A separation between semantic and syntactic cutting planes

Yuval Filmus
University of Toronto
yuvalf@cs.toronto.edu

Massimo Lauria
KTH Royal Institute of Technology
lauria@kth.se

March 31, 2013

Abstract

In this note we argue that semantic cutting planes refutations are stronger than syntactic ones. In particular, we give a formula for which any refutation in syntactic cutting planes requires exponential length, while there is a polynomial length refutation in semantic cutting planes. This means that syntactic cutting planes does not p-simulate (nor simulate) semantic cutting planes.

We also give a pair of incompatible cutting planes lines which require exponential length to be refuted in syntactic cutting planes.

1 Introduction

Cutting planes is the second most popular proof system after resolution. It has been introduced as a technique of solving (linear) integer programs (see [Gom58, Chv73]). The idea is to compute the optimum of the program as if it were a linear program. If the optimum is achieved at a fractional point, it is possible to deduce an inequality which can be "rounded" in order to remove that point from the set of feasible solutions.

Another way to describe the rounding rule is as follows: if the inequality $\sum_i a_i x_i \leq A$ is valid and all a_i are integers divisible by c, then any integer solution would also satisfy $\sum_i \frac{a_i}{c} x_i \leq \lfloor \frac{A}{c} \rfloor$. The latter inequality is not necessarily satisfied by fractional solutions.

Cutting planes was later proposed as a proof system in [CCT87], indeed it is possible to view the previous optimization process as a sequence of inferences: a new inequality is either a positive combination or a rounding of previously derived inequalities. Cutting planes can prove the false inequality "$0 \leq -1$" from any unsatisfiable integer program with a finite sequence of deductions.

Studying the length of such proofs is a way of studying the running time of integer programming solvers based on the rounding rule. Unfortunately this seems to be difficult, and all known lower bounds are either for relatively artificial formulas or for proofs of restricted forms (e.g. when the numeric coefficients are small [BPR97] or the proof is tree-like [IPU94]). Indeed the only lower bound known for *unrestricted* cutting planes refutations is due to Pudlák [Pud97], and it based on efficient interpolation techniques. This strongly limits the type of formulas for which lower bounds are provable. For further information about cutting planes refutations and the notion of rank (also called Chvátal rank) we refer the reader to [Juk12, Chapter 19].

It is interesting that the lower bound for tree-like cutting planes mentioned before holds for any kind of deduction rule, no matter how strong. Indeed we may consider the stronger proof system *semantic cutting planes* for which the deduction rule is the following: from any two linear inequalities with integer coefficients $a^T x \leq \alpha$ and $b^T x \leq \beta$ it is possible to deduce any $d^T x \leq \delta$ which is a consequence over $\{0,1\}$ assignments. This system it is clearly as strong as *syntactic* cutting planes, and the main contribution of this paper is to exhibit a formula for which semantic cutting planes has a short refutation while syntactic cutting planes has none.

Theorem (Main Theorem)**.** *There is an unsatisfiable CNF in n variables which requires syntactic cutting planes refutations of length $2^{\Omega((n/\log n)^{1/3})}$, but has a polynomial size semantic cutting planes refutation using only one application of semantic inference.*

1.1 Proof strategy

The plan is to consider a formula of the form $\mathit{Clique}(C, x) \wedge \mathit{Color}(x, \chi)$. $\mathit{Clique}(C, x)$ claims that the graph encoded by variables in x contains a clique of k vertices, and $\mathit{Color}(x, \chi)$ claims that the graph has a vertex coloring using $k-1$ colors. Clearly $\mathit{Clique}(C, x) \wedge \mathit{Color}(x, \chi)$ is unsatisfiable. We claim that for suitable values of k, any refutation in syntactic cutting planes must be long (this will follow almost directly from [Pud97]), while there is a short refutation in semantic cutting planes.

The separating formula is going to be defined in several steps. We need to find the right balance of hardness (for syntactic cutting planes) and easiness (for semantic cutting planes). A key step is to write the formula as a set of linear equations over $\{0,1\}$. Now we list the many steps of the construction, most of which are rather simple. The most important steps are the construction of formulas F and T. In the end, the CNF which separates syntactic and semantic cutting planes is formula U.

Formula	type	Description	Reference
O	CNF	original hard formula for CP	See [Pud97]
F	CNF	extension variables	Section 3
S	integer inequalities	the cutting planes encoding of F	Section 3
T	integer equations	adds $\{0,1\}$ slack variables to S	Section 3.1
U	CNF	CNF encoding of T	Section 3.2
Z	integer equation	equivalent to T	Section 5

2 Preliminaries

Given a set of linear inequalities of the form

$$\sum_{i=1}^{n} a_i x_i \leq b \tag{1}$$

where the coefficients a_i are integers, we want to know whether the polytope generated by the inequalities contains any solution in $\{0,1\}^n$. Standards tools from linear programming only tell us whether the system has a solution in $\mathbb{R}^n$ or not. Deciding the presence of a boolean solution is NP-complete, so unless $\mathsf{P} = \mathsf{NP}$ there is no efficient way of doing that.

2.1 Cutting planes proof systems

The *cutting planes* proof systems are inference systems for integer programming. If the system of linear inequalities is incompatible in $\{0,1\}^n$ then it is possible to deduce the contradictory inequality $0 \leq -1$.

Consider a set of m inequalities on n variables $x_1 \ldots x_n$ of the form

$$\sum_{i=1}^{n} a_{r,i} x_i \leq b_r,$$

where $a_{r,i} \in \mathbb{Z}$ and $1 \leq r \leq m$. A cutting planes inference is a sequence of proof lines (i.e. linear inequalities)

$$L_1, L_2, \ldots, L_l$$

such that each L_i is either one of the initial m inequalities, one of the axioms $x_i \leq 1$, $x_i \geq 0$, or it is obtained from previous inequalities by applying one inference rule. The nature of the inference rules is the main topic of this paper.

In the *semantic cutting planes* proof system, we can use any inference of the form

$$\frac{L_{j_1} \qquad L_{j_2}}{L_{j_3}} \qquad \text{where } j_3 > \max\{j_1, j_2\} \tag{2}$$

such that L_{j_3} is true on every point in $\{0,1\}^n$ in which L_{j_1} and L_{j_2} are true. Notice that this rule is very powerful, and there is no efficient way to witness the soundness of the inference, unless $\mathsf{NP} = \mathsf{coNP}$. To see that consider the NP-complete problem *subset sum*: given a linear equation $\sum_i a_i x_i = A$, decide whether there is a solution of the equation in which $x_i \in \{0,1\}$ for all i. Assuming that the equation has no such solution, in semantic cutting planes it is possible to have the inference:

$$\frac{\sum_i a_i x_i \leq A \qquad \sum_i -a_i x_i \leq -A}{0 \leq -1}, \tag{3}$$

so either $\mathsf{NP} = \mathsf{coNP}$ or there is no short witness for this inference. This means that semantic cutting planes is not a proof system in the sense of Cook and Reckhow [CR79], because there is no polynomial time algorithm which verifies the proof.

In *syntactic cutting planes* proofs, the inference rules have a particular structure that allows checking their correctness in polynomial time. More precisely, we allow the following inference rules:

$$\text{(Sum)} \qquad \frac{\sum_i a_i x_i \leq A \qquad \sum_i b_i x_i \leq B}{\sum_i (\alpha a_i + \beta b_i) x_i \leq \alpha A + \beta B} \qquad \text{for } \alpha, \beta \in \mathbb{N}, \tag{4}$$

$$\text{(Division)} \qquad \frac{\sum_i a_i x_i \leq A}{\sum_i \frac{a_i}{c} x_i \leq \lfloor \frac{A}{c} \rfloor} \qquad \text{when } c \text{ divides all } a_i. \tag{5}$$

The inference rule described in (5) only holds for integer values of x_i, so it may cut away unwanted fractional solutions.

The *length* of a cutting planes proof is the number of lines (i.e. inference steps) in it. It is important to stress that the coefficients appearing in the refutation may carry a lot of information, so the size of their bit representation must be taken into account. Fortunately, it has been proven by [BC96] that any syntactic cutting planes refutation can be transformed into another one in which the coefficients are at most exponential in the number of variables. Thus each coefficient can be represented with a linear number of bits. For semantic cutting planes, we can use a more general argument: every threshold function over $\{0,1\}^n$ can be represented as a linear inequality with coefficients of bit length $O(n \log n)$ [MTT61].

2.2 Syntactic simulation of semantic inferences

It turns out that the rules of syntactic cutting planes are sufficient to prove unsatisfiability for an arbitrary unsatisfiable system of equations. In particular, syntactic cutting planes simulates semantic cutting planes, and this follows immediately from [CCH89] and [ES99].

Theorem (Theorem 6.2 in [CCH89], rephrased)**.** *Let $P \subseteq [0,1]^n$, $P \neq \emptyset$ be a polytope in the 0/1-cube with at least one integer point which is defined by integer linear inequalities. Let $cx \leq \delta$ be an integer inequality of Chvátal rank at most d relative to P. Then there is a cutting planes proof of $cx \leq \delta$ from the inequalities defining P, of length at most $(n^{d+1} - 1)/(n-1)$.*

Theorem (Theorem 4.6 in [ES99], rephrased)**.** *Let $P \subseteq [0,1]^n$, $P \neq \emptyset$ be a nonempty polytope in the 0/1-cube, and let $cx \leq \delta$ be an inequality which is valid for all integer points in P, with $c \in \mathbb{Z}^n$. The Chvátal rank of $cx \leq \delta$ with respect to P is at most $||c||_1$.*

Corollary 1. *Any semantic cutting planes inference rule (even with an unbounded number of premises) can be simulated in syntactic cutting planes with a proof whose size depends only on the number of variables and on the magnitude of the coefficients of the target inequality.*

The simulation obtained using the Corollary 1 is general and thus very inefficient. The main result of this paper shows that in general this simulation is not possible in polynomial size.

2.3 Using equations in cutting planes

In this paper, we consider an extension of (both syntactic and semantic) cutting planes which both equations and inequalities. This will make some upper bounds easier to explain. In the new systems, a proof line can also have the form $a^T x = b$, where a^T is an integer vector. The coefficient of an equation in an application of the Sum rule is not restricted to be positive.

$$\text{(Eq+Eq)} \qquad \frac{\sum_i a_i x_i = A \qquad \sum_i b_i x_i = B}{\sum_i (\alpha a_i + \beta b_i) x_i = \alpha A + \beta B} \qquad \text{for } \alpha, \beta \in \mathbb{Z}; \tag{6}$$

$$\text{(Eq+In)} \qquad \frac{\sum_i a_i x_i = A \qquad \sum_i b_i x_i \leq B}{\sum_i (\alpha a_i + \beta b_i) x_i \leq \alpha A + \beta B} \qquad \text{for } \alpha \in \mathbb{Z} \text{ and } \beta \in \mathbb{N}. \tag{7}$$

The division rule can be extended to equations as well. Notice that either the constant term is integral after the division, or the proof line is already contradictory.

$$\text{(Div)} \qquad \frac{\sum_i a_i x_i = A}{\sum_i \frac{a_i}{c} x_i = \lfloor \frac{A}{c} \rfloor} \qquad \text{when } c \text{ divides all } a_i \text{ and } A; \tag{8}$$

$$\text{(Frac)} \qquad \frac{\sum_i a_i x_i = A}{0 \leq -1} \qquad \text{when } c \text{ divides all } a_i \text{ and } c \text{ does not divide } A. \tag{9}$$

Regular cutting planes can p-simulate this extended version with equations. Each equation is represented by two opposite inequalities. It is a simple exercise to show how to simulate the new rules. As an example, consider the linear combination of two proof lines $ax \leq b$ with coefficient $A > 0$ and $cx = d$ with coefficient $-C$ and $C > 0$: the result $(Aa - Cc)x \leq (Ab - Cd)$ is the sum of $ax \leq b$ and $-cx \leq -d$ with positive coefficients A and C respectively. Such a sum is a rule in standard cutting planes. To simulate Frac consider $ax = A$ with $c \mid a$ and $c \nmid A$. From $ax \leq A$ and $-ax \leq -A$, the division rule gives $\frac{a}{c} x \leq \left\lfloor \frac{A}{c} \right\rfloor$ and $-\frac{a}{c} x \leq \left\lfloor -\frac{A}{c} \right\rfloor$, respectively. Their sum gives $0 \leq \left\lfloor -\frac{A}{c} \right\rfloor + \left\lfloor \frac{A}{c} \right\rfloor$. Since $c \nmid A$, the last inequality is $0 \leq -1$.

3 The Clique vs Coloring formula

A graph G with a clique of size k cannot be colored using $k-1$ colors. Thus a CNF that claims that a graph simultaneously has a clique of size k and can be colored using $k-1$ colors is unsatisfiable. We encode a graph on n vertices using variables $x_{u,v}$ for each $\{u,v\} \in \binom{[n]}{2}$, and two mappings $C\colon [k] \to [n]$ and $\chi\colon [n] \to [k-1]$ to encode the clique and the coloring, respectively. The mapping C is encoded with variables $C_{i,v}$ for $i \in [k]$ and $v \in [n]$, and the mapping χ is encoded with variables $\chi_{v,c}$ for $v \in [n]$ and $c \in [k-1]$. The basic formula F that we consider is the conjunction of two CNFs $\mathit{Clique}(C,x)$ and $\mathit{Coloring}(x,\chi)$. The formula F is a variant of a simpler formula O that has been used in [Pud97]: in F we include some extension variables to represent the conjunction of two clique variables or two coloring variables. Here we define F directly, with its extension variables:

$$C_{(i,j),(u,v)} = C_{i,u} \wedge C_{j,v} \tag{10}$$

$$\chi_{(u,v),c} = \chi_{u,c} \wedge \chi_{v,c}. \tag{11}$$

Formula F: $\mathit{Clique}(C,x)$ is composed of the following propositions, easily written in clausal form:

$$\bigvee_{v\in[n]} C_{i,v} \qquad \text{for every } i \in [k]; \tag{12}$$

$$\neg C_{i,u} \vee \neg C_{i,v} \qquad \text{for all } i \in [k] \text{ and for all } u \neq v \in [n]; \tag{13}$$

$$\neg C_{i,v} \vee \neg C_{j,v} \qquad \text{for all } i \neq j \in [k] \text{ and for all } v \in [n]; \tag{14}$$

$$\neg C_{i,u} \vee \neg C_{j,v} \vee C_{(i,j),(u,v)} \qquad \text{for all } i \neq j \in [k] \text{ and for all } u \neq v \in [n]; \tag{15}$$

$$C_{i,u} \vee \neg C_{(i,j),(u,v)} \qquad \text{for all } i \neq j \in [k] \text{ and for all } u \neq v \in [n]; \tag{16}$$

$$C_{j,v} \vee \neg C_{(i,j),(u,v)} \qquad \text{for all } i \neq j \in [k] \text{ and for all } u \neq v \in [n]; \tag{17}$$

$$\neg C_{(i,j),(u,v)} \vee x_{u,v} \qquad \text{for all } i \neq j \in [k] \text{ and for all } u \neq v \in [n]. \tag{18}$$

Clauses (12)(13)(14) claim that the relation encoded by the C variables is an injective function, i.e. identify a set of k distinct vertices. Clauses (15)(16)(17) represent equation (10) in clausal form. Finally, clauses (18) claim that the chosen vertices form a clique.

$\mathit{Coloring}(x,\chi)$ is composed of the following clauses:

$$\bigvee_{c\in[k-1]} \chi_{v,c} \qquad \text{for every } v \in [n]; \tag{19}$$

$$\neg\chi_{v,c} \vee \neg\chi_{v,d} \qquad \text{for all } v \in [n] \text{ and for all } c \neq d \in [k-1]; \tag{20}$$

$$\neg\chi_{u,c} \vee \neg\chi_{v,c} \vee \chi_{(u,v),c} \qquad \text{for all } u \neq v \in [n] \text{ and for all } c \in [k-1]; \tag{21}$$

$$\chi_{u,c} \vee \neg\chi_{(u,v),c} \qquad \text{for all } u \neq v \in [n] \text{ and for all } c \in [k-1]; \tag{22}$$

$$\chi_{v,c} \vee \neg\chi_{(u,v),c} \qquad \text{for all } u \neq v \in [n] \text{ and for all } c \in [k-1]; \tag{23}$$

$$\neg\chi_{(u,v),c} \vee \neg x_{u,v} \qquad \text{for all } u \neq v \in [n] \text{ and for all } c \in [k-1]. \tag{24}$$

Clauses (19)(20) claim that the relation encoded in χ is a function from the set of vertices to the set of colors. Clauses (21)(22)(23) represent equation (11) in clausal form. Finally, clauses (24) claim that the coloring is legal (i.e. do not color two adjacent vertices with the same color).

Formula S: We now encode (12)–(24) in the form of integer inequalities. Observe that the linear inequalities which encode the clauses in (12)–(13) and in (19)–(20) can be written as linear equations (25) and (31), respectively. Such equations can be efficiently inferred from the original inequalities (see Claim 10 in the appendix), so for our purposes we can just write them in this compact form.

The integer program corresponding to $\mathit{Clique}(C, x)$ is:

$$\begin{aligned}
\sum_{v\in[n]} C_{i,v} &= 1 && \text{for every } i \in [k]; && (25)\\
C_{i,v} + C_{j,v} &\le 1 && \text{for all } i \ne j \in [k] \text{ and for all } v \in [n]; && (26)\\
C_{i,u} + C_{j,v} - C_{(i,j),(u,v)} &\le 1 && \text{for all } i \ne j \in [k] \text{ and for all } u \ne v \in [n]; && (27)\\
C_{(i,j),(u,v)} - C_{i,u} &\le 0 && \text{for all } i \ne j \in [k] \text{ and for all } u \ne v \in [n]; && (28)\\
C_{(i,j),(u,v)} - C_{j,v} &\le 0 && \text{for all } i \ne j \in [k] \text{ and for all } u \ne v \in [n]; && (29)\\
C_{(i,j),(u,v)} &\le x_{u,v} && \text{for all } i \ne j \in [k] \text{ and for all } u \ne v \in [n]. && (30)
\end{aligned}$$

The integer program corresponding to $\mathit{Coloring}(x, \chi)$ is:

$$\begin{aligned}
\sum_{c\in[k-1]} \chi_{v,c} &= 1 && \text{for every } v \in [n]; && (31)\\
\chi_{u,c} + \chi_{v,c} - \chi_{(u,v),c} &\le 1 && \text{for all } u \ne v \in [n] \text{ and for all } c \in [k-1]; && (32)\\
\chi_{(u,v),c} - \chi_{u,c} &\le 0 && \text{for all } u \ne v \in [n] \text{ and for all } c \in [k-1]; && (33)\\
\chi_{(u,v),c} - \chi_{v,c} &\le 0 && \text{for all } u \ne v \in [n] \text{ and for all } c \in [k-1]; && (34)\\
\chi_{(u,v),c} &\le 1 - x_{u,v} && \text{for all } u \ne v \in [n] \text{ and for all } c \in [k-1]. && (35)
\end{aligned}$$

We refer to the entire integer program as S.

3.1 Adding $\{0, 1\}$ slack variables

We need to modify the formula in order to make it easy for semantic cutting planes but still hard for the syntactic system. We plan to reduce to a subset sum problem, so it seems a good idea to turn as many inequalities as possible into integer equations with variables in $\{0, 1\}$. The plan is to add explicit slack variables to the integer inequalities. Since we are in cutting planes, we need these slack variables to be in $\{0, 1\}$, otherwise the proof system cannot manipulate them correctly. The reason we use extension variables in the formulation of *Clique* and *Coloring* is that we need to enforce all resulting inequalities to have slack $\{0, 1\}$ when they are satisfied. Indeed it is the case for inequalities (26)–(30) and (32)–(35) that the slack is always either 0 or 1 in any satisfying assignment.

Formula T: To define the new integer program T, consider an arbitrary indexing I of the inequalities in (26)–(30) or (32)–(35). For each inequality

$$l_i \equiv a^T z \le b, \qquad i \in I, \tag{36}$$

we add a new variable σ_i and we substitute l_i with an equation

$$a^T z + \sigma_i = b. \tag{37}$$

Now we have three objects: formula F is the original CNF, formula S is the corresponding set of inequalities, and formula T is the set of equations after the addition of slack variables. We now argue that the slack variables do not make T easier than S for syntactic cutting planes.

Proposition 2. *Let $P = P_0 \cup \{ax \leq b\}$ be an integer program where $P \vdash a^T x \geq b - 1$ with a syntactic proof of length L. Consider the alternative integer program*

$$P' = P_0 \cup \{a^T x + \sigma = b\},$$

where σ does not appear in P_0. In syntactic cutting planes, the lengths of the shortest refutations of P' and P differ at most by a additive term $O(L)$.

Proof. Consider a refutation of $P \vdash 0 \leq -1$. We want to get a refutation $P' \vdash 0 \leq -1$ of similar length. The only missing axiom in $P \setminus P'$ is $a^T x \leq b$, which can be derived from $a^T x + \sigma = b$ and $-\sigma \leq 0$ which are axioms in F'.

In the opposite direction we start with a refutation $P' \vdash 0 \leq -1$ and we consider a substitution $\sigma \mapsto b - a^T x$ applied to its lines. After this substitution we get a refutation of P.

Axioms not mentioning σ stay the same. The substitution in the remaining axioms is

$$\begin{aligned} a^T x + \sigma = b \quad & \mapsto \quad 0 = 0; \\ -\sigma \leq 0 \quad & \mapsto \quad a^T x \leq b; \\ \sigma \leq 1 \quad & \mapsto \quad b - 1 \leq a^T x; \end{aligned}$$

and each of these formulas can be inferred in at most L steps by hypothesis. After the substitution the sum of two lines and the product by a scalar remain correct inferences step. For the division step, consider $\alpha \in \mathbb{N}$ and a proof line $\alpha r x + \alpha s \sigma \leq t$.

$$\alpha r x + \alpha s \sigma \leq t \quad \mapsto \quad \alpha r x + \alpha s(b - a^T x) \leq t \quad \equiv \quad \alpha(r - s a^T)x \leq t - \alpha s b. \tag{38}$$

For the consequents of the inference, we get

$$r x + s\sigma \leq \left\lfloor \frac{t}{\alpha} \right\rfloor \quad \mapsto \quad r x + s(b - a^T x) \leq \left\lfloor \frac{t}{\alpha} \right\rfloor \quad \equiv \quad (r - s a^T)x \leq \left\lfloor \frac{t}{\alpha} \right\rfloor - sb. \tag{39}$$

Since b is integer, $\left\lfloor \frac{t - \alpha s b}{\alpha} \right\rfloor = \left\lfloor \frac{t}{\alpha} \right\rfloor - sb$, and so substitution after rounding is the same as rounding after substitution. □

Using Proposition 2 we remove the slack variables of T one by one, to get in the end a refutation of S.

Corollary 3. *Consider a syntactic cutting plane refutation of T of length L_T. There is a syntactic cutting plane refutation of F of length $L_F \leq L_T + n^{O(1)}$.*

Proof. Take a refutation of T. Observe that T uses equations (25) and (31) as axioms, which are not cutting planes encodings of axioms from F. Nevertheless, using Claim 10 we can prove (25) and (31) in polynomial length from the cutting planes encodings of (12), (13), (19), and (20).

To eliminate the $O(|F|)$ slack variables from the proof we use Proposition 2. For each substitution $\sigma \mapsto b - ax$ we can deduce $b - 1 \leq ax$ from F in length $O(1)$, as the structure of the formula F shows. Indeed, the needed inequalities for all but (27) and (32) follow directly from the boolean axioms, and the former follow from (28) and (33), respectively.

The result is a refutation of F of length $L_F = L_T + O(|F|)$. □

3.2 The separating CNF

We want to separate syntactic and semantic cutting planes with a CNF formula, and so far we have just designed an integer program T. Later we will show that this program has a short semantic cutting planes refutation but requires a large syntactic one. Now we want to design a CNF with the same characteristic, that we call formula U.

Formula U: The plan is to have a CNF which encodes exactly T as a CNF. U is made by

- clauses (12), (13);
- the trivial CNF encoding of the equations obtained adding slack variables to the inequalities (26), (27), (28), (29), (30);
- clauses (19), (20);
- the trivial CNF encoding of the equations obtained adding slack variables to the inequalities, (32), (33), (34), (35).

The "brute force encoding" of an equation of l variables is the CNF made as follow: for each boolean assignment falsifying the equation there is a clause with l literals which is falsified by that assignment. So the CNF has at most 2^l clauses. The formula U has length $O(|F|)$ since the equations encoded as CNFs have a constant number of variables each. We observe that T and U mutually deduce each other.

Proposition 4. *Formula T and the cutting planes encoding of U mutually deduce each other with a polynomial length syntactic cutting planes derivation.*

Proof. Fix a clique index i; clause (12) is encoded as $\sum_v C_{i,v} \geq 1$, and using Claim 10 we know that clauses (13) deduce $\sum_v C_{i,v} \leq 1$. Thus from clauses (12)–(13) of formula U it is possible to deduce equation (25) of formula T, and vice versa. A similar relation holds between clauses (19)–(20) and equation (31).

For the rest of the clauses, we observe that the remaining clauses of U are the CNF encodings of the constant size equations of T. Each of them has at most four variable, including slack variables. By definition, the inequalities in U and the corresponding equation in T are semantically equivalent. Using Corollary 1 we can derive one from the other using a syntactic derivation of constant size, but in order to have a self contained proof we show how to do such deductions for this particular case. Another advantage of our proof over using Corollary 1 is that the proofs we construct are shorter than the ones given by the corollary: for equations involving ℓ variables, our proofs consist of $O(2^\ell \ell)$ lines, while the corollary only gives proofs of length $O(\ell^\ell)$.

Consider any such equation $E = 0$ in T. For any assignment α for which $E(\alpha) \neq 0$, formula U has a clause which is only falsified by α. Denote its linear encoding as $C_\alpha \geq 1$.

Notice that the coefficients of $E = 0$ are in $\{-1, 0, 1\}$: without loss of generality, we can assume that $E = 0$ is of the form $\sum_{i=1}^{\ell} x_i = K$ for some $K \in \mathbb{Z}$ and some variables $x_1 \ldots x_\ell$. This is because it is always possible to apply a variable substitution $x \leftrightarrow 1 - x'$ to any variable x (where x' is a new variable), preserving the soundness of syntactic cutting planes refutations.

We actually show that the inequality $\sum_{i=1}^{\ell} x_i \geq K$ and the set of inequalities $C_\alpha \geq 1$ for which $\sum_{i=1}^{\ell} \alpha(x_i) < K$ mutually deduce each other. The dual relation, between inequality $\sum_{i=1}^{\ell} x_i \leq K$ and $C_\alpha \geq 1$ such that $\sum_{i=1}^{\ell} \alpha(x_i) > K$, is omitted since it has the same proof.

From T to U: Let $|\alpha| = \{i : \alpha(x_i) = 1\}$. We want to prove every inequality $C_\alpha \geq 1$ such that $|\alpha| < K$, using $\sum_{i=1}^{l} x_i \geq K$ and standard variable axioms. Add to $\sum_{i=1}^{\ell} x_i \geq K$ the inequality $-2x_j \geq -2$ for every j such that $\alpha(x_j) = 1$. The result it $\sum_{i:\alpha(x_i)=0} x_i - \sum_{i:\alpha(x_i)=1} x_i \geq K - 2|\alpha|$. By adding the tautology $0 \geq |\alpha| - K + 1$[1] we obtain $\sum_{i:\alpha(x_i)=0} x_i - \sum_{i:\alpha(x_i)=1} x_i \geq 1 - |\alpha|$, which is exactly $C_\alpha \geq 1$.

This strategy uses $O(\ell)$ lines to deduce each C_α. In total, deducing all C_α requires $O(2^\ell \ell)$ lines.

From U to T: We want to deduce $\sum_{i=1}^{\ell} x_i \geq K$ from the clauses $C_\alpha \geq 1$ for all α such that $\sum_{i=1}^{\ell} \alpha(x_i) < K$.

By induction on $0 \leq k \leq K$ we will prove $\sum_{i=1}^{\ell} x_i \geq k$. For $k = 0$ the inequality follows from variable axioms. For $k = 1$, consider the clause $C_\alpha \geq 1$ for $\alpha(x_i) = 0$: this is exactly the inequality $\sum_{i=1}^{\ell} x_i \geq 1$.

Now prove $\sum_{i=1}^{l} x_i \geq k + 1$ as follows:

$$\sum_{i=1}^{\ell} x_i \geq k \qquad \text{inductive hypothesis.} \tag{40}$$

$$\sum_{i:\alpha(x_i)=0} x_i - \sum_{i:\alpha(x_i)=1} x_i \geq 1 - k \qquad C_\alpha \text{ for } |\alpha| = k. \tag{41}$$

$$\sum_{i \notin S} x_i \geq 1 \qquad \text{for any } S \subseteq \binom{[\ell]}{k}. \tag{42}$$

$$\sum_{i \notin S} x_i \geq 1 + a \qquad \text{for any } S \subseteq \binom{[\ell]}{k-a}. \tag{43}$$

$$\sum_{i=1}^{\ell} x_i \geq k + 1 \qquad \text{which is inequality (43) for } a = k. \tag{44}$$

Equation (40) is the inductive hypothesis, (41) are the rewriting of clauses $C_\alpha \geq 1$ for any α with exactly k ones: such clauses are available since $k < K$. Summing (40) and (41) and dividing by two we get equations (42), with $S = \{i : \alpha(x_i) = 1\}$. These inequalities say that however you remove k variables from the sum, at least one of the remaining variables is positive. Inequalities (43) say that however $k - a$ variables are removed from the sum, at least $1 + a$ variables are set to 1, so they are a generalization of (42) for $0 < a \leq k$. They are proved by induction on a: given any S of size $k - a$, sum the inequalities corresponding to all its supersets of size $k - (a - 1)$. The resulting inequality is

$$\sum_{i \notin S} (\ell - k + a - 1) x_i \geq (\ell - k + a) a,$$

which after division by $\ell - k + a - 1$ gives (43) for the set S. Going all the way to $a = k$ gives (44). The length of the resulting proof is $O(2^\ell \ell)$ lines.

□

[1] This tautology is an integer combination of the axioms $x \geq 0$ and $-x \geq -1$ for any variable x.

4 A lower bound for syntactic cutting planes

We show that the integer program F requires exponential length refutations in syntactic cutting planes. This in turn implies that U requires exponential length refutations as well.

One of the most useful tools in proof complexity is the interpolation theorem, which gives a way to extract computation from a proof. In some cases, from a small proof it is possible extract a computation which is too efficient to be possible, and indeed to prove a lower bound on proof size.

Theorem 5 (Interpolation theorem [Pud97])**.** *Let $F = A(x,z) \vee B(y,z)$ be an unsatisfiable CNF such that variables z appear in A only positively. If F has a cutting planes refutation of size S then there is a* monotone real circuit *$\mathcal{C}$ of size $O(S)$ such that*

$$\mathcal{C}(v) = \begin{cases} 0 & \textit{when } A(x,v) \textit{ is unsatisfiable;} \\ 1 & \textit{otherwise;} \end{cases} \qquad \textit{for each assignment } v \textit{ to } z\textit{-variables.} \tag{45}$$

This theorem has been used to prove the first lower bound for syntactic cutting planes. Notice that the original lower bound in [Pud97] studies a formula O which is very similar to F but does not use extension variables. Now we go all the way to the CNF U in order to get the lower bound there.

Corollary 6. *Consider the CNF U, for graphs of n vertices and for $k = \Theta\left(\left(\frac{n}{\log n}\right)^{2/3}\right)$. Any syntactic cutting planes refutation of U has length at least $2^{\Omega((n/\log n)^{1/3})}$.*

Proof. Let L_U, L_T, L_F be the lengths of the shortest syntactic refutations of U, T and F, respectively. By Proposition 4 it holds that $L_T \leq L_U + n^{O(1)}$, and by Corollary 3 it holds that $L_F \leq L_T + n^{O(1)}$.

To conclude the proof we need to lower bound L_F. This follows from Theorem 5: any syntactic cutting plane refutation of F of length L_F implies the existence of an interpolant circuit $\mathcal{C}(x)$ of size $O(L_F)$ which separates graphs with cliques of size k from graphs with $k-1$ colorings, and is also a monotone real circuit. For $k = \Theta\left(\left(\frac{n}{\log n}\right)^{2/3}\right)$, such a circuit requires size $2^{\Omega((n/\log n)^{1/3})}$, as shown in [Pud97]. □

5 Upper bound for semantic cutting planes

In this section we show that the formula U has a short proof in semantic cutting planes. We already argued in Proposition 4 that formulas U and T have the same proof complexity in syntactic and semantic cutting planes. So it is sufficient to show that T has a short refutation in semantic cutting planes. We already observed in Section 2.1 that the semantic rule can refute unsatisfiable instances of subset sum. The structure of T is that of a set of integer equations with no common $\{0,1\}$ solutions. We show that it is possible to pack two or more integer equations into a single equation which is satisfiable if and only if the former are.

Proposition 7. *Consider a sequence of m integer equations $\sum_i a_{j,i}x = A_j$ for $j \in [m]$. If there is no $\{0,1\}$ assignment which satisfies all equations then there is a polynomial size refutation of these equations which uses the semantic inference rule exactly once.*

Proof. Given an equation $\sum_i a_{j,i}x_i = A_j$ let $M \geq 2$ be the smallest number satisfying $M > |A_j| + \sum_i |a_{j,i}|$ for all j. From the given equations we can deduce the following equation, formula Z:

$$\sum_{j=1}^{m} \left(\sum_i a_{j,i}x_i - A_j \right) M^{j-1} = 0 \tag{Z}$$

using only integer scalar multiplications and sums. Notice that the coefficients in these equation have length polynomial in the length of the original ones. Such an equation is satisfiable if and only if all the m initial equations are, as we show below. By assumption then this equation is false for every boolean assignment, so we can deduce $0 \leq -1$ in a single step of semantic refutation.

It remains to show that formula Z is satisfiable if and only if the original equations are. If the original equations are satisfiable, then the same truth assignment satisfies formula Z. Now suppose that some truth assignment satisfies formula Z, and let $s_j = \sum_i a_{j,i}x_i - A_j$. By construction, $|s_j| < M$ and $\sum_{j=1}^{m} s_j M^{j-1} = 0$. Since $|s_j| < M$,

$$\left| \sum_{j=1}^{m-1} s_j M^{j-1} \right| \leq (M-1) \sum_{j=1}^{m-1} M^{j-1} < M^{m-1}.$$

This shows that $s_m = 0$, and similarly we can deduce that all other equations are satisfied. □

By applying the previous proposition to T, and knowing that formulas T and U deduce each other in semantic cutting planes, we get the following upper bound.

Corollary 8. *Formula U has a polynomial size refutation in semantic cutting planes, using a single application of the semantic rule.*

As another corollary, we get that formula Z is contradictory but hard to refute in syntactic cutting planes.

Corollary 9. *There are two incompatible inequalities in n variables which require length $2^{\Omega((n/\log n)^{1/3})}$ to be refuted in syntactic cutting planes.*

6 Open problems

We leave several open problems. The most pressing one is to prove lower bounds for semantic cutting planes. It is natural to ask whether semantic cutting planes has some form of interpolation, even in the case of large coefficients. The second open problem that stands after this note is whether it is possible for syntactic cutting planes to p-simulate (or at least simulate) semantic cutting planes with small coefficients. Notice that subset sum with polynomial coefficients (in the number of variables) can be solved with easily by dynamic programming.

Acknowledgments

The research was done while the first author was visiting KTH Royal Institute of Technology. The first author received funding from the European Union's Seventh Framework Programme (FP7/2007–2013) under grant agreement no 238381. The second author was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no 279611.

References

[BC96] Samuel R. Buss and Peter Clote. Cutting planes, connectivity, and threshold logic. *Archive for Mathematical Logic*, 35(1):33–62, 1996. 3

[BPR97] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. *The Journal of Symbolic Logic*, 62(3):pp. 708–728, 1997. 1

[CCH89] Vašek Chvátal, William Cook, and M. Hartmann. On cutting-plane proofs in combinatorial optimization. *Linear Algebra and its Applications*, 114:455–499, 1989. 4

[CCT87] William Cook, Collette R. Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987. 1

[Chv73] Vašek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973. 1

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979. 3

[ES99] Friederich Eisenbrand and Andreas S. Schulz. Bounds on the Chvátal rank of polytopes in the 0/1-cube. *Integer Programming and Combinatorial Optimization*, pages 137–150, 1999. 4

[Gom58] Ralpha E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958. 1

[IPU94] Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *Logic in Computer Science, 1994. LICS'94. Proceedings., Symposium on*, pages 220–228. IEEE, 1994. 1

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*. Springer-Verlag, 2012. 1

[MTT61] S. Muroga, I. Toda, and S. Takasu. Theory of majority decision elements. *Journal of the Franklin Institute*, 271(5):376–418, 1961. 3

[Pud97] Pavel Pudlák. Lower bounds for Resolution and Cutting Plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, 1997. 1, 2, 5, 10

[Rho09] Martin Rhodes. On the Chvátal rank of the pigeonhole principle. *Theoretical Computer Science*, 410(27-29):2774–2778, 2009. 12

A Appendix

Claim 10 ([Rho09])**.** *Let $x_1, \dots, x_n$ be boolean variables. The inequality $\sum_{i=1}^{n} x_i \leq 1$ has a polynomial length syntactic cutting planes proof from the inequalities $x_i + x_j \leq 1$ for each $1 \leq i < j \leq n$. The opposite direction also holds.*

Proof. For the forward direction, we prove by induction on $b - a$ that $\sum_{i=a}^{b} x_i \leq 1$. The cases $b - a \leq 2$ follow directly from the axioms. For $b - a > 2$, consider the sum of $\sum_{i=a}^{b-1} x_i \leq 1$, $\sum_{i=a+1}^{b} x_i \leq 1$ and $x_a + x_b \leq 1$, which is $\sum_{i=a}^{b} 2x_i \leq 3$. A division step concludes the proof.

The opposite direction is easier: given $\sum_{i=1}^{n} x_i \leq 1$ and two indices a and b, we can sum $-x_i \leq 0$ for each $i \notin \{a, b\}$ to get $x_a + x_b \leq 1$. □