both associated expressions to be evaluated. In the case $\oplus$, proponent chooses which of the two expressions will be evaluated. In the case $\&$, opponent chooses by a fair coin toss which of the two expressions will be evaluated. In the case of a primitive sequent, verifier simply computes the value. Each sequent containing the constant $\top$, each identity axiom, and each primitive sequent containing only the constant 1 is scored 1 by verifier. All other primitive sequents are scored 0. Each completed play of the game is scored as the *minimum* of the scores of terminal sequents obtained in the play. Note that the number of moves is finite; indeed, it is polynomial in the size of a given MALL sequent. Proponent wins when each encountered primitive sequent is an identity axiom or the constant 1.

Let us define the function μ, which represents the expected score when proponent plays optimally.

$$\mu(\Gamma) = max\{\mu(\Gamma'; A) \mid \Gamma = \Gamma', A\},$$

$$\mu(\Gamma; A \otimes B) = max\{min\{\mu(\Delta, A), \mu(\Theta, B)\} \mid \Delta \uplus \Theta = \Gamma\},$$

$$\mu(\Gamma; A \parr B) = \mu(\Gamma, A, B),$$

$$\mu(\Gamma; A \oplus B) = max\{\mu(\Gamma, A), \mu(\Gamma, B)\},$$

$$\mu(\Gamma; A\&B) = \frac{1}{2}[\mu(\Gamma, A) + \mu(\Gamma, B)],$$

$$\mu(\Gamma; \bot) = \mu(\Gamma),$$

$$\mu(\Gamma; \top) = 1,$$

$$\mu(\Sigma) = \begin{cases} 1 & \text{if } \Sigma \text{ is 1 or an axiom,} \\ 0 & \text{otherwise.} \end{cases}$$

Let us emphasize that, for any MALL sequent $\vdash \Xi$, the value $\mu(\Xi)$ is the maximum possible value satisfying these recursive conditions. Specifically, if any encountered sequent contains composite formulas, then several clauses regarding $\mu(\Gamma; A)$ might be applicable. The following proposition is proved by induction on the number of symbols in Ξ.

Proposition 2.1 *A play of the simple linear logic proof game is won by proponent iff the score of the play is equal to 1. Furthermore, a* MALL *sequent $\vdash \Xi$ is provable iff $\mu(\Xi) = 1$. In addition, if $\vdash \Xi$ is unprovable and does not contain $\&$, then $\mu(\Xi) = 0$.*

However, note that $\mu(\Xi)$ may be arbitrarily close to 1 if $\vdash \Xi$ is unprovable and contains $\&$.

The more involved, *weighted* version of the linear logic proof game against a randomized opponent may also be presented as an avg/max game. The players' moves and the winning condition are the same as in the simple game just described. However, in this version of the game, proponent also attempts to use as many certain preferred axioms as possible. Preferred axioms are, say, instances of a distinguished axiom $\vdash d, d^{\perp}$, where the propositional atom d is fixed in advance. In this version, proponent gets one point for each instance of the distinguished axiom $\vdash d, d^{\perp}$ encountered in a play, but no points are awarded if a primitive sequent $\vdash \Sigma$ is any other identity axiom

$\vdash p, p^{\perp}$, or contains $\top$, or consists of a single constant $\vdash 1$. If Σ is any other multiset of literals or constants, that is, a possibly empty multiset of literals or constants $1, 0$ other than an identity axiom or the single constant 1, then proponent receives a penalty of -2^n points, where n is length of the original sequent. Each completed play of the game is now scored as the *sum* of the scores of primitive sequents obtained at the end of the play. Note that proponent wins iff this sum is ≥ 0.

Let us define the function ρ, which represents the expected score for proponent when proponent plays optimally.

$$\rho(\Gamma) = \rho_n(\Gamma), \ \ where \ n = |\Gamma|,$$

$$\rho_n(\Gamma) = max\{\rho_n(\Gamma'; A) \mid \Gamma = \Gamma', A\},$$

$$\rho_n(\Gamma; A \otimes B) = max\{\rho_n(\Delta, A) + \rho_n(\Theta, B) \mid \Delta \uplus \Theta = \Gamma\},$$

$$\rho_n(\Gamma; A \parr B) = \rho_n(\Gamma, A, B),$$

$$\rho_n(\Gamma; A \oplus B) = max\{\rho_n(\Gamma, A), \rho_n(\Gamma, B)\},$$

$$\rho_n(\Gamma; A \& B) = \frac{1}{2}[\rho_n(\Gamma, A) + \rho_n(\Gamma, B)],$$

$$\rho_n(\Gamma; \perp) = \rho_n(\Gamma),$$

$$\rho_n(\Gamma; \top) = 0,$$

$$\rho_n(\Sigma) = \begin{cases} 1 & \text{if } \Sigma \text{ is the distinguished axiom } d, d^{\perp}, \\ 0 & \text{if } \Sigma \text{ is another axiom or } 1, \\ -2^n & \text{otherwise.} \end{cases}$$

Observe again that, for any MALL sequent $\vdash \Xi$, the value $\rho(\Xi)$ is the maximum possible value satisfying these recursive conditions, now with n fixed as the length of Ξ, that is, the number of symbols in Ξ.

Proposition 2.2 *Let $\vdash \Xi$ be a* MALL *sequent $\vdash \Xi$ and let d be a distinguished propositional atom. Let n be an integer no smaller than the length of Ξ, let k be the maximal &-depth of Ξ, and let j be the number of positive occurrences of d in Ξ. A play of the weighted linear logic proof game is won by proponent iff the score of the play is ≥ 0. Furthermore, if Ξ is provable, then $0 \leq \rho_n(\Xi) \leq j$. If Ξ is unprovable, then $\rho_n(\Xi) \leq -2^{n-k} + j \leq -1$. In particular, if Ξ is unprovable and does not contain any occurrences of d, then $\rho_n(\Xi) \leq -2^{n-k} \leq -2$.*

Proof If Ξ is provable, then proponent can play as in a cut-free proof of Ξ and so achieve a non-negative score. But then for an optimal strategy certainly $\rho(\Xi) \geq 0$. Furthermore, $n \geq 3k + j$ for any sequent Ξ because since each & comes with a pair of parentheses. Thus $-2^{n-k} + j \leq -1$ and if $j = 0$, then $-2^{n-k} \leq -2$. The other two upper bounds may be established by simultaneous induction on the lentgh of Ξ. ∎

Game score functions considered here are intrinsic to the proof system MALL. In particular, they are invariant with respect to certain *permutability* properties, important "structural" properties of MALL. That is, our game score functions are invariant with respect to invertible inference rules of MALL. The following theorem is proved by induction on the length of Γ.

Theorem 2.3 *The following equalities hold, with n the length of the longest sequent in each equality:*

- $\rho_n(\Gamma, A \parr B) = \rho_n(\Gamma, A, B)$,
- $\rho_n(\Gamma, A \& B) = \frac{1}{2}(\rho_n(\Gamma, A) + \rho_n(\Gamma, B))$,
- *If Γ does not contain $\&$, then* $\rho_n(\Gamma, A \oplus B) = max\{\rho_n(\Gamma, A), \rho_n(\Gamma, B)\}$,

and similarly for μ.

Let us also observe that many isomorphisms of linear logic are respected by the score functions we consider, *e.g.*, $A \parr (B \parr C) \cong (A \parr B) \parr C$, $A \parr (B \& C) \cong (A \parr B) \& (A \parr C)$, $A \parr \perp \cong A$, $A \parr B \cong B \parr A$, $A \otimes B \cong B \otimes A$, $A \otimes 1 \cong A$, $A \otimes (B \otimes C) \cong (A \otimes B) \otimes C$, etc. Notable exceptions are $A \& \top \cong A$ and $A \& (B \& C) \cong (A \& B) \& C$.

3 Lower bounds for optimal strategies

The NP-hardness of MLL$\top$ and Propositions 2.1 and 2.2 imply that, in either version of the proof game, the optimal score functions μ, ρ and the corresponding optimal strategies for proponent are NP-hard to compute on MLL$\top$ sequents (in fact, already on MLL sequents.) Furthermore, because the values of μ, ρ are discrete, it is just as hard to approximate them as it is to compute them exactly. For the weighted version of the proof game, this will turn out to be the case even on provable formulas, and even if they contain $\&$.

The formal definition follows. One usually approximates a function f by computing a function in some "neighborhood" of f. Let $\mathcal{D} \subseteq \{0,1\}^*$, and let $\mathcal{R}$ be the set of real numbers. For a function $f: \mathcal{D} \to \mathcal{R}$, the neighborhood of f consists of all functions $g: \mathcal{D} \to \mathcal{R}$ such that, for every string $x \in \mathcal{D}$, the difference between $f(x)$ and $g(x)$ is relatively small. One measure of error that appears in the literature [10, 4, 3, 8, 7] is

Definition 3.1 *Let $\mathcal{D} \subseteq \{0,1\}^*$, and let f be a real-valued function on $\mathcal{D}$. Let $0 < \epsilon < 1$, where ϵ may depend on $|x|$, the length of a string $x \in \mathcal{D}$. The ϵ-neighborhood of f is the set*

$$\epsilon\text{-}nbhd(f) = \left\{ g: \mathcal{D} \to \mathcal{R} \mid \forall x \in \mathcal{D}, \quad \epsilon(|x|) \leq \frac{g(x)}{f(x)} \leq \frac{1}{\epsilon(|x|)} \right\}$$

For any $g \in \epsilon$-$nbhd(f)$ one says that g approximates f within multiplicative ratio ϵ.

Intuitively, this is approximation up to factor $1/\epsilon$. Also this is basically approximation with relative error at most $1 - \epsilon$.

Theorem 3.2 *Let $0 < \epsilon < 1$. Let ρ be the optimal score function for the weighted linear logic proof game. If P$\neq$NP, then there is no polynomial-time computable rational-valued function that approximates the function ρ on provable MLL$\top$ sequents within multiplicative ratio ϵ.*

This property extends to provable MALL formulas, where it is appropriate to require approximations to be PSPACE-hard. More precisely[1],

Definition 3.3 *Let $\mathcal{D} \subseteq \{0,1\}^*$, and let g be a rational-valued function on $\mathcal{D}$. The function g is* PSPACE*-hard if, for any language $L \in$ PSPACE and its associated characteristic function χ_L, there exist polynomial-time computable functions h_1, h_2 such that $\chi_L = h_2 \circ g \circ h_1$.*

Definition 3.4 *Let $\mathcal{D} \subseteq \{0,1\}^*$, and let f be a real-valued function on $\mathcal{D}$. Let ϵ be a real-valued function on the natural numbers such that $\epsilon(n) < 1$, for all n. The function f is* PSPACE*-hard to approximate within multiplicative ratio ϵ if every rational-valued function $g \in \epsilon$-nbhd(f) is* PSPACE*-hard.*

Theorem 3.5 *Let ρ be the optimal score function for the weighted linear logic proof game. The function ρ on provable* MALL *sequents is* PSPACE*-hard to approximate within any multiplicative ratio.*

That is, computing any function g in any ϵ-neighborhood of ρ would allow us to decide membership in any language L in PSPACE, using time and space at most a polynomial greater than the time and space needed to compute g.

In order to obtain Theorems 3.2 and 3.5 it suffices to show the following lemma.

Lemma 3.6 *Let ρ be the optimal score function for the weighted linear logic proof game. Let A be a* MALL *formula that does not contain a distinguished propositional atom d. Let $A^{\#}$ be the formula*

$$((A \otimes (d ⅋ d^{\perp})) \otimes \perp) ⅋ \top.$$

Then $\rho(A^{\#}) = 1$ if A is provable, else 0.

Proof Let n be the length of $A^{\#}$. In any case, the first move must be ⅋ and $\rho_n(A) \leq 0$ by Proposition 2.2. $\rho(A^{\#}) \geq 0$ because proponent can always achieve a score of 0 by playing the axiom $\top$ as the second move. However, if A is provable, proponent can achieve a score of 1 by instead playing $\otimes$, where $\perp$ is paired with $\top$, and then $\otimes$, placing $d ⅋ d^{\perp}$ by itself on one side, and on the other side continuing as in a proof of A. $\rho_n(A) = 0$, $\rho_n(d ⅋ d^{\perp}) = 1$, and $\rho_n(\perp, \top) = 0$. The strategy just described is optimal. Indeed, placing $\perp$ by itself in the first $\otimes$ move would yield a negative overall score: on the side of $\perp$ one has -2^n, while on the other side $\rho_n((A \otimes (d ⅋ d^{\perp})), \top) = 1$ by playing $\otimes$ and placing $d ⅋ d^{\perp}$ by itself on one side, and on the other side immediately playing the axiom $\top$. (Note that this does not assume that A is provable.) We have already observed that playing the axiom $\top$ instead of the first $\otimes$ would yield the score of 0.

If A is not provable, then playing the axiom $\top$ in the second move is optimal for proponent. Indeed, suppose proponent plays $\otimes$ instead. If in that move $\perp$ is placed by itself, then, as we have seen in the previous paragraph, the overall score would be negative. If $\perp$ is placed with $\top$, which yields 0 on that side, then the overall score is also negative because on the other side $\rho_n(A) \leq -2$ by Proposition 2.2, hence $\rho(A \otimes (d ⅋ d^{\perp})) \leq -1$. ■

While a hardness of approximation property analogous to Theorem 3.5 does hold for the simple version of the linear logic proof game as well, we do not know whether it is a direct consequence of the PSPACE-hardness of MALL. Instead, we shall rely on recent results in complexity theory [5, 14, 28, 34, 11, 7].

[1]See [25] for a further discussion of the relevant background in complexity theory.

Theorem 3.7 *Let μ be the optimal score function for the simple linear logic proof game. There exists a positive real number c such that the function μ on* MALL *sequents is* PSPACE*-hard to approximate within the multiplicative ratio 2^{-n^c}, where n is the length of a sequent.*

In other words, computing any function g that, for every MALL formula F, satisfies

$$2^{-|F|^c} \leq \frac{g(F)}{\mu(F)} \leq 2^{|F|^c}$$

would amount to the ability to recognize every PSPACE language. As an illustration of the meaning of the condition, note that even a very large constant g fails to satisfy the condition, since the values of μ can be very close to zero. On the other hand, the theorem implies that the function μ is PSPACE-hard to approximate up to any constant factor.

The proof of Theorem 3.7 will occupy most of Sections 5 and 7. But first we present in Section 4 a direct encoding of a combinatorial optimization problem in the weighted version of the linear logic proof game, which will yield another proof of Theorem 3.2.

4 Constrained matching problem as a linear logic proof game

In this section we concentrate on the *multiplicative fragment* MLL *extended with additive constants* $\top, 0$. This fragment consists of formulas built from literals and constants $\perp$, 1, 0, and $\top$ by using multiplicative connectives $\otimes, \invamp$, and the fragment has the corresponding rules of inference. Let us call this fragment MLL$\top$. It is easy to see that provability in MLL$\top$ is in NP. It follows from the work of Kanovich [18] and Lincoln and Winkler [22] on the NP-completeness of MLL that MLL$\top$ is NP-complete.

When the input sequent of a linear logic proof game is an MLL$\top$ sequent, the game reduces to a kind of solitaire game because the opponent is absent. However, even this special case has interesting features. We show that an optimization problem known to be NP-hard to approximate can be represented in polynomial time as the weighted linear logic proof game described in Section 2, where the input is an MLL$\top$ sequent. Thus we obtain another proof of Theorem 3.2.

4.1 The Problem

Informally, one can think of the matching problem as the problem faced by the host of a dinner party. Given a set of people P, a set of tables each of which has N place settings (in "three dimensional matching" $N = 3$), and a list of topics τ (each described by a set of N people who get along and a topic of interest that they have in common), the gracious host must find an arrangement where every person is sitting at a table with others who get along sharing some topic of interest. The constrained-maximum version adds the constraint that there are some 'good' topics G the host wants to encourage. The problem then becomes one of finding a solution where the maximum number of good topics are used.

Formally, consider

CONSTRAINED MAXIMUM 3 DIMENSIONAL MATCHING (CM3DM)

INSTANCE: A hypergraph (P, τ), $\tau \subseteq P \times P \times P$, a $G \subseteq \tau$, and a solution 3-dimensional matching τ', that is, $\tau' \subseteq \tau$, $|\tau'| = |P|/3$ and no two elements of τ' agree in any coordinate.

SOLUTION: A 3-dimensional matching τ', that is, $\tau' \subseteq \tau$, $|\tau'| = |P|/3$ and no two elements of τ' agree in any coordinate.

MEASURE: $|\tau' \bigcap G|$

HARDNESS: For some constant $\epsilon > 0$, maximal measure is NP-hard to approximate within multiplicative ratio $|J|^{-\epsilon}$, where $|J|$ is the size of problem instance J [36].

Note that a problem instance J supplies an example solution. One might assume that this would be a rather poor solution. The existence of the one solution simplifies the statement of the measure of all problem instances, which otherwise would need a special case for unsolvable instances. We also use this fact to simplify our proof of correctness of our encoding.

Note that a hardness theorem in [36] says that even estimating the number of good topics that can be used is difficult. Unless P=NP, this number cannot be estimated with any reasonable degree of accuracy in deterministic polynomial time.

4.2 The Encoding

Let us describe the encoding of CM3DM into MLL$\top$. Each person $p_i \in P$ will be represented by a unique negated propositional atom $p_i^\perp$. Let n be the number of people and m be the number of topics. Let $\star$ abbreviate the formula $(d \parr d^\perp)$ for a distinguished propositional atom d.

$$\begin{array}{rrcl}
p_i \in P & [p_i] & = & p_i^\perp \\
 & [P] & = & p_1^\perp \parr p_2^\perp \parr \ldots \parr p_n^\perp \\
\tau_x = (p_i, p_j, p_k) \in \tau - G & [\tau_x] & = & (p_i \otimes (p_j \otimes p_k)) \\
\tau_x = (p_i, p_j, p_k) \in \tau \bigcap G & [\tau_x] & = & (p_i \otimes (p_j \otimes (p_k \otimes \star))) \\
 & [\tau] & = & [\tau_1] \parr [\tau_2] \parr \ldots \parr [\tau_m] \\
 & [J] & = & ([P] \otimes \top) \parr [\tau]
\end{array}$$

In other words, the encoding $[P]$ of the set of people is the $\parr$ of the encodings of each person in P. The encoding $[\tau]$ of the set of topics is the $\parr$ of the encodings of each element of τ. The overall encoding $[J]$ of a CM3DM problem instance J is the formula $([P] \otimes \top) \parr [\tau]$. Note that the encoding $[J]$ is computable in polynomial time in $|J|$, the size of a problem instance J.

Note that the game score function ρ distinguishes $(p \otimes (q \otimes \star)) \parr \top$ from $((p \otimes q) \otimes \star) \parr \top$. On the former the optimal score is 0 while on the latter the optimal score is 1. Also, although one might expect it, it is not the case that if $A \multimap B$ and $B \multimap A$ are provable and $\rho(A) = \rho(B)$, then for any context C $\rho(C(A)) = \rho(C(B))$. Consider $(p \otimes (q \otimes \star))$ and $((p \otimes q) \otimes \star)$ which satisfy the conditions of equality, but the counterexample just mentioned shows that in some contexts they behave differently under ρ. However, many isomorphisms of linear logic are respected by the score functions we consider, see Section 2.

4.3 Correctness

A sequent is said to be *balanced* if the number of occurrences of propositional atoms p_i with positive polarity and negative polarity are equal. Otherwise, we say a sequent is *unbalanced*. All provable MLL sequents are balanced, and therefore all unbalanced sequents are not provable. This property was previously discussed, *e.g.*, in [18, 21].

Proposition 4.1 *An* MLL *sequent is provable only if it is balanced.*

Proof By induction on the depth of assumed cut-free proof. ∎

Note that this property fails for other fragments of linear logic such as MALL which include the additive connectives and constants: $\&$, $\oplus$, $\top$, and 0, and also fails to hold in the presence of exponential connectives ! and ?. In particular, the property fails for the fragment in question here, MLL$\top$.

Recall the weighted version of the linear logic proof game and its optimal score ρ discussed in Section 2.

Theorem 4.2 *The solutions to a constrained maximum 3-dimensional matching problem instance J induce the winning plays of the weighted linear logic proof game beginning with $[J]$ so that a maximum solution induces an optimal winning play. In particular, the measure of a maximum solution is equal to $\rho([J])$.*

It will be clear from the proof that the multiplicative constants 1 and $\perp$ and the additive constant 0 are not needed for the argument.

Proof In one direction, assume a solution τ' to the CM3DM instance J. Beginning with the formula $([P] \otimes \top) ⅋ [\tau]$, the proponent first plays the ⅋ moves and then uses τ' to form the partition required in the $\otimes$ move on $[P] \otimes \top$. Writing $\tau'' = \tau - \tau'$ and using $[\tau]$ ambiguously for the formulas $[\tau_1], \cdots, [\tau_n]$ separated by commas or by ⅋s, the opening moves can be presented as

$$\dfrac{\dfrac{\dfrac{\vdots}{\vdash [P], [\tau']} \qquad \dfrac{}{\vdash \top, [\tau'']}\top}{\dfrac{\vdash [P] \otimes \top, [\tau]}{\vdots}}\otimes \; ⅋}{\vdash ([P] \otimes \top) ⅋ [\tau]}⅋$$

This play can be completed by analyzing each tensor in $[\tau']$ in any order. This results in a play that never encounters an unprovable sequent. The score on $\vdash [P], [\tau']$is exactly the number of uses of the distinguished axiom $\vdash d, d^{\perp}$ on that branch, which is equal to the number of $\star$s in the conclusion sequent of that branch. The number of $\star$s in the sequent is the same as the number of good topics included in the assumed solution to CM3DM τ'. The overall score on $\vdash ([P] \otimes \top) ⅋ [\tau]$ is the same, since the right hand branch above scores zero. Thus the maximum solution to the CM3DM instance J does not exceed $\rho([J])$.

In the other direction, since a solution is provided as a part of a CM3DM instance, by the construction above one can assume without loss of generality that $\rho([J]) \geq 0$. Thus we need not consider any plays of the proof game that end in an unprovable sequent in any branch, since all such plays lead to an overall negative score.

Given an encoding of a CM3DM problem, $\rho(([P] \otimes \top) ⅋ [\tau])$ is equal to $\rho(([P] \otimes \top), [\tau_1], \cdots, [\tau_n])$ by Theorem 2.3. In other words, an optimal play might as well begin with the ⅋ moves. There are then two possible moves that could come next: to split the $\otimes$ involving $\top$ or to split a $\otimes$ in some $[\tau_i]$. In the latter case, where the $\otimes$ in $[\tau_i]$ is analyzed first, say τ_i is a good topic, $\tau_i \in G$ (the other case is similar.) Let $\tau = \tau'' \bigcup \tau''' \bigcup \{\tau_i\}$ be the required partition. This game position can be presented as

$$\dfrac{\dfrac{\vdots}{\vdash [P] \otimes \top, (p_j \otimes (p_k \otimes \star)), [\tau'']} \qquad \dfrac{?}{\vdash p_i, [\tau''']}}{\vdash [P] \otimes \top, (p_i \otimes (p_j \otimes (p_k \otimes \star))), [\tau_2], \cdots, [\tau_n]}\otimes$$

But now the proponent cannot win. The problematic right-hand branch (the one without the formula $[P] \otimes \top$) is an unbalanced MLL sequent and hence unprovable, and thus the score would be negative. Therefore, the $\otimes$ moves involving the $[\tau_i]$'s should come only after the $\otimes$ move involving the formula $[P] \otimes \top$.

In that case, the game position can be presented as

$$\dfrac{\dfrac{\vdots}{\vdash [P], [\tau']} \qquad \overline{\vdash [\tau''], \top}^{\,\top}}{\vdash [P] \otimes \top, [\tau]}\otimes$$

where $[\tau'], [\tau'']$ is a partition of $[\tau]$. The formula $[P]$ appears on the left-hand branch. On that branch, since the sequent $\vdash [P], [\tau']$ is a pure MLL sequent, some subset of $[\tau]$ must appear if the sequent is to be balanced. The subset of $[\tau]$ must be chosen so that each proposition p_i (representing a person) appears exactly once (in order to balance the single occurrence of each $p_i^\perp$ in $[P]$). This leaves the formula $\top$ and the (possibly empty) remaining subset of $[\tau]$ on the right-hand branch. This branch can be won immediately with the use of the $\top$ move, resulting in the zero score. Any other attempted move on the right-hand branch leads to defeat: analyzing any $[\tau_i]$ results in two branches, neither of which is balanced. There is only one occurrence of $\top$ that can be used to complete the play in one branch, so the other branch must end in defeat. Thus the branch containing $\vdash [\tau''], \top$ will obtain a score of zero. Returning attention to the main branch, this branch is concerned with a pure MLL sequent, and thus each formula is analyzed exactly once. Thus the score is equal to the number of $\star$s appearing as subformulas in the conclusion sequent of this branch. From any such play one can obtain a solution to the given CM3DM instance by considering all the formulas of $[\tau']$ that appear in the main branch as defining a subset of topics τ'. This play represents a correct solution to the given CM3DM instance because all propositions must appear exactly once in that main branch of the proof. The score achieved will be the same as the number of good topics used in the CM3DM solution derived from this winning play. Thus the maximum solution to the CM3DM instance J is not exceeded by $\rho([J])$. ■

Because the encoding is polynomial, we obtain another proof of Theorem 3.2.

We do not know if there are natural linear logic invariants that are NP-hard to approximate on provable sequents of pure MLL, where additive constants 0 and $\top$ are not allowed.

5 Stochastic quantified boolean formulas and games

We consider boolean matrices in conjunctive normal form. Prenex boolean formulas are defined as usual, but we allow the "random" quantifier ¶ in addition to $\exists$ and $\forall$. A formula is a $k-CNF$ if every clause in its matrix has exactly k literals. For the purposes of establishing a succinct terminology, we say a formula is *classical* if it is closed and all of the quantifiers are $\forall$ or $\exists$, *existential* if it is closed and all of the quantifiers are $\exists$, and *stochastic* if it is closed and all of the quantifiers are ¶ or $\exists$. It is possible to consider formulas that contain other combinations of quantifiers (such as $\forall$ and ¶), but we will not need these other classes of formulas.

A classical formula is either valid or invalid, according to the usual interpretation. One way of understanding the value of a classical formula that will be useful in comparing classical and stochastic formulas is through a very simple game with two players called "$\forall$" and "$\exists$". For a formula $Q_0x_0Q_1x_1\ldots Q_nx_nM$, the play follows the quantifier order $Q_0, Q_1, \ldots, Q_n$ from left to right, with each player selecting a truth value (*true* or *false,* or, equivalently, 1 or 0) for each variable identified with that player. Informally, the goal of player "$\exists$" is to choose values for the existentially-quantified variables so that the matrix is true. Player "$\forall$" tries to do the opposite, choosing values for the universally-quantified variables that will make the formula false.

It is easy to see that, for any classical formula, if both players continue until the quantifier prefix is exhausted, player "$\exists$" has the ability to win against any possible "$\forall$" opponent precisely if the formula is valid. When player "$\exists$" has a way of winning, regardless of how player "$\forall$" plays, we say player "$\exists$" has a winning strategy, and similarly for player "$\forall$." More formally, a *strategy* is a function from positions (which may be represented by the sequence of moves made so far in the game) to moves. A strategy is a *winning strategy* for a given player if this player is guaranteed a win by following the strategy. Using this standard terminology from game theory, we say that a classical formula is valid iff player "$\exists$" has a winning strategy. We can think of the player "$\exists$" as "proponent" and the player "$\forall$" as "opponent."

For stochastic formulas, we associate a probability PROB-STOC(ϕ) with each formula ϕ. One way of explaining this probability is using a variant of the classical formula game described above, this time between players called "¶" (opponent) and "$\exists$" (proponent). The game is played in essentially the same way as the classical game, except for the way that player "¶"chooses the values of variables. Specifically, the play follows the quantifier order from left to right, with each player selecting a truth value for each variable with the appropriate quantifier. The value of a formula is computed by associating a specific strategy with each player. The simpler of the two is player "¶", who chooses truth values at random. That is, the player "¶" assigns independently to each variable 1 or 0, each with probability 1/2. Player "$\exists$" chooses truth values so as to maximize the probability that the formula is satisfied. In other words

$$\text{PROB-STOC}(\phi) \stackrel{def}{=} \text{Probability that } \phi \text{ is satisfied if "}\exists\text{" plays optimally.}$$

This informal description may be made more precise by defining PROB-STOC(ϕ) by recursion on the length of the quantifier sequence. Specifically, suppose $Q_{i+1}x_{i+1}\ldots Q_nx_nM$ is a formula with ¶ and $\exists$ quantifiers, and free boolean variables $x_0, \ldots, x_i$. The value $\mathcal{V}(b_0, \ldots, b_i \mid Q_{i+1}x_{i+1}\ldots Q_nx_nM)$ of formula $Q_{i+1}x_{i+1}\ldots Q_nx_nM$ given truth values $b_0, \ldots, b_i$ is defined inductively as follows:

$$\begin{array}{lcl}
\mathcal{V}(b_0, \ldots, b_n \mid M) & = & \begin{cases} 1 & \text{if } M \text{ is satisfied by } b_0, \ldots, b_n \\ 0 & \text{if } M \text{ is not satisfied by } b_0, \ldots, b_n \end{cases} \\
\mathcal{V}(b_0, \ldots, b_i \mid \P x_{i+1}\ldots Q_nx_nM) & = & 1/2(\mathcal{V}(b_0, \ldots, b_i, 0 \mid Q_{i+2}x_{i+2}\ldots Q_nx_nM) \\
 & & \quad +\mathcal{V}(b_0, \ldots, b_i, 1 \mid Q_{i+2}x_{i+2}\ldots Q_nx_nM)) \\
\mathcal{V}(b_0, \ldots, b_i \mid \exists x_{i+1}\ldots Q_nx_nM) & = & max\{\mathcal{V}(b_0, \ldots, b_i, 0 \mid Q_{i+2}x_{i+2}\ldots Q_nx_nM), \\
 & & \quad \mathcal{V}(b_0, \ldots, b_i, 1 \mid Q_{i+2}x_{i+2}\ldots Q_nx_nM)\}
\end{array}$$

Another view of the value of a stochastic formula game is that this is the probability that player "$\exists$" will win a game played on the classical formula obtained by replacing each ¶ by $\forall$, when $\forall$ follows a random strategy and $\exists$ plays optimally.

In [7] it is observed that

Theorem 5.1 ([7]) *There exists a positive constant c such that* PROB-STOC *on stochastic formulas is* PSPACE*-hard to approximate within multiplicative ratio 2^{-n^c}, where n is the length of a stochastic formula.*

Let us also describe another, related game on stochastic formulas, where a useful intuition is to think of a matrix $M = C_1 \wedge \ldots \wedge C_j$ with j clauses as stating a multiset of j conditions to be satisfied simultaneously. While it would be best to satisfy all conditions, this may not be possible. In this case, one would like to know how close one can come to this goal. This time, player "$\exists$" (*i.e.*, proponent) tries to satisfy as many clauses as possible, against an opponent that plays randomly. The number function MAX-STOC gives the expected number of clauses that "$\exists$" will be able to satisfy (when playing optimally), *i.e.*,

$$\text{MAX-STOC}(\phi) \stackrel{def}{=} \sum_k k\, Prob(\text{exactly } k \text{ clauses of } M \text{ are satisfied, when "}\exists\text{" plays optimally}).$$

This has a precise recursive definition as above, except that the base case is replaced by:

$$\mathcal{V}(b_0, \ldots, b_n \mid M) \quad = \quad \text{the number of clauses of } M \text{ satisfied by } b_0, \ldots, b_n.$$

Intuitively, MAX-STOC is the expected score for "$\exists$" when "$\exists$" plays optimally against a random adversary. Note that this use of boolean formulas does *not* respect logical equivalence: It is easy to find logically equivalent purely existential formulas ϕ and ϕ' with $\text{MAX-STOC}(\phi) \neq \text{MAX-STOC}(\phi')$.

The following theorem is proved in [7].

Theorem 5.2 ([7]) *There exists a positive real constant $\epsilon < 1$ such that the function* MAX-STOC *on 3-CNF stochastic formulas is* PSPACE*-hard to approximate within multiplicative ratio ϵ.*

The games just described and their associated optimal score functions readily extend to a more general setting where the conditions to be satisfied simultaneously are boolean matrices rather than disjunctive clauses. That is, one counts the number of simultaneously satisfiable boolean matrices from a given multiset $\{M_1, \ldots, M_j\}$. The game is played on *generalized prenex formulas* $Qx_0Qx_1 \ldots Qx_n\{M_1, \ldots, M_j\}$. The terminology regarding formulas is also used for generalized formulas. Also, for instance, the base case of the two previous recursive definitions is replaced by

$$\mathcal{V}(b_0, \ldots, b_n \mid \{M_1, \ldots, M_j\}) \quad = \quad \text{the number of matrices in } \{M_1, \ldots, M_j\} \text{ satisfied by } b_0, \ldots, b_n.$$

The game played on formulas is a special case when the boolean matrices $M_1, \ldots, M_j$ are disjunctive clauses.

6 Game repetition

Any of the games discussed in Section 5 can be played on several boards at once, similarly to a simultaneous chess tournament. We shall see that playing the simultaneous PROB-STOC game in this way is closely related to playing one MAX-STOC game on one board. This relationship yields another approximation bound for the MAX-STOC function.

Theorem 6.1 *There exists a positive real number c such that the function* MAX-STOC *on generalized stochastic formulas is* PSPACE*-hard to approximate within multiplicative ratio n^{-c}, where n is the length of the generalized stochastic formula.*

This appears to be a new observation, but it follows an analogous result in [16] regarding a counting game on generalized classical formulas. Note the trade-off with Theorem 5.2: Theorem 6.1 obtains a better bound but on a much larger class.

The proof of Theorem 5.1 in [7] uses the methods of [11] to obtain a class of stochastic formulas ϕ for which PROB-STOC(ϕ) is either equal to 1, or it is at most 2^{-n^d}, where n is the length of ϕ, such that the decision between the two options is PSPACE-hard. Our proof of Theorem 6.1 will rely on the same class of stochastic formulas.

Using the same technique as the one used in [16] for classical formulas, we will copy the matrix of the stochastic formula ϕ k times, and derive that the function MAX-STOC on the generalized stochastic formula obtained in this way is either k or $k \times 2^{-n^d}$ otherwise.

Let ϕ be a stochastic formula $Q_1 x_1 \ldots Q_m x_m M$. For any fixed k and any $i = 1, \ldots, k$ let M_i be the result of replacing the veriable x_j with the fresh variable x_j^i. Consider the generalized stochastic formula ψ defined as

$$Q_1 x_1^1 \ldots Q_m x_m^1 \ldots Q_1 x_1^k \ldots Q_m x_m^k \{M_1, \ldots, M_k\}$$

Let us assume that ϕ encodes the special version of IP mentioned above. If PROB-STOC(ϕ) $= 1$, then the player "$\exists$" in the PROB-STOC game has an outright winning strategy. Thus in k copies of the game the player "$\exists$" results in a guaranteed score of k along every branch of the game obviously resulting in an overall score of k. The same is true for the MAX-STOC game on ψ. On the other hand, if PROB-STOC(ϕ) < 1, then necessarily PROB-STOC(ϕ) $< 2^{-n^d}$, *i.e.*there is an an upper bound of 2^{-n^d} on any strategy of the player "$\exists$" in the PROB-STOC game. The lemma below implies that k simultaneous games of this kind, as well as one MAX-STOC game on ψ, have an overall expected score of at most $k \times 2^{-n^d}$.

Lemma 6.2 *Let $Q_1 x_1 \ldots Q_n x_n \{M_1, \ldots, M_k\}$ be any generalized stochastic formula. Then*

$$\textsc{max-stoc}(Q_1 x_1 \ldots Q_n x_n \{M_1, \ldots, M_k\}) \leq \sum_{i=1}^{k} \textsc{prob-stoc}(Q_1 x_1 \ldots Q_n x_n M_i)$$

Proof First, we consider the binary case:

$$\textsc{max-stoc}(Q\colon M_1, M_2) \leq \textsc{prob-stoc}(Q\colon M_1) + \textsc{prob-stoc}(Q\colon M_2).$$

By induction on the length of quantifier prefix. Base case: no quantifiers: $\textsc{max-stoc}(M_1, M_2) = \textsc{max-stoc}(M_1) + \textsc{max-stoc}(M_2) = \textsc{prob-stoc}(M_1) + \textsc{prob-stoc}(M_2)$ That is, the game boards are evaluated in the same way for MAX-STOC and for PROB-STOC (zero or one in our case).

Inductive step: Assume for all M_1, M_2,

$$\textsc{max-stoc}(Q\colon M_1, M_2) \leq \textsc{prob-stoc}(Q\colon M_1) + \textsc{prob-stoc}(Q\colon M_2)$$

and show that $\textsc{max-stoc}(\P x, Q\colon M_1, M_2) \leq \textsc{prob-stoc}(\P x, Q\colon M_1) + \textsc{prob-stoc}(\P x, Q\colon M_2)$ and $\textsc{max-stoc}(\exists x, Q\colon M_1, M_2) \leq \textsc{prob-stoc}(\exists x, Q\colon M_1) + \textsc{prob-stoc}(\exists x, Q\colon M_2)$.

Consider $\textsc{max-stoc}(\P x, Q\colon M_1, M_2)$, that is

$$average \begin{pmatrix} \textsc{max-stoc}(Q\colon M_1, M_2\{x \leftarrow \mathit{True}\}), \\ \textsc{max-stoc}(Q\colon M_1, M_2\{x \leftarrow \mathit{False}\}) \end{pmatrix}.$$

By induction, this is not more than

$$1/2 \times \left(\begin{array}{l} (\text{PROB-STOC}(Q\colon M_1\{x \leftarrow \mathit{True}\}) + \text{PROB-STOC}(Q\colon M_2\{x \leftarrow \mathit{True}\}))+ \\ (\text{PROB-STOC}(Q\colon M_1\{x \leftarrow \mathit{False}\}) + \text{PROB-STOC}(Q\colon M_2\{x \leftarrow \mathit{False}\})) \end{array} \right).$$

Rearranging, this is equal to

$$\left(\begin{array}{l} 1/2 \times (\text{PROB-STOC}(Q\colon M_1\{x \leftarrow \mathit{True}\}) + \text{PROB-STOC}(Q\colon M_1\{x \leftarrow \mathit{False}\}))+ \\ 1/2 \times (\text{PROB-STOC}(Q\colon M_2\{x \leftarrow \mathit{True}\}) + \text{PROB-STOC}(Q\colon M_2\{x \leftarrow \mathit{False}\})) \end{array} \right),$$

which by definition of PROB-STOC on ¶ quantifiers, brings us to our goal

$$\text{PROB-STOC}(\P x, Q\colon M_1) + \text{PROB-STOC}(\P x, Q\colon M_2).$$

In other words, we have shown in the case of a ¶ quantifier that MAX-STOC is bounded above by the sum of PROB-STOC values.

Regarding $\exists$, consider $\text{MAX-STOC}(\exists x, Q\colon M_1, M_2)$, that is

$$\mathit{max} \left(\begin{array}{l} \text{MAX-STOC}(Q\colon M_1, M_2\{x \leftarrow \mathit{True}\}), \\ \text{MAX-STOC}(Q\colon M_1, M_2\{x \leftarrow \mathit{False}\}) \end{array} \right).$$

By induction, this is not more than

$$\mathit{max} \left(\begin{array}{l} (\text{PROB-STOC}(Q\colon M_1\{x \leftarrow \mathit{True}\}) + \text{PROB-STOC}(Q\colon M_2\{x \leftarrow \mathit{True}\})), \\ (\text{PROB-STOC}(Q\colon M_1\{x \leftarrow \mathit{False}\}) + \text{PROB-STOC}(Q\colon M_2\{x \leftarrow \mathit{False}\})) \end{array} \right).$$

For the next step, we appeal to the fact that for any A, B, C, D, $\mathit{max}((A+B),(C+D)) \leq \mathit{max}(A,C) + \mathit{max}(B,D)$. Indeed, note that $A \leq \mathit{max}(A,C)$ and $B \leq \mathit{max}(B,D)$, so $A + B \leq \mathit{max}(A,C) + \mathit{max}(B,D)$. By similar reasoning, $C + D \leq \mathit{max}(A,C) + \mathit{max}(B,D)$. Thus one has the desired result that $\mathit{max}((A+B),(C+D)) \leq \mathit{max}(A,C) + \mathit{max}(B,D)$. Simply instantiating this fact, one sees that the above displayed expression does not exceed the sum

$$\left(\begin{array}{l} \mathit{max}(\text{PROB-STOC}(Q\colon M_1\{x \leftarrow \mathit{True}\}), \text{PROB-STOC}(Q\colon M_1\{x \leftarrow \mathit{False}\}))+ \\ \mathit{max}(\text{PROB-STOC}(Q\colon M_2\{x \leftarrow \mathit{True}\}), \text{PROB-STOC}(Q\colon M_2\{x \leftarrow \mathit{False}\})) \end{array} \right),$$

which by definition of PROB-STOC on $\exists$ quantifiers, brings us to our goal

$$\text{PROB-STOC}(\exists x, Q\colon M_1) + \text{PROB-STOC}(\exists x, Q\colon M_2).$$

In other words, we have shown in the case of a $\exists$ quantifier, that MAX-STOC is bounded above by the sum of PROB-STOC values.

In general, given that $\text{MAX-STOC}(Q\colon M_1, M_2) \leq \text{PROB-STOC}(Q\colon M_1) + \text{PROB-STOC}(Q\colon M_2)$, we ought to show for any k that

$$\text{MAX-STOC}(Q\colon M_1, M_2, \ldots, M_k) \leq \sum_{i=1}^{k} \text{PROB-STOC}(Q\colon M_i).$$

This can be done by induction on k, using intermediate results of the form $\text{MAX-STOC}(Q\colon M_1, M_2) \leq \text{MAX-STOC}(Q\colon M_1) + \text{MAX-STOC}(Q\colon M_2)$, which are proved exactly along the lines of the binary case above. Alternatively, one could repeat the binary proof above with k game-boards M_i and obtain the result directly. ∎

Proof of Theorem 6.1 Continuing with the notation introduced earlier in this section, let $k = |\phi|$. Then $|\psi|$ is roughly k^2. Let $b > 0$ be such that $n^b < 2^{n^d}$ for all n large enough, and let $0 < c \leq b/4$. Now let $n = |\psi|$ and suppose g belongs to the n^{-c}-neighborhood of MAX-STOC, that is

$$|\psi|^{-c} \leq \frac{g(\psi)}{\text{MAX-STOC}(\psi)} \leq |\psi|^c.$$

Then one alternative is $\text{PROB-STOC}(\phi) = 1$, in which case $\text{MAX-STOC}(\psi) = |\psi|^{1/2}$ and thus $g(\psi) \geq |\psi|^{-c+1/2}$. The other alternative is that $\text{PROB-STOC}(\phi) \leq 2^{-n^d}$, in which case by Lemma 6.2 and by our choice of b, $\text{MAX-STOC}(\psi) \leq |\psi|^{-b+1/2}$ and hence $g(\psi) \leq |\psi|^{c-b+1/2}$. But by our choice of c, $-c + 1/2$ exceeds $c - b + 1/2$ (by at least $b/2$.) ∎

7 Representing games in the linear logic proof game

In this section, stochastic (generalized) formula games discussed in Section 5 will be represented by linear logic games introduced in Section 2. The simple game on formulas will be represented by the simple version of the linear logic proof game. The clause-counting games will be represented by means of the weighted version of the linear logic proof game. Of course, the representations themselves will be computable in polynomial time.

In the instances we consider, a natural way to define game representations is to emphasize optimal strategies, that is, optimal with respect to score, rather than winning strategies:

Definition 7.1 *A game representation is an assignment that maps*

- *inputs to inputs,*
- *proponent's moves to sequences of proponent's moves,*
- *opponent's moves to sequences of moves containing exactly one opponent's move,*
- *proponent's strategies to proponent's strategies, and*
- *proponent's optimal strategies to proponent's optimal strategies.*

In the instances we consider, these conditions imply that the game score is preserved.

The reader will observe that the proof of the PSPACE-hardness of MALL in [20] in fact establishes a polynomial-time representation of the simple evaluation game on stochastic formulas (discussed here at the beginning of Section 5) by means of the simple linear logic proof game with a *universal* opponent, who may make clever choices in blocking one of the premises of a & rule instance.

Let us summarize the main results of this section.

Theorem 7.2 *There exists a polynomial-time representation of the* PROB-STOC *game by the simple linear logic proof game with the randomized opponent. In particular, the associated polynomial-time encoding* $[\![\cdot]\!]$ *of stochastic formulas by* MALL *sequents satisfies the condition* $\text{PROB-STOC}(\phi) = \mu([\![\phi]\!])$ *for all closed* ϕ.

Theorem 7.3 *There exists a polynomial-time representation of the* MAX-STOC *game by the weighted linear logic proof game with the randomized opponent. In particular, the associated polynomial-time encoding* $[\![\cdot]\!]$ *of generalized stochastic formulas by* MALL *sequents satisfies the condition* $\text{MAX-STOC}(\phi) = \rho([\![\phi]\!])$ *for all closed* ϕ.

7.1 Encoding generalized stochastic formulas in MALL

We assume without loss of generality that all the quantified propositional variables are renamed apart. The encoding $[\![\phi]\!]$ depends on a positive integer m (standing for the multiplicity of the "locks" and "keys") and a MALL formula A (which stands as an alternative to each "clause"). Two important cases are when $A = \bot$ or $A = \top$.

The notation B^n will be used to indicate a sequence of n B's, separated by $\otimes$, in other words:

$$B^n \triangleq \overbrace{B \otimes B \otimes \cdots \otimes B}^{n}$$

The special case B^0 is defined to be 1, the unit of $\otimes$. The notation $B^{\langle n \rangle}$ stands for the formula built from n copies of the formula B using only the connective $⅋$.

$$B^{\langle n \rangle} \triangleq \overbrace{B ⅋ B ⅋ \cdots ⅋ B}^{n}$$

The special case of $B^{\langle 0 \rangle}$ is defined to be $\bot$, the unit of $⅋$. The formula $(d ⅋ d^\perp)$, where $\vdash d, d^\perp$ is a distinguished axiom, will be abbreviated by $\star$. The atom d is distinct from any other atoms used in the encoding.

Generalized prenex formulas described in Section 5 can be encoded in MALL as follows, where k is chosen fresh for each occurrence of a quantifier.

$$\begin{array}{rcl}
[\![\P x.F]\!] & = & ((x^{\langle Count(x,F) \rangle} ⅋ k) \,\&\, ((x^\perp)^{\langle Count(x,F) \rangle} ⅋ k)) ⅋ (k^\perp \otimes [\![F]\!]) \\
[\![\exists x.F]\!] & = & ((x^{\langle Count(x,F) \rangle} ⅋ k) \oplus ((x^\perp)^{\langle Count(x,F) \rangle} ⅋ k)) ⅋ (k^\perp \otimes [\![F]\!]) \\
[\![\{M_1, \ldots, M_j\}]\!] & = & (([\![M_1]\!] \otimes \star) \oplus A) \otimes \ldots \otimes (([\![M_j]\!] \otimes \star) \oplus A) \\
[\![C \wedge M]\!] & = & [\![C]\!] \otimes [\![M]\!] \\
[\![L \vee C]\!] & = & [\![L]\!] \oplus [\![C]\!] \\
[\![x]\!] & = & (x^\perp \otimes \top) \\
[\![\neg x]\!] & = & (x \otimes \top)
\end{array}$$

where the auxiliary function $Count(x, F)$ is defined by

$$\begin{array}{rcll}
Count(x, x) & = & 1 & \\
Count(x, y) & = & 0 & \text{for atomic } y \neq x \\
Count(x, \neg z) & = & Count(x, z) & \text{for atomic } z \\
Count(x, L \vee C) & = & max(Count(x, L), Count(x, C)) & \\
Count(x, C \wedge M) & = & Count(x, C) + Count(x, M) & \\
Count(x, \{M_1, \ldots, M_j\}) & = & Count(x, M_1) + \ldots + Count(x, M_j) & \\
Count(x, \P x.A) & = & 0 & \\
Count(x, \P y.A) & = & Count(x, A) & \text{for atomic } y \neq x \\
Count(x, \exists x.A) & = & 0 & \\
Count(x, \exists y.A) & = & Count(x, A) & \text{for atomic } y \neq x
\end{array}$$

7.2 Example

Consider the stochastic formula $\phi = \exists x \P y \P z \{(x \vee y \vee z), (\neg x \vee \neg y \vee z)\}$. The truth-value assignment $x \rightarrow True, y \rightarrow True, z \rightarrow False$ satisfies the first clause but not the second. The truth-value assignment $x \to False, y \to False, z \to False$ satisfies the second clause but not the first. All other six truth-value assignments satisfy both clauses. Thus PROB-STOC$(\phi) = 3/4$ and MAX-STOC$(\phi) = 7/4$. The entire formula ϕ is encoded in MALL as

$$((x \parr x \parr k_1) \oplus (x^\perp \parr x^\perp \parr k_1)) \parr (k_1^\perp \otimes D),$$

where D is $[\![\P y \P z\{(x \vee y \vee z), (\neg x \vee \neg y \vee z)\}]\!]$, that is

$$\begin{array}{l} ((y \parr y \parr k_2) \,\&\, (y^\perp \parr y^\perp \parr k_2)) \parr \\ k_2^\perp \otimes \left[\begin{array}{l} ((z \parr z \parr k_3) \,\&\, (z^\perp \parr z^\perp \parr k_3)) \parr \\ k_3^\perp \otimes \ [\![\{(x \vee y \vee z), (\neg x \vee \neg y \vee z)\}]\!] \end{array} \right] \end{array}$$

and $[\![\{(x \vee y \vee z), (\neg x \vee \neg y \vee z)\}]\!]$ is

$$\Big((((x^\perp \otimes \top) \oplus (y^\perp \otimes \top) \oplus (z^\perp \otimes \top)) \otimes \star) \oplus A \Big) \otimes \Big((((x \otimes \top) \oplus (y \otimes \top) \oplus (z^\perp \otimes \top)) \otimes \star) \oplus A \Big).$$

Let us consider the proponent's strategy in the linear logic proof game that begins with the sequent $\vdash [\![\exists x \P y \P z\{(x \vee y \vee z), (\neg x \vee \neg y \vee z)\}]\!]$. The first move must be a $\parr$ move, since there is only one formula in the sequent and this formula has $\parr$ as the main connective.

A play of the game can then continue in two ways: with the $\oplus$ move or with the $\otimes$ move. The $\oplus$ option is better for the proponent. Indeed, the $\otimes$ option leads to two sequents. One contains $k_1^\perp$ and the other contains D, *i.e.*$[\![\P y \P z\{(x \vee y \vee z), (\neg x \vee \neg y \vee z)\}]\!]$. The formula $((x \parr x \parr k_1) \oplus (x^\perp \parr x^\perp \parr k_1))$ may occur in either of the two sequents. In either case the proponent cannot win on the sequent containing $k_1^\perp$, that is, the sequent containing $k_1^\perp$ is not provable. Hence the proponent makes the $\oplus$ move. The two possibilities lead to very similar results. Consider the left-hand alternative at the $\oplus$, as illustrated in the figure below. This alternative corresponds to the truth value assignment $x \to True$ for the boolean variable x. By Theorem 2.3, the following two moves might as well be $\parr$ moves. This opening sequence of moves by the proponent can be represented as

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash x, x, k_1, (k_1^\perp \otimes D)}}{\vdash x, x \parr k_1, (k_1^\perp \otimes D)}\parr}{\vdash x \parr x \parr k_1, (k_1^\perp \otimes D)}\parr}{\vdash ((x \parr x \parr k_1) \oplus (x^\perp \parr x^\perp \parr k_1)), (k_1^\perp \otimes D)}\oplus}{\vdash ((x \parr x \parr k_1) \oplus (x^\perp \parr x^\perp \parr k_1)) \parr (k_1^\perp \otimes D)}\parr$$

If the right-hand alternative at the $\oplus$ had been chosen instead, the resulting sequent would be the same except for the replacement of two copies of x with $x^\perp$, corresponding directly to the truth assignment of *False* to the boolean variable x. After that the only possible move is $\otimes$ because there are no other applicable MALL proof rules. This $\otimes$ move leads to two sequents. If the proponent is to win the sequent containing $k_1^\perp$ must also contain k_1 and no other formulas. Note that the earlier attempted $\otimes$ move would have led into proponent's defeat exactly because k_1 could not be separated from x or $x^\perp$.

Thus one can see that the propositional formula k_1 in this MALL representation is playing the part of a "key", and $k_1^\perp \otimes B$ is playing the part of a "lock" on the formula B. The mechanism locking the remainder of the MALL formula may not be opened until the key is available without completely spoiling the proponent's chances of winning.

The next part of proponent' strategy follows very similar reasoning, where the locks and keys k_2 and k_3 enforce the order of the moves which corresponds to the order of the moves in the stochastic (generalized) formula game. That is, the proponent's $\oplus$ moves in the linear logic proof game correspond to the moves of the player "$\exists$" in the stochastic formula game and the opponent's moves in the linear logic proof game correspond to the moves of the player "¶" in the stochastic formula game. In other words, the proponent can win only if the $\oplus$ and & moves happen in the same order as the corresponding quantifiers $\exists$ and ¶ are evaluated in a given stochastic formula.

After such $\oplus$ and & moves, which correspond to the choice of a truth value assignment, the game is basically finished, except that the proponent has to demonstrate that the two clauses are validated by the chosen truth assignment. That is, the cases differ only in the truth assignment, being of the form $\vdash x, x, y, y, z, z, \Phi$, or $\vdash x, x, y, y, z^\perp, z^\perp, \Phi$, or $\vdash x, x, y^\perp, y^\perp, z, z, \Phi$, etc. In each case the formula Φ encoding the set of clauses is the same. Also note that the truth-value assignment leaves each variable duplicated just the right number of times for its use in Φ.

Consider the case where the opponent chooses the left-hand alternative at each of the two & moves, corresponding to the truth value assignment $y \to True, z \to True$ for the boolean variables y, z. (Recall that proponent has already made the $\oplus$ choice corresponding to $x \to True$.)

$$\dfrac{\dfrac{\cdots \quad \dfrac{\vdots}{\vdash x, x, y, y, z, z, [\![\{(x \vee y \vee z), (\neg x \vee \neg y \vee z)\}]\!]} \quad \cdots}{\vdots}}{\vdash ((x \parr x \parr k_1) \oplus (x^\perp \parr x^\perp \parr k_1)) \parr (k_1^\perp \otimes D)}\parr$$

The next move must be the $\otimes$ move and the best option for the proponent is to partition the truth assignment appropriately between the two resulting sequents, according to the number of occurrences of each variable in each clause:

$$\dfrac{\vdash x, y, z, ([\![x \vee y \vee z]\!] \otimes \star) \oplus A \qquad \vdash x, y, z, ([\![\neg x \vee \neg y \vee z]\!] \otimes \star) \oplus A}{\vdash x, x, y, y, z, z, [\![\{(x \vee y \vee z), (\neg x \vee \neg y \vee z)\}]\!]}\otimes$$

If $A = \perp$ then for example the left-hand side of this $\otimes$ partition is $\vdash x, y, z, (((x^\perp \otimes \top) \oplus (y^\perp \otimes \top) \oplus (z^\perp \otimes \top)) \otimes \star) \oplus \perp$. The only nonatomic formula in this sequent has main connective $\oplus$. The proponent has no hope winning by choosing $\perp$, so instead the proponent chooses the other option, and then immediately makes a $\otimes$ move involving $\star$:

$$\dfrac{\dfrac{\vdash x, y, z, (x^\perp \otimes \top) \oplus (y^\perp \otimes \top) \oplus (z^\perp \otimes \top) \qquad \overline{\vdash \star}}{\vdash x, y, z, (((x^\perp \otimes \top) \oplus (y^\perp \otimes \top) \oplus (z^\perp \otimes \top)) \otimes \star)}\otimes}{\vdash x, y, z, (((x^\perp \otimes \top) \oplus (y^\perp \otimes \top) \oplus (z^\perp \otimes \top)) \otimes \star) \oplus \perp}\oplus$$

On the other hand, if $A = \top$ then the left-hand side of the $\otimes$ partition just discussed is $\vdash x, y, z, (((x^\perp \otimes \top) \oplus (y^\perp \otimes \top) \oplus (z^\perp \otimes \top)) \otimes \star) \oplus \top$. The only nonatomic formula in this sequent has main connective $\oplus$. Choosing the $\top$ leads to a win for the proponent with the score of zero. Choosing the other option leads to the following moves.

$$\dfrac{\dfrac{\vdash x, y, z, (x^\perp \otimes \top) \oplus (y^\perp \otimes \top) \oplus (z^\perp \otimes \top) \qquad \overline{\vdash \star}}{\vdash x, y, z, (((x^\perp \otimes \top) \oplus (y^\perp \otimes \top) \oplus (z^\perp \otimes \top)) \otimes \star)}\otimes}{\vdash x, y, z, (((x^\perp \otimes \top) \oplus (y^\perp \otimes \top) \oplus (z^\perp \otimes \top)) \otimes \star) \oplus \top}\oplus$$

With either $A = \bot$ or $A = \top$ proponent wins in this branch and likewise the boolean clause $x \vee y \vee z$ is satisfied by the assignment $x \to True, y \to True, z \to True$. For instance, proponent may choose to finish with

$$\dfrac{\dfrac{\dfrac{}{\vdash x, x^{\perp}}\,I \qquad \dfrac{}{\vdash y, z, \top}\,\top}{\vdash x, y, z, x^{\perp} \otimes \top}\,\otimes}{\vdash x, y, z, (x^{\perp} \otimes \top) \oplus (y^{\perp} \otimes \top) \oplus (z^{\perp} \otimes \top)}\,\oplus$$

However, if $A = \bot$, then proponent does not have an overall winning strategy and $\mu([\![\phi]\!]) = 3/4$, intuitively for the same reason as PROB-STOC$(\phi) = 3/4$. The simple linear logic game against a randomized opponent described in Section 2 admits no winning strategy for the proponent when played on MALL formula $[\![\phi]\!]$ exactly because there is no full MALL proof of $[\![\phi]\!]$.

On the other hand, if one is allowed to end some proof branches with the sequent $\vdash A$ then one can build a correspondence between MAX-STOC(ϕ) and the expected number of sequents of the form $\vdash \star$ one can guarantee along any main branch of a MALL proof of $\vdash [\![\phi]\!]$. This is the technique used to derive Theorem 7.3.

For the specific example $\phi = \exists x \P y \P z\{(x \vee y \vee z), (\neg x \vee \neg y \vee z)\}$, one obtains MAX-STOC$(\phi) = 7/4$. Observe that if in the linear logic encoding one specifies $A = \top$, then the weighted version of the linear logic game against a randomized opponent described in Section 2, admits a winning strategy for the proponent when played on formula $[\![\phi]\!]$ exactly because there is a MALL proof of $[\![\phi]\!]$, which necessarily does not break any locks. However, the value of $\rho([\![\phi]\!])$ is $7/4$ for the same reason as MAX-STOC$(\phi) = 7/4$. That is, the only positive scores are generated in the linear logic proof game from sequents of the form $\vdash \star$, which arise exactly when a clause is satisfied by a truth assignment. Other potential proof attempts that 'break a lock' are penalized so heavily in ρ (scoring -2^n), that the resulting score in such cases is negative.

7.3 Boolean matrix validity as exact provability in MALL

The encoding of stochastic formulas discussed above is related but not identical to the encoding of classical formulas used previously to demonstrate the PSPACE-completeness of provability in MALL [20]. The main differences are that here we invert the meaning of MALL propositions in the truth assignment (using x for the case of x being *True*, rather than $x^{\perp}$), and that the scope of the $\otimes$'s used for locks includes the remainder of the formula, rather than just the next quantifier. We also use a new encoding of the quantifier-free matrix, and accomplish the fanout directly in the quantifier encoding, rather than utilize additional explicit copying or fanout mechanisms. Where the earlier work used truth tables for a circuit-like representation of the boolean matrix, here we use a simpler logical encoding.

In order to facilitate the demonstration of correctness, we extend our encoding to *assignments* I which map a set of stochastic Boolean variables to the truth values $\{True, False\}$.

$$\begin{array}{rcl} [x \to True, I]^F & = & x, x, \ldots, x, [I]^F \\ [x \to False, I]^F & = & x^{\perp}, x^{\perp}, \ldots, x^{\perp}, [I]^F \end{array}$$

where in the first case there are $Count(x, F)$ copies of x on the right-hand side, while in the second case there are $Count(x, F)$ copies of $x^{\perp}$ on the right-hand side.

The following three propositions establish that for any boolean matrix M and truth-value assignment I on the variables of M, I satisfies M iff the sequent $\vdash [I], [\![M]\!]$ is provable.

Proposition 7.4 *Let M be a boolean matrix. If $\vdash \Gamma, [\![M]\!]$ is provable for some Γ consisting only of literals, then for any context Δ the sequent $\vdash \Gamma, \Delta, [\![M]\!]$ is provable.*

Proof By induction on the structure of the cut-free proof of the sequent. If M is of the form $M_1 \wedge M_2$ then by induction we have the desired result as follows:

$$\frac{\vdash \Gamma_1, [\![M_1]\!] \qquad \vdash \Gamma_2, [\![M_2]\!]}{\vdash \Gamma, [\![M_1]\!] \otimes [\![M_2]\!]}\otimes \qquad \Rightarrow \qquad \frac{\vdash \Gamma_1, \Delta, [\![M_1]\!] \qquad \vdash \Gamma_2, [\![M_2]\!]}{\vdash \Gamma, \Delta, [\![M_1]\!] \otimes [\![M_2]\!]}\otimes$$

If M is of the form $M_1 \vee M_2$, by induction we have the desired result as follows:

$$\frac{\vdash \Gamma, [\![M_i]\!]}{\vdash \Gamma, [\![M_1]\!] \oplus [\![M_2]\!]}\oplus \qquad \Rightarrow \qquad \frac{\vdash \Gamma, \Delta, [\![M_i]\!]}{\vdash \Gamma, \Delta, [\![M_1]\!] \oplus [\![M_2]\!]}\oplus$$

If M is a propositional variable x, then $[\![M]\!] = (x^\perp \otimes \top)$. When this formula is analyzed, one branch of the proof contains a top-level formula $\top$. We can add Δ to the context of that sequent, now proved with the use of the $\top$ rule, and complete the required proof with the use of $\otimes$ as below.

$$\frac{\vdash \Gamma_1, x^\perp \qquad \vdash \Gamma_2, \top}{\vdash \Gamma, [\![x]\!]}\otimes \qquad \Rightarrow \qquad \frac{\vdash \Gamma_1, x^\perp \qquad \overline{\vdash \Gamma_2, \Delta, \top}^{\,\top}}{\vdash \Gamma, \Delta, [\![x]\!]}\otimes$$

The case of a negative literal is similar. ∎

It is readily seen that the restriction on Γ is not necessary in Proposition 7.4, but we do not need the stronger version here.

Proposition 7.5 *If I is an assignment of truth values to the variables of a boolean matrix M, then I satisfies M only if $\vdash [I]^M, [\![M]\!]$ is provable in* MALL.

Proof We assume that I satisfies M, and prove this proposition by structural induction on the boolean matrix M.

If M is of the form $M_1 \wedge M_2$ then I satisfies both M_1 and M_2. By induction we can assume that we can obtain a MALL proof of the encoding of each sequent $\vdash [I]^{M_1}, [\![M_1]\!]$ and $\vdash [I]^{M_2}, [\![M_2]\!]$. We can combine these together with the application of the $\otimes$ rule to obtain $\vdash [I]^{M_1}, [I]^{M_2}, [\![M_1]\!] \otimes [\![M_2]\!]$ which is equivalent to our obligation, $\vdash [I]^M, [\![M]\!]$ since for all y, $Count(y, M) = Count(y, M_1) + Count(y, M_1)$. This proof appears as follows:

$$\frac{\vdash [I]^{M_1}, [\![M_1]\!] \qquad \vdash [I]^{M_2}, [\![M_2]\!]}{\vdash [I]^M, [\![M_1]\!] \otimes [\![M_2]\!]}\otimes$$

If M is of the form $M_1 \vee M_2$ then I satisfies M_1 or I satisfies M_2. Without loss of generality we assume that I satisfies M_1. By induction we can assume that we can obtain a MALL proof of the encoding $\vdash [I]^{M_1}, [\![M_1]\!]$. We continue the proof with the application of the $\oplus$ rule to obtain

$\vdash [I]^{M_1}, [\![M_1 \vee M_2]\!]$. The result follows by Proposition 7.4 and the $\parr$ rule. The proof could appear as follows with the proposition being used as a derived rule of inference.

$$\dfrac{\dfrac{\vdash [I]^{M_1}, [\![M_1]\!]}{\vdash [I]^{M_1}, [\![M_1]\!] \oplus [\![M_2]\!]}\oplus}{\vdash [I]^{M}, [\![M_1]\!] \oplus [\![M_2]\!]}\text{Prop. 7.4 and }\parr$$

In the case that M is some propositional variable x, then we know that in order for x to hold in I, I must assign *True* to x, and thus x appears in the encoding sequent. Since x appears exactly once in the formula x, the encoding of this situation is thus of the form $\vdash x, x^{\perp} \otimes \top$, which is provable in MALL using the $\otimes$, identity, and $\top$ rules as follows:

$$\dfrac{\dfrac{}{\vdash x, x^{\perp}}I \quad \dfrac{}{\vdash \top}\top}{\vdash [I]^{x}, [\![x]\!]}\otimes$$

The case of a negative literal is similar. ∎

Proposition 7.6 *If I is an assignment of truth values to variables of a boolean matrix M, then for any boolean matrix M', I satisfies M if $\vdash [I]^{M}, [I]^{M'}, [\![M]\!]$ is provable in* MALL.

Proof By structural induction on the boolean matrix M.

Suppose M is of the form $M_1 \wedge M_2$. Because $[I]^{M}, [I]^{M'}$ consists only of atomic literals, the assumed cut-free proof must end in an application of $\otimes$ as follows:

$$\dfrac{\vdash [I]^{N_1}, [\![M_1]\!] \qquad \vdash [I]^{N_2}, [\![M_2]\!]}{\vdash [I]^{M}, [I]^{M'}, [\![M_1]\!] \otimes [\![M_2]\!]}\otimes$$

where $[I]^{M}, [I]^{M'}$ is multiset-equal to $[I]^{N_1}, [I]^{N_2}$. By Proposition 7.4, it is then the case that both $\vdash [I]^{M_1}, [I]^{N_1}, [\![M_1]\!]$ and $\vdash [I]^{M_2}, [I]^{N_2}, [\![M_2]\!]$ are provable. Appealing to the induction hypothesis, one obtains that I satisfies M_1 and that I also satisfies M_2. By the semantic meaning of $\wedge$ it follows that I satisfies M.

If M is of the form $M_1 \vee M_2$ then the assumed cut-free proof is of the form

$$\dfrac{\vdash [I]^{M}, [I]^{M'}, [\![M_i]\!]}{\vdash [I]^{M}, [I]^{M'}, [\![M_1]\!] \oplus [\![M_2]\!]}\oplus$$

By rearranging $[I]^{M}, [I]^{M'}$ into $[I]^{M_i}, [I]^{M''}$, an appealing to the induction hypothesis, we know that I satisfies M_i, and thus I satisfies $M_1 \vee M_2$.

In the case that M is some propositional variable x, the sequent $\vdash [I]^{x}, [I]^{M'}, x^{\perp} \otimes \top$ is provable in MALL only if x occurs in $[I]^{x}, [I]^{M'}$. By inspection of the translation of I, I must assign x to *True*. The case of M being a negated literal $\neg x$ is similar. ∎

When the linear logic encoding is computed with $A = \perp$ or $A = \top$, Proposition 7.4 extends to generalized classical formulas if the quantifier $\forall$ is encoded in the same way as the quantifier $\P$. Proposition 7.5 so extends for any choice of A. Proposition 7.6 so extends if $A = \perp$. We do not need these stronger versions here.

7.4 Game representations induced by MALL encodings

In this subsection it is shown that MALL encodings introduced above induce game representations stated in Theorems 7.2, and 7.3. In particular, it is shown that the PROB-STOC game corresponds directly to the simple version of the linear logic game against a randomized opponent. This correspondence uses the form of the linear logic encoding where $A = \perp$. On the other hand, if $A = \top$, it is shown that the MAX-STOC game corresponds directly to the weighted version of the linear logic game against a randomized opponent.

Let us discuss the MAX-STOC case in detail. The PROB-STOC case is similar.

The reader will observe that the function MAX-STOC can be readily extended to include a truth assignment. Namely, MAX-STOC(I, ϕ) is MAX-STOC$(\phi\{I\})$ where I has been applied to ϕ as a substitution of truth values for propositional variables.

Lemma 7.7 *The linear logic encoding computed with $A = \top$ induces a game representation of the* MAX-STOC *counting game by the weighted linear logic proof game against a randomized opponent. In particular, if ϕ is a generalized stochastic formula with all quantified propositional variables are named apart and I is an assignment of truth values to free variables of ϕ, and if n is at least the length of $[\![\phi]\!]$, then* MAX-STOC$(I, \phi) = \rho_n([I]^\phi, [\![\phi]\!])$.

Proof The game representation is defined and the required properties proved simultaneously by structural recursion on the generalized stochastic formula ϕ.

If ϕ is of the form $\exists x.\phi'$, then the player "$\exists$" (proponent) moves in the MAX-STOC game. MAX-STOC(I, ϕ) is the maximum of MAX-STOC$(I\{x \leftarrow True\}, \phi')$ and MAX-STOC$(I\{x \leftarrow False\}, \phi')$. Assume the maximum is reached when x gets *True*. (The case for *False* is similar.) In the weighted linear logic game, the proponent first makes the $\parr$ move in the position $[\![\phi]\!]$ (the only option since $\parr$ is the only top-level connective.) Then the proponent has two options: playing the $\otimes$ or playing the $\oplus$. The proponent plays $\oplus$ with the choice of $x^{\langle Count(x,\phi')\rangle} \parr k$, which corresponds to the truth value *True* chosen in the MAX-STOC game by the player "$\exists$".

This choice is optimal for the proponent in the linear logic proof game. Indeed, playing the $\otimes$ instead of the $\oplus$ would lead to proponent's defeat and a negative score independently of the future moves because the resulting sequent containing $k^\perp$ is not provable in MALL (Proposition 2.2.)

The linear logic game continues with proponent playing the $\parr$ of the linear logic formula corresponding to the truth value and then the top-level locking $\otimes$ $k^\perp \otimes [\![\phi']\!]$. These moves are optimal by Theorem 2.3. The result is one sequent with an easy win for the proponent with zero score $\vdash k, k^\perp$ and another sequent of the form $\vdash [I]^\phi, x^{\langle Count(x,\phi')\rangle}, [\![\phi']\!]$. By Theorem 2.3 it is optimal for the proponent to replace the $\parr$s in $x^{\langle Count(x,\phi')\rangle}$ by commas, which the proponent does in $Count(x, \phi') - 1$ $\parr$ moves, if any. Because variables are renamed apart in ϕ, it is the case that $[I]^\phi = [I]^{\phi'}$. Thus the resulting sequent can be written as $\vdash [I']^{\phi'}, [\![\phi']\!]$. One now appeals to the inductive hypothesis to ensure that MAX-STOC$(I', \phi') = \rho_n([I']^{\phi'}, [\![\phi']\!])$.

If ϕ is of the form $\P x.\phi'$, then the player "$\P$" (opponent) moves in the MAX-STOC game. MAX-STOC(I, ϕ) is the average of MAX-STOC$(I\{x \leftarrow True\}, \phi')$ and MAX-STOC$(I\{x \leftarrow False\}, \phi')$. In the linear logic game the corresponding sequence of moves is defined as in the case just discussed, except that instead of the proponent's move $\oplus$ the opponent plays $\&$. The surrounding moves by the proponent are optimal for the same reason as above. In this case, depending on the choice the opponent randomly makes in the $\&$ move, the resulting position involves the sequent $\vdash [I]^\phi, x^{\langle Count(x,\phi')\rangle}, [\![\phi']\!]$ or the sequent $\vdash [I]^\phi, (x^\perp)^{\langle Count(x,\phi')\rangle}, [\![\phi']\!]$. In either case after the further

$Count(x, \phi') - 1$ ⅋ moves the sequent is of the form $\vdash [I']^{\phi'}, [\![\phi']\!]$ for the same reason as above, and thus one can apply the inductive hypothesis. The function ρ_n takes the average of these two cases.

In the base case ϕ is a multiset of boolean matrices, say $\{M_1, \ldots, M_j\}$. This is the endgame. The proponent plays the $\otimes$s in $\vdash [I]^{\phi}, [\![\{M_1, \ldots, M_j\}]\!]$ by partitioning $[I]^{\phi}$ into $[I]^{M_1}, \ldots, [I]^{M_j}$ and then for each $i = 1, \ldots, j$ the proponent plays the $\oplus$ in $([\![M_i]\!] \otimes \star) \oplus \top$, choosing $[\![M_i]\!] \otimes \star$ if I satisfies M_i else $\top$. By Proposition 7.5, in this play for each i the proponent obtains the score of 1 if I satisfies M_i else 0. Propositions 7.4 and 7.6 ensure that this is an optimal play for the proponent. Indeed, in any other partition $[I]^{N_1}, \ldots, [I]^{N_j}$ of $[I]^{\phi}$ the score on the branch that begins with $\vdash [I]^{N_i}, ([\![M_i]\!] \otimes \star) \oplus \top$ is 1 if the sequent $\vdash [I]^{N_i}, [\![M_i]\!]$ is provable, else 0. But if $\vdash [I]^{N_i}, [\![M_i]\!]$ is provable, then by Proposition 7.4 $\vdash [I]^{N_i}, [I]^{M_i}, [\![M_i]\!]$ is provable and hence by Proposition 7.6 I satisfies M_i. ■

Let us note that in the PROB-STOC case with the linear logic encoding computed with $A = \bot$ generalized formulas are not necessary. The argument, which closely follows the argument just above, can be made directly on stochastic formulas.

Theorem 3.7 is now immediate from Theorems 5.1 and 7.2, where the encoding $[\![\cdot]\!]$ defined in Section 7.1 is computed with $A = \bot$.

Similarly, Theorems 6.1 and 7.3 yield another proof of Theorem 3.5. This proof, however, yields an additional feature that the set of provable MALL sequents on which the function ρ is PSPACE-hard to approximate may be restricted further. Another such restriction follows from the result from [7] cited here as Theorem 5.2. Let us say that a *constraint* is a MALL formula of the form $(A \otimes z_1)⅋(z_1^\perp \otimes z_2) \ldots ⅋(z_n^\perp \otimes \star)$, where & does not occur in A, each z_i is a literal, and where $\star$ abbreviates the formula $d⅋d^\perp$ related to the distinguished axiom $\vdash d, d^\perp$. Note that the *size* of a constraint includes the size of the formula A. Let us say that a sequent is *constrained* if the distinguished formula $\star$ occurs only in constraints. Note that provability of constraints is in NP, in contrast with the general MALL provability, which is PSPACE-complete.

Theorem 7.8 *There exists a positive real number c such that the function ρ is* PSPACE*-hard to approximate within multiplicative ratio n^{-c} on provable constrained* MALL *sequents, where n is the length of the sequent. Furthermore, there exists a positive real number $\epsilon < 1$ such that the function ρ is* PSPACE*-hard to approximate within multiplicative ratio ϵ on provable constrained* MALL *sequents where every constraint is of constant size.*

On the other hand, proof-theoretic techniques of Mints and Kanovich may be used to restrict provable MALL sequents to those that contain only formulas of depth at most 2.

Proof The first part follows from Theorems 6.1 and 7.3, where the encoding $[\![\cdot]\!]$ defined in Section 7.1 is computed with $A = \top$. The second part follows from Theorem 5.2 by the same encoding. ■

Theorems 7.3, 3.5, and 7.8 have analogs in the case in which the game is played against a universal opponent, but that case is omitted here. The reader is referred to [26].

8 Lower bounds for local proof heuristics

We have shown in the previous sections that the optimal score functions μ, ρ and the corresponding optimal strategies for proponent are PSPACE-hard to approximate on MALL-formulas and NP-hard to approximate on MLL$\top$ formulas. Some of these properties follow directly from the NP-hardness of MLL$\top$ [18, 22] and the PSPACE-hardness of MALL [20], while others involve recent complexity-theoretic techniques for proving lower bounds on optimization problems [4, 3, 36, 8, 7, 16].

But how hard is it to make *one* good move? In this section we investigate lower bounds on such local proof search heuristics. For instance, we show that on MLL$\top$ formulas, it is NP-hard to compute a local proof search heuristic up to any constant factor. That is, unless P=NP, there is no polynomial-time strategy for proponent satisfying the condition that, given a MLL$\top$ formula A, chooses a proof rule instance so that even if one continued playing the game from then on optimally, the achieved score would be within a factor of the true optimal score $\rho(A)$. Such a heuristic could be used to solve any problem in NP in polynomial time. Therefore such heuristics are not likely to exist. As in Section 3, it suffices to require the above condition only for provable formulas A.

The formal definition follows along the lines of Definition 3.1.

Definition 8.1 *Let $0 < \epsilon < 1$. An ϵ-heuristic is a function from* MALL *sequents to proof rule instances such that, for every sequent Γ, the score $h(\Gamma)$ achieved by playing optimally from then on is within multiplicative ratio ϵ of the optimal achievable score $OPT(\Gamma) \geq 0$, that is,*

$$\epsilon \cdot OPT(\Gamma) \leq h(\Gamma) \leq OPT(\Gamma).$$

where in the simple version of the linear logic proof game, $OPT(\Gamma) = \mu(\Gamma)$, and in the weighted version on provable formulas, $OPT(\Gamma) = \rho(\Gamma)$.

Note that the emphasis is on the first inequality. The second inequality holds for *any* strategy of proponent.

The following theorem is a direct consequence of Lemma 3.6 and the NP-hardness of MLL$\top$ [18, 22].

Theorem 8.2 *Let $0 < \epsilon < 1$. If* P$\neq$NP, *then there are no polynomial-time computable ϵ-heuristics for the weighted linear logic proof game on provable* MLL$\top$ *formulas.*

Proof Let $0 < \epsilon < 1$ and let H be an ϵ-heuristic. Recall the proof of Lemma 3.6. Consider a formula of the form

$$A^{\#} = ((A \otimes (d \parr d^{\perp})) \otimes \perp) \parr \top,$$

where d is the distinguished propositional atom and A does not contain d. The first move must be $\parr$.

Consider the second move. We claim that if H indicates that the second move is $\otimes$, then the formula A is provable, and if H indicates that the second move is the axiom $\top$, then A is not provable. Indeed, let h be the score achieved by playing that move according to the heuristic H and optimally from then on. By the main assumption, $\epsilon \cdot \rho(A^{\#}) \leq h \leq \rho(A^{\#})$. Suppose H indicates the axiom $\top$. Then clearly $h = 0$ because this is the last move. If A is nevertheless provable, then $\rho(A^{\#}) = 1$ by Lemma 3.6, and hence by the main assumption $0 < \epsilon < h$, contradiction. Now suppose H indicates $\otimes$. If A is nevertheless unprovable, then $\rho(A^{\#}) = 0$ by Lemma 3.6, and hence by the main assumption $h = 0$. But we have shown in the proof of Lemma 3.6 that if A is unprovable, then playing $\otimes$ yields a negative score, *i.e.*, $h < 0$, contradiction. ∎

Theorem 8.2 may be strengthened to provable formulas of depth at most 2, by proof-theoretic techniques of Mints and Kanovich.

The analogs for either version of the game on all MLL$\top$ formulas, and even on all formulas of the pure multiplicative fragment MLL, may be obtained from the NP-hardness of MLL$\top$ [18, 22] and Propositions 2.1 and 2.2. Indeed, since the values of μ, ρ on MLL$\top$ formulas are discrete, it is just as hard to approximate them as it is to compute them exactly. But iterating the presumed polynomial-time heuristic would yield a close approximation of μ, ρ computable in polynomial time.

In the simple version of the proof game, the reasoning outlined above cannot be directly lifted to MALL, where the additive connectives & and $\oplus$ are allowed as well. The main obstruction is the exponential size of entire MALL proof tree. A part of this difficulty has already been mentioned in the remark just after Proposition 2.1. However, instead of simply building the entire proof tree, one could sample into the tree of possible plays. In this context it makes sense to ask that a proof search heuristic be computable in randomized polynomial time, BPP [30, 29].

Using such a heuristic, one could simulate a single play of the simple version of the proof game against a random opponent. Considering this play a single data point, one could repeatedly sample a polynomial number of times. The law of averages yields a bound on the likelihood of sampling error. The result is that such a heuristic may be used a polynomial number of times to closely approximate the optimal score, which is known from previous sections to be PSPACE-hard.

Theorem 8.3 *Let $0 < \epsilon < 1$. Let H be a heuristic for the simple linear logic proof game on* MALL *formulas such that the expected score resulting from proponent's use of the heuristic throughout a play of the game is within multiplicative ratio ϵ from the optimal score μ. Then H is* PSPACE*-hard.*

The proof of Theorem 8.3 will use the following exercise from probability theory (a simple form of the Law of Averages) [35].

Proposition 8.4 *Suppose an event A occurs with probability p. Let $Y(n)$ be the binomial random variable denoting number of times the event A occurs in n independent repetitions. Then for any $\delta > 0$ and any positive integer n,*

$$Pr\left[\left|\frac{1}{n}Y(n) - p\right| > \delta\right] \leq 2e^{-\frac{n\delta^2}{4}}.$$

Proof By normalized version of IP [11], for any small $\epsilon > 0$ there exists a class of stochastic formulas ϕ such that either PROB-STOC(ϕ) $= 1$ or PROB-STOC(ϕ) $\leq \epsilon/4$, and such that the decision between the two is PSPACE-hard. A similar property is used in [7] to obtain the result cited in this paper as Theorem 5.1 and is also used here in Section 6 in the proof of Theorem 6.1. By our game representation, there exists a class of MALL sequents $\vdash \Gamma$ such that either $\mu(\Gamma) = 1$ or $\mu(\Gamma) \leq \epsilon/4$, and such that the decision between the two is PSPACE-hard. Let us consider the plays of the simple proof game beginning with such sequents $\vdash \Gamma$.

Let X_H be a random variable given by a single play of the simple linear logic proof game against a randomized opponent in which all the moves of proponent are given by the supposed ϵ-heuristic H, $0 < \epsilon < 1$, and the moves of the randomized opponent by flipping a fair coin. Note that the values of X_H are either 0 or 1. Here and throughout this argument we suppress the dependence on a starting sequent Γ in order to simplify notation. Consider the average result of n independent plays,

$$X = \frac{1}{n}\sum_{i=1}^{n} X_H.$$

Note that the expected value is not affected, *i.e.,* $E[X] = E[X_H]$. By Proposition 8.4 with $\delta = \epsilon/4$, for any positive integer n,

$$Pr\left[E[X] - \frac{\epsilon}{4} \leq X \leq E[X] + \frac{\epsilon}{4}\right] \geq 1 - 2e^{-\frac{n\epsilon^2}{64}}.$$

By the main assumption

$$\epsilon \cdot \mu \leq E[X] \leq \mu.$$

But either $\mu = 1$ or $\mu \leq \epsilon/4$, hence *either* $\epsilon \leq E[X]$ and thus

$$Pr\left[\frac{3\epsilon}{4} \leq X\right] \geq 1 - 2e^{-\frac{n\epsilon^2}{64}},$$

or

$$Pr\left[X \leq \frac{\epsilon}{2}\right] \geq 1 - 2e^{-\frac{n\epsilon^2}{64}}.$$

Fix $n \geq 192/\epsilon^2$. Then the probabilities are $\geq 3/4$. Consider nX, the number of proponent's wins in n independent repetitions of plays. If this number is large, then $\mu = 1$. If it is small, then $\mu \leq \epsilon/4$. But in each play, and hence in all n of them (n is fixed; it does not depend on the length of the starting sequent $\vdash \Gamma$,) there are only polynomially many iterations of H.

QUESTION: Can this be formulated in terms of our notion of PSPACE-hardness or does it require Turing reducibility? ∎

Theorem 8.3 relies on recent complexity-theoretic techniques for proving PSPACE lower bounds on optimization problems [5, 14, 28, 34, 11, 7]. In contrast, the following theorem may be obtained directly from the PSPACE-hardness of MALL [20] and the proof of Lemma 3.6, as in the proof of Theorem 8.2.

Theorem 8.5 *Let* $0 < \epsilon < 1$. *Every* ϵ*-heuristic* H *for the weighted linear logic proof game on provable* MALL *formulas is* PSPACE*-hard.*

9 Further Work

We do not know any significant positive results on proof search heuristics in linear logic. It is not known whether any problem related to linear logic proof search heuristics is MAXSNP-complete in the sense of [31, 30]. Optimization problems in MAXSNP, such as MAX-SAT, the problem of maximizing the number of satisfiable clauses in a 3-CNF, have a nontrivial approximation threshold $c > 1$ in the sense that they are polynomial-time approximable within any factor $c' > c$, but NP-hard to approximate within any factor $1 < c' < c$.

Related questions in intuitionistic logic are completely open. Are there lower bounds analogous to our results?

Another potential direction for future research is the consideration of average case complexity, that is, the possibility of proof search heuristics that on "most" sequents provide a good next proof rule instance, but on some sequents choose a rather poor proof rule instance. Our results above only show that it is hard to build a heuristic that is "never too bad," while one may be as interested in heuristics that are "usually good."

Acknowledgements: We would like to thank Mitsu Okada and the local organizers of the Linear Logic '96 Tokyo Meeting for the splendid arrangements at the conference. We would also like to thank Madhu Sudan for directing us to [8, 7] and Sanjeev Arora and Joan Feigenbaum for their informative email correspondence and for their help with the literature. Patrick Lincoln would like to thank the University of Pennsylvania for hosting him during his visit in the spring of 1996. Scedrov would like to thank Stanford University and SRI International for hosting him during his visit in the spring of 1995. All the authors would like to thank the Isaac Newton Institute for Mathematical Sciences in Cambridge, England for hosting them at various times in the fall of 1995.

References

[1] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59:543–574, 1994.

[2] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF (Extended abstract). In *Proc. TACS '94, Springer LNCS Vol. 789*, pages 1–15, 1994.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proc. 33-rd Annual IEEE Symposium on Foundations of Computer Science*, pages 14–23. IEEE Computer Society Press, Los Alamitos, California, 1992.

[4] S. Arora and S. Safra. Probabilistic checking of proofs; A new characterization of NP. In *Proc. 33-rd Annual IEEE Symposium on Foundations of Computer Science*, pages 2–13. IEEE Computer Society Press, Los Alamitos, California, 1992.

[5] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *J. Computer System Sciences*, 36:254–276, 1988.

[6] A. Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56:183–220, 1992.

[7] A. Condon, J. Feigenbaum, C. Lund, and P. Shor. Random debaters and the hardness of approximating stochastic functions. In *Proc. 9-th Annual IEEE Conference on Structure in Complexity Theory*, pages 280–293. IEEE Computer Society Press, Los Alamitos, California, 1994. Full version to appear in SIAM Journal of Computing.

[8] A. Condon, J. Feigenbaum, C. Lund, and P. Shor. Probabilistically checkable debate systems and nonapproximability of PSPACE-hard functions. *Chicago Journal of Theoretical Computer Science*, page http://www.cs.uchicago.edu/publications/cjtcs/articles/1995/4/contents.html, 1995, No. 4. Extended abstract in Proc. 25-th ACM Symposium on Theory of Computing, pages 305-314, ACM, New York, 1993.

[9] V. Danos, H. Herbelin, and L. Regnier. Game semantics and abstract machines. In *Proc. 11-th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, NJ*, July 1996.

[10] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proc. 32-nd Annual IEEE Symposium on Foundations of Computer Science*, pages 2–12. IEEE Computer Society Press, Los Alamitos, California, 1991.

[11] M. Furer, O. Goldreich, Y. Mansour, M. Sipser, and S. Zachos. On completeness and soundness in interactive proof systems. In S. Micali, editor, *Randomness and Computation*, pages 429–442, Greenwich, Connecticut, 1989. Advances in Computing Research, vol. 5, JAI Press.

[12] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[13] J.-Y. Girard. Linear logic: its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in linear logic.* London Mathematical Society Lecture Note Series, Volume 222, Cambridge University Press, 1995.

[14] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Computing*, 18:186–208, 1989.

[15] J. M. E. Hyland and C.-H. L. Ong. Pi-calculus, dialogue games and PCF. In *Proceedings of the 7th* ACM *Conference on Functional Programming Languages and Computer Architecture*, pages 96–107. ACM Press, 1995.

[16] H.B. Hunt III, M.V. Marathe, and R.E. Stearns. Generalized CNF satisfiability problems and non-efficient approximability. In *Proc. 9-th Annual IEEE Conference on Structure in Complexity Theory*, pages 356–366. IEEE Computer Society Press, Los Alamitos, California, 1994.

[17] A. Joyal. Free lattices, communication, and money games. In: Proceedings of the 10-th International Congress of Logic, Methodology, and Philosophy of Science, Firenze, 1995.

[18] M.I. Kanovich. The complexity of Horn fragments of linear logic. *Annals of Pure and Applied Logic*, 69:195–241, 1994.

[19] F. Lamarche. Games semantics for full propositional linear logic. In *Proc. 10-th Annual IEEE Symposium on Logic in Computer Science, San Diego*, pages 464–473, June 1995.

[20] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56:239–311, 1992. Extended abstract in Proc. 31-st Annual IEEE Symposium on Foundations of Computer Science, pages 662-671, 1990.

[21] P. Lincoln and A. Scedrov. First order linear logic without modalities is NEXPTIME-hard. *Theoretical Computer Science*, 135:139–154, 1994.

[22] P. Lincoln and T. Winkler. Constant-Only Multiplicative Linear Logic is NP-Complete. *Theoretical Computer Science*, 135:155–169, 1994.

[23] P.D. Lincoln. *Computational Aspects of Linear Logic.* MIT Press. to appear.

[24] P.D. Lincoln, J.C. Mitchell, and A. Scedrov. Stochastic interaction and linear logic. In J.-Y Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 147–166. London Mathematical Society Lecture Notes, Volume 222, Cambridge University Press, 1995.

[25] P.D. Lincoln, J.C. Mitchell, and A. Scedrov. Linear logic proof games and optimization. *Bulletin of Symbolic Logic*, 2:322–338, 1996.

[26] P.D. Lincoln, J.C. Mitchell, and A. Scedrov. Linear logic proof games and optimization. Preliminary report, January 1996. Available by anonymous ftp from ftp.cis.upenn.edu as pub/papers/scedrov/approx.[dvi,ps].Z. See pub/papers/TableOfContents.

[27] P.D. Lincoln, J.C. Mitchell, and A. Scedrov. The complexity of local proof search in linear logic. Extended abstract. In *Proceedings Linear Logic '96, Tokyo Meeting*, page http://www.elsevier.nl:80/mcs/tcs/pc/Menu.html. Electronic Notes in Theoretical Computer Science, Volume 3, 1996.

[28] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39:859–868, 1992.

[29] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[30] C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[31] C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. In *Proc. 20-th Annual ACM Symposium on Theory of Computing*, pages 229–234, 1988. Also in J. Computer System Sciences, 1991.

[32] A. Scedrov. A brief guide to linear logic. In G. Rozenberg and A. Salomaa, editors, *Current Trends in Theoretical Computer Science*, pages 377–394. World Scientific Publishing Co., 1993.

[33] A. Scedrov. Linear Logic and Computation: A Survey. In H. Schwichtenberg, editor, *Proof and Computation, Proceedings Marktoberdorf Summer School 1993*, volume 139, pages 379–395. NATO Advanced Science Institutes, Series F, Springer-Verlag, Berlin, 1995.

[34] A. Shamir. IP = PSPACE. *J. ACM*, 39:869–877, 1992.

[35] D. Stirzaker. *Elementary Probability.* Cambridge University Press, 1994.

[36] D. Zuckerman. NP-complete problems have a version that's hard to approximate. In *Proc. 8-th Annual IEEE Conference on Structure in Complexity Theory*, pages 305–312. IEEE Computer Society Press, Los Alamitos, California, 1993. Full paper under the title "On Unapproximable Versions of NP-Complete Problems" to appear in SICOMP.