1 Introduction

The main reference for this talk is Raz's paper "Elusive Functions and Lower Bounds for Arithmetic Circuits". An excellent reference for the background (which will occupy much of the talk) is Wigderson's book *Mathematics and Computation*, especially Chapter 12 (a free pdf of the book is available on his website).

For the rest of the talk, we will be working over $\mathbb{C}$. Essentially everything can be made to work over other fields (even non-algebraically closed fields), but for concreteness we will stick with $\mathbb{C}$. We begin with the following basic fact from linear algebra:

Proposition. *Let $f : \mathbb{C} \to \mathbb{C}^m$ be the rational normal curve (aka the moment curve), defined by*

$$f(t) = (t, t^2, \dots, t^m)$$

Then its image $\mathrm{im}(f)$ *is not contained in any affine hyperplane in $\mathbb{C}^m$.*

Proof. Since the origin is in $\mathrm{im}(f)$, it suffices to consider only hyperplanes through the origin. Picking m distinct non-zero numbers $t_1, \dots, t_m \in \mathbb{C}$, the calculation of the Vandermonde determinant tells us that their images $f(t_1), \dots, f(t_m)$ are linearly independent in $\mathbb{C}^m$. Therefore, they are not contained in any $(m-1)$-dimensional subspace, or equivalently any hyperplane through the origin. □

We can reformulate this fact as follows:

Proposition. *For any polynomial mapping $\Gamma : \mathbb{C}^{m-1} \to \mathbb{C}^m$ of degree at most 1 (i.e. a mapping where each coordinate function is a polynomial of degree at most 1), we have*

$$\mathrm{im}(f) \nsubseteq \mathrm{im}(\Gamma)$$

This property, of not lying in any low-degree affine variety, is the one that we will study throughout this talk. We formalize this notion as follows:

Definition. A polynomial mapping $f : \mathbb{C}^n \to \mathbb{C}^m$ is called *r-elusive* if for every polynomial mapping $\Gamma : \mathbb{C}^{m-1} \to \mathbb{C}^m$ of degree at most r,

$$\mathrm{im}(f) \nsubseteq \mathrm{im}(\Gamma)$$

The example above shows that the moment curve is 1-elusive.

The key result of this talk (which, as stated here, is quite imprecise) is the following:

"Theorem" (Raz)**.** *If there is an* ***explicit*** *2-elusive function, then* "P $\neq$ NP"*.*

A few remarks need to be made. First, and most glaringly, we need to specify what *explicit* means above, and also to remove the scare quotes around P $\neq$ NP. Second, it's worth observing that regardless of its precise meaning, the word *explicit* is crucial, since one can verify that if we allow sufficiently high degree, then a generic choice of f will be elusive; loosely, it suffices to write down in a human-readable way an actual example, rather than appealing to any sort of genericity argument. This sort of phenomenon, where the veracity of a theorem depends on our ability to write down an example (rather than just knowing it exists) is rather unusual in mathematics, but actually not that uncommon in computer science. Finally, it's worth remarking that (to my knowledge) one can't use this technique to actually prove that P $\neq$ NP, only the closely related result that we'll soon discuss. That question is, as far as anyone knows, logically independent from the P vs. NP problem; nevertheless, it is closely related, and solving it would be a very big deal.

2 Arithmetic Circuits

Arithmetic complexity theory is concerned, essentially, with the question of ways to express polynomials "concisely". The basic definition is the following:

Definition. An *arithmetic circuit* is a directed graph without directed cycles, where each root vertex is labeled by either a variable x_i or a field element $\alpha \in \mathbb{C}$, and each non-root vertex is labeled by either a $+$ or $\times$ sign. Each node computes a polynomial in a natural way: every $+$ node computes the sum of its children, and every $\times$ node computes the product of its children.

The *size* of an arithmetic circuit is the number of edges in the graph.

Arithmetic circuits are a natural way of expressing polynomials, and the size is a natural measure of how "concise" a given representation is.

Example (Elementary symmetric polynomials)**.** Consider the elementary symmetric polynomials,

$$S_n^d(x_1, \dots, x_n) = \sum_{\substack{A \subseteq \{1,\dots,n\} \\ |A|=d}} \prod_{i \in A} x_i$$

This representation, as a sum of products of variables, can be thought of as a simple arithmetic circuit, with a product node for each such subset A, and then a single sum node adding them all up. The number of such sets A is $\binom{n}{d}$, so the size of this circuit will be (up to constants) roughly $(n/d)^d$. When d depends on n (e.g. $d = n/2$), then this is exponential in n, and thus huge.

Ben-Or found a beautiful trick to get a circuit computing S_n^d with size only $O(n^2)$. Ben-Or's proof is based on the fact that S_n^d is the coefficient of t^d in the univariate polynomial

$$P(t) = \prod_{i=1}^{n} (t + x_i)$$

He then uses the fact that this polynomial can be computed using a small arithmetic circuit to evaluate $P(t)$ at $n+1$ different choices of t, and then uses polynomial interpolation to recover the coefficients S_n^d.

On the other hand, Baur and Strassen proved a lower bound, which remains the best lower bound known to this day (for *any* polynomial), proving that any circuit computing S_n^d has size at least $\Omega(n \log d)$. For $d = n/2$, this gives the slightly superlinear bound $\Omega(n \log n)$; despite being a tiny improvement over linear, it has never been beaten.

Example (Determinant)**.** Given n^2 variables $\{x_{ij}\}_{i,j=1}^n$, the determinant of the symbolic matrix they define is a polynomial. Again, its naive representation as a circuit, coming from the definition

$$\det([x_{ij}]_{i,j}) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_{i=1}^{n} x_{i\pi(i)}$$

uses $n!$ product gates, which is again exponential in n. However, it turns out that Gaussian elimination (which is the method one actually uses when one wants to compute a big determinant) can be turned into an arithmetic circuit of size only $O(n^3)$. This requires a bit of care to do correctly, since the usual description of Gaussian elimination requires us to divide by our matrix entries (which we don't allow in arithmetic circuits), but it turns out that this can be avoided. In fact, Strassen proved that allowing division "never helps", in the sense that any circuit that uses division gates can be turned into one that doesn't while suffering only a polynomial loss in size.

Example (Permanent)**.** The permanent polynomial is defined almost like the determinant, except with no signs:

$$\operatorname{per}([x_{ij}]_{i,j}) = \sum_{\pi \in S_n} \prod_{i=1}^{n} x_{i\pi(i)}$$

so again, our first guess at an arithmetic circuit will have size $\Omega(n!)$. Ryser found a smaller circuit for the permanent, of size $O(n^2 2^n)$, which is still exponential. But unlike the previous examples, there is no known method to compute the permanent using only polynomially-sized circuits. In fact, it was proved by Valiant

that the permanent is "complete for the class VNP", which means that the permanent can be computed by polynomially-sized circuits if and only if *all* "natural" polynomials can be computed by polynomially-sized circuits. Equivalently, if we can find *any* polynomial that is hard to compute, then the permanent is hard to compute.

This question considered in all these examples, of which polynomials can be efficiently computed, is the so-called VP vs. VNP question (the V stands for Valiant, who defined these classes and started this theory). Here are the foraml definitions of these two classes:

Definition. A sequence of polynomials $\{f_n\}_{n\to\infty}$, with $f_n \in \mathbb{C}[x_1, \dots, x_n]$, is said to be in the class VP if, for all n, there is an arithmetic circuit computing f_n whose size is $\text{poly}(n)$.

A sequence of polynomials $\{f_n\}_{n\to\infty}$ is said to be in VNP if there exists another sequence $\{g_n\}$ with $g_n \in \mathbb{C}[x_1, \dots, x_n, y_1, \dots, y_\ell]$ such that $\{g_n\} \in \text{VP}$, $\ell = \text{poly}(n)$, and

$$f_n(x_1, \dots, x_n) = \sum_{a_1, \dots, a_\ell \in \{0,1\}} g_n(x_1, \dots, x_n, a_1, \dots, a_\ell)$$

This definition, especially of the class VNP, is probably a bit surprising at first. Getting intuition for why this is a natural definition is a bit hard, but here are some attempts at doing so:

- In the case where f is a multilinear polynomial (which turns out to be essentially equivalent to the general case, basically thanks to the Veronese embedding), the question of whether $f \in \text{VNP}$ is essentially equivalent to the following question: given a monomial, can we compute its coefficient in f in polynomial time? This is a reasonable model for a "natural" polynomial, namely one that we can "describe" efficiently.

- In standard (Boolean) complexity theory, a problem is said to be in P if it can be solved by a polynomial-time algorithm, and is said to be in NP if it can be solved in polynomial time once a "hint" is given. More concretely, a problem is in NP if there is some *witness* a of polynomial size and some polynomial-time algorithm whose input is both a and the input to the problem, which can solve the problem. An essentially prototypical example is Sudoku: determining whether a given Sudoku puzzle has a solution might be hard, but if I give you a filled-in grid, then you can confirm quickly that it is indeed a valid solution, and thus confirm that the puzzle has a solution. In this case, the filled-in grid would be the witness.

  Since the existence of the witness a is defined by an existential quantifier, we can think of this property as an OR over all possible witnesses, namely an OR over all Boolean strings of polynomial length. The algebraic analogue of an OR is a sum, so in the definition of VNP we sum over all possible witness-analogues.

- Every single polynomial that arises naturally in mathematics and physics is in the class VNP. This includes things as disparate as the Jones polynomials of knots, partition functions in statistical mechanics, Schubert polynomials in the theory of flag varieties, and so on. So whether or not one accepts that this definition has some intrinsic merits, it's certainly proven useful by its wide scope.

Before we can return to the actual proof, we need to define the relevant notion of an explicit polynomial map $\mathbb{C}^n \to \mathbb{C}^m$. Essentially, this notion is just being in VNP, except that we need all the coordinate functions to be computable "simultaneously":

Definition. A polynomial map $f : \mathbb{C}^n \to \mathbb{C}^m$ is called *explicit* if for some $\ell = \text{poly}(n)$ and for $k = \lceil \log_2 m \rceil$, there is some $g \in \mathbb{C}[x_1, \dots, x_n, y_1, \dots, y_\ell, w_1, \dots, w_k]$ with $\deg g = \text{poly}(n)$ and $g \in \text{VP}$, such that for every $1 \le i \le m$,

$$f_i(x_1, \dots, x_n) = \sum_{a_1, \dots, a_\ell \in \{0,1\}} g(x_1, \dots, x_n, a_1, \dots, a_\ell, j_1, \dots, j_k)$$

where $(j_1, \dots, j_k)$ is the binary representation of $i - 1$.

Again, the precise definition is not so important: the key properties are that this is essentially equivalent to having the coefficient of each monomial of each coordinate function be efficiently computable, and that any polynomial with a human-readable description will fit into this formalism of explicitness.

3 A Formal Statement and Proofs

Now that we've defined all the necessary concepts, we can formally state (a special case of) Raz's theorem:

Theorem (Raz)**.** *Let n be an integer tending to infinity, and let m grow exponentially in n. If there is an* ***explicit*** *2-elusive function $f : \mathbb{C}^n \to \mathbb{C}^m$, then* VP $\neq$ VNP*.*

The first idea we need in order to prove this result is that of a *universal circuit:*

Definition. A *pre-circuit*[1] is a circuit where we have not yet assigned what constants will be inputted on the leaves (recall that the leaves of a circuit are labeled by either variables x_i or by constants $\alpha \in \mathbb{C}$). In other words, the leaves in a pre-circuit are labeled by two kinds of variables, x_i and y_j, and we can turn a pre-circuit into a circuit by assigning a complex number to each y_j.

Proposition. *For every $n, s, r \geq 1$, there is a pre-circuit $\widetilde{C}$ of size $O(sr^4)$ that is* universal *for all n-input circuits of size s that compute a polynomial of degree $\leq r$. Specifically, for every polynomial $g \in \mathbb{C}[x_1, \dots, x_n]$ with $\deg g \leq r$ that can be computed by a circuit of size $\leq s$, there is an assignment of constants to the y_j leaves of $\widetilde{C}$ such that the resulting circuit computes g.*

Idea of proof. Basically, this proposition boils down to the fact that all circuits can be put in some sort of normal form without increasing their size too much. As intuition, consider the fact that every polynomial of bounded degree can be be written as a sum of monomials; this means that if we make a new variable for the coefficient of each monomial, we get a "universal pre-polynomial", and we can get any polynomial by assigning constants to the dummy variables.

For circuits specifically, it turns out that given any circuit C computing a polynomial of degree $\leq r$, we can turn it into another circuit C' with the following properties:

- C' is not much bigger than C (namely, the size increases by a factor of at most r^4)
- The nodes of C' come in $2r$ layers, where the first layer consists of all input nodes, the next layer consists of sum nodes, and after that all layers alternate between sum and product nodes
- The children of any sum node come from the previous level, and the children of any product node come from some lower level
- Every node labeled by a constant is used at most once (i.e. it has at most one outgoing edge), and there are at most s such nodes

We can construct C' from C more or less greedily (i.e. just adding extra nodes whenever one of the above properties is not satisfied), and one can check that doing so will not increase the size too much.

Now, once we know that every circuit has a normal form as above, the result follows fairly straightforwardly. Simply take $\widetilde{C}$ to be the "largest" possible pre-circuit in this normal form (i.e. with the maximal possible number of nodes and edges). Then any circuit in normal form can be found as an assignment of labels to $\widetilde{C}$ (some care needs to be taken in order to destroy unnecessary edges, but this can be done by assigning a bunch of constants to be zero, and then multiplying them by the relevant edge outputs in product gates, which is equivalent to not using the edges at all). □

[1] This is not standard terminology.

Now, fix n, r, s, and let m be the number of monomials of degree $\leq r$ in n variables, namely $m = \binom{n+r}{r}$. We then identify $\mathbb{C}^m$ with the vector space of all polynomials of degree $\leq r$, denoted $\mathbb{C}[x_1, \dots, x_n]_{\leq r}$. Recall that in the universal pre-circuit $\widetilde{C}$, there are at most s nodes labeled by the dummy variables y_j, so we define a map $\Gamma : \mathbb{C}^s \to \mathbb{C}^m$ as follows. Each vector $(\alpha_1, \dots, \alpha_s) \in \mathbb{C}^s$ defines an assignment to each dummy variable in $\widetilde{C}$, and thus defines an arithmetic circuit, which computes some polynomial $g_{\alpha_1,\dots,\alpha_s}$ of degree $\leq r$. By our identification, we can think of this polynomial as an element of $\mathbb{C}^m$, and define it as the image under Γ. In other words,

$$\Gamma(\alpha_1, \dots, \alpha_s) = g_{\alpha_1,\dots,\alpha_s} \in \mathbb{C}[x_1, \dots, x_n]_{\leq r} \cong \mathbb{C}^m$$

Equivalently, we can think of the pre-circuit $\widetilde{C}$ as computing a polynomial not just in the "ordinary" input variables x_i, but also in the dummy variables y_j. Then the map Γ is just a restriction map of this universal polynomial to the space of all polynomials in the x_i, gotten by plugging in values to the y_j. This idea implies the following result:

Lemma. *$\Gamma : \mathbb{C}^s \to \mathbb{C}^m$ is a polynomial map of degree $\leq 2r - 1$.*

Proof. The only thing that remains to be proved is the degree bound. But this follows from our normal form condition, that each y_j is used at most once, together with the fact that $\widetilde{C}$ has only $2r - 1$ non-input layers. So at most $2r - 1$ multiplications can happen to the y_j, giving the desired bound. □

Lemma. *Let $g \in \mathbb{C}[x_1, \dots, x_n]_{\leq r}$. If g can be computed by a circuit of size s, then $g \in \mathrm{im}(\Gamma)$. Conversely, if $g \in \mathrm{im}(\Gamma)$, then g can be computed by a circuit of size $O(sr^4)$.*

Proof. This follows from the universality of $\widetilde{C}$ and the definition of Γ. If g can be computed by a size-s circuit, then there is an assignment of the dummy variables in $\widetilde{C}$ that gives such a circuit, and thus $g \in \mathrm{im}(\Gamma)$. Conversely, if $g \in \mathrm{im}(\Gamma)$, then pick a preimage $(\alpha_1, \dots, \alpha_s)$. Substituting these into $\widetilde{C}$ gives a circuit computing g, whose size is at most $O(sr^4)$, the size of $\widetilde{C}$. □

Now, suppose we have a map $f : \mathbb{C}^n \to \mathbb{C}^m$ that is $(2r-1)$-elusive. In particular, $\mathrm{im}(f) \nsubseteq \mathrm{im}(\Gamma)$, so there is some $(z_1, \dots, z_n) \in \mathbb{C}^n$ such that $f(z_1, \dots, z_n) \notin \mathrm{im}(\Gamma)$. Viewing $f(z_1, \dots, z_n) \in \mathbb{C}^m \cong \mathbb{C}[x_1, \dots, x_n]_{\leq r}$ as a polynomial f' of degree $\leq r$, we get a polynomial that cannot be computed by size-s circuits, by the previous lemma. Moreover, one can check using the definition of explicitness that if f were an explicit map $\mathbb{C}^n \to \mathbb{C}^m$, then the resulting polynomial f' will be in VNP (since explicitness was essentially equivalent to being in VNP, and the map giving f' from f can be represented by a small circuit). Thus, we have found a polynomial in VNP, together with a lower bound of s on the size of circuits that compute it. If we're careful with the parameters, we can ensure that s is superpolynomial in n, thus proving that $f' \notin$ VP, and concluding that VP $\neq$ VNP.

Finally, how do we get from this result, which was about $(2r-1)$-elusive functions, to a statement about 2-elusive functions? This uses a very important result of Agrawal and Vinay (based on a lot of earlier work) that says that our normal form (and thus our universal pre-circuit $\widetilde{C}$) can be asked to have a much stronger structure. Specifically, we can ensure that this normal form will have only four layers, two of addition and two of multiplication (this is called a $\Sigma\Pi\Sigma\Pi$-circuit), while not increasing the size too much (though we must now tolerate superpolynomial losses in size, rather than just the polynomial ones from before). When we do this, since we have only two multiplication layers, we see that the map Γ we get from the resulting pre-circuit $\widetilde{C}$ will be of only degree 2, since there will be at most two multiplications of the dummy variables y_j. Thus, once one is again careful with the choice of parameters (now one needs to be even more careful because of the larger size-increase in the reduction to a $\Sigma\Pi\Sigma\Pi$-circuit), we get the result about 2-elusive functions that was promised.