has to at least be read (see Lecture 4 for more details on this). We will now give a randomized algorithm (in co-RP) which takes only $O(n^2)$ time.

Let us introduce some notation for the rest of the course: $x \in_R X$ means "choose x uniformly at random from the set X".

The algorithm is as follows:

- Pick a vector $x \in_R \{0,1\}^n$.
- Compare ABx with Cx (note this takes $O(n^2)$ time since we can compute in the order $A(Bx)$).
- If $ABx = Cx$, then output *Yes*, otherwise output *No*.

Note that if the matrices are over the field $\mathcal{F}$, then the computations are also carried out over the field $\mathcal{F}$. Now if $AB = C$, the algorithm always outputs the correct answer; only if $AB \neq C$, the algorithm may output the wrong answer. We now bound the probability of such an error.

Theorem 2.2.2 (Freivald) *If $AB \neq C$, then the above algorithm fails with probability at most $\frac{1}{2}$.*

Proof: We first prove a simpler claim:

Claim 2.2.3 *Given n-digit strings $a, b \in \mathbb{R}^n$ and $x \in_R \{0,1\}^n$, $Pr[a \cdot x = b \cdot x] \leq \frac{1}{2}$.*

Proof: [Claim] Suppose $a_i \neq b_i$. Let $\alpha = \sum_{j \neq i} a_j x_j$ and $\beta = \sum_{j \neq i} b_j x_j$. We can write $a \cdot x = \alpha + a_i x_i$ and $b \cdot x = \beta + b_i x_i$. This gives us

$$a \cdot x - b \cdot x = (\alpha - \beta) + (a_i - b_i) x_i.$$

We can invoke the Principle of Deferred Decisions (see Section 3.5 of M&R) to assume that we are given x_j for $j \neq i$. Then we can write

$$Pr[a \cdot x - b \cdot x = 0] = Pr\left[x_i = \frac{\alpha - \beta}{b_i - a_i}\right] \leq \frac{1}{2}.$$

■

We now use this claim to prove the theorem: if $AB \neq C$, then there is at least one row in AB, say $(AB)_i$, that differs from the corresponding row C_i in C. Now we apply the above claim with $a = (AB)_i$ and $b = C_i$. The probability that $a \cdot x = b \cdot x$ is at most 1/2, but in order for the algorithm to output *Yes*, we must have $a \cdot x = b \cdot x$. Therefore, the probability of failure for the algorithm is at most 1/2. ■

(See Section 7.1 in M&R for more on this problem.)

2.2.4 Polynomial identity checking

In the polynomial identiy checking problem, we are given two multi-variate polynomials $f(x_1, \ldots, x_n)$ and $g(x_1, \ldots, x_n)$ each with degree d (again we are computing in some field $\mathcal{F}$). We are not given the polynomials explicity (we cannot read the polynomials in poly-time). Instead, we have "black-box" access for evaluating a polynomial (for example, see the definition of a *Vandermonde matrix* on page 165 of M&R). Given these two polynomials, the problems is to determine if the polynomials are equal (i.e. $f = g$ or $f - g = 0$). Letting $Q = f - g$, we can check if $Q = 0$.

There is no known poly-time algorithm for this problem. We can show that it is in co-RP.

First consider the univariate case. We can pick $d+1$ distinct, arbitrary values at random from $\mathcal{F}$. If $Q(x) = 0$ for all $d+1$ values for x, then $Q = 0$. (With degree d it can have at most d roots.)

This approach does not directly apply to the multivariate case because there can be exponentially many roots. Roughly speaking, we can handle the multivariate case by fixing $n-1$ variables and applying the result from the univariate case. Consider the following algorithm, which assumes we have some subset $S \subset \mathcal{F}$ with $|S| \geq 2d$.

- Pick $r_1, \ldots, r_n \in_R S$
- Evaluate $Q(r_1, \ldots, r_n)$
- If 0, return $Q = 0$

Theorem 2.2.4 (Schwartz-Zippel) *If, in the above algorithm, $Q \neq 0$, we have*

$$Pr[Q(r_1, \ldots, r_n) = 0] \leq \frac{d}{|S|}.$$

Proof: By induction on n. The base case is the univariate case described above. With $Q \neq 0$, we want to compute $Pr[Q(r_1, \ldots, r_n) = 0]$. Let k be the largest power of x_1. We can rewrite

$$Q(x_1, \ldots, x_n) = x_1^k Q_1(x_2, \ldots, x_n) + Q_2(x_1, \ldots, x_n)$$

for some polynomials Q_1 and Q_2. Now we consider two events. Let A be the event that $Q(r_1, \cdots, r_n)$ is 0, and B be the event that $Q_1(r_2, \cdots, r_n)$ is 0.

We can rewrite the probability that $Q_1(r_2, \cdots, r_n)$ is 0 as:

$$\begin{aligned} \Pr[Q(r) = 0] = \Pr[A] &= \Pr[A \mid B]\Pr[B] + \Pr[A \mid \neg B]\Pr[\neg B] \\ &\leq \Pr[B] + \Pr[A \mid \neg B] \end{aligned}$$

Let us first bound the probability of B, or the probability that $Q_1(r_2, \cdots, r_n) = 0$. Q_1 has degree $d-k$ and so we can use the inductive hypothesis to obtain

$$\Pr[B] = \Pr[Q(r_2, \ldots, r_n) = 0] \leq \frac{d-k}{|S|}$$

.

Similarly, given $\neg B$ (or $Q_1(r_2, \cdots, r_n) \neq 0$), the univariate polynomial $Q(x_1, r_2, \ldots, r_n)$ has degree k. Therefore, again by inductive hypothesis,

$$\Pr[A \mid \neg B] = \Pr[Q(x_1, r_2, \ldots, r_n) = 0 \mid Q_1(r_2, \ldots, r_n) \neq 0] \leq \frac{k}{|S|}.$$

Thus,

$$\begin{aligned} Pr[Q(r)=0] &\leq Pr[B] + Pr[A \mid \neg B] \\ &\leq \frac{d-k}{|S|} + \frac{k}{|S|} \\ &= \frac{d}{|S|} \end{aligned}$$

■

As mentioned before, there is no known poly-time algorithm for this problem. Recent results by Impagliazzo and Kabanets (2003) show that proving that p.i.c. is in P would imply that either $NEXP$ cannot have poly-size non-uniform circuits, or *Permanent* cannot have poly-size non-uniform circuits. Little is known about these lower bound results, and the Impagliazzo-Kabanets result indicates that proving p.i.c. $\in$ P may require new techniques in complexity theory.

(For more on the polynomial identity checking problem, see Section 7.2 in M&R.)

References

L. Adleman and M-D A. Huang (1992). Primality Testing And Two Dimensional Abelian Varieties Over Finite Fields. In *Springer Verlag Lecture Notes In Mathematics*, 1992.

Impagliazzo and Kabanets (2003). Derandomizing polynomial identity checking means proving circuit lower bounds. In *ACM Symposium on the Theory of Computation*, 2003.