CS 40: Computational Complexity

Sair Shaikh

November 9, 2025

Problem 27. Let UNSAT be the language $\{\langle\varphi\rangle : \varphi \text{ is an unsatisfiable CNF formula}\}$. Note that I said "CNF" and not "3CNF" (not that this will matter much, but please stick to the given definition).
Describe an interactive proof (IP) protocol for UNSAT that has perfect completeness, a soundness error of at most 1/3, and where the prover and verifier exchange only $O(n/\log n)$ messages in total, n being the number of variables in the input formula φ. As usual, the verifier needs to run in time polynomial in the length of the input.

Solution. The protocol follows a structure similar to the protocol for #SAT described in the class and in Sipser 10.4, except instead of removing one variable at each step, we will remove $\log n$ variables.

First, the verifier arithmetizes the CNF formula in the input w to obtain a polynomial $f(x_1, \cdots, x_n)$ over some finite field $\mathbb{F}_q$ with $q > 2^{|w|}$. As the arithmetization equivalents for $\wedge$ and $\vee$ result in summing the degrees of their operands, the degree of each term in f (expanded out) is at most the input length, $|w|$. Call the total degree of this polynomial $d \leq |w|$.

Let

$$f_i(x_1, \cdots, x_i) := \sum_{x_{i+1}, \cdots, x_n \in \{0,1\}} f(x_1, \cdots, x_n)$$

To show that the CNF has no satisfying assignments, the prover needs to prove that:

$$f_0() = 0$$

in $O(n/\log n)$ interactions. In the ith phase of interaction, the Prover persuades the verifier that $f_{(i-1)\log n}(r_1, \cdots, r_{(i-1)\log n})$ is correct if $f_{i\log n}(r_1, \cdots, r_{i\log n})$ is correct for $1 \leq i \leq n/\log n$ where r_i are random elements in $\mathbb{F}_q$ chosen by the verifier. This is done as follows:

1. The prover sends the coefficients of $f_{i\log n}(r_1, \cdots, r_{(i-1)\log n}, z_1, \cdots, z_{\log n})$ (a polynomial in $\log n$ variables) over to the prover. Note that each coefficient is in $\mathbb{F}_q$, which is finite.

Moreover, the number of coefficients is polynomial in the input length as each term of the polynomial must have degree at most $d \leq |w|$ (reject if not), thus we can use stars and bars to determine that there are only polynomially many monomials possible. Thus this is a polynomial-sized message.

2. The verifier evaluates the provided polynomial at all values $s_1, \cdots, s_{\log n} \in \{0,1\}^{\log n}$ (which is polynomially many) and adds up the results (mod q). If this does not agree with $f_{(i-1)\log n}(r_1, \cdots, r_{(i-1)\log n})$ (take $f_0() = 0$ if $i = 1$) from the previous interaction step, reject.

3. Finally, the verifier sends the prover $r_{(i-1)\log n+1}, \cdots, r_{i \log n}$ uniformly from $\mathbb{F}_q$ and computes $f_{i \log n}(r_1, \cdots, r_{i \log n})$ for the next interaction step.

Finally, the verifier checks $f_n(r_1, \cdots, r_n) = f(r_1, \cdots, r_n)$ himself.

It is clear that the verifier operates in polynomial time. We argue the correctness of this protocol similarly to in class. If $f_0() = 0$, then a prover sending the correct f_is at every step can clearly make the verifier accept, thus we have perfect completeness. Thus, we only need to the analyze soundness error.

If $f_0() \neq 0$ and we do not reject in the first round, then the polynomial the prover sent in the first round, $\tilde{f}_{\log n}$ must disagree with the "true" polynomial $f_{\log n}$ at at least one value for $(z_1, \cdots, z_{\log n}) \in \{0,1\}^n$, i.e. $f_{\log n} \not\equiv \tilde{f}_{\log n}$. Thus, by the Schwartz-Zippel lemma,

$$\mathbf{Pr}_{\vec{r}}[\tilde{f}_{\log n}(r_1, \cdots, r_{\log n}) - f_{\log n}(r_1, \cdots, r_{\log n}) = 0] \leq \frac{d}{|\mathbb{F}^q|}$$

The upshot of this is that the value of $\tilde{f}_{\log n}(r_1, \cdots, r_{\log n})$ that the verifier uses in the subsequent step will be wrong with high probability (we'll show $\frac{d}{|\mathbb{F}^q|}$ is small). We can repeat the same logic in general: If the value $\tilde{f}_{(i-1)\log n}(r_1, \cdots, r_{(i-1)\log n}) \neq f_{(i-1)\log n}(r_1, \cdots, r_{(i-1)\log n})$ (is not correct) and we do not reject in the ith step, then it must be that $\tilde{f}_{i \log n} \not\equiv f_{i \log n}$. Thus, we have two cases:

- If $\forall i : \tilde{f}_{(i-1)\log n}(r_1, \cdots, r_{(i-1)\log n}) \neq f_{(i-1)\log n}(r_1, \cdots, r_{(i-1)\log n})$, then at the final step, the verifier will directly evaluate $f(r_1, \cdots, r_n)$ and catch this. Thus, we accept with probability 0 in this case.

- Otherwise there exists some step k where

$$\tilde{f}_{(i-1)\log n}(r_1, \cdots, r_{(i-1)\log n}) = f_{(i-1)\log n}(r_1, \cdots, r_{(i-1)\log n})$$

Taking the union bound, and applying Schwarz-Zippel,

$$\begin{aligned}
\mathbf{Pr}\left[P * V(w,r) = 1\right] &\leq \mathbf{Pr}\left[\bigcup_{i=1}^{n/\log n} \tilde{f}_{i\log n}(r_1, \cdots, r_{i\log n}) - f_{i\log n}(r_1, \cdots, r_{i\log n}) = 0\right] \\
&\leq \sum_{i=1}^{n/\log n} \mathbf{Pr}\left[\tilde{f}_{i\log n}(r_1, \cdots, r_{i\log n}) = f_{i\log n}(r_1, \cdots, r_{i\log n})\right] \\
&\leq \frac{n}{\log n} \cdot \frac{d}{|\mathbb{F}_q|} \\
&\leq \frac{|w|}{\log |w|} \cdot \frac{|w|}{2^{|w|}} = o(1)
\end{aligned}$$

Thus, for large enough messages, this probability is $< \frac{1}{3}$. If we want to improve this, we could pick an even bigger field $\mathbb{F}_q$. The question of finding a big prime of that form is not answered in this problem, but can be done to a high accuracy by the verifier using probabilisitic primality testing methods seen before.