Lecture 18

In this and the next lecture we will see that **IP** = **PSPACE**. In this lecture we give an interactive proof system for UNSAT, and therefore show that **coNP** $\subseteq$ **IP**. In fact, as we will see, the same proof system with minor modifications can be used for #SAT. Since #SAT is #**P**-complete, and since $\mathbf{PH} \subseteq \mathbf{P}^{\#\mathbf{P}}$, then **PH** $\subseteq$ **IP**. In the next lecture we will generalize this protocol in order to solve QSAT, and thus show that **PSPACE** $\subseteq$ **IP**.

1 UNSAT $\subseteq$ IP

Let ϕ be a formula with m clauses over the variables $x_1, \ldots, x_n$. Let $N > 2^n \cdot 3^m$ be a prime number. We translate ϕ to a polynomial p over the field $\pmod N$ in the following way. Literals are translated as follows: $x_i \to x_i$, $\overline{x_i} \to (1 - x_i)$. A clause is translated to the sum of the (at most 3) expressions corresponding to the literals of the clause. Finally, p is taken as the product of the m expressions corresponding to the m clauses. Notice that each literal is of degree 1, and therefore p is of degree at most m. Furthermore, for a 0-1 assignment, p evaluates to 0 if this assignment does not satisfy ϕ, and to a non-zero number if this assignment does satisfy ϕ where this number can be at most 3^m. Hence, ϕ is unsatisfiable iff

$$\sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} p(x_1, \ldots, x_n) \equiv 0 \pmod N$$

Also, the translation to p can be done in polynomial time, and if we take $N = 2^{O(n+m)}$ then elements in the field can be represented by $O(n+m)$ bits, and thus one evaluation of p in any point in $\{0, \ldots, N-1\}^m$ can be computed in polynomial time.

We now describe a protocol that enables the prover to convince the verifier that ϕ is unsatisfiable (see also Figure 1 for a sketch of the protocol). At the first round, the prover sends N, a proof that it is prime, and an m-degree univariate polynomial $q_1(x) = \sum_{x_2, \ldots, x_n \in \{0,1\}} p(x, x_2, \ldots, x_n)$. At this point, the verifier verifies the primality proof as well as checking that $q_1(0) + q_1(1) = 0$. If any of the checks fail then the verifier rejects. Otherwise, it continues by choosing a random $r_1 \in \{0, \ldots, N-1\}$ and sending it to the prover. The prover responds by sending an m-degree univariate polynomial $q_2(x) = \sum_{x_3, \ldots, x_n \in \{0,1\}} p(r_1, x, x_3, \ldots, x_n)$. Now, the verifier checks if $q_2(0) + q_2(1) = q_1(r_1)$. If not, it rejects, and otherwise it continues by choosing a random $r_2 \in \{0, \ldots, N-1\}$ and sending it to the prover. The protocol continues in this way where the ith message that the prover sends is a polynomial $q_i(x) = \sum_{x_{i+1}, \ldots, x_n \in \{0,1\}} p(r_1, \ldots, r_{i-1}, x, x_{i+1}, \ldots, x_n)$. The verifier responds by checking that $q_i(0) + q_i(1) = q_{i-1}(r_{i-1})$, rejecting if it is not, and otherwise, choosing a random $r_i \in \{0, \ldots, N-1\}$ and sending it to the prover. At the last round, the prover sends $q_n(x) = p(r_1, \ldots, r_{n-1}, x)$. The verifier checks that $q_n(0) + q_n(1) = q_{n-1}(r_{n-1})$, chooses a random $r_n \in \{0, \ldots, N-1\}$, and checks that $q_n(r_n) = p(r_1, \ldots, r_n)$. The verifier rejects if either of the checks fail, and accepts otherwise. The verifier can indeed perform the last check since as mentioned before, it can

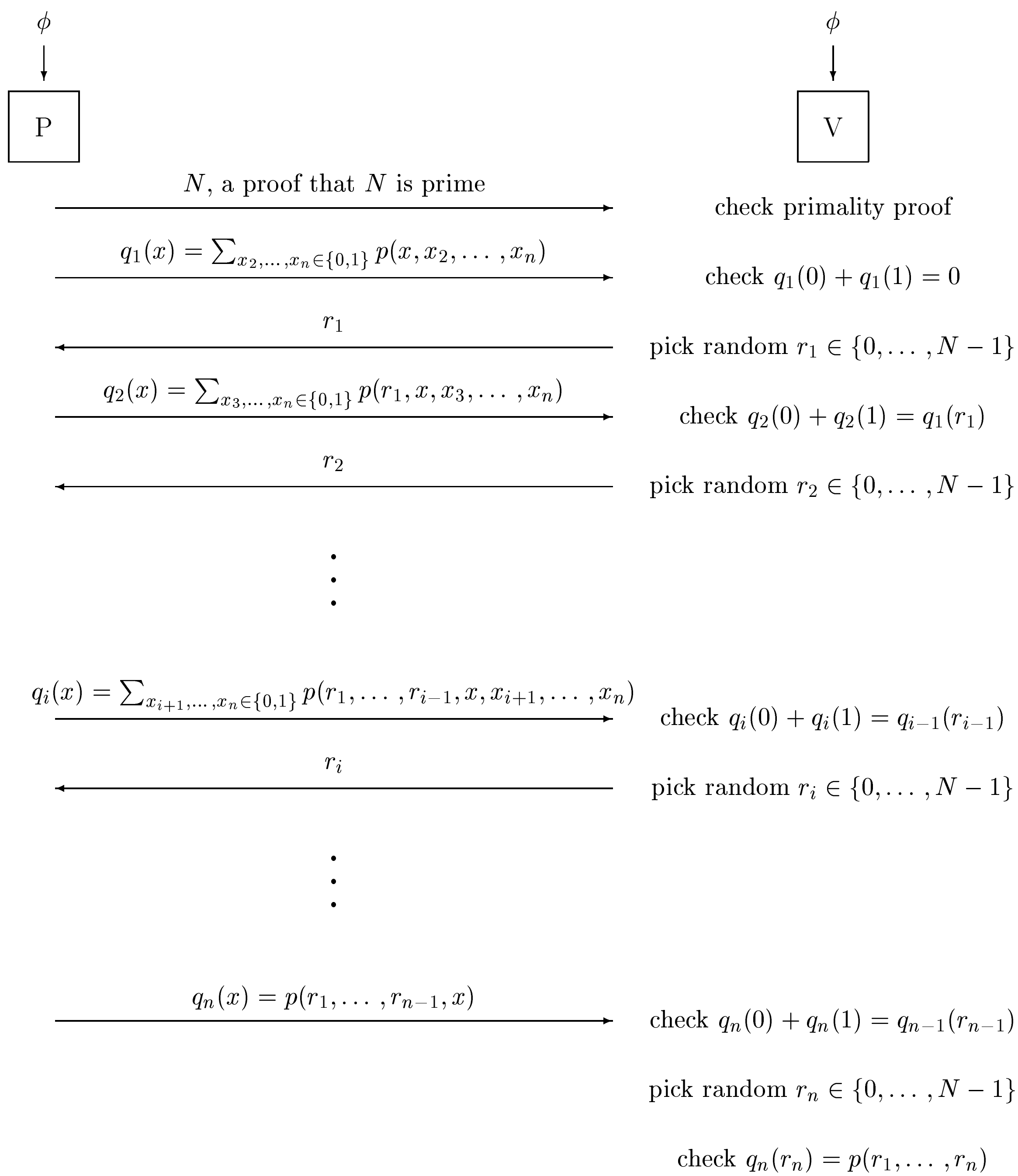

Figure 1: The protocol for proving unsatisfiability

translate ϕ to p and evaluate it at the point $(r_1, \ldots, r_n)$ in polynomial time. Also, the messages are of polynomial length because elements of the field can be written with $O(m+n)$ bits, and since each polynomial has m coefficients.

We now show the correctness of this proof system. If ϕ is unsatisfiable then the prover can make the verifier accept w.p. 1 by just following the protocol since $q_1(0) + q_1(1) = \sum_{x_1,\ldots,x_n \in \{0,1\}} p(x_1, \ldots, x_n) \equiv 0 \pmod N$. It is clear that the rest of the checks will succeed if the prover send the q_i's accordingly. We have to show that if ϕ is satisfiable then no matter what polynomials the prover sends, the verifier will reject with high probability. We write $p_i(x)$ for the polynomial that a prover who follows the protocol sends in round i. Since ϕ is satisfiable then $\sum_{x_1,\ldots,x_n \in \{0,1\}} p(x_1, \ldots, x_n) \not\equiv 0 \pmod N$. Therefore, $p_1(0) + p_1(1) \neq 0$, and hence, if the prover sends $q_1 = p_1$ then the verifier will reject. If the prover sends $q_1 \neq p_1$ then since both are degree-m polynomials then they agree on at most m places. Thus, there is probability $\geq 1 - \frac{m}{N}$ that $p_1(r_1) \neq q_1(r_1)$. Suppose that indeed $p_1(r_1) \neq q_1(r_1)$. If the prover then sends $q_2 = p_2$ then the verifier will reject because $q_2(0)+q_2(1) = p_1(r_1) \neq q_1(r_1)$. Thus, the prover must send $q_2 \neq p_2$. Again, in that case, q_2 and p_2 will differ on a fraction $\geq 1 - \frac{m}{N}$ of the elements of $\{0, \ldots, N-1\}$, so $p_2(r_2) \neq q_2(r_2)$ with probability $\geq 1 - \frac{m}{N}$. We continue the argument in a similar way. If $q_i \neq p_i$ then w.p. $\geq 1 - \frac{m}{N}$, r_i is such that $p_i(r_i) \neq q_i(r_i)$. If so, then the prover must send $q_{i+1} \neq p_{i+1}$ in order for the verifier not to reject. At the end, if the verifier has not rejected before the last check, then w.p. $\geq 1 - (n-1)\frac{m}{N}$, $q_n \neq p_n$. If so, then w.p. $\geq 1 - \frac{m}{N}$ the verifier will reject since, again, $q_n(x)$ and $p(r_1, \ldots, r_{n-1}, x)$ differ on at least that fraction of the points. Thus, the total probability that the verifier will accept is at most $\frac{nm}{N}$.

2 A Proof System for #SAT

We now show that with a few minor modifications, the previous proof system can be used to prove that a formula has a given number of satisfying assignments. Suppose ϕ has k satisfying assignments. We wish to give a protocol s.t. if the prover gives k as an answer, he is able to continue with the protocol s.t. the verifier will accept w.p. 1, but if it gives another number as an answer, then no matter what messages the prover sends afterwards, the verifier will reject with high probability.

We first change the way we translate ϕ to the polynomial p. The change is only in the way we translate clauses. Instead of translating a clause to the sum of the literals, we translate the clause $(z_1 \vee z_2 \vee z_3)$ to $1-(1-z_1)(1-z_2)(1-z_3)$. Notice that if a 0-1 assignment to the variables satisfies the clause, then the value of this expression will be 1, and if the assignment does not satisfy the clause, this value will be 0. Hence, 0-1 assignments that satisfy ϕ will be evaluated to 1 by p, while as before, the 0-1 assignments that do not satisfy ϕ will be evaluated to 0. Notice that now the degree of p is $3m$ instead of m, but now the sum over all evaluations of 0-1 assignments is equal to the number of satisfying assignments of ϕ. For the same reason, it is now enough to take $N > 2^n$.

In the current protocol, the prover starts by sending k, and then continues as in the previous protocol (with the updated definition of p). After receiving the first message, the verifier checks if $q_1(0) + q_1(1) = k$ instead of checking $q_1(0) + q_1(1) = 0$, and follows the rest of the previous protocol. If k is indeed the correct number of satisfying assignments, then the prover can continue by sending $q_i = p_i$ at each round, and the verifier will accept with probability 1. If k is not the correct number of satisfying assignments, then according to the same analysis as in the previous section, the verifier will accept with probability at most $\frac{3mn}{N}$.