Lower Bounds for Cutting Planes (Extended Abstract)

Yuval Filmus, Toniann Pitassi
University of Toronto

May 15, 2011

Abstract

Cutting Planes (CP) is a refutation propositional proof system whose lines are linear inequalities in Boolean variables. Exponential lower bounds for CP refutations of a clique/coclique dichotomy have been proved by Bonet, Pitassi and Raz, and by Krajíček, using similar methods. Their proofs only apply when all the coefficients in the proof are small. Pudlák proved a lower bound for the same statement without the restriction on the coefficients, but with restrictions on the allowed derivation rules.

The basic plan of all arguments is to reduce the CP proof to a monotone circuit distinguishing between cliques and cocliques, and then employ a monotone circuit lower bound. The techniques of Bonet et al. and of Krajíček are very similar, constructing a monotone circuit to simulate a virtual *proof search*. Bonet et al. describe the construction as part of their argument, while Krajíček invokes a more general theorem of Razborov in an awkward way.

We present an abstraction of this construction, clearly separating it from the rest of the argument. This abstraction allows us to point the differences between the two proofs, differences which can be traced to the somewhat different translation of the statement into the language of CP.

In our view, this new presentation is clearer and cleaner than the previous ones. Our goal is to extend the proof to handle arbitrary coefficients (ongoing work).

1 Introduction

1.1 Cutting Planes

Cutting Planes (CP for short) is a refutation propositional proof system. There are variables x_i, which are assumed to be Boolean (0 or 1). The *lines* in the proof are linear inequalities (hyperplanes) with integer coefficients

$$\sum_i a_i x_i \geq b.$$

There are two sets of rules for CP, forming two different proof systems: Syntactic CP and Semantic CP. Syntactic CP includes an explicit list of derivation rules. Semantic CP allows the derivation of a line ℓ from a set L of up to two lines as long as ℓ follows semantically from L, that is if every $0/1$ assignment to the x_i which satisfies L, also satisfies ℓ. Every derivation valid for Syntactic CP is also valid for Semantic CP.

CP proofs involve "concepts" which are hyperplanes. Syntactic CP can simulate resolution since a resolution clause can be expressed as a hyperplanes with coefficients which are $0, \pm 1$. However, a general hyperplane involving n variables may involve coefficients of magnitude $n!$, see Muroga [6] and Håstad [4]. Buss and Clote [3] showed that in the case of Syntactic CP proofs, one can assume that the coefficients have magnitude only $2^{O(n)}$.

1.2 Lower Bounds

There are several strong lower bounds known for CP. We will only describe lower bounds which apply to dag-like proofs (rather than tree-like proofs). These lower bounds are by Bonet, Pitassi and Raz [2], Krajíček [5] and Pudlák [7].

All lower bounds apply to the same contradiction, although the contradiction is translated to the language of CP in two different ways. The contradiction states that a graph on n vertices has an m-clique while being $(m-1)$-colorable, where $m = \sqrt[3]{n}$. A CP refutation of this contradiction is converted into a monotone circuit which distinguishes between graphs containing an m-clique and graphs which are $(m-1)$-colorable. Alon and Boppana [1] proved a lower bound of $2^{\Omega(n^{1/3})}$ on such circuits.

The proofs by Bonet et al. and by Krajíček are very similar, their main difference being the way that the clique/coclique contradiction is translated into CP. The account of Bonet et al. is somewhat simpler. Both proofs replace each line in the proof by roughly $O(M)$ gates, where M is a bound on the magnitude of the coefficients in the proof. Hence a non-trivial lower bound is only obtained when $M = 2^{o(n^{1/3})}$. Both proofs work for Semantic CP.

Pudlák's account, while similar in spirit, differs in details. The overhead in conversion is only $O(1)$, but the result is a monotone *real* circuit. Also, the proof only works for Syntactic CP. Pudlák extends the monotone circuit lower bound to the real case, and thus obtains a lower bound of $2^{\Omega(n^{1/3})}$ on the number of lines in the proof.

The proofs by Bonet et al. and by Krajíček are quite general and apply in principle to other proof systems and other contradictions. However, for simplicity of presentation we only consider Semantic CP and the clique/coclique contradiction.

1.3 Our Contribution

We restate the proofs of Bonet et al. and Krajíček in terms of a game between two players, similar but not identical to the Karchmer-Wigderson construction.

In our view, our novel presentation is cleaner than previous ones, and should facilitate generalization into wider settings. In particular, we hope to extend the lower bounds to the setting of Semantic CP without restriction on the size of the coefficients.

2 Games

Consider some Boolean formula φ with binary $\wedge, \vee$ and constants $0, 1$. We can think of φ as describing a game between two players, the $\vee$-player and the $\wedge$-player. Picture φ as a binary tree. The players start at the root of φ and work their way down to leaves. Each time the node is $\vee$, the $\vee$-player chooses which way to go. Each time the node is $\wedge$, the $\wedge$-player chooses which way to go. The goal of the $\vee$-player is to reach a 1-leaf. The goal of the $\wedge$-player is to reach a 0-leaf.

Lemma 2.1. *The formula φ evaluates to 1 iff the $\vee$-player has a winning strategy.*

The formula φ evaluates to 0 iff the $\wedge$-player has a winning strategy.

Proof. Easy induction. □

In this section we reverse the construction, obtaining a monotone circuit from the description of the game. First, let us define what a game is.

Definition 2.1. *A* game *is defined by the following data:*

1. *A dag G with a single source where each internal node has out-degree 2.*
2. *Each internal node is labeled with either $\vee$ or $\wedge$.*
3. *Each leaf is labeled with either 0 or 1.*

Two players, the $\vee$-player and the $\wedge$-player, play the game by starting at the source and working their way down. At each internal node, the player whose label is on the node chooses the next node. If the game reaches a leaf labeled with 1, the $\vee$-player winds. Otherwise the $\wedge$-player wins.

In the sequel we will allow nodes to have out-degree 1. Such a game can easily be converted to an equivalent game which has out-degree 2.

Lemma 2.2. *Convert a game G into a circuit C by replacing each internal node by an $\wedge$ or an $\vee$ gate, according to its label.*

Th circuit C evaluates to 1 iff the $\vee$-player has a winning strategy for the game G. The circuit C evaluates to 0 iff the $\wedge$-player has a winning strategy for the game G.

Proof. Easy induction. □

3 Method of Bonet, Pitassi and Raz

3.1 Formulation

The proof of Bonet et al. involves the contradiction stating that a graph on n vertices both has an m-clique and is $(m-1)$-colorable. This contradiction is translated to CP by including a description of the clique and the coloring. A further constraint states that any two vertices of the clique are colored differently. There is no explicit reference to the graph itself.

The description of the clique is a *clique-witness* x_{vi}, where $v \in [n]$ and $i \in [m]$. The variable x_{vi} indicates that vertex v is the ith vertex of the clique. The relevant constraints are as follows:

- Some vertex is the ith vertex: $x_{1i} + \cdots + x_{ni} \geq 1$.
- A vertex v cannot be both the ith and the jth vertex, for $i \neq j$: $-x_{vi} - x_{vj} \geq -1$.
- Two vertices $v \neq u$ cannot both be the ith vertex: $-x_{vi} - x_{ui} \geq -1$.

A clique-witness is *legal* if it satisfies all these constraints.

The description of the coloring is a *coclique-witness* y_{vc}, where $v \in [n]$ and $c \in [m-1]$. The variable y_{vc} indicates that vertex v is colored c. The relevant constraints are as follows:

- Vertex v gets some color: $y_{v1} + \cdots + y_{v(m-1)} \geq 1$.
- A vertex v cannot get two colors $c \neq d$: $-y_{vc} - y_{vd} \geq -1$.

A coclique-witness is *legal* if it satisfies all these constraints.

Finally, we have the *edge-constraints*, stating that two vertices in the clique are colored differently: $-x_{vi} - x_{uj} - y_{vc} - y_{uc} \geq -3$. We say that such a constraint involves the edge (v, u).

3.2 Game

Given a proof of the contradiction $0 \geq 1$ out of the constraints previously listed, we construct a game between two players, the clique-player and the coclique-player. We call the game the *clique/coclique game*. We think of the clique-player as being in charge of the clique-witness x, while the coclique-player is in charge of the coclique-witness y. The two players traverse the proof dag from its root down to a leaf which states an edge-constraint. The clique-player wins if the edge involved is in the graph, and otherwise the coclique-player wins.

The players will repeatedly use a communication protocol to decide which way to go. The protocol evaluates the truth value of a hyperplane involving the x variables (held by the clique-player) and y variables (held by the coclique-player). The protocol itself is described below. The proof doesn't depend on the specifics of the protocol, but only on the number of bits exchanged. It must have the property that the completed transcript determines a set of (x, y) pairs for which the the line has the stated truth-value.

Here are the rules of the game:

- The game starts at the root $0 \geq 1$.
- Whenever the game is at an internal line $\ell_1, \ell_2 \vdash \ell$: The two players run the protocol twice to decide the truth values of ℓ_1, ℓ_2. If ℓ_1 is false, the game continues at ℓ_1. Otherwise, the game continues at ℓ_2.
- At each given point in time, there must be some *legal* pair (x, y) which is consistent with the transcript of the protocols for all of ℓ, ℓ_1, ℓ_2 (at the same time).
- When the game reaches an edge-constraint leaf, the clique-player wins if the edge is in the graph, and the coclique-player wins if the edge is not in the graph. When the game reaches any other leaf, the outcome is undefined (we show below that this never happens).

Lemma 3.1. *Whenever the game is at a line ℓ, each pair (x, y) consistent with the transcript of the protocol for ℓ falsifies ℓ.*

Proof. The proof is by induction. The claim is vacuously true at the root. Suppose it is true for an internal line ℓ which is derived from ℓ_1, ℓ_2. The protocols for ℓ, ℓ_1, ℓ_2 are consistent with some pair (x, y). This pair (x, y) falsifies ℓ by induction, and so it must falsify either ℓ_1 or ℓ_2. The rules of the game force the next line to be one which is falsified by (x, y). □

Corollary 3.2. *The game terminates at an edge-constraint leaf.*

Proof. By the lemma, the leaf is falsified by some legal pair (x, y). All other constraints are satisfied by legal pairs. □

Using Lemma 3.1, we can determine who wins the game.

Lemma 3.3. *If the graph contains an m-clique, the clique-player has a winning strategy. If the graph is $(m-1)$-colorable, the coclique-player has a winning strategy.*

Proof. Suppose the graph contains an m-clique. The clique-player fixes some clique-witness x for an m-clique in the graph. Her strategy is to always use x in the various protocols. Lemma 3.1 shows that the game must reach an edge-constraint which is falsified for some y. An edge-constraint can only be falsified by (x, y) if x states that the edge is in the clique. Hence the clique-player wins.

The proof of the dual case is similar. □

Finally, we convert this game into a game as defined in Section 2.

Lemma 3.4. *Let G be a dag whose nodes are tuples (ℓ, h, h'), where:*

- *ℓ is a line in the proof.*

- *h is a transcript of the protocol for deciding the truth value of ℓ.*
- *h' is a partial transcript of an ongoing protocol for deciding the truth value of the two lines implying ℓ.*
- *h, h' are consistent with some legal pair (x, y).*

A node is labeled $\vee$ if the next player to speak is the clique-player. It is labeled $\wedge$ if the next player to speak is the coclique-player. Further nodes correspond to all edge-constraints in the proof. An edge-constraint involving (u, v) is labeled by the edge e_{uv}. The nodes are combined into a dag by simulating the clique/coclique game.

The $\vee$-player has a winning strategy for G iff the clique-player has a winning strategy for the clique/coclique game. The $\wedge$-player has a winning strategy for G iff the coclique-player has a winning strategy for the clique/coclique game.

Proof. Easy induction. Local consistency is enforced by "remembering" the previous history h while running the protocol for determining the next line. □

Everything is now in place. The proof is completed by presenting the communication protocol.

Theorem 3.5. *A Semantic CP refutation of the clique/coclique contradiction requires $2^{\Omega(\sqrt[3]{n})}$ lines, given that all coefficients are bounded in magnitude by $M = 2^{o(\sqrt[3]{n})}$.*

Proof. If all coefficients are bounded in magnitude by M, there is a communication protocol for deciding whether a line is true which requires $O(\log(nM))$ bits. Given a line $\sum_i a_i x_i + \sum_j b_j y_j \geq c$, the x-player sends $\sum_i a_i x_i$, and the y-player sends $\sum_j b_j y_j$.

Given a proof with L lines, the construction in Lemma 3.4 results in a monotone circuit of size LnM which distinguishes cliques from cocliques. The theorem follows since such a circuit must contain $2^{\Omega(\sqrt[3]{n})}$ gates. □

4 Method of Krajíček

4.1 Formulation

Krajíček's formulation differs from that of Bonet et al. by explicitly including variables e_{uv} for the graph. The edge-constraints are replaced by constraints stating that the graph contains the clique x, and that y is a legal coloring.

The clique-constraints state that if two vertices u, v are in the clique, then they're connected: $e_{uv} - x_{ui} - x_{vj} \geq -1$.

The coclique-constraints state that two vertices u, v colored by the same color c are not connected: $-e_{uv} - y_{uc} - y_{vc} \geq -2$.

4.2 Game

Given a proof of $0 \geq 1$, we define a game between the clique-player and the coclique-player, along the lines of the game defined in Section 3.2. This time the game is complicated by the fact that there are graph-variables.

We say that a line ℓ is clique-falsifiable by (x, y) if there's a graph containing the clique described by x that falsifies ℓ. A line is coclique-falsified by (x, y) if the coclique (complete $(m-1)$-partite graph) described by y falsifies ℓ.

The game proceeds as follows:

- The game starts at the root $0 \geq 1$.
- Whenever the game is at an internal line $\ell_1, \ell_2 \vdash \ell$: The two players run protocols to determine whether each of ℓ_1, ℓ_2 is clique-falsifiable and coclique-falsified (in total, four protocols).
- If one of ℓ_1, ℓ_2 is coclique-falsified but not clique-falsifiable, a further protocol determines some edge which is in the clique but not in the coclique. The clique-player wins if the edge is in the graph, the coclique-player wins if the edge is not in the graph.
- Otherwise, the game continues with ℓ_1 if it is coclique-falsified, and with ℓ_2 otherwise.
- At each given point in time, there must be some legal pair (x, y) consistent with the protocols for ℓ, ℓ_1, ℓ_2.
- If the game ever reaches a clique-constraint or a coclique-constraint, the corresponding edge determines the winner. If it reaches any other leaf, the result is undefined.

The intuition is that we want to test the truth value of the lines under two assignments of the graph variables: clique and coclique. As long as the outcome is the same, we can proceed as in Section 3.2. If the outcome is are different, we can find an edge which differs between the clique and coclique, and let it determine the outcome of the game. In order to get a monotone circuit, we must guarantee that this edge belong to the clique and not belong to the coclique. We ensure that by being more flexible with the definition of truth value.

The proof that the construction works is similar to the proof in Section 3.2, along the following lines.

Lemma 4.1. *Whenever the game is at a line ℓ, each pair (x, y) consistent with the transcript of the protocol for ℓ renders ℓ clique-falsifiable and coclique-falsified.*

Corollary 4.2. *If the game terminates at a leaf, the leaf is either a clique-constraint or a coclique-constraint.*

Lemma 4.3. *If the graph contains an m-clique, the clique-player has a winning strategy. If the graph is $(m-1)$-colorable, the coclique-player has a winning strategy.*

Lemma 4.4. *The clique/coclique game can be converted into a dag-game with an overhead of* $2^{O(C)}$*, where* C *is the communication complexity of the relevant protocols.*

Theorem 4.5. *A Semantic CP refutation of the clique/coclique contradiction requires* $2^{\Omega(\sqrt[3]{n})}$ *lines, given that all coefficients are bounded in magnitude by* $M = 2^{o(\sqrt[3]{n})}$.

Proof. We have to show how to determine if a line is clique-falsifiable and coclique-falsified, and how to find a distinguishing edge in case a line is coclique-falsified but not clique-falsifiable.

By communicating all parts involving x and y, the problem reduces to consideration of lines of the form $\sum_i a_i e_i \geq b$. Divide the indices (which correspond to edges) into two parts $I^{\pm}$ according to sign, so that the line becomes

$$\sum_{i \in I_+} a_i e_i + \sum_{i \in I_-} a_i e_i \geq b.$$

Such a line is clique-falsifiable if there is an $e \geq e^x$ that falsifies it, where e^x is the clique represented by x. Clearly the minimal value that the left-hand side attains is when $e_i = 1$ for $i \in I^-$, and $e_i = e_i^x$ for $i \in I^+$. Hence it is enough for the clique-player to send $\sum_{i \in I^+} a_i e_i^x$.

The line is coclique-falsified if e^y falsifies it, where e^y is the coclique represented by y. This can be determined by having the coclique-player send $\sum_{i \in I} a_i e_i^y$.

Finally, suppose the line is coclique-falsifiable but not clique-falsifiable. Therefore

$$\sum_{i \in I^+} a_i e_i^x + \sum_{i \in I^-} a_i \cdot 1 \geq b,$$
$$\sum_{i \in I^+} a_i e_i^y + \sum_{i \in I^-} a_i e_i^y < b.$$

These inequalities imply that $\sum_{i \in I^+} a_i(e_i^x - e_i^y) > 0$. Using binary search on the interval I_+, the two players can determine an edge i such that $e_i^x > e_i^y$. This involves exchanging $\log n \log(nM)$ bits. □

References

[1] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7:1–22, 1987.

[2] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. *Journal of Symbolic Logic*, 62:708–728, 1997.

[3] Samuel R. Buss and Peter Clote. Cutting planes, connectivity, and threshold logic. *Archive for Mathematical Logic*, 35:33–62, 1996.

[4] Johan Håstad. On the size of weights for threshold gates. *SIAM Journal on Discrete Mathematics*, 7(3):484–492, 1994.

[5] Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *Journal of Symbolic Logic*, 62(2):457–486, 1997.

[6] Saburo Muroga, Iwao Toda, and Satoru Takasu. Theory of majority switching elements. *Journal of the Franklin Institute*, 271:376–418, 1961.

[7] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, 1997.