On extracting computations from propositional proofs (a survey)

Pavel Pudlák [*]

September 16, 2010

Abstract

This paper describes a project that aims at showing that propositional proofs of certain tautologies in weak proof system give upper bounds on the computational complexity of functions associated with the tautologies. Such bounds can potentially be used to prove (conditional or unconditional) lower bounds on the lengths of proofs of these tautologies and show separations of some weak proof systems. The prototype are the results showing the feasible interpolation property for resolution. In order to prove similar results for systems stronger than resolution one needs to define suitable generalizations of boolean circuits. We will survey the known results concerning this project and sketch in which direction we want to generalize them.

1 Introduction

Proof complexity studies problems about formal systems that are related to similar problems in computational complexity. In particular, some complexity classes can be associated in a natural way with formal systems and one can translate problems about these classes to problems about the formal systems. The formal systems that we have in mind are weak arithmetical theories formalized in predicate logic and proof systems for propositional calculus. Our original reason for studying weak arithmetical theories was to show the unprovability of some open problems in computational complexity theory. Since the attempts to show that, say, $P \neq NP$ is unprovable in Peano Arithmetic completely failed, researchers focused on the study of much weaker theories. Unfortunately we are still unable to show such independence results even for the weakest theory in which polynomial time computations are formalizable. Nevertheless, a number of interesting results have been proven and new proof methods have been introduced.

In computational complexity the most important problems can be reduced to proving lower bounds on the circuit size of boolean functions. Similarly, in proof complexity one can

[*] Institute of Mathematics, Academy of Sciences, Prague, Czech Republic; also supported by the Insitiue for Theoretical Computer Science (project 1M0545) and grant IAA100190902.

reduce problems about unprovability in weak theories of arithmetic to proving lower bounds on the lengths of proofs of tautologies in certain proof systems. In the 1980s the pioneering work in the area of lower bounds on propositional proofs was done by Armin Haken, who proved exponential lower bounds on proofs in Resolution [6], and Miklos Ajtai, who proved superpolynomial lower bounds (later extended to exponential) on proofs in bounded depth Frege systems [1]. In the early 1990s Jan Krajíček introduced a new method for proving lower bounds on propositional proofs, which we now call *feasible interpolation* [8]. This enables one to reduce the task of proving lower bounds on the lengths of propositional proofs to the task of proving lower bounds on the circuit size of boolean functions defined from tautologies. Although the problem of proving nontrivial lower bounds on the size of boolean circuits is a notoriously open problem, this reduction is very useful for the following two reasons. First, one can often show that the reduction gives *monotone* boolean circuits and then one can use the well-known exponential lower bounds on the size of such circuits. Secondly, even when the reduction to monotone boolean circuits is not possible and we only get general circuits, the reduction gives us important information that, for example, can be used to show lower bounds based on some conjectures from computational complexity theory.

The method of feasible interpolation has been successfully applied to obtain exponential lower bounds on the lengths of proofs in Resolution [9] and several other proof systems for propositional logic. However, this method fails for systems that are only a little stronger than resolution, unless some commonly accepted conjectures in computational complexity are false [13, 4, 3]. Therefore we started a project whose aim is to find structures that are more general than boolean circuits and prove for them generalized forms of feasible interpolation. Our motivation is not only to find new ways of proving lower bounds on the lengths of proofs, but also to study a question important per se: *can one extract computational information from any propositional proof?*

There are other approaches to the fundamental question of proving lower bounds on the lengths of propositional proofs, most notably the approach based on *proof complexity generators*, [10, 12]. Proof complexity generators are inspired by the concept of pseudorandom generators and the conjecture is that some pseudorandom generators can actually be used to construct tautologies with no polynomial proofs. However, if this is true than one must find connections between propositional proofs and computational complexity for strong proof systems. It is likely that before such connections are found for strong proof systems, they will be discovered for moderately strong systems. Therefore we focus on such propositional proof systems.

Another approach, also studied by Krajíček, is based on using a different *form* of tautologies that are used in the feasible interpolation theorems [11]. For such tautologies, it is conceivable that the associated computational problem is solvable in polynomial time even for stronger systems. In [11] such a connection is proved for a very special form of tautologies and proofs in bounded depth Frege systems.

Although our project has already yielded some preliminary results, it would be premature to try to describe them in this paper. We will rather focus on describing the results that we want to generalize, hoping that our presentation will be more understandable than the original ones in [17, 9].

2 Preliminaries

2.1 Propositional proof systems

The *Resolution* propositional proof system is a proof system for proving tautologies that are in DNF form. Given a tautology ϕ in DNF form, we take its negation, which is in CNF form, and treat it as a set of disjunctions, which are called *clauses.* A proof of ϕ in Resolution is a proof of contradiction from the clauses. In Resolution we treat clauses as sets of literals; a literal is a propositional variable or negated propositional variable. The single rule used in Resolution is:

$$\frac{\Gamma, p \qquad \Delta, \neg p}{\Gamma, \Delta}.$$

The contradiction, which is the last clause in the proof, is represented by the empty set.

A *Frege system* is any sound and complete propositional proof system that is based on a finite number of rules. No nontrivial lower bounds are known for Frege systems. A depth d Frege system is a Frege system in which only formulas of depth d are allowed. We want to present bounded depth Frege systems as generalizations of Resolution. Thus we will define a depth d Frege system as a *refutation* proof system based on sequents that use formulas of depth at most d. We will only use formulas with negations at variables; so $\neg\phi$ denotes the formula obtained from ϕ by switching conjunctions and disjunctions and switching literals. Defining the depth of a literal to be 1, we get that Resolution is the depth 1 Frege system.

For $d > 1$, the *cut rule* of Resolution is generalized to arbitrary formulas ϕ of depth at most d:

$$\frac{\Gamma, \phi \qquad \Delta, \neg\phi}{\Gamma, \Delta}.$$

Further, we also have *rules for introducing conjunctions and disjunctions:*

$$\frac{\Gamma, \phi \qquad \Delta, \bigwedge A}{\Gamma, \Delta, \phi \wedge \bigwedge A} \qquad \frac{\Gamma, \phi, \bigvee A}{\Gamma, \phi \vee \bigvee A},$$

and the *weakening rule:*

$$\frac{\Gamma}{\Gamma, \phi}.$$

We treat sequents as sets, and conjunctions and disjunctions as set operations in order not to have to introduce structural rules.

2.2 Polynomial search problems

Polynomial search problems, also called *Total Function Nondeterministic Polynomial search problems* and abbreviated by **TFNP,** are given by a binary relation R such that

1. $R(x, y)$ is decidable in deterministic polynomial time;
2. there exists a polynomial p such that for all x and y, if $R(x, y)$, then $|y| \leq p(|x|)$;
3. for every x there exists y such that $R(x, y)$.

Given such a relation and x, the task is to find y such that $R(x, y)$. There is a natural concept of polynomial reduction of one polynomial search problem to another one. Also many natural classes of polynomial search problems have been defined and they play an important role in proof complexity.

We will only mention one of these classes, *Polynomial Local Search,* or **PLS**. An instance of a Polynomial Local Search problem for a given input x is determined by a *search space* S, a *feasibility predicate* $F \subseteq S$, a *neighborhood function* $N : S \to S$, and a *cost function* $c : S \to \mathbb{N}$. The search space is $S = \{0, 1\}^m$, where m is polynomial in $|x|$. The functions N, c and the predicate F are computable in polynomial time. Formally, this means that the predicate F is parametrized by the input x and $F(s, x)$ is a binary relation in **P**, etc. for the other notions. The functions and the predicate should satisfy:

1. $F(\bar{0})$;
2. if $F(s)$, then $F(N(s))$;
3. if $F(s)$ and $N(s) \neq s$, then $c(s) < c(N(s))$.

The task is to find a "local maximum", which is an $s \in S$ such that $s = N(s)$.

2.3 The Karchmer-Wigderson game

Karchmer and Wigderson found a characterization of the circuit depth of boolean functions using communication complexity.

Theorem 1 ([7]) *The minimal depth of a circuit computing a boolean function $f(x_1, \ldots, x_n)$ is equal to the minimal number of bits that two players need to communicate in the worst case in the following game. Player I gets an input u such that $f(u) = 0$ and Player II gets an input v such that $f(v) = 1$. By sending messages, they should determine an index i such that $u_i \neq v_i$.*

This theorem also holds for *partial* boolean functions.

3 Razborov's characterization of circuit complexity

Note that the task in the Karchmer-Wigderson game can be viewed as a communication complexity analogue of search problems. Given u and v such that $f(u) = 0$ and $f(v) = 1$, there always exists an index i such that $u_i \neq v_i$ and the task is to find such an i. In communication complexity theory one speaks about computing *relations*, but the analogy with search problems is more appropriate.

When analogues of the usual complexity concepts are defined in communication complexity, $O(\log n)$ communication bits correspond to polynomial time. Thus in the Karchmer-Wigderson Theorem the boolean functions of $\mathbf{NC_1/poly}$ are characterized by the communication complexity class corresponding to search problems solvable in polynomial time. Razborov came up with the idea to characterize **P/poly**, a probably larger class of functions, by a communication complexity analogue of a probably larger class of polynomial search problems. He showed that such a class of search problems is **PLS**.

To state his theorem we have to translate the definition of **PLS** into a communication complexity problem. Let a partial boolean function $f(x_1, \ldots, x_n)$ be given. Again, Player I gets an input u such that $f(u) = 0$ and Player II gets an input v such that $f(v) = 1$. So the predicate F and the functions N and c will now depend on u and v. But we are not interested in the computational complexity of these functions, only in their communication complexity. Roughly speaking, we want, for every $s \in S$, the communication complexity of computing $F(s, u, v), N(s, u, v)$ and $c(s, u, v)$ to be small.

The goal of the players is again to determine an index i such that $u_i \neq v_i$, but now they want to do it by first computing a local maximum. Therefore, we need another function $p : S \to \{1, 2, \ldots, n\}$ that tells the players the index, given a local maximum s. The function only depends on $s \in S$, thus it does not play any role in defining the complexity of the problem. What however does play an important role is the size of the set of feasible solutions, $\{s \in S;\ F(s, u, v)\}$.

We say that the (f, F, N, c, p) is a **PLS** *communication protocol* if for every u, v such that $f(u) = 0$ and $f(v) = 1$ and every local maximum s (with respect to the parameters u, v), the number $p(s)$ is an index such that $u_{p(s)} \neq v_{p(s)}$.

The complexity of a protocol (f, F, N, c, p) is defined to be the number

$$C = \left| \bigcup_{f(u)=0, f(v)=1} \{s \in S;\ F(s, u, v)\} \right| \cdot 2^{2CC(F,c)+CC(N)},$$

where $2CC(F, c)$ is the maximal communication complexity of computing simultaneously $F(s, u, v)$ and $c(s, u, v)$ and $CC(N)$ is the maximal communication complexity of computing $N(s, u, v)$.

This is a rather technical definition that enables Razborov to state his theorem in a strong form. However, if we were only interested in the communication complexity analogue of **PLS**, we would only require that $|S|$ be of polynomial size, which would correspond to the exponential size of the search space in the usual **PLS**, and that $CC(F), CC(c), CC(N)$ be $O(\log n)$, which would correspond to F, c, N being computable in polynomial time. Then, clearly, the number C would be bounded by a polynomial in n.

Theorem 2 ([17]) *For a given partial boolean function f, the smallest complexity C of* **PLS** *communication protocols (f, F, N, c, p) is, up to a constant factor, equal to the circuit complexity of f.*

We will consider a special case of the theorem that has a more transparent proof. We restrict the above protocols (f, F, N, c, p) as follows.

1. To compute $F(s, u, v)$, the players only need to send one bit to each other independently on each other.
2. For every $s \in S$, either $N(s, u, v) = s$ independently of u, v, or there are two elements $s_0, s_1 \in S$ and one assigned player such that, given u, v, $N(s) \in \{s_0, s_1\}$, and the assigned player knows $N(s)$; thus the player only needs to send one bit to the other player.
3. c only depends on s, not on u, v.

We will call such protocols *restricted* **PLS** *communication protocols.* Essentially, these are protocols in which the players need to send the minimal possible number of bits. Note that condition 1. can be stated more explicitly as follows.

1. There are two predicates $F_I(s, u)$ and $F_{II}(s, v)$ such that $F(s, u, v) \equiv F_I(s, u) \wedge F_{II}(s, v)$.

Lemma 1 *The smallest $|S|$ in restricted protocols for f is equal to the circuit complexity of f.*

We will sketch the proof of this lemma.

1. First, assume a circuit D computing f is given. Define S to be the nodes of the circuit, except that we have to rename the output node of the circuit to $\bar{0}$. Given u, v such that $f(u) = 0$, $f(v) = 1$, the predicate $F(s, u, v)$ is defined to be true if, for the function f_s computed at the gate s, $f_s(u) \neq f_s(v)$. The cost function c is an arbitrary antimonotone function from the DAG of the circuit to natural numbers. Given a node s, if it is an input, then $N(s) = s$, otherwise s_0 and s_1 are its input nodes. The assigned player is Player I if the gate at s is $\wedge$ and Player II if the gate is $\vee$. For an input node s labeled by x_i or $\neg x_i$, $p(s) = i$; otherwise it is defined arbitrarily.

We leave the verification of the properties to the reader.

2. Consider a protocol (f, F, N, c, p). Let $U = \{u;\ f(u) = 0\}$ and $V = \{v;\ f(v) = 1\}$. Let $s \in S$ be feasible for some $u \in U$ and $v \in V$, which means that it satisfies $F(s, u, v)$. The condition 1. concerning F says that the set $\{(u, v) \in U \times V;\ F(s, u, v)\}$ is a combinatorial rectangle $U_s \times V_s$, where $U_s = \{u \in U;\ F_I(s, u)\}$ and $V_s = \{v \in V;\ F_{II}(s, v)\}$. We will construct a circuit whose nodes will be the elements of S such that, for every $s \in S$, the function f_s computed at the node s will satisfy

$$f_s(u) = 0 \text{ and } f_s(v) = 1, \text{ for all } u \in U_s \text{ and } v \in V_s. \qquad (1)$$

If s is such that $N(s) = s$ and $p(s) = i$, then we label s by x_i or $\neg x_i$. If s is feasible for some u, v, then exactly one of the two labels is correct; otherwise we do not care. To see that only one label is correct, suppose that s is feasible for u and v, and $u_i = 0$ and $v_i = 1$.

Then the label should be x_i. It is not possible that s is feasible for some u' and v' such that $u'_i = 1$ and $v'_i = 0$. If it were, we would also have $F(s, u', v)$, because it is equivalent to $F_I(s, u') \wedge F_{II}(s, v)$. But this is impossible, because $u'_i = v_i$.

Now suppose that s is such that $N(s, u, v) \in \{s_0, s_1\}$ is associated with Player I. It may still happen that, for some $i = 0, 1$, $N(s, u, v) = s_i$ for all $(u, v) \in U_s \times V_s$. In such a case we just join s by a wire to s_i and $f_s = f_{s_i}$. Otherwise we label s by $\wedge$ and connect it to s_0 and s_1. The verification that condition (1) is preserved reduces to proving the following two implications:

$$F(s, u, v) \rightarrow F_I(s_0, u) \vee F_I(s_1, u),$$
$$F(s, u, v) \rightarrow F_{II}(s_0, v) \wedge F_{II}(s_1, v).$$

The first implication follows trivially from $F(s, u, v) \rightarrow F(N(s), u, v) \rightarrow F(s_0, u, v) \vee F(s_1, u, v)$. To prove the second one, suppose w.l.o.g. that $N(s, u, v) = s_0$. Since $N(s, u, v)$ only depends on u and since the value s_1 is also possible, there must be some $u' \in U_s$ such that $N(s, u', v) = s_1$. Thus we have $F(s_0, u, v) \wedge F(s_1, u', v)$, whence $F_{II}(s_0, v) \wedge F_{II}(s_1, v)$. ■

Theorem 2 can be now proved from Lemma 1 by showing that general communication protocols can be reduced to the restricted protocols of Lemma 1. For example, if the players need $k > 1$ bits to compute N, we introduce, for every s, $2^k - 1$ new vertices and replace the arrows $s \rightarrow s_0, s \rightarrow s_1$ by a tree. Instead of going directly from s to s_0 or s_1, the players will go to these vertices in k steps. Similarly, we have to replace each vertex by 2^ℓ vertices if the feasibility predicate F needs $\ell > 1$ communication bits to be decided, etc.

4 Feasible interpolation

Suppose $\phi(\bar{x}, \bar{y}) \vee \psi(\bar{x}, \bar{z})$ is a tautology where $\bar{x}$ is the string of propositional variables that ϕ and ψ share and $\bar{y}$ and $\bar{z}$ are disjoint strings. Suppose we substitute a string of truth values $\bar{a}$ for $\bar{x}$. Then the terms in the tautology $\phi(\bar{a}, \bar{y}) \vee \psi(\bar{a}, \bar{z})$ have disjoint sets of propositional variables, hence either $\phi(\bar{a}, \bar{y})$ is a tautology, or $\psi(\bar{a}, \bar{z})$ is a tautology, or both. Thus such tautologies give us a computational problem: given $\bar{a}$, determine which of the two formulas $\phi(\bar{a}, \bar{y})$ or $\psi(\bar{a}, \bar{z})$ is a tautology. In general, this problem is not in **P**, unless **P** = **NP**∩**coNP**. But what if we not only know that $\phi(\bar{x}, \bar{y}) \vee \psi(\bar{x}, \bar{z})$ is a tautology, but also have a proof of it? Krajíček's important discovery is that in some cases we can solve this task if we have a proof. Exactly when this is possible depends on the proof system.

Definition 1 *We say that a propositional proof system P has the feasible interpolation property, if there exists a polynomial time algorithm A such that given a P-proof d of $\phi(\bar{x}, \bar{y}) \vee \psi(\bar{x}, \bar{z})$ and an assignment $\bar{a}$ for the common variables $\bar{x}$,*

1. *if $A(d, \bar{a}) = 0$, then $\phi(\bar{a}, \bar{y})$ is a tautology,*
2. *if $A(d, \bar{a}) = 1$, then $\psi(\bar{a}, \bar{z})$ is a tautology.*

Theorem 3 ([9]) *The Resolution proof system has the feasible interpolation property.*

We will reproduce Krajíček's proof and show how it can be done only using Lemma 1. The basic idea is to use the given refutation d to define a communication search problems in the sense of Razborov and from it to construct a circuit C that satisfies 1. and 2. of the theorem. Since C is constructed in polynomial time from d, the construction gives us the polynomial algorithm A.

Let d be a Resolution derivation of the empty clause from two sets of clauses $\Phi(\bar{x}, \bar{y})$ and $\Psi(\bar{x}, \bar{z})$, where $\bar{x}$ are the common variables of the two sets.

Let f be the partial function defined by:

$$\begin{array}{lll} f(\bar{u}) = 0 & \text{if} & \exists \bar{y} \bigwedge \Phi(\bar{u}, \bar{y}), \\ f(\bar{v}) = 1 & \text{if} & \exists \bar{z} \bigwedge \Psi(\bar{v}, \bar{z}), \end{array}$$

and otherwise undefined. For every $\bar{u}$ such that $f(\bar{u}) = 0$ we choose a $\bar{w}_u$ such that $\bigwedge \Phi(\bar{u}, \bar{y})$ is true, and similarly we define $\bar{t}_v$ for every $\bar{v}$ such that $f(\bar{v}) = 1$.

The search space S is the set of clauses of the proof d. The initial element $\bar{0} \in S$ is the empty clause. The cost function c is the distance from the empty clause.

Given a clause Σ, the feasibility predicate $F(\Sigma, \bar{u}, \bar{v})$ is satisfied, if both assignments $\bar{u}, \bar{w}_u \bar{t}_v$ and $\bar{v}, \bar{w}_u, \bar{t}_v$ falsify Σ.

If Σ is an initial clause, we put $N(\Sigma) = \Sigma$; the definition of $p(\Sigma)$ is irrelevant, because such a Σ is never feasible. Suppose, now, that $\Sigma = \Gamma, \Delta$ and it was derived from clauses $\Sigma_0 = \Gamma, p$ and $\Sigma_1 = \Delta, \neg p$. We distinguish several cases according to to which set of variables p belongs.

1. If $p = y_j$, then we assign Σ to Player I and $N(\Sigma, \bar{u}, \bar{v}) = \Sigma_{(w_u)_j}$.
2. If $p = z_l$, then we assign Σ to Player II and $N(\Sigma, \bar{u}, \bar{v}) = \Sigma_{(t_v)_l}$.
3. If $p = x_i$, then the players tell each other the values u_i and v_i. Then
   (a) if $u_i = v_i$, then $N(\Sigma, \bar{u}, \bar{v}) = \Sigma_{u_i}$;
   (b) otherwise $N(\Sigma, \bar{u}, \bar{v}) = \Sigma$ and $p(\Sigma) = i$.

This is a protocol in the sense of Razborov, so one can apply his Theorem 2. However, one can also easily reduce it to the simpler Lemma 1. Note that the above protocol almost satisfies the restrictions needed in Lemma 1. Only in 3. the players have to send two bits instead of one. This can be rectified as follows.

Add to the search space S also all x_i and $\neg x_i$, if they are not already present. Put $N(x_i) = x_i$, $N(\neg x_i) = \neg x_i$ and $p(x_i) = p(\neg x_i) = i$. Define $F(x_i, \bar{u}, \bar{v})$ to be true if $u_i = 0$ and $v_i = 1$ and dually for $\neg x_i$.

For $\Sigma, \Sigma_0, \Sigma_1$ as above such that the resolved variable is $p = x_i$, add two new vertices Σ_0' and Σ_1' to S. Associate Σ with Player I and both Σ_0' and Σ_1' with Player II. Then define:

- $N(\Sigma, \bar{u}, \bar{v}) = \Sigma'_{u_j}$;
- if $u_i = v_i$, put $N(\Sigma'_\nu, \bar{u}, \bar{v}) = \Sigma_\nu$, for $\nu = 0, 1$;

- if $u_i \neq v_i$, put $N(\Sigma'_0, \bar{u}, \bar{v}) = \neg x_i$ and $N(\Sigma'_1, \bar{u}, \bar{v}) = x_i$;
- $F_I(x_i, \bar{u}) \equiv u_i = 0$ and $F_{II}(x_i, \bar{v}) \equiv v_i = 1$;
- $F_I(\neg x_i, \bar{u}) \equiv u_i = 1$ and $F_{II}(\neg x_i, \bar{v}) \equiv v_i = 0$;
- $F_I(\Sigma'_i, \bar{u}) \equiv F_I(\Sigma, \bar{u}) \wedge N(\Sigma, \bar{u}, \bar{v}) = i$ for $i = 0, 1$;
- $F_{II}(\Sigma'_i, v) \equiv F_{II}(\Sigma, v)$ for $i = 0, 1$.

We leave the verification of the properties to the reader. ■

We have shown that there exists a circuit C with the properties required by the theorem whose size is at most $2n + 3|d|$, where $|d|$ denotes the number of clauses in the Resolution proof d. In [15] we gave a different, more direct proof of Theorem 3. In our construction the number of vertices in the circuit C is at most $2n + |d|$; however, the circuit uses on top of the usual gates $\wedge$ and $\vee$ (together with literals x_i and $\neg x_i$) also the ternary gate *selector*. If the selector gates are replaced by circuits in the basis $\wedge, \vee$, we get the same bound as above $2n + 3|d|$. But not only that: the circuits are identical.

5 From communication protocols to proofs

We are studying three things: circuits, **PLS** communication protocols and Resolution proofs. We have shown how to construct a protocol from a circuit, a circuit from a protocol and a protocol from a proof. Further, this gives us a construction of a circuit from proof, by transitivity, but a direct construction was shown in [15]. To complete the picture, it remains to construct proofs from protocols and circuits. We will only show the construction of a proof from a protocol. The other remaining construction follows by transitivity, but, certainly, a direct construction is also possible. To simplify the presentation, we will only consider protocols for the Karchmer-Wigderson games and restricted **PLS** protocols.

Proposition 4 *Let P be a protocol in the form of a Karchmer-Wigderson game for computing a partial boolean function $f(\bar{x})$ that uses k communication bits. Then one can construct a tautology of the form $\phi(\bar{x}, \bar{y}) \vee \psi(\bar{x}, \bar{z})$, with $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ in DNF, and a Resolution proof d of it such that*

1. *the size of the proof is $|d| = O(|2^k|)$ and*
2. *the formulas $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ define a partial boolean function that is at least as much defined as f, which means*
   (a) *if $f(\bar{u}) = 0$, then there exists $\bar{w}$ such that $\phi(\bar{u}, \bar{w})$ is false, and*
   (b) *if $f(\bar{v}) = 1$, then there exists $\bar{t}$ such that $\psi(\bar{v}, \bar{t})$ is false.*

Furthermore, the proof has the form of a tree.

To prove the proposition, consider all possible situations that may appear when playing according to the protocol P. We will inductively assign a clause to each of them. For the initial situation when they start, we take the empty clause. Suppose we are in a situation s with a clause Σ and Player I is to speak. Then we choose a new variable y_i and assign $\Sigma \vee y_i$ to the situation after Player I sent the bit 0, respectively, $\Sigma \vee \neg y_i$ to the situation after Player I sent the bit 1. If it is Player II to speak we do the same with a variable z_j instead of y_i. Suppose that the game ends in a situation where the players learn that the ith bit of Player I is 0 while the ith bit of Player II is 1. Let Σ be the clause assigned to this situation, let Σ_I, respectively Σ_{II}, be the subclause of Σ consisting of y_is, respectively z_js. Then we introduce two clauses

$$\Sigma_I \vee \neg x_i \quad \text{and} \quad \Sigma_{II} \vee x_i.$$

If it is 1 and 0, we take x_i in the first clause and $\neg x_i$ in the second.

One can see immediately that the clauses form a Resolution refutation. Let us check condition 2.(a). Let $\bar{u}$ such that $f(u) = 0$ be given. We want to find an assignment that makes all initial clauses made of x_is and y_js false. Given u, the protocol P determines how Player I plays in each situation, so we can set the values w according to what the protocol says. Since the protocol is correct, the clause $\Sigma_I \vee \neg x_i$ (respectively $\Sigma_I \vee x_i$) must be satisfied.

Again, we leave the details to the reader. ■

Proposition 5 *Let P be a restricted* **PLS** *communication protocol for computing a partial boolean function $f(\bar{x})$ with a search space S. Then one can construct a tautology of the form $\phi(\bar{x}, \bar{y}) \vee \psi(\bar{x}, \bar{z})$, with $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ in DNF, and its Resolution proof d such that*

1. *the size of the proof is $|d| = O(|S|)$ and*
2. *the formulas $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{x}, \bar{z})$ define a partial boolean function that is at least as much defined as f (in the sense of the previous proposition).*

The idea of the proof is essentially the same: we introduce a variable for every possible bit sent by the players and describe the dependences between them by clauses. Here are more details.

For every node s in the search space S, we introduce variables $y_{F,s}$ and $z_{F,s}$. The meaning is that s is feasible iff $y_{F,s} \wedge z_{F,s}$ is true. If s is assigned to Player I (respectively Player II), we also introduce $y_{N,s}$ (respectively $z_{N,s}$) for the neighborhood function N.

Let s be such that $N(s) = s$ and suppose that the function p determines that $u_i = 0$ and $v_i = 1$ (where $\bar{u}$ is the input of Player I and $\bar{v}$ is the input of Player II).[1] Then we introduce the clauses

$$y_{F,s} \rightarrow \neg x_i \quad \text{and} \quad z_{F,s} \rightarrow x_i. \tag{2}$$

Again, if it is 1 and 0, then we switch the negation at x_i.

[1] According to the definition p only tells the index, but we have already noticed that the actual values are also determined.

Let s be such that $N(s) \neq s$, $N(s) \in \{s_0, s_1\}$ and s is assigned to Player I. Then we introduce the following clauses:

$$\begin{aligned}
&(y_{F,s} \wedge y_{N,s}) \to y_{F,s_0},\\
&(y_{F,s} \wedge \neg y_{N,s}) \to y_{F,s_1},\\
&z_{F,s} \to z_{F,s_0},\\
&z_{F,s} \to z_{F,s_1}.
\end{aligned}$$

If s is assigned to Player II we take the same clauses with ys and zs switched.

Finally we take two clauses for the initial node $\bar{0}$:

$$y_{F,\bar{0}} \quad \text{and} \quad z_{F,\bar{0}}.$$

In order to derive the empty clause from these clauses, first derive $\neg y_{F,s} \vee \neg z_{F,s}$ from every pair of clauses (2). Then continue deriving such clauses for all $s \in S$ (going in the direction of decreasing cost). Once we have $\neg y_{F,\bar{0}} \vee \neg z_{F,\bar{0}}$, we resolve with the last two clauses to obtain the empty clause.

The verification of the condition 2. of the proposition is the same as in the previous proof. ∎

Combining the proof of Theorem 2 and Proposition 5 we get a construction of a tautology and its Resolution proof from a function and its circuit. Unfortunately, this is not the converse to feasible interpolation. In particular, it does not give us a reduction of proving lower bounds on circuit complexity to proving lower bounds on the length of proofs. Such a reduction would require a construction that, for a given partial boolean function f and a suitable representation of f by a tautology τ, would transform any boolean circuit for f into a proof of τ. It seems rather unlikely that such a reduction is possible. We only have the following trivial observation.

Corollary 6 *Let $f_n(x_1, \ldots, x_n)$ $n = 1, 2, \ldots,$ be a sequence of booelan functions. Suppose that there is no sequence of DNF tautologies $\phi_n(\bar{x}, \bar{y}) \vee \psi_n(\bar{x}, \bar{z})$ such that they represent the boolean functions f_n in the sense of Proposition 4 and have polynomial size Resolution proofs. Then f_n do not have polynomial size circuits.*

6 Generalizations

A natural question connected with Razborov's characterization of circuit complexity is: *What happens when we replace* **PLS** *by a different class of polynomial search problems?* One can consider subclasses of **PLS** and get characterizations of classes of circuits with certain restrictions. This line of research may be interesting, but we are interested in the opposite direction: replacing **PLS** by larger classes of polynomial search problems.

Our motivation is to generalize the feasible interpolation property and show that it holds true for stronger proof systems. There are results [13, 4, 3] showing that if certain bit commitment schemas are secure, then sufficiently strong proof systems do not have the feasible interpolation property. Therefore we need a concept of computation that is stronger than boolean circuits, but not too strong, otherwise it would not provide us with any new information about the complexity of the interpolation problem. Generalizations of Razborov's theorem seems to be the right place to look for such concepts.

The classes of search problems that we want to use instead of **PLS** are those that characterize provably total polynomial search problems in fragments of Bounded Arithmetic. The fragments form a hierarchy, denoted by $T^0_2, T^1_2, T^2_2, \ldots$, where, roughly speaking, T^n_2 is a theory that is based on the induction axioms for sets of complexity Σ^p_n from the Polynomial Time Hierarchy. These theories have a tight connection to bounded depth Frege propositional proof systems. The provably total polynomial search problems of T^0_2 are solvable in polynomial time; for T^1_2 they belong to **PLS**. Characterizations for higher fragments were found relatively recently, see e.g. [14, 2, 16] (but there are more papers about it).

Very recently, working with Neil Thapen, we obtained a kind of generalization of boolean circuits that can be used for a generalized feasible interpolation theorem for depth 2 Frege systems. This result is too fresh to be included in this paper. We will present it on the conference.

The negative results about feasible interpolation apply to bounded depth Frege of a sufficiently large depth, but it is not clear where the border is. In particular, we do not have an argument implying that depth 2 Frege systems do not have the feasible interpolation property. This is one more reason for studying generalizations.

How do the search problems come into play?

Suppose that

$$\bigwedge \Phi(\bar{x}, \bar{y}) \ \wedge \ \bigwedge \Psi(\bar{x}, \bar{z}) \tag{3}$$

is not satisfiable. This is equivalent to saying that, for every $\bar{u}, \bar{w}, \bar{v}, \bar{t}$, if $\bar{u}, \bar{w}$ satisfies $\bigwedge \Phi(\bar{x}, \bar{y})$ and $\bar{v}, \bar{t}$ satisfies $\bigwedge \Psi(\bar{x}, \bar{z})$, then $\bar{u} \neq \bar{v}$. The last condition is equivalent to the statement that $u_i \neq v_i$ for some i. So the following is a search problem associated with a formula of the form (3):

> *Given a proof d in a proof system P of unsatisfiability of (3) and assignments $\bar{u}, \bar{w}, \bar{v}, \bar{t}$ such that $\bigwedge \Phi(\bar{u}, \bar{w}) \ \wedge \ \bigwedge \Psi(\bar{v}, \bar{t})$ is true, find i such that $u_i \neq v_i$.*

For this problem to be nontrivial, we have to scale it up to an exponentially large structure. Think of the formula, the proof and the assignments as being exponentially large. For example, we can represent the assignments $\bar{u}$ and $\bar{v}$ as *boolean functions* $\bar{u}, \bar{v} : \{0,1\}^n \rightarrow \{0,1\}$. In Bounded Arithmetic this is formalized by second order theories, [5].

In order to prove that the search problem above always has a solution we need to prove that the proof system is sound. The strength of the theory needed to prove it depends on the strength of the proof system P; the stronger the proof system is the stronger the theory must be.

For proving soundness of Resolution, T^1_2 suffices. All provably total polynomial search problems in T^1_2 are reducible to **PLS**. Hence the problem above is reducible to a **PLS** problem. The functions and the predicate in such a **PLS** problem can be viewed as polynomial time algorithms that use $\bar{u}, \bar{v}$ as oracles. What is only important for us that they only ask a polynomial number of queries. If we now scale it down from the exponential domain to the polynomial one, we see that to compute these functions and this predicate we only need a logarithmic number of communication bits. This gives us the **PLS** communication protocol.

In a similar fashion we can associate classes of search problems with depth d Frege systems for $d > 1$.

Acknowledgment I would like to thank to Jan Krajíček and Neil Thapen for their comments on the draft of this paper.

References

[1] M. Ajtai. The complexity of the Pigeonhole Principle, Combinatorica Volume 14, Number 4, (1994), 417-433.

[2] A. Beckmann and S. Buss. Characterizing Definable Search Problems in Bounded Arithmetic via Proof Notations. Preliminary manuscript, 2009

[3] M. L. Bonet, C. Domingo, R. Gavaldá, A. Maciel, and T. Pitassi. Non-Automatizability of Bounded-Depth Frege Proofs, Computational Complexity 13:1-2 (2004), 47-68.

[4] M. L. Bonet, T. Pitassi, R. Raz. On Interpolation and Automatization for Frege Systems, SIAM Journal of Computing 29(6) (2000), pp. 1939-1967

[5] S. Cook and P. Nguyen. Logical Foundations of Proof Complexity, ASL and Cambridge University Press, 2010.

[6] A. Haken. The intractability of resolution. Theoretical Computer Science, 39:297308, 1985.

[7] M. Karchmer, A. Wigderson. Monotone Circuits for Connectivity require Super-Logarithmic Depth, SIAM Journal on Discrete Mathematics, vol. 3, no. 2, pp. 255-65, 1990.

[8] Lower Bounds to the Size of Constant-Depth Propositional Proofs, J. of Symbolic Logic, 59(1), (1994), pp.73-86.

[9] J. Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic, Journal of Symbolic Logic, 62(2), (1997), pp.457-486.

[10] J. Krajíček. Tautologies from pseudo-random generators, Bulletin of Symbolic Logic, 7(2), (2001), pp.197-212.

[11] J. Krajíček. A form of feasible interpolation for constant depth Frege systems , Journal of Symbolic Logic, 75(2), (2010), pp. 774-784.

[12] J. Krajíček. Forcing with random variables and proof complexity, London Mathematical Society Lecture Note Series, No.382, Cambridge University Press, to appear in 2010, 280pp.

[13] J. Krajíček and P. Pudlák. Some Consequences of Cryptographical Conjectures for S_2^1 and EF, Information and Computation, Volume 140, Number 1, January 1998 , pp. 82-94(13)

[14] J. Krajíček, A. Skelley and N. Thapen. NP search problems in low fragments of bounded arithmetic, in Journal of Symbolic Logic, Vol 72:2, 2007

[15] P. Pudlák: Lower bounds for resolution and cutting planes proofs and monotone computations, Journal of Symb. Logic 62(3), 1997, pp.981-998.

[16] P. Pudlák and N. Thapen. Alternating minima and maxima, Nash equilibria and Bounded Arithmetic, preprint, 2009.

[17] A. A. Razborov. Unprovability of Lower Bounds on the Circuit Size in Certain Fragments of Bounded Arithmetic, in Izvestiya of the Russian Academy of Science, mathematics, Vol. 59, No 1, 1995, pages 201-224.