IP = PSPACE

Zhewei Xu

In this report, we aim to provide a high-level idea on $IP = SPACE$. The original proof is presented by Adi Shamir, and later simplified by Alexander Shen. Our report will not follow either paper rigorously, but provide an overall explanation referencing multiple sources. We will jump straight to the main proof, and not bore readers with the introduction of IP.

IP $\subseteq$ PSPACE

We will first show the easier direction. In IP, a polynomial-bounded verifier V interacts with the prover P for at most polynomial rounds, and on each round, processes strings of at most polynomial length. Therefore, if a tree recording all possible interactions is constructed, the tree has maximum polynomial depth, corresponding to the number of rounds. Each node in the tree has 2^{poly} child nodes, corresponding to all possible strings exchanged between P and V.

To show that IP $\subseteq$ PSPACE, we need to show that a correct proof can accepted w.p. $\geq \frac{2}{3}$ in PSPACE, or a wrong proof is only accepted w.p. $\leq \frac{1}{3}$ in PSPACE.

We perform a DFS on the tree. If at the end of some exchange, V accepts, the matching leaf node will have value 1, else 0. If an edge represents a $P \to V$ exchange (P can produce a string that V accepts), the *maximum* value $\{0, 1\}$ over all its child nodes is taken. If an edge represents a $V \to P$ exchange (the probability that V acknowledges the current round of interaction is valid), the *average* value over all its child nodes is taken. In the end, if the root has probability $\geq \frac{2}{3}$, the language is in IP. If the root has probability $\leq \frac{1}{3}$, the language is not in IP.

This algorithm can be computed in PSPACE. While there are an exponential number of nodes, the algorithm only needs to keep track of the number

of recursions, and the maximum or average value, using $log(2^{poly}) = poly$ space. $\square$

PSPACE $\subseteq$ IP

In the reverse direction, we will show that a PSPACE-complete problem, in particular TQBF, can be reduced to IP. Two major techniques are used: arithmetization and the sum check protocol.

All of the TQBF's clauses are rewritten in 3CNF format. The rewritten formula is then arithmetized. Any $\vee$ operation is treated as $+$; any $\wedge$ is treated as $\times$; any negation of a variable x is treated as $(1-x)$. Furthermore, any $\forall$ operation of a variable x is expanded as the product of all possible assignments of x:

$$\forall x_n \phi(x_1, ..., x_{n-1}, x_n) = \phi(x_1, ..., x_{n-1}, 0)\phi(x_1, ..., x_{n-1}, 1)$$

Any $\exists$ operation is expanded as follows:

$$\exists x_n \phi(x_1, ..., x_{n-1}, x_n) = 1 - (1 - \phi(x_1, ..., x_{n-1}, 0))(1 - \phi(x_1, ..., x_{n-1}, 1))$$

However, before performing the sum check protocol, there is one more issue. The expansion of each quantifier doubles the power of the formula, resulting in an exponential power in the end. To counter this effect, after each expansion, a power reduction operation is added before resolving the next quantifier. Since $x \in \{0, 1\}$, it follows that $x^k \in \{0, 1\}$. Thus we reduce all powers of x^k to x.

Finally, the sum check protocol is carried out, with final value $= 0$ representing an unsatisfiable TQBF formula, and > 0 otherwise. $\square$

Conclusions

Shamir's paper is a milestone in computation complexity. Although it is certainly not a proof to $P \neq NP$, it reinforces the idea of a polynomial hierarchy. It also heavily influences areas of cryptography, and the soon-to-come PCP theorem, which marks another breakthrough in theoretical computer science.

The open question followed by the paper is obviously the $P = NP$ question. Having proved that both diagonalization and relativization does not work, perhaps the existence of NPI from Ladner's theorem can finally prove such a result.

Sources

- A. Chiesa, I. Shinkar. Introduction to Interactive Proofs & The Sum-check Protocol
- J. Katz. CMSC 652, Fall 11, Lecture 19.