# Feasible interpolation as games

Yuval Filmus

December 2012

**Abstract**

This note is a transcript of a lecture given in Prague on December 1st, 2012. Krajíček [4] and Bonet, Pitassi and Raz [2] have independently described a construction of a monotone boolean circuit out of a refutation proving that two NP-sets are disjoint. Also known as monotone feasible interpolation, this construction can be used to prove lower bounds on systems such as resolution and cutting planes.

Bonet, Pitassi and Raz constructed the circuit directly, while Krajíček appealed to a theorem of Razborov [7]. We give an alternate account, in terms of games.

## 1 General setting

Consider two disjoint NP sets $A(x,y)$ and $B(x,z)$, where furthermore the set $A$ is upwards-closed and the set $B$ is downwards-closed. For example, $x$ may code the vertices of a graph, $y$ may code a $k$-clique, and $z$ may code a $k-1$-coloring of the vertices of the graph. The formula $A(x,y)$ states that $y$ represents a clique in the graph, and the formula $B(x,z)$ states that $z$ represents a coloring of the graph.

More explicitly, let $n$ be an integer. For each $i < j \in [n]$ there is a variable $x(i,j)$, which codes the existence or non-existence of the edge $(i,j)$. The variables $y(i,t)$, where $i \in [n]$ and $t \in [k]$, are the graph of a function $[k] \rightarrowtail [n]$ representing a clique in the graph. The variables $z(i,c)$, where $i \in [n]$ and $c \in [k-1]$, are the graph of a function $[n] \to [k-1]$ describing a $k-1$-coloring of the graph. The formula $A(x,y)$ is the conjunction of the following clauses:

| | |
|---|---|
| $\bigvee_{i=1}^{n} y(i,t)$ | some vertex is the $t$th vertex |
| $\neg y(i,t) \vee \neg y(j,t)$ | vertices $i,j$ cannot both be the $t$th vertex |
| $\neg y(i,t) \vee \neg y(i,s)$ | vertex $i$ cannot be both the $t$th vertex and the $s$th vertex |
| $y(i,t) \wedge y(j,s) \Rightarrow x(i,j)$ | if vertices $i,j$ are in the clique then $i,j$ are connected |

The formula $B(x,z)$ is the conjunction of the following clauses:

| | |
|---|---|
| $\bigvee_{c=1}^{k-1} z(i,c)$ | vertex $i$ gets some color |
| $\neg z(i,c) \vee \neg z(i,d)$ | vertex $i$ is not colored both $c$ and $d$ |
| $x(i,j) \Rightarrow \neg z(i,c) \vee \neg z(j,c)$ | if vertices $i,j$ are connected then not both $i,j$ are colored $c$ |

We call this the *explicit version* of the contradiction $A(x,y) \wedge B(x,z)$. Our construction become simpler if we eliminate the common variables $x$ by cutting over them. For $A, B$ described above, the new axioms are

$$y(i,t) \wedge y(j,s) \Rightarrow \neg z(i,c) \vee \neg z(j,c).$$

These axioms state that vertices in the clique are colored by distinct colors. We call this the *implicit version*.

A third version was suggested by Neil Thapen. We use two different copies for the $x$ variables, and postulate that one implies the other. In the running example, this amounts to

$$y(i,t) \wedge y(j,s) \Rightarrow x(i,j)$$
$$w(i,j) \Rightarrow \neg z(i,c) \vee \neg z(j,c)$$
$$x(i,j) \Rightarrow w(i,j)$$

We call this *Neil's version*. Since every proof of the explicit version can be converted to a proof of Neil's version (by resolving all $w$ variables, say), a lower bound on Neil's version implies a lower bound on the explicit version.

Our goal is to show that in certain proof systems, refuting the contradiction $A \wedge B$ is hard. We will do this by showing that each such proof translates to a monotone circuit which separates $A$ and $B$. The circuit gets as input the common variables $x$, outputs True when $A(x,y)$ is satisfiable, and False when $B(x,z)$ is satisfiable. If both $A, B$ are unsatisfiable, then we don't claim anything about the output of the circuit. Alon and Boppana [1] and others have proved exponential lower bounds on the size of such circuits, for an appropriate choice of $k$.

## 2 Monotone circuits as games

Instead of constructing a monotone circuit directly, we will describe a game between two players $A$ and $B$. Formally, a game is given by the following:

- A set $S$ of states.
- An initial state $s_0 \in S$.
- For each non-terminal state $s \in S$, one of the players A, B is designated. That player can choose either of two states $s_1, s_2 \in S$.
- Terminal states are labeled with either 0, 1 or an edge $x(i,j)$.

The play of the game is as follows. The players start at the initial state $s_0$. At each point, the player whose turn it is to speak chooses which direction to go among the two options given. When a terminal state is reached, if it is annotated with 0 then player B wins, if it is annotated with 1 then player A wins, and otherwise the edge $x(i,j)$ is revealed and determines the winner (A if the edge exists, B if it doesn't exist).

To convert a game into a monotone circuit, replace all A-states with $\vee$ gates and all B-states with $\wedge$ gates. The circuit outputs 1 if A has a winning strategy, and it outputs 0 if B has a winning strategy. This is the so-called game semantics of monotone circuits.

## 3 Resolution (implicit version)

We start by considering resolution refutations of $A(y) \wedge B(z)$ (the version without the $x$ variables). The two players play a game on the proof dag. They start at the final contradiction $\emptyset$, and work their way toward the axioms. Whenever a variable $y(i,t)$ is resolved, player A chooses which way to go. Whenever a variable $z(i,c)$ is resolved, player B chooses which way to go. Eventually they will reach an axiom. If it is an A-axiom, then player B wins. If it is a B-axiom, then player A wins. Otherwise, it is an axiom of the form

$$y(i,t) \wedge y(j,s) \Rightarrow \neg z(i,c) \vee \neg z(j,c).$$

The corresponding edge $x(i,j)$ is revealed and determines the winner: player A if the edge exists, player B if the edge doesn't exist.

Suppose that the graph has a $k$-clique. We show that player A has a winning strategy. Player A chooses some clique in the graph and codes it as an assignment to $y$. She will play the game, all the time maintaining

the invariant that the current clause is falsified by $y$ and by *some* $z$. Whenever the game reaches some clause $C \vee D$ derived from $C \vee y(i,t)$ and $D \vee \neg y(i,t)$, if $y(i,t)$ is true then she chooses to continue with $D \vee \neg y(i,t)$, and if $y(i,t)$ is false then she chooses to continue with $C \vee y(i,t)$.

Player A's strategy ensures that her invariant is maintained. It is clearly true at the beginning of the game (at the empty clause). Consider a step $C \vee y(i,t), D \vee \neg y(i,t) \vdash C \vee D$, and suppose that $y(i,t)$ is true. We know that $C \vee D$ is falsified by $y$ and by some $z$. On the other hand, $C \vee y(i,t)$ is clearly true. So $D \vee \neg y(i,t)$ must be falsified by $y$ and the same $z$, and that's exactly the clause that she chooses.

Finally, consider a step $C \vee z(i,c), D \vee \neg z(i,c) \vdash C \vee D$. We know that $C \vee D$ is falsified by $y$ and by some $z$. The value of $z(i,c)$ is immaterial here, since $z(i,c)$ does not appear in $C \vee D$. Therefore whichever way player B chooses to go, there is as assignment of $z(i,c)$ that falsifies the clause. For example, if player B goes to $C \vee z(i,c)$, then we set $z(i,c)$ to false. Since $D \vee \neg z(i,c)$ is true, $C \vee z(i,c)$ must be false.

The game ends at an axiom which is falsified by $y$ and some $z$. This axiom cannot be an A-axiom since $y$ was chosen so that all these axioms are satisfied. If it is a B-axiom then player A wins. The final case is when it's an axiom $y(i,t) \wedge y(j,s) \Rightarrow \neg z(i,c) \vee \neg z(j,c)$. Since this clause is false, its premise $y(i,t) \wedge y(j,s)$ must be true, i.e. $i, j$ are both in the clique. But that implies that the edge $(i,j)$ is in the graph, and so player A wins.

Exactly the same argument shows that if the graph is $k-1$-colorable then player B has a winning strategy. Therefore we can convert our game into a monotone circuit which separates $A$ and $B$. The size of the circuit is exactly the number of lines in the proof. Therefore an exponential lower bound on the size of the circuit translates to an exponential lower bound on the size of the proof.

# 4 Resolution (explicit version)

We turn next to the original version of the contradiction $A(x,y) \wedge B(x,z)$. The game is defined similarly with two differences: we need to handle resolving of $x$ variables, and the axioms are different.

The two players start at the final contradiction, and work their way toward the axioms. Whenever a $y$ variable is resolved, player A chooses which way to go. Whenever a $z$ variable is resolved, player B chooses which way to go. The game becomes more interesting when an $x$ variable is resolved, say $x(i,j)$:

- Player A is asked for the value of $x(i,j)$. If she says that $x(i,j)$ is false, the game continues at the clause containing $x(i,j)$.
- Otherwise, player B is asked for the value of $x(i,j)$. If he agrees with player A that $x(i,j)$ is true, the game continues at the clause containing $\neg x(i,j)$.
- Otherwise, there is a showdown. The edge $x(i,j)$ is revealed and determines the winner.

If the game does not end in a showdown, then eventually an axiom will be reached. If it is an A-axiom then player B wins, if it is a B-axiom then player A wins.

Suppose that the graph has a $k$-clique. The winning strategy for player A is the same as before. She picks some particular encoding $y$ of a clique in the graph. Whenever a variable $y(i,s)$ is resolved, she chooses the branch containing $y(i,s)$ if $y(i,s)$ is false, and the branch containing $\neg y(i,s)$ if $y(i,s)$ is true. When asked for the value of the variable $x(i,j)$, she always answers truthfully.

Player A maintains the same invariant that the current clause is falsified by $y$ and some $z$. This is clearly true at the beginning. Resolving over $y$ and $z$ variables is handled as before. Consider next a clause $C \vee D$ derived from $C \vee x(i,j)$ and $D \vee \neg x(i,j)$. Player A reports the value of $x(i,j)$ truthfully, so if she is disputed by player B then she wins (this can only happen if $x(i,j)$ is true). Otherwise, the game proceeds to a clause in which the literal corresponding to $x(i,j)$ is falsified; the other clause is satisfied, so this clause must be false.

If the game doesn't end in a showdown, it ends at some axiom. This axiom cannot be an A-axiom since all A-axioms actually hold. So it must be a B-axiom, and player A wins.

The argument for player B is a bit more delicate. Suppose that the graph is $k-1$-colorable. Player B picks some legal coloring $z$. His winning strategy is the same as the one described for player A. However,

the invariant he maintains is different: the current clause is falsified by $z$ and some $y$, *and some subgraph* $x'$ *of the real graph* (denoted $x' \leq x$ for short). He needs this extra degree of freedom to handle the case in which player A lies about an edge which exists in the graph.

As before, the invariant is maintained whenever $y$ and $z$ variables are resolved. Consider next a clause $C \vee D$ derived from $C \vee x(i,j)$ and $D \vee \neg x(i,j)$. Recall that we assume that $C \vee D$ is satisfied by $z$ and by some $y$ and $x' \leq x$. There are several different cases. If player A claims that $x(i,j)$ is true when it's in fact false, player B calls her bluff and immediately wins in the ensuing showdown. If player A is truthful, then the invariant will be maintained if we set $x'(i,j) = x(i,j)$.

The remaining case is when $x(i,j)$ is true but player A claims that it's false. The rules of the game don't allow player B to call her bluff this time around (if the game allowed that possibility, the resulting circuit wouldn't be monotone). Player B sets $x'(i,j)$ to false, and then the invariant is maintained: $C \vee D$ is still false, $D \vee \neg x'(i,j)$ is true, and so $C \vee x'(i,j)$, where the game continues, must be false.

If the game doesn't end in a showdown, an axiom is reached. This axiom is falsified by $z$ and by some $y$ and $x' \leq x$. Since $z$ is syntactically correct, we can't possible reach a B-only axiom. We also can't reach an axiom $x(i,j) \Rightarrow \neg z(i,c) \vee \neg z(j,c)$. Indeed, this axiom can only be falsified if the succedent is false, which means that $i, j$ get the same color. Since $z$ is a legal coloring, the edge $(i,j)$ is not in the graph, and so $x(i,j)$ is false. This forces $x'(i,j)$ to be false, so the axiom actually holds. We conclude that the game ends at an A-axiom, and so B wins.

The new circuit contains at most two gates per each line in the proof, and so the same exponential lower bound still holds. Note that we only used the fact that $B(x,z)$ is closed downwards. A simple modification (changing the order of speakers) handles the case when we only know that $A(x,y)$ is closed upwards.

# 5 Resolution (Neil's version)

Finally, we consider Neil's version of the contradiction, $A(x,y) \wedge B(w,z) \wedge x \Rightarrow w$. The game is very similar to the one defined for the implicit version. Contrary to the situation with the explicit version, now we don't have to consider $x$ variables in any special way.

The two players start at the final contradiction, and work their way toward the axioms. Whenever an $x$ or a $y$ variable is resolved, player A chooses which way to go. Whenever a $z$ or a $w$ variable is resolved, player B chooses which way to go. Eventually the players will reach an axiom. If it's an A-axiom, then player B wins. If it's a B-axiom, then player A wins. If it's the axiom $x(i,j) \Rightarrow w(i,j)$, then the edge $(i,j)$ is revealed and determines the winner.

Suppose that the graph has a $k$-clique. Player A fixes some encoding $y$ of the clique, and uses the same clique as $x$. The winning strategy of player A is the same as before. Whenever a variable $y(i,s)$ is resolved, if the variable is true then she chooses the $\neg y(i,s)$ branch, and if it is false that she chooses the $y(i,s)$ branch. She uses the same rule for $x$ variables.

Player A maintains the same invariant as before, namely the current clause is falsified by $x, y$ and by some $w, z$. This is clearly true at the beginning, and resolving is handled as for the implicit version. The game cannot end at an A-axiom, since all of these are true. If it ends at a B-axiom, then player A wins. Finally, suppose it ends at an axiom $x(i,j) \Rightarrow w(i,j)$. Since the axiom is false, $x(i,j)$ must be true. This means that the edge $(i,j)$ belongs to the clique encoded by $y$, which is actually in the graph, and so player A wins.

We mentioned in the introduction that Neil's version simulates the explicit version. We can use the construction described here directly for the explicit version $A(x,y) \wedge B(x,z)$. We give the $x$ variables to one of the players, say player A, and sort it out with player B's version at the end.

The two players start at the final contradiction, and work their way toward the axioms. Whenever an $x$ or a $y$ variable is resolved, player A chooses which way to go. Whenever a $z$ variable is resolved, player B chooses which way to go. The game eventually reaches an axiom. If it is an A-axiom, then player B wins. If it is a B-only axiom, then player A wins. If it is a B-axiom of the form $x(i,j) \Rightarrow \varphi(z)$, the edge $(i,j)$ is revealed and determines the winner.

Suppose that the graph has a $k$-clique. The winning strategy for player A is the same as above. She picks some particular encoding $y$ of a clique in the graph, and uses that clique for $x$. Whenever an $x$ or a $y$ variable is resolved, she chooses the branch in which the corresponding literal is falsified. She maintains the invariant that the current line is falsified by $x, y$ and by some $z$. Eventually the game reaches an axiom. It cannot be an A-axiom since they are all true. If it's a B-only axiom then she wins. In the remaining case, it is an axiom $x(i,j) \Rightarrow \varphi(z)$. Since the axiom is false, $x(i,j)$ must be true. Since the clique coded in $x$ is actually in the graph, player A wins when the edge $(i,j)$ is revealed.

The argument for player B is very similar, though the players are not symmetric here. The winning strategy for player B is the same, though he need only worry about $z$ variables. He maintains the invariant that the current line is falsified by $z$ and by some $x, y$. Eventually the game reaches an axiom. If it's an A-axiom, he wins. It cannot be a B-only axiom since they are all true. So it must be an axiom $x(i,j) \Rightarrow \varphi(z)$. Since the axiom is false, $\varphi(z)$ must be false. Since $z$ codes a legal coloring, this means that the edge $(i,j)$ is actually not in the graph, and so player B wins.

# 6 Communication complexity

In order to handle more general proof systems, we will need to borrow concepts from communication complexity. The setting of communication complexity involves a cooperative game between two players A and B trying to compute a function $f(y,z)$ (which is going to be the truth value of a proof line in our case). The difficulty is that player A only has access to $y$, and player B only to $z$. The players communicate with each other, until finally any outside observer can conclude the value of $f(y,z)$.

More formally, a communication protocol for $f(y,z)$ is given by the following:

- A set $S$ of states.
- An initial state $s_0 \in S$.
- For each non-terminal state $s \in S$, one of the players A, B is designated. That player sends a bit to the other player, which results in the protocol transitioning to one of two states $s_1, s_2 \in S$.
- Terminal states are labeled with the purported value of the function (a boolean value).
- For each $y$ there is a winning strategy $\sigma_y$ for player A which tells her which bit to send in any of her states. For each $z$ there is a winning strategy $\sigma_z$ for player B which tells him which bit to send in any of his states. If the two players follow the strategies $\sigma_y, \sigma_z$ then the terminal state they reach is labeled $f(y,z)$.

An outside observer can record the transcript $h$ of the protocol, which is the sequence of bits sent. The transcript is partial until the players reach a terminal state, at which point it becomes complete.

We will be interested in the size (number of states) in the communication protocol. If the communication complexity (maximal number of bits sent) is $cc$, then the size of the protocol is at most $2^{cc}$.

# 7 General construction (implicit version)

We generalize the construction of Section 3 to arbitrary proof systems. As a running example, we will consider semantic cutting planes (CP). Lines in CP are linear inequalities in the variables $y, z$ (or $x, y, z$ for the explicit version) with integer coefficients. A derivation $\ell_1, \ell_2 \vdash \ell$ is valid if any 0/1 assignment that satisfies $\ell_1, \ell_2$ also satisfies $\ell$.

We can use the following trivial communication protocol to decide whether a line $a_y \cdot y + a_z \cdot z \geq b$ is satisfied: player A sends $a_y \cdot y$, and player B sends $a_z \cdot z$. The size of this protocol is at most $\|a_y\|_1 \|a_z\|_1$, so the smaller the coefficients are, the better[1].

[1] Håstad [3] showed that some CP lines require coefficients as large as $2^{O(N \log N)}$, where $N$ is the number of variables (in our case, $n(2k-1)$). However, the players can always make do by exchanging at most $2N$ bits: $a_y \cdot y$ and $a_z \cdot z$ each have at most $2^N$ possible values, and it is enough to send an index into a sorted list of these values.

We can now describe the game. The game starts at the final contradiction $0 \geq 1$, and proceeds toward the axioms. When at a line $\ell$ derived from $\ell_1, \ell_2$, the two players run the communication protocol to decide which of $\ell_1, \ell_2$ is false (under an important restriction described below). We show below that at least one of the lines will always be false, and the game continues with one of the false lines ($\ell_1$ for definiteness if both are false). The game eventually reaches an axiom. If it is an A-axiom, player B wins. If it is a B-axiom, player A wins. Otherwise it's an axiom corresponding to some $x(i,j)$. The corresponding edge is revealed and determines the winner.

When the players go through the protocols for $\ell_1, \ell_2$, their actions are constrained so that at all times, there must be some assignment $y, z$ which is consistent with the completed transcript for $\ell$ and the ongoing transcripts for $\ell_1, \ell_2$. This local consistency condition ensures that the protocol can always choose a line which is falsified: since there are some $y, z$ consistent with the transcripts of the protocols for $\ell, \ell_1, \ell_2$ and the protocol declared $\ell$ to be false, one of $\ell_1, \ell_2$ must be declared false.

The constraints might force the players to send a specific bit at some point. Note, however, that if a player (say player A) is following her winning strategy $\sigma_y$ then she never gets stuck, that is she can always play along this strategy, as long as she sticks to the same $y$ for all of $\ell, \ell_1, \ell_2$.

Suppose the graph has a $k$-clique. Player A chooses some encoding $y$ of the clique. She will always use the winning strategy $\sigma_y$ for the current line at stake. This ensures that the game progresses along lines which are falsified by $y$ and by some $z$. When the game reaches an axiom, it can't be an A-axiom, since all of them are true. If it's a B-axiom then she automatically wins. If it's an axiom associated with an edge $x(i,j)$, then the fact that the axiom is falsified shows that the edge is in the graph (as in Section 3), and so she wins again.

Similar arguments apply to player B. So the game can be turned into a monotone circuit separating $A$ from $B$. Each state of the game corresponds to a line $\ell$ in the proof, the completed transcript $h$ for the protocol for $\ell$, and two partial transcripts $h_1, h_2$ for the protocols for $\ell_1, \ell_2$. So if the proof has $L$ lines and each communication protocol has size at most $C$, then the resulting circuit has size at most $LC^3$.

For an arbitrary cutting planes proof, we have $C \leq 2^{O(nk)}$ and so the circuit has size $L2^{O(nk)}$, which is too big to imply any meaningful bound on $L$. In order to get meaningful bounds on the size of the proof, we must constrain all coefficients used in the proof to be small (say, polynomial).

# 8 General construction (explicit version)

Generalizing the construction of Section 4 is more complicated. We put player A in charge of the $x$ variables. Throughout the game, we maintain the invariant that the current line is falsified by $x, y, z$.

When at a line $\ell$ derived from $\ell_1, \ell_2$, we first determine whether $\ell$ is falsified by $x', y, z$ for some $x'$ for which $B(x', z)$ holds (so $z$ is a valid coloring for $x'$). If so, we determine which of $\ell_1, \ell_2$ is falsified, and continue as before. Otherwise, there is a showdown: the players determine a variable $x(i,j)$ which player A claims to be true, but the B-axioms force it to be false (such a variable always exists). The edge is revealed and determines the winner.

If the game reaches an axiom $\ell$, we first determine whether $\ell$ is falsified by $x', y, z$ for some $x'$ for which $B(x', z)$ holds. If so, if it is an A-axiom then player B wins, if it is a B-axiom then player A wins. Otherwise, there is a showdown as before.

Suppose the graph has a $k$-clique. Player A chooses some encoding $y$ of one such clique, and uses the real graph for $x$. If there is ever a showdown she will win, since $x$ corresponds to the real graph. Otherwise, the game eventually ends at an axiom. This axiom cannot be an A-axiom since all of them hold. So the game must end at a B-axiom, and player A wins.

The argument for the case that the graph is $k-1$-colorable is slightly different. Player B chooses some encoding $z$ of a legal coloring. If there is ever a showdown then he will win, since in this case the edge in question is one which is forced by the B-axioms not to be in the graph, and all such edges are indeed missing from the graph since $z$ codes a legal coloring. Otherwise, the game ends at some axiom. Since a showdown was averted, the axiom in question is falsified by $x', y, z$ for some $x'$ for which $B(x', z)$ holds. In particular, it can't be a B-axiom. So the game must end at an A-axiom, and player B wins.

Given a cutting planes line $a_x \cdot x + a_y \cdot y + a_z \cdot z \geq b$, we need to decide whether some $x'$ for which $B(x', z)$ holds falsifies the line. Player B can calculate the minimum value of $a_x \cdot x'$: for any coordinate $(i, j)$ in which $a_x(i, j)$ is positive, he puts $x'(i, j) = 0$, and for any coordinate in which $a_x(i, j)$ is negative, he puts $x'(i, j) = 1$ if $i$ and $j$ are colored differently, and $x'(i, j) = 0$ otherwise. He sends $\min_{x'} a_x \cdot x' + a_z \cdot z$ to player A and she replies with $a_y \cdot y$, at which point their joint mystery can be solved.

If no such $x'$ falsifies the line, we need to find a coordinate $(i, j)$ such that $x(i, j) = 1$ and $i, j$ get the same color. Let $x' = \operatorname{argmin} a_x \cdot x'$. Since $x$ falsifies the line while $x'$ satisfies it, $a_x \cdot x' > a_x \cdot x$. The same must be true if we restrict to the set of indices $I$ corresponding to coordinates $(i, j)$ in which $a_x(i, j)$ is negative (since on the positive coordinates $x'$ is always 0). Using binary search on $I$, the players can determine a single coordinate $(i, j)$ such that $a_x(i, j)x'(i, j) > a_x(i, j)x(i, j)$. Since $a_x(i, j)$ is negative, this implies that $x(i, j) = 1$ whereas $x'(i, j) = 0$.

Concluding, given a proof of size $L$ and coefficients of magnitude at most $M$, we can transform the proof into a monotone circuit of size at most $LO(knM \log(kn))^3$, where the factor $\log(kn)$ corresponds to binary search. This bounds is only marginally worse than the bound we got for the implicit version (for comparison with the notation there, $C = O(knM)$).

# 9 General Construction (Neil's version)

Adapting the construction in Section 7 to Neil's version $A(x, y) \wedge B(w, z) \wedge x \Rightarrow w$ is straightforward.

The game starts at the final contradiction $0 \geq 1$, and progresses down to the axioms. During inference steps, the game is identical to the implicit version, with player A in charge of $x, y$ and player B in charge of $w, z$. Eventually the game reaches an axiom. If it's an A-axiom the player B wins. If it's a B-axiom then player A wins. If it's the axiom $x(i, j) \Rightarrow w(i, j)$ (coded as the line $w(i, j) - x(i, j) \geq 0$), the edge $(i, j)$ is revealed and determines the winner.

Suppose that the graph has a $k$-clique. Player A chooses an encoding $y$ of the clique, and uses the same clique for $x$. She maintains the invariant that the current line is falsified by $x, y$ and by some $w, z$. Eventually the game reaches an axiom. It cannot be an A-axiom since all of these are true. If it's a B-axiom then player A wins. If it's an axiom $x(i, j) \Rightarrow w(i, j)$, then since the axiom is false, $x(i, j)$ must be true, and so the revealed edge $(i, j)$ belongs to a clique which is actually found in the graph.

The same argument holds for player B, and we get the same bounds that we got for the implicit version[2].

As in the case of resolution, we can also apply this construction directly to the explicit version. We put one of the players (say player A) in charge of the $x$ variables, and the game proceeds as before. If the game reaches an A-axiom then player B wins. If it reaches a B-only axiom then player A wins. Otherwise, it reaches an axiom of the form $x(i, j) \Rightarrow \varphi(z)$. The edge $(i, j)$ is revealed and determines the winner.

The argument for Neil's version can be easily adapted to show that the construction just described works for the explicit version.

# 10 Randomized games

The arguments in the previous two sections break for cutting plane proofs with arbitrary coefficients. Pudlák [6] was able to give a different argument which does work for cutting plane proofs with arbitrary coefficients, albeit only for the syntactic version (with predefined rules). His proof constructs a monotone real circuit, and so the lower bound needs to be generalized to such circuits. This still leaves open the case of semantic cutting planes.

It is tempting to replace the long deterministic protocols used to decide whether a CP line is true with Noam Nisan's celebrated $O(\log(nk))$-bit randomized communication protocol [5]. At key points during this communication protocol, coins are tossed publicly and control the flow of the algorithm. The protocol is guaranteed to result in the correct value with good probability, which can be amplified further at the cost of communicating more bits.

[2] Curiously, the same suggestion is found in my own old notes, where it is deemed not to work.

We can adjust our framework to handle randomized games. Our original games had A-states, B-states and terminal states. The new type of game additionally has R-states. Those are states which lead to either of two states $s_1, s_2$, the first with probability $p$, the second with probability $1-p$, for some arbitrary $p$.

We originally translated a game to a circuit by replacing A-states with $\vee$ gates and B-states with $\wedge$ gates. For the new game, we replace A-states with max gates, B-states with min gates and R-states with avg gates: the latter compute a weighted average (with weights $p, 1-p$) of the two subcircuits corresponding to the two successor states.

It is straightforward to show that the circuit outputs the probability (over coin tosses at R-states) that player A wins under optimal play. More formally, a strategy $\sigma_A$ for player A tells her which way to go at each A-state, and a strategy $\sigma_B$ for player B tells him which way to go at each B-state. Let $P(\sigma_A, \sigma_B)$ be the probability that the game reaches a terminal state in which player A wins. Then the value that the circuit computes is

$$\max_{\sigma_A} \min_{\sigma_B} P(\sigma_A, \sigma_B).$$

One would hope that by using the same construction as in the previous sections, we could construct a circuit with the following properties. If the graph contains a $k$-clique, then player A has a strategy that ensures she will win with probability more than $1/2$, whereas if the graph is $k-1$-colorable, then player B has a strategy that ensures he will win with probability more than $1/2$. Putting a threshold gate at the root of the circuit, we get a boolean value and can apply Pudlák's lower bound.

Unfortunately, this idea doesn't work. The problem is that Nisan's protocol (or any other randomized protocol) is designed to help two *cooperating* players compute some function. It guarantees the answer to be correct with high probability only if the two players follow the strategies corresponding to their inputs. In our setting, the two players are *competing*, and so this guarantee is useless.

# 11 Open questions

Two open questions naturally arise. First, is there a way to fix the argument described in the preceding version? Second, is semantic cutting planes actually stronger than syntactic cutting planes?

Regarding the second question, we can focus on the following simple question. Suppose that two lines $\ell_1, \ell_2$ imply a line $\ell$ semantically. Can we drive some line $\ell' \vdash \ell$ syntactically from $\ell_1, \ell_2$ in polynomial length?

Recognizing whether a line $\ell$ follows (semantically) from $\ell_1, \ell_2$ is coNP-complete. Indeed, if $\ell$ does *not* follow from $\ell_1, \ell_2$ then a proof of the fact constitutes a 0/1-assignment that satisfies $\ell_1, \ell_2$ but not $\ell$. So semantic validity is in coNP. In order to show that it is coNP-complete, we use a reduction from subset sum. Given an instance $S = \{s_1, \ldots, s_t\}; A$ of subset sum (the task is to decide whether $A$ is a sum of elements from $S$), we construct the following derivation:

$$\sum s_i x_i \geq A, -\sum s_i x_i \geq -A \vdash 0 \geq 1.$$

The instance of subset sum has no solution if and only if the derivation is valid.

The same argument shows that recognizing validity of an entire cutting planes proof is also coNP-complete.

If syntactic cutting planes p-simulates semantic cutting planes, then every semantic derivation $\ell_1, \ell_2 \vdash 0 \geq 1$ has a polynomial size syntactic cutting planes derivation. Using the reduction from subset sum, this shows that NP=coNP. In other words, unless NP=coNP, syntactic cutting planes does not p-simulate semantic cutting planes.

It would be interesting to find, assuming NP$\neq$coNP, an explicit example of a semantic derivation $\ell_1, \ell_2 \vdash 0 \geq 1$ which is not provable in polynomial size in syntactic cutting planes.

# 12 Sources

The constructions described in Sections 3 and 7 are taken from Bonet, Pitassi and Raz [2]. The constructions described in Sections 4 and 8 are taken from Krajíček [4], which in turn relies on Razborov [7] for the actual

construction of the circuit. The constructions described in Sections 5 and 9 are by Neil Thapen.

The relation between the syntactic and the semantic versions of cutting planes described in Section 11 is by Kaveh Ghasemloo.

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

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

[5] Noam Nisan. The communication complexity of threshold gates. In *Combinatorics, Paul Erdős is Eighty*, pages 301–315, 1993.

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

[7] Alexander A. Razborov. Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic. *Izvestiya of the R.A.N*, 59:201–224, 1995.