Circuit complexity
(CMI student seminar)

Bijayan Ray

November 6, 2023

Contents

Perfect matching of a graph (motivating example) I

Definition 1.1

Given a graph $G = (V, E)$ a perfect matching in G is a subset M of edge set E, such that every vertex $v \in V$ is adjacent to exactly one edge in M.

It need not always exist. For example, consider the K_3 (triangle) graph does not have a perfect matching.

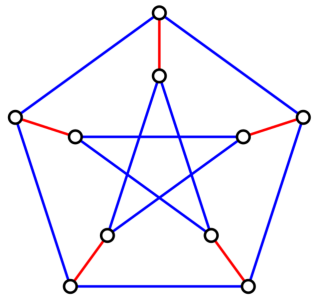

Figure: Perfect matching in Petersen graph

Perfect matching of a graph (motivating example) II

Problem 1.1

Given a graph $G = (V, E)$, how will you check if the graph has a perfect matching?

Well one of the computationally efficient ways is to consider something called the Tutte matrix of the graph and check whether the determinant of the matrix is zero or not. But determinants are polynomials that can be represented in form of a "small" arithmetic circuit, so checking whether a graph has a perfect matching or not reduces to the problem of PIT as we shall discuss and make more precise in subsequent slides.

Arithmetic circuit

Definition 1.2

It is a directed acyclic graph with internal nodes that are arithmetic operations $+, \times$ and the leaf nodes are variables x_is or field elements. Note that this evaluates to a polynomial in $\mathbb{F}[x_1, \cdots, x_n] = \mathbb{F}[x]$ and every polynomial in $\mathbb{F}[x]$ can be written in this form.

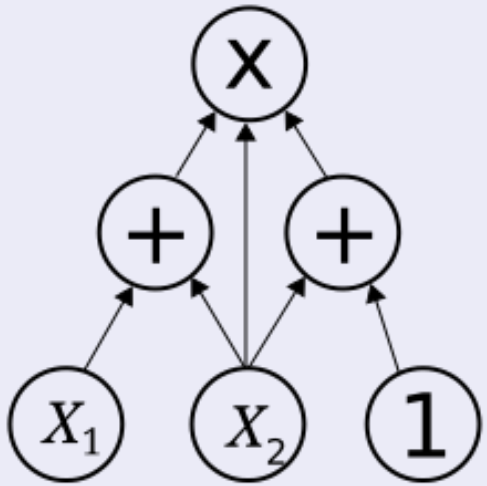

Figure: *Arithmetic circuit*

Determinants I

Theorem 1.1

An $n \times n$ determinant can be represented by an arithmetic circuit of size $poly(n)$.

We proof using simple induction on order n. The $n = 1$ case is quite easy to check. Now assume that the $n \times n$ determinant has $poly(n)$ sized arithmetic circuit, then $(n+1) \times (n+1)$ has a $poly(n+1)$ sized circuit too. Consider the matrix $A = [a_{ij}]_{0 \leq i \leq n+1, 0 \leq i \leq n+1}$

$$det(A) = \sum_{j=1}^{n+1} (-1)^{1+j} a_{1,j} A_{1,j}$$

where A_{ij} denotes the ijth minor of the matrix that $=$ determinant of the matrix we obtain by removing the ith row and jth column from the matrix A.

Determinants II

Now by induction hypothesis since A_{1j} is $n \times n$ order matrix it a has $poly(n)$ sized circuit for every $j \in 1, \cdots n+1$ which means the determinant $det(A)$ from the equation (minor expansion) also has a $poly(n)$ sized circuit that is $poly(n+1)$ sized circuit. As it is a linear combination of $n+1$ many $poly(n)$ sized circuits. This proves the theorem by induction.

A slight detour: Permanents I

It is worth mentioning about another (*relevant*) polynomial somewhat similar to determinant, for which if there is a polynomial sized circuit, is still open. Recall the Leibniz formula of determinant of matrix $A = [a_{ij}]_{0 \leq i \leq n, 0 \leq i \leq n}$:

$$det(A) = \sum_{\sigma \in S} sgn(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

On a similar line the permanent is defined as:

$$perm(A) := \sum_{\sigma \in S} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

We saw that the determinant has a small sized arithmetic circuit, but is that the case of the permanent as well? This is still unsolved, the circuit size lower bound of the permanent:

A slight detour: Permanents II

Problem 1.2 (open)

Does there exist a polynomial sized arithmetic circuit for permanent of an $n \times n$ matrix.

We used the Laplace expansion for determinant to get the small sized circuit for the determinant, however nothing of such is known for permanent. And directly using the definition of permanent to construct a circuit will lead to circuit size to be exponential in n (Note that the number of summands in the RHS is $n!$).

VP, VNP classes

$VP$and VNP are referred to as the algebraic version of the P and NP class respectively.

Definition 1.3

VP class is a class of polynomials f that have a polynomial size circuit over a fixed field. VNP class is the class of polynomials f of polynomial degree such that given a monomial one can determine its coefficient in f efficiently, with a polynomial size circuit.

Valiant showed [Val79] that permanent is in VNP and if the permanent has a poly sized circuit then all polynomials in VNP will have a poly sized circuit. So showing permanent has a "small" circuit would imply $VP = VNP$ but it has been a long standing open problem whether that is the case.

Problem 1.3 (open)

$$VP \stackrel{?}{=} VNP$$

Questions one might ask...

1. Evaluation at a point.
2. Identity testing.
3. Differentiation.
4. Interpolation.
5. Sparse interpolation.
6. Univariate PIT

Polynomial identity testing

Problem 3.1 (open)

Given an arithmetic circuit C (in the form of graph (white box) or as a blackbox where know the evaluation at a point is fast) find if it evaluates to 0 in polynomial time (polynomial in s, n, d where s is size of circuit, n is number of variables and d is the degree of the polynomial).

P, BPP classes

Definition 3.1

Complexity class P is a collection of problems that can be solved in polynomial time.

Definition 3.2

Complexity class BPP is a collection of problems that can be solved in randomized polynomial time. More precisely this means: there is a probabilistic Turing machine such that

1. *The machine runs in polynomial time for all inputs.*
2. *The machine gives the correct answer with* $\geq 2/3$ *probability.*
3. *The machine gives an incorrect answer with* $\leq 1/3$ *probability.*

PIT in *BPP* I

There is a randomized polynomial time algorithm solving the PIT.

Lemma 3.1 (Schwartz Zippel lemma [Sax09])

$P \in F[x_1, \cdots x_n]$ be a non-zero polynomial of degree $d \geq 0$ over a field F and take S a finite subset of F. Then,

$$Prob_{r_1, \cdots r_n \in S}[P(r_1, \cdots, r_n) = 0] \leq \frac{d}{|S|}$$

From the lemma one can see that if the polynomial is nonzero then with high probability it is not going to vanish on a finite subset of F, that is PIT is solvable in randomized polynomial time. In other words, PIT is in *BPP*.

PIT in *BPP* II

Proof of lemma 3.1

Inducting on n: the case $n = 1$ is easy to check since the polynomial can have at most d roots and hence the probability of hitting a root is at most $d/|S|$.

Now, assuming the statement is valid for all polynomials in x_1 by writing it as $P(x_1, \cdots x_n) = \sum_{i=0}^{d} x_1^i P_i(x_2, \cdots x_n)$ and since P is a nonzero polynomial, $\exists i$ such that P_i is nonzero. Thus taking the largest such i, $deg\ P_i \leq (d - i)$. Now we randomly pick $r_2, \cdots r_n$ from S and applying inductive hypothesis we have $Prob[P_i(r_2, \cdots r_n) = 0] \leq \frac{d-i}{|S|}$. Note if $P_i(r_2, \cdots r_n) \neq 0$ then $P(x_1, r_2, \cdots r_n)$ is of degree i so by univariate case:

$$Prob[P(r_1, \cdots r_n) = 0 | P_i(r_2, \cdots r_n) \neq 0] \leq \frac{i}{|S|}$$

PIT in *BPP* III

Now using probabilistic bounds one can see that

$$\begin{aligned} & Prob_{r_1,\cdots,r_n \in S}[P(r_1,\cdots r_n)=0] \\ & \leq Prob[P_i(r_2,\cdots r_n)=0] + Prob[P(r_1,\cdots r_n)=0 | P_i(r_2,\cdots r_n) \neq 0] \\ & \leq \frac{d-i}{|S|} + \frac{i}{|S|} \leq \frac{d}{|S|} \end{aligned} \tag{1}$$

Significance of PIT $\overset{?}{\in} P$

Problem 3.2 (open)

Is $BPP = P$?

This is a harder problem than if $PIT \overset{?}{\in} P$, but proving PIT in P if at all that is true will be a substantial progress towards this question.

PIT for bounded depth circuits I

Theorem 3.1

Depth 2-PIT that is PIT for $\Sigma\Pi$ *and* $\Pi\Sigma$ *circuits are solved. PIT for* $\Pi\Sigma\Pi$ *is reducible to* $\Sigma\Pi$ *so is also solved.*

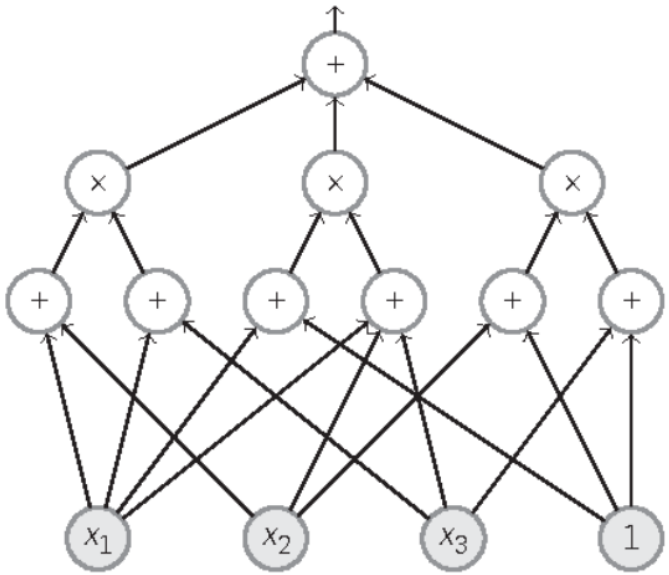

Figure: Depth 3 $\Sigma\Pi\Sigma$ circuit

PIT for bounded depth circuits II

Problem 3.3 (open)

Does there exist a polynomial time solution for solving PIT for $\Sigma\Pi\Sigma$ circuit.

How about a special case with a restricted number of in-edges on the top gate? For example consider the case $\Sigma^2\Pi\Sigma$ that is checking whether

$$\Pi_j(\Sigma a_{ij}x_i) + \Pi_j(\Sigma_i a'_{ij}x_i) = 0 \iff \Pi_j(\Sigma_i a_{ij}x_i) = -\Pi_j(\Sigma_i a'_{ij}x_i)$$

but using the property of *unique factorization* it reduces to a brute force checking if the linear combinations are equal or not and that can be done fast (polynomial in s, n time). You can ponder upon the case of constant number of such in-edges of the top gate, in fact for which PIT is solved:

Theorem 3.2 (Bounded top-fanin depth 3)

PIT for $\Sigma^{[d]}\Pi\Sigma$ circuit is solved.

PIT for bounded depth circuits III

Solving the PIT for depth 3 $\Sigma\Pi\Sigma$ case would be a big progress towards solving the general PIT since there are certain efficient depth reduction algorithms ([SY+10]) reducing to the $\Sigma\Pi\Sigma$ case. To read more about it one can refer to Nitin Saxena's survey [Sax08], Kayal and Saxena's paper on PIT for depth 3 circuits [KS07] and Shpilka's survey [SY+10].

Some applications of PIT I

1. A graph has no perfect matching if and only if the determinant of its *Tutte matrix* is zero. A Tutte matrix of graph $G = (V, E)$ is $n \times n$ matrix A :

$$A_{i,j} = \begin{cases} x_{i,j} & (i,j) \in E \text{ and } i < j \\ -x_{i,j} & (i,j) \in E \text{ and } i > j \\ 0 & \text{otherwise} \end{cases}$$

   So the perfect matching problem is reduced to the PIT of the determinant polynomial of the Tutte matrix.

2. Primality testing: It was observed by Agrawal and Biswas [AB03] that

$$n \text{ is prime } \iff (x+1)^n = (x^n + 1)(mod\ n)$$

   which can be used as:
   Define $P(x) = (x+1)^n - (x^n - +1)$ then the primality of n question reduces to testing whether $P(x)$ is a zero polynomial or not over the

Some applications of PIT II

ring $\mathbb{Z}/n\mathbb{Z}$. However $P(x)$ has degree n and we want time complexity being polynomial in $\log n$, we cannot naively expand $P(x)$. The idea that was used in [AB03] was to test $P(x) = 0(mod\ n, Q(x))$ instead, for a randomly chosen Q of degree $O(\log n)$ and as Q has a small degree we can do this in $poly(\log n)$ time, using repeated squaring of $(x+1)$ and x.

This was derandomized by Agrawal, Kayal, Saxena (AKS) [AKS04] to give the first polynomial time algorithm for primality testing.

What to do?

"If there is a problem you can't solve, then there is an easier problem you can solve: find it."

– G. Polya

Some terminology I

Definition 3.3 (Hitting set)

Given a collection $\mathcal{C}$ of n variate circuits f over a field F call a set $\mathcal{H} \subset F^n$ a hitting set of $\mathcal{C}$ when $\forall f \in \mathcal{C}, f \equiv 0 \iff f(p) = 0 \quad \forall p \in \mathcal{H}$.

Definition 3.4 (Sparsity)

The number of monomials m in a polynomial is called the sparsity of the polynomial and the polynomial is called m-sparse polynomial.

Some terminology II

Definition 3.5 (Fan-out and fan-in)

The maximum number of outputs of $+$ or $\times$ gates in the arithmetic circuit is called fan-out of the arithmetic circuits. Similarly, the maximum number of inputs of $+$ or $\times$ gates in the arithmetic circuit is called fan-out of the arithmetic circuits.

Definition 3.6 (Noncommutative circuits)

A circuit $C(x_1, \cdots, x_n)$ over an algebra over a field F is called is noncommutative if each of its multiplication gate has ordered inputs and the variables do not commutate that $x_i x_j \neq x_j x_i$.

Some known special cases

1. Sparse polynomials.
2. Depth-3 diagonal circuits $\Sigma\Lambda\Sigma$ (whitebox).

Sparse case I

Theorem 3.3 ([Sax09])

$p(x_1, \cdots x_n)$ is a nonzero polynomial over field F with degree in each variable $< d$ and sparsity $< m$ then there is an $1 \leq r \leq (mn \log d)^2$ such that $p(y, y^d, \cdots, y^{d^n-1})(mod\ y^r - 1) \neq 0$

Proof:

Consider $q(y) := p(y, y^d, \cdots, y^{d^n-1})$ in $F[y]$ and note that in p the monomial $x_1^{i_1} \cdots x_n^{i_n}$ is mapped to monomial $y^{i_1 + i_2 d + \cdots i_n d^{n01}}$ in q and this is one-one map since we have assumed $i_1, \cdots, i_n < d$ (and for a number d base representation is unique). Therefore $q(y) \neq 0$. Take y^a a monomial with nonzero coefficient in q and look at $q(y)(mod\ y^r - 1)$.
If $q(y) = 0(mod\ y^r - 1)$ then $\exists$ monomial y^b such that $y^b \neq y^a$ in q for which $y^b = y^a(mod\ y^r - 1)$. But this is possible if and only if $r|(b - a)$

Sparse case II

(because $(y^r - 1)|(y^s - 1) \iff r|s$). Thus, to avoid picking such a "bad" r we need one that satisfies

$$r \nmid \prod_{y^b \in q(y), b \neq a} (b - a) =: R$$

Note, integer R can be at most $(d^n)^m$ in value. Since R has at most $\log R$ prime factors and since we would consider encounter at least $(\log R + 1)$ primes in the range $1 < r \leq (\log R)^2 = (mn \log d)^2$ which implies that we have the required (prime) r such that $q(y) \neq 0 (mod\ y^r - 1)$.

Sparse case III

Corollary 3.1

Sparse PIT can be solved in poly(s, n, m) where m is the sparsity of the polynomial.

This follows immediately from the theorem above: Given an $m-$sparse circuit $C(x_1, \cdots, x_n)$ over F, fix $d := 2^{size(C)}$ and for every $1 \leq r \leq (mn \log d)^2$: compute $d, d^2, \cdots d^{n-1} (mod\ r)$ using repeated squaring and then find the evaluations of $C(y, y^d, \cdots y^{d^n-1})$ over $F[y]/(y^r - 1)$ and we declare C is zero if and only if all the evaluations are zero.

Depth-3 diagonal whitebox circuit I

Definition 3.7

Depth-3 diagonal circuits $\Sigma \wedge \Sigma$ circuits are arithmetic circuits that computes polynomials of the form

$$p(x) = \sum_{j=1}^{k} c_i (\sum_{i=1}^{n} a_{ij} x_i)^{d_j}$$

Basically, the Λ gate computes the power of its input using product gate.

Depth-3 diagonal whitebox circuit II

Theorem 3.4 ([Sax08])

Over characteristic zero field, whitebox PIT for diagonal circuits $\sum_{j=1}^{k}(\sum_{i=1}^{n} a_{ij}x_i)^{d_j}$ *can be done in deterministic polynomial* $poly(nkd)$ *time where* $d = \max_j d_j$.

This theorem was proven by Nitin Saxena in his 2008 paper [Sax08] using duality trick which we state here as a lemma.

Duality trick I

Lemma 3.2 ([Sax08])

Take $a_0, \cdots a_n \in \mathbb{F}$ of zero characteristic. Then we can compute univariate polynomials $f_{i,j}$'s in poly(nd) field operations such that for $t = (nd + d + 1)$

$$(a_0 + a_1 x_1 + \cdots a_n x_n)^d = \sum_{i=1}^{t} f_{i,1}(x_1) \cdots f_{i,n}(x_n)$$

Proof:

Consider the exponential formal power series,

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots$$

Duality trick II

The degree d truncation of e^x be done as $T_d(x) = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^d}{d!}$.
Observe that:

$$\begin{aligned}
&(d!)^{-1} \cdot (a_0 + a_1 x_1 + \cdots a_n x_n)^d \\
&= \text{coefficient of } z^d \text{ in } e^{((a_0 + a_1 x_1 + \cdots + a_n x_n) \cdot z)} \\
&= \text{coefficient of } z^d \text{ in } e^{a_0 z} \cdot e^{a_1 x_1 z} \cdots e^{a_n x_n z} \\
&= \text{coefficient of } z^d \text{ in } T_d(a_0 z) \cdot T_d(a_1 x_1 z) \cdots T_d(a_n x_n z)
\end{aligned} \tag{2}$$

Now the product of $T_d(a_0 z) \cdot T_d(a_1 x_1 z) \cdots T_d(a_n x_n z)$ can be viewed as a

Duality trick III

univariate polynomial in z of degree $(nd + d)$. Thus its coefficient of z^d can be computed by evaluating the polynomials at $(nd + d + 1)$ distinct points $\alpha_1, \cdots \alpha_{nd+d+1} \in \mathbb{F}$ and by interpolation we can compute $\beta_1, \cdots, \beta_{nd+d+1}$ such that

$$\begin{aligned} &\text{coefficient of } z^d \text{ in } T_d(a_0 z) \cdot T_d(a_1 x_1 z) \cdots T_d(a_n x_n z) \\ &= \sum_{i=1}^{nd+d+1} T_d(a_0 \alpha) \cdot T_d(a_1 x_1 \alpha) \cdots T_d(a_n x_n \alpha) \end{aligned} \tag{3}$$

This is the *dual form* of the expression $(a_0 + a_1 x_1 + \cdots a_n x_n)^d$ as required. It can be seen that all the polynomials T_ds can be computed in $poly(nd)$ time (i.e. field operations).

Proof of theorem 3.4[Sax08] I

The proof uses the following theorem which we do not prove here:

Theorem 3.5 ([RS05] theorem 5 section 2)

Given a noncommutative arithmetic formula (circuit with fan-out of every gate= 1) of size we s we can verify in time polynomial in s whether the formula is identically zero or not.

Proof of theorem 3.4:

Suppose we are given a diagonal circuit C:

$$C(x_1, \cdots x_n) = \sum_{i=1}^{k} c_i l_i^{d_i}$$

Proof of theorem 3.4[Sax08] II

where $l_i = \sum_{i=1}^{n} a_{ij}x_i$ are linear combinations. Then using the lemma 3.2 we can compute the dual form of each of the k multiplication gates such that

$$C(x_1, \cdots x_n) = \sum_{i=1}^{k} \sum_{j=1}^{nd_i+d_i+1} f_{i,j,1}f_{i,j,1}(x_1) \cdots f_{i,j,n}(x_n)$$

where the univariate polynomials $f_{i,j,j'}$s are of degree at most d_i.
Now observing that the variables in the circuit on the RHS of equation above can be assumed to be noncommutative without affecting the output (because $\sum_k \prod_i p_{ik}(x_i)$ commutative is zero if and only if it is zero considering x_is to be noncommutative), i.e. to the circuit C we can apply Raz Sphilka's identity testing algorithm ([RS05] theorem 5 section 2: refer to 3.5) to the circuit (arithmetic formula) on the RHS of the equation above we know deterministically whether C is zero or not. Hence PIT for C is solved in $poly(nkd)$ time (field operations).

Some interesting open special cases

1. Depth-3 Diagonal circuits (blackbox).
2. Orbit of Sparse polynomial.

Diagonal circuits

Problem 3.4 (open)

Given a blackbox access to the circuit C_P of the polynomial of the form $P = \sum_j (\sum_i a_{ij})^{d_j}$ find if the polynomial P is zero polynomial or not in polynomial time (i.e. polynomial in size of circuit, number of variables and degree).

Since we are not given the whitebox access to the circuit (that is the circuit is not explicitly given) the duality trick doesn't work anymore.

Orbit of Sparse polynomials

Problem 3.5 (open)

Given $C_{f \circ A}$ the circuit of a polynomial $f(A(x))$ in the orbit of sparse polynomial f where $A \in GL_n(\mathbb{R})$, find if the f is identity or not in polynomial time (i.e. polynomial in size of circuit, number of variables and degree).

As an easier exercise check why does the PIT solving algorithm for sparse polynomial doesn't work here directly.

Other questions: Differentiation of circuits

Theorem 4.1 ([SY+10; BS83])

Given a n-variate polynomial f coming from arithmetic circuit C_f of size s and depth d. Then there exist a circuit C'_f of size $O(s)$ and depth $O(d)$ computing the polynomials $\partial_{x_1}(f), \cdots \partial_{x_n}(f)$ simultaneously.

That is the first order partial derivatives can be computed with linear blowup in circuit size and depth. It is however not known if it can be done with an arithmetic circuit of the same size. And what about the case of the second derivative?

Problem 4.1 ((open) [SY+10])

Does an analog of theorem 4.1 hold for second order partial derivatives?

Other questions: Interpolation

Problem 4.2

Given blackbox or whitebox access to a circuit find the polynomial corresponding to the circuit in polynomial in s, n, d time.

Note that it is stronger than finding the PIT.

Other questions: Sparse interpolation

Problem 4.3

Given a blackbox or whitebox access to a circuit of a sparse polynomial, find the sparse polynomial in polynomial in s, n, d time.

One can read more about it in Grenet's recent workshop talk [Gre23].

Other questions: Univariate PIT I

Problem 4.4 (open)

Given a blackbox or whitebox access to a circuit of a univariate polynomial $p(x)$ of size s and degree d. Determine if $p(x)$ is zero or not in poly(s) time.

Note that the general PIT can be reduced to univariate PIT using a univariate substitution for example with a exponential degree blowup we can have

$$x_i \to x^{d^i} \qquad \text{where } d = \max_i d_i$$

This works because if there exist monomials $\prod x_i^{d_i} - \prod x_i^{d_i'} \neq 0$ in a polynomial they can never cancel out after the substitution because if they cancels out that would imply there exist d_i, d_i' such that:

$$\sum_i d^i d_i = \sum_i d^i d_i' \iff d_i = d_i' \text{ since } d \text{ base representation is unique}$$

Other questions: Univariate PIT II

which is a contradiction since $\prod x_i^{d_i} - \prod x_i^{d_i'} \neq 0$.
Solving univariate PIT in $poly(d)$ is trivial since a degree d univariate polynomial over a field can not have more than d roots. However s sized univariate circuit can have $O(exp(s))$ degree polynomial for example:

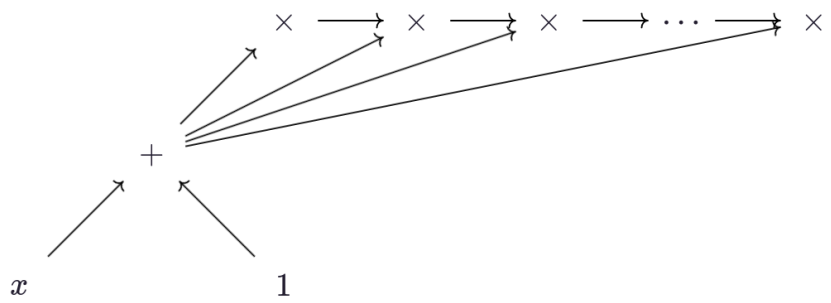

Figure: s sized circuit with degree 2^s

Acknowledgement

I am thankful to Prof. Amit Kumar Sinhababu, CMI, who introduced me to this world of seemingly accessible long open problems through a summer project.

Bibliography I

[AB03] Manindra Agrawal and Somenath Biswas. "Primality and identity testing via Chinese remaindering". In: *Journal of the ACM (JACM)* 50.4 (2003), pp. 429–443.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. "PRIMES is in P". In: *Annals of mathematics* (2004), pp. 781–793.

[BS83] Walter Baur and Volker Strassen. "The complexity of partial derivatives". In: *Theoretical computer science* 22.3 (1983), pp. 317–330.

[Gre23] Bruno Grenet. *Sparse polynomial interpolation.* Slides: https://www.dcs.warwick.ac.uk/~u2270030/act2023icalp/slides/ICALP2023ACT_grenet.pdf. 2023.

[KS07] Neeraj Kayal and Nitin Saxena. "Polynomial identity testing for depth 3 circuits". In: *computational complexity* 16 (2007), pp. 115–138.

Bibliography II

[RS05] Ran Raz and Amir Shpilka. "Deterministic polynomial identity testing in non-commutative models". In: *computational complexity* 14 (2005), pp. 1–19.

[Sax08] Nitin Saxena. "Diagonal circuit identity testing and lower bounds". In: *Automata, Languages and Programming: 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I 35*. Springer. 2008, pp. 60–71.

[Sax09] Nitin Saxena. "Progress on Polynomial Identity Testing.". In: *Bull. EATCS* 99 (2009), pp. 49–79.

[SY+10] Amir Shpilka, Amir Yehudayoff, et al. "Arithmetic circuits: A survey of recent results and open questions". In: *Foundations and Trends® in Theoretical Computer Science* 5.3–4 (2010), pp. 207–388.

Bibliography III

[Val79] Leslie G Valiant. "Completeness classes in algebra". In: *Proceedings of the eleventh annual ACM symposium on Theory of computing*. 1979, pp. 249–261.

Thank you!