CS 2429 - Propositional Proof Complexity

Lecture #8: 31 October 2002

Lecturer: Toniann Pitassi

Scribe Notes by: Phuong Nguyen

1 Automatizability and Feasible Interpolation

We have seen in the last lecture that Resolution and Cutting Plane have feasible interpolation. Note that the same result holds for the Cut–free Sequent Calculus proof system.

A natural question is if this result also holds for stronger proof systems. For example do Frege, and Extended Frege systems have feasible interpolation? We will show that under cryptographic assumptions, the answer is no.

We will first show that for any proof system P which is closed under restrictions, that there is a relationship between feasible interpolation for P and automatizability for P. It can be seen from this relationship that there is an explicit tradeoff between proof theoretic strength and proof search. In strong proof systems, there is no feasible interpolation, and it is hard to find short proofs. On the other hand, weaker proof systems have feasible interpolation, and hence lower bounds apply.

Definition A proof system P is said to be *closed under restrictions* if for all formulas f and for all restrictions ρ, from a P-proof S of f, one can get a P-proof of $f\lceil_\rho$ in time polynomial in $|S|$, where ρ is a setting of some of the underlying variables to 0 and 1.

Note that although all of the proof systems we consider in this course are closed under restriction, there are proof systems (e.g., linear resolution) which are not.

Theorem 1 *Let a proof system P be closed under restriction. If P is automatizable then P has feasible interpolation.*

Proof Suppose that S is a P–proof for the unsatisfiable formula $A(\overline{p},\overline{q}) \wedge B(\overline{p},\overline{r})$. We will give a polynomial-time algorithm $C(\alpha, S)$ such that

$$\begin{cases} C(\alpha,S) = 0 \Longrightarrow A(\alpha,\overline{q}) \text{ is unsatisfiable} \\ C(\alpha,S) = 1 \Longrightarrow B(\alpha,\overline{r}) \text{ is unsatisfiable} \end{cases}$$

$C(\alpha, S)$ is obtained as follows. First, we run the automatizability algorithm on $A(\overline{p},\overline{q}) \wedge B(\overline{p},\overline{r})$ to get a P–proof S' of $A(\overline{p},\overline{q}) \wedge B(\overline{p},\overline{r})$. Here $|S'| = s(|S|)$ for some polynomial s (by automatizibility). Now suppose that $B(\alpha,\overline{r})$ is satisfiable, then let γ be such that $B(\alpha,\gamma) = 1$. Since P is closed under restriction, $S'\restriction_{\alpha,\gamma}$ will be a proof of $A(\alpha,\overline{q})$. Also, the size of $S'\restriction_{\alpha,\gamma}$ is $s(|S|)$. Therefore, we run the automatizibility algorithm on $A(\alpha,\overline{q})$ for $s(|S'|)$ steps. If the algorithm terminates with a proof of $A(\alpha,\overline{q})$ we output 0 (i.e., $A(\alpha,\overline{q})$ is unsatisfiable). Otherwise $B(\alpha,\overline{r})$ is guaranteed to be unsatisfiable, and we output 1.

Note that the above proof gives a uniform algorithm, but both α and S are part of the input. It is not hard to convert this algorithm into a polynomial-size circuit with just α as input.

2 No Feasible Interpolation for Frege Systems

In this section, we will sketch the argument showing that under some assumption, strong proof systems like Frege, Extended Frege do not have feasible interpolation. First, we introduce some notations.

The first negativer result for interpolation was given by Krajicek and Pudlak who showed that Extended Frege does not have feasible interpolation, under the assumption that RSA is secure. We will not give the details of the proof here, but will sketch the basic idea.

Assume that there exists a one-to-one function f from n bit strings to n bit strings, and such that f is one-way. (This is a complicated technical definition, but intuitively it means that $f(x)$ can be computed in polynomial-time but for any string y, it is hard to compute the inverse, $f^{-1}(y)$. Now let $A(\overline{p}, \overline{q})$ be a propositional formula expressing that $f(\overline{q}) = \overline{p}$ and the i^{th} bit of q is 0. Similarly, let $B(\overline{p}, \overline{r})$ be a propositional formula expressing that $f(\overline{r}) = \overline{p}$ and the i^{th} bit of r is 1. For an explicit one-to-one one-way function f it is possible to give an Extended Frege refutation of $A(\overline{p}, \overline{q}) \wedge B(\overline{p}, \overline{r})$ since f is one-to-one. On the other hand, if Extended Frege has feasible interpolation, then this implies the existence of a polynomial-size circuit $C(\alpha)$ such that $C(\alpha) = 0$ whenever the i^{th} bit of $f^{-1}(\alpha)$ is zero, and $C(\alpha) = 1$ whenever the i^{th} bit of $f^{-1}(\alpha)$ is one.

This shows that we can find any bit of $f^{-1}(\alpha)$ with a polynomial-size circuit. Thus feasible interpolation for Extended Frege implies that f is not one-way. For a specific f, such as RSA, the proof shows that EF does not have feasible interpolation unless RSA is secure.

The next theorem shows the similar result for Frege proof systems.

Definition An integer n is called a *Blum integer* if $n = pq$ for prime numbers p and q such that p and q are equivalent to 3 mod 4.

Theorem 2 (Bonet, Pitassi, Raz) *If factoring Blum integers is hard, then Frege proof systems do not have feasible interpolation*

Proof [Sketch] We will use the idea of a bit commitment scheme that is behind the Diffie Hellman secret key exchange scheme. Let

$$DH_n = A(P, g, X, Y, a, b) \wedge B(P, g, X, Y, c, d)$$

be a propositional formula where X, Y are integers, P is a prime number with length $|P| = n$, and g is a string in $\mathbb{Z}_p^*$. Here P, g, X, Y represent the public part of the key, while a, b and c, d are the private part of the key. The propositional formula $A(P, g, X, Y, a, b)$ codes the sentence

"$g^a \bmod P = X, g^b \bmod P = Y$ and $g^{ab} \bmod P$ is even".

and similarly $B(P, g, X, Y, c, d)$ is a propositional formula coding the sentence

"$g^c \bmod P = X, g^d \bmod P = Y$ and $g^{cd} \bmod P$ is odd".

It is not too hard to see that DH_n is unsatisfiable, since

$$g^{ab} \bmod P = (g^a \bmod P)^b \bmod P = X^b \bmod P = (g^c \bmod P)^b \bmod P$$

$$= (g^{bc}) \bmod P = (g^b \bmod P)^c \bmod P = Y^c \bmod P = (g^d \bmod P)^c \bmod P = g^{cd} \bmod P$$

It can be shown that there exists a polynomial-size Frege refutation for DH_n as well. This is a lot of work, and involves showing how to carry out integer multiplication, powering (using the Chinese remainder theorem), and modular arithmetic via Frege proofs, and to show that the usual properties of these arithmetic operations hold, thereby allowing Frege to give a feasible proof along the lines of the above informal argument.

On the other hand, if Frege systems have feasible interpolation, then an interpolant function computes one bit of the secret key exchanged by the Diffie-Hellman protocol. Thus if Frege has feasible interpolation, then all bits of the secret key exchanged by the Diffie-Hellman procedure can be broken using polynomial-size circuits and hence the Diffie-Hellman cryptographic scheme is not secure. It was proved that breaking the Diffie-Hellman cryptographic scheme is harder than factoring Blum integers. Thus, it follows that Frege does not have feasible interpolation, if Blum integers are hard to factor. (We have left the definitions of "hard" undefined, but the formal definitions can be found in the paper by Bonet, Pitassi and Raz.)