Math 267a - Propositional Proof Complexity

Lecture #3: 23 January 2002

Lecturer: Sam Buss

Scribe Notes by: Reid Andersen

1 p-Simulation

Definition Let f and g be proof systems in the same language. We say f *p-simulates* g if there exists a poly-time computable function $H(x)$ such that $\forall x$, $g(x) = f(H(x))$. We say f *simulates* g if there exists a polynomial $p(n)$ such that $\forall x \exists y$, $|y| \leq p(|x|)$ and $f(y) = g(x)$.

Definition A proof system f is *maximal* if f simulates g for any proof system g. A proof system f is *super* if there exists a polynomial $p(n)$ such that $\forall \varphi \in TAUT$, $\exists x$ such that $|x| \leq p(|\varphi|)$ and $f(x) = \varphi$. Note that any super proof system is maximal.

Open Question Is there a super or maximal proof system?

Theorem 1 *[1] [Cook] There exists a super proof system* $\iff NP = co - NP$.

Homework 1 *Prove the above theorem for a homework excercise.*

Definition A Frege system is a proof system given by a finite set of of schematic axioms and inference rules, and must be implicationally sound and implicationally complete.

Theorem 2 *[2] [Cook-Reckhow] If* $\mathcal{F}_1$, $\mathcal{F}_2$ *are Frege systems, then* $\mathcal{F}_1$ *p-simulates* $\mathcal{F}_2$.

Proof For the proof we will assume $\mathcal{F}_1$ and $\mathcal{F}_2$ have the same language, but the statement is true in general. Consider a rule of $\mathcal{F}_2$, $\frac{A_1 \ldots A_k}{B}$. $\mathcal{F}_1$ can prove $A_1 \ldots A_k \vdash B$ by the implicational completeness of Frege proof systems. Consider an $\mathcal{F}_2$-proof $\varphi_1 \ldots \varphi_n$. We convert to an $\mathcal{F}_1$-proof as follows: φ_i follows from an inference rule $\frac{A_1\sigma \ldots A_k\sigma}{B\sigma}$, where $A_1\sigma = \varphi_{i_1}, \ldots, A_k\sigma = \varphi_{i_k}$, with $i_1 \ldots i_k < i$, and $B\sigma = \varphi_i$. Assuming $\varphi_{i_1} \ldots \varphi_{i_k}$ already proved, use the substitution σ on the $\mathcal{F}_1$-proof $A_1 \ldots A_k \vdash B$ to get an $\mathcal{F}_1$-proof $\varphi_{i_1} \ldots \varphi_{i_k} \vdash \varphi_i$. Combining this proof and the proof of $\varphi_{i_1} \ldots \varphi_{i_k}$ yields an $\mathcal{F}_1$-proof of ϕ_i.

Proof Complexity This is a polynomial time procedure. For each line of the $\mathcal{F}_2$-proof, there are $O(1)$ lines in the $\mathcal{F}_1$-proof. If the $\mathcal{F}_2$ proof has n lines and m total symbols, the $\mathcal{F}_1$ proof has $O(n)$ lines, and each line has $O(m)$ symbols. So the $\mathcal{F}_1$-proof contains $O(n)$ lines, and $O(mn)$ total symbols. Since $n \leq m$, the size of the $\mathcal{F}_1$-proof is bounded by a polynomial in the size of the $\mathcal{F}_2$-proof.

Open Question Can the bound of $O(mn)$ symbols in the preceeding proof be improved to $O(m)$? It can if we assume that $\mathcal{F}_1$ has modus ponens, but is it true in general?

Open Question Are Frege systems super? or maximal?

Open Question Is there a "natural" proof system stronger than Frege systems?

2 Extended Frege Sytems

Definition Here we define an extended Frege system, $e\mathcal{F}$. An $e\mathcal{F}_0$-proof is the same as an $\mathcal{F}_0$-proof, except the size of the proof is computed differently. The size of an extended Frege proof of A is (# of lines in the proof) + $|A|$.

Example In a previous lecture we saw that any formula $A \to A$ has an $\mathcal{F}_0$-proof of five lines. So there is an $e\mathcal{F}_0$-proof of $A \to A$ of size $5 + |A|$.

The catch is that an extended Frege system as defined above is not an abstract proof system, since an abstract proof system defines the size of a proof x to be the number of symbols in x. For this reason we will present an encoding where an $e\mathcal{F}_0$ proof with size n in the extended Frege sense can be encoded by a string of length $O(poly(n))$. We also present a polynomial time decoding algorithm to verify that a string encodes a valid $e\mathcal{F}_0$ proof. This decoding algorithm defines an abstract proof system with the notion of size that we desire, within a polynomial.

Encoding [3] [Parikh] Number the rules of inference. The axioms take values 0...9, and modus ponens takes 10. We represent an $e\mathcal{F}_0$-proof $\varphi_1, \ldots, \varphi_n = \varphi$ by a tuple $\langle e_1, \ldots, e_n, \varphi \rangle$ where if φ_i is an instance of axiom k then $e_i = k$, and if φ_i is inferred from $\varphi_{j_i}, \varphi_{k_i}$ by modus ponens, $e_i = \langle 10, j_i, k_i \rangle$.

The size of this proof skeleton is $O(nlogn + |\varphi|)$, where n is the size of the $e\mathcal{F}_0$ proof.

Claim There is a polynomial time algorithm to decide if an encoding corresponds to a valid $e\mathcal{F}_0$-proof.

Proof We convert the skeleton into a unification problem which has a solution iff the proof skeleton is valid. We create new "metavariables" $y_1 ... y_n$, and z^i_j, and search for a substitution $\sigma : y_i \mapsto \varphi_i$ which must satisfy the following equations:

1. $y_n \doteq \varphi$. (means $\sigma y_n = \varphi$)
2. if $e_i = \langle 10, j_i, k_i \rangle$, we require that $y_{k_i} \doteq (y_{j_i} \to y_i)$
3. for $0 \le e_i \le 9$ let A be the e_i-th axiom. Replace each x_j in A by z^i_j, and denote this instance of the axiom A by A^i. We require that $y_i \doteq A^i$.

A substitution σ that satisfies these requirements is called a unifier, and the encoding corresponds to a valid $e\mathcal{F}_0$-proof of φ if and only if such a σ exists. More on this next time.

References

[1] S. A. Cook, *Feasibly constructive proofs and the propositional calculus*, in Proceedings of the Seventh Annual ACM Symposium on Theory of Computing, 1975, pp. 83–97.

[2] S. A. Cook and R. A. Reckhow, *The relative efficiency of propositional proof systems*, Journal of Symbolic Logic, 44 (1979), pp. 36–50.

[3] R. J. Parikh, *Some results on the lengths of proofs*, Transactions of the American Mathematical Society, 177 (1973), pp. 29–36.