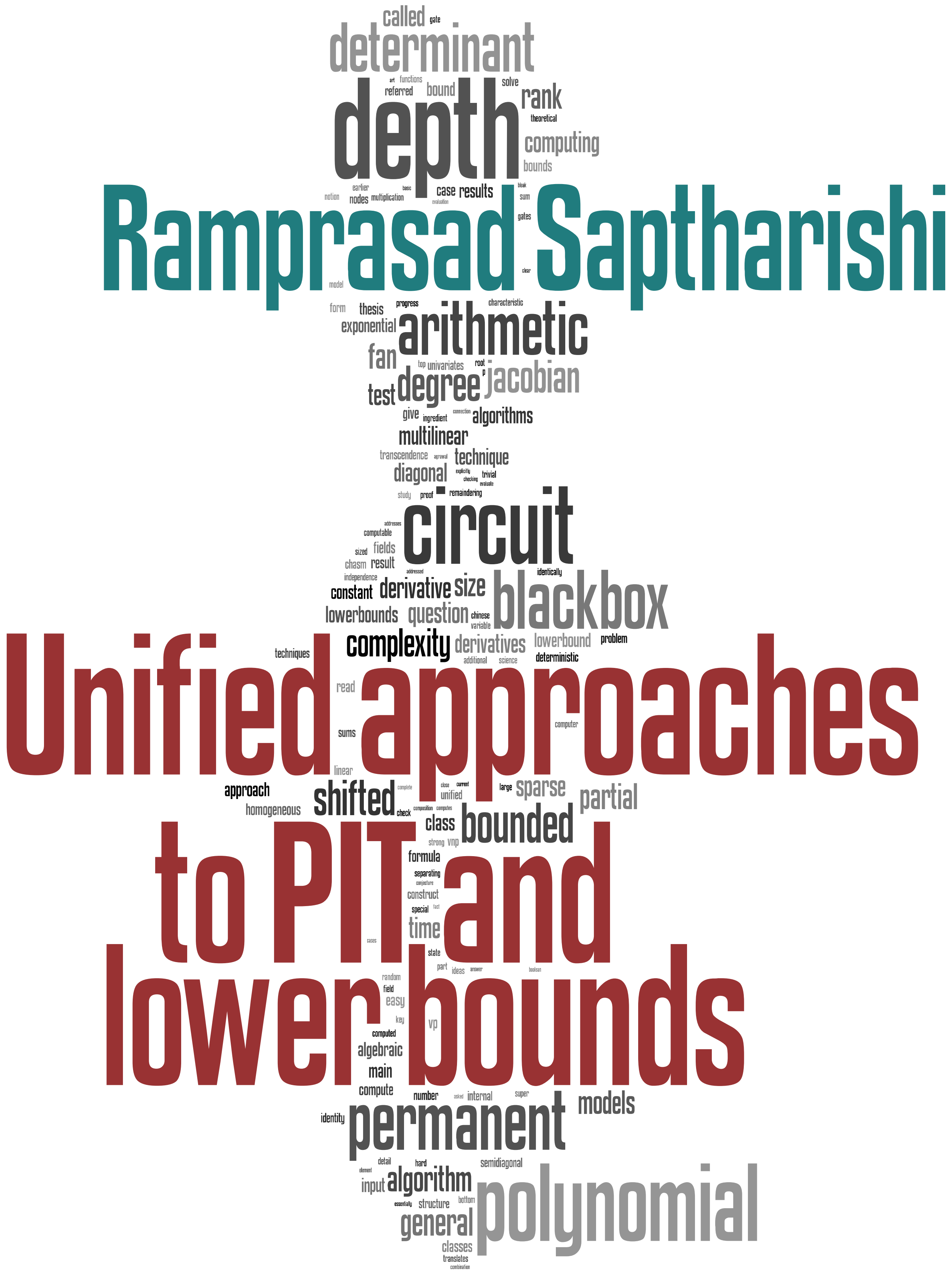

called
gate
determinant
art
functions
referred
bound
solve
rank
depth
theoretical
computing
bounds
earlier
basic
notion
nodes
multiplication
case
results
bleak
sum
evaluation
Ramprasad Saptharishi
gates
clear
model
form
thesis
progress
characteristic
exponential
arithmetic
fan
top
univariates
root
jacobian
test
degree
give
ingredient
connection
algorithms
multilinear
transcendence
agrawal
technique
diagonal
explicitly
trivial
checking
evaluate
study
proof
remaindering
addresses
computable
circuit
fields
sized
chasm
result
addressed
identically
independence
constant
derivative
size
blackbox
lowerbounds
question
chinese
variable
complexity
derivatives
lowerbound
problem
additional
science
deterministic
techniques
Unified approaches
read
computer
sums
linear
approach
complete
close
current
large
sparse
unified
partial
shifted
homogeneous
composition
computes
check
class
bounded
strong
vnp
to PIT and
formula
separating
conjecture
construct
special
fact
time
cases
state
part
ideas
answer
boolean
lower bounds
random
field
easy
key
vp
computed
algebraic
main
compute
number
asked
internal
super
models
identity
permanent
detail
hard
semidiagonal
element
input
algorithm
essentially
structure
bottom
polynomial
general
classes
translates
combination

Unified Approaches to

Polynomial Identity Testing

and Lower Bounds

Thesis submitted in
Partial Fulfilment of the
Requirement of the Degree of

Doctor of Philosophy (Ph.D)
in Computer Science

by

Ramprasad Saptharishi

Chennai Mathematical Institute
Plot H1 SIPCOT IT Park
Padur PO, Siruseri 603 103

April 2013

Declaration

I declare that the thesis "Unified Approaches to Polynomial Identity Testing and Lower Bounds" submitted by me for the degree of Doctor of Philosophy is the record of work carried out by me during the period from August 2007 to April 2013 under the guidance of Prof. Manindra Agrawal. This work has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this or any other university or other similar institution of higher learning.

April 2013 Ramprasad Saptharishi

Chennai Mathematical Institute
Plot H1, SIPCOT IT Park, Siruseri,
Kelambakkam 603103
India

Certificate

This is to certify that the Ph.D. thesis submitted by Ramprasad Saptharishi to Chennai Mathematical Institute, titled "Unified Approaches to Polynomial Identity Testing and Lower Bounds" is a record of *bona fide* research work done during the period 2007 – 2013 under my guidance and supervision. The research work presented in this thesis has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this institute or any other university or institution of higher learning.

It is further certified that the thesis represents independent work by the candidate and collaboration when existed was necessitated by the nature and scope of problems dealt with.

Manindra Agrawal
(Thesis superviser)

Chennai Mathematical Institute
Plot H1, SIPCOT IT Park, Siruseri,
Kelambakkam 603103
India

&

Indian Institute of Technology
Department of CSE
Kanpur, 208016
India

Synopsis

The field of arithmetic circuit complexity aims towards understanding the complexity of polynomials with respect to the number of additions and multiplications required to compute it. The most important problem in the field of arithmetic circuit complexity is to find an explicit polynomial that requires super-polynomially many operations to compute it. The *permanent* is widely conjectured to be such a polynomial, though its illustrious sibling – the *determinant* – can indeed be computed by polynomial sized circuits. Separating the complexity of the determinant from the permanent is a question of foremost important in this field. In fact, this question was formalized by Valiant [Val79] as an algebraic analogue of the "P vs NP" question.

Over the last few decades, this problem has received a lot of attention by several researchers but has been resilient to numerous attacks. Apart from direct attempts at obtaining a lower bound, a very surprising connection was established with another problem of prime importance — *polynomial identity testing* (PIT). PIT is the task of check if the polynomial computed by a given input circuit is identically zero or not. Though this problem admits a very straightforward randomized algorithm (which is to evaluate the circuit at a random point), a deterministic polynomial time solution has eluded us so far. The connection of PIT (in spirit) says that efficient deterministic algorithms for PIT would yield arithmetic (and boolean) circuit lower bounds [KI03, Agr05].

For both lower bounds and PIT, there has been limited progress in various restricted models of circuits/formulas. Super-polynomial lower bounds were shown for *multilinear* circuits computing the determinant or permanent [Raz09]. Exponential lower bounds for the determinant and permanent were also shown for depth-3 circuits over finite fields [GK98], *homogeneous* depth-3 circuits [NW97], and for *monotone* circuits [JS82]. However, none of the above results separate the complexity of the determinant and permanent. Towards this, there has only been a $\Omega(n^2)$ lower bound for the *determinantal complexity* of the permanent [MR04].

Polynomial time algorithms for PIT have been obtained for depth-2 circuits [AB99, KS01], bounded fan-in depth-3 circuits [KS07, SS11], diagonal circuits [Kay10, Sax08], bounded transcendence degree depth-4 circuits [BMS11a], bounded fan-in multilinear depth-4 circuits [SV11], multilinear read-k formulas [AvMV11].

Progress for both PIT and lower bounds seem to halt before depth-4 circuits (without additional restrictions like multilinearity, read etc.). This "chasm" was explained [AV08, Koi10] by showing that depth-4 circuits are almost the general case.

Contributions of this thesis

Though quite a lot of progress for PIT has been made for these restricted models, one issue is the solution in each model involves a different technique tailored to work for the models — univariate substitutions, chinese remaindering over local rings, algebraic independence, duality transforms, shattering of partial derivatives, sparsity bounds etc. As a result, it is not clear if all these special cases throw insight into the general problem at all. This thesis address the following natural questions — What is the central core that makes PIT on these models easy? Is there a unified technique that encompasses all these models? Can PITs of two different models be "composed" to work on "combinations" of these models? Do lower bound technique for depth-3 circuits lift to larger depth?

Composition of PITs

If we know how to perform PIT on two different classes $\mathscr{C}_1$ and $\mathscr{C}_2$, can perform PIT on circuits from $\mathscr{C}_1 + \mathscr{C}_2$? This question depends on the classes $\mathscr{C}_1$ and $\mathscr{C}_2$ of course. If f is a depth-2 circuit, and $g = g_1 \dots g_t$ is a product of depth-2 circuits, then checking if $f = g_1 \dots g_t$ is exactly the problem of validating *sparse factorization* (cf. [vzG83]). The simplest models for which PITs are known are bounded fan-in depth-3 circuits and diagonal circuits. Can the two seemingly disparate techniques glue together to give a PIT for the composed circuits? In Chapter 3, we answer this question in the affirmative. Our technique also applies to a special case of the depth-4 problem $f - \prod_{i=1}^{t} g_i \stackrel{?}{=} 0$ where each of the g_i's are sums of univariates.

A unified technique for PITs

The next question the thesis addresses is a unification of the PITs for different models. In Chapter 4, we present a unified approach via *algebraic independence* and the *jacobian*.

We show that the Jacobian is a powerful tool to give one unified approach for *blackbox* PITs for all the classes for which polynomial time *blackbox* PITs were known, namely bounded fan-in depth-3, bounded fan-in depth-4 multilinear circuits , bounded read bounded depth multilinear circuits, and bounded transcendence degree depth-4 circuits. The approach however has one caveat that the method works over all fields with zero or large chracteristic.

In the process of unifying the varied techniques, we strengthen the earlier results significantly thus giving the first blackbox PIT for these generalized models. We construct blackbox PITs for not only bounded fan-in depth-3 circuits, but also for circuits of the form $C(T_1, \cdots, T_m)$ where C is *any* polynomial of low degree and T_i's are products of linear functions with bounded transcendence degree. Further, we remove the multilinear restriction completely from the constant-depth constant-read models. The notion of 'read' is also replaced by a general notion of '*occur*', which additionally generalizes PIT on $\Sigma\Pi\Sigma\Pi^{[\mathrm{mult}]}(k)$ circuits as well.

The connection between PIT and lowerbounds is present in this unification as well. We present lowerbounds for the Det_n and Perm_n for almost all the models we construct PITs for, again via the Jacobian.

Approaching the chasm at depth-4

The final question addressed in this thesis is a direct attempt towards a lower bound for the permanent. A result of Agrawal and Vinay [AV08] states that we only need to prove a strong enough lowerbound for the class of depth-4 circuits. This was subsequently strengthened by Koiran [Koi10] to show that it suffices to prove a lower bound of $\exp(\omega(\sqrt{n}\log^2 n))$ for *homogeneous* depth-4 circuit with the bottom multiplication gates having $O(\sqrt{n})$ fan-in to obtain super-polynomial lower bounds for general circuits.

In Chapter 5 we present a lower bound of $\exp(\Omega(\sqrt{n}))$ for homogeneous depth-4 circuits where the fan-in of the bottom multiplication gates having $O(\sqrt{n})$ fan-in that compute Det_n or Perm_n. In the light of above results of Agrawal-Vinay [AV08] and Koiran [Koi10] that a $\exp\left(\omega\left(\sqrt{n}\log^2 n\right)\right)$ lower bound suffices for a superpolynomial circuit lower bound, our result does get *very* close to the chasm. Further, we also exhibit that our method has the potential to show non-trivial separations between the complexities of the determinant and permanent (albeit the above result works for both). It is conceivable that this indeed might be one of the ingredients in separating VP and VNP.

Acknowledgements

I have been very fortunate to have Manindra Agrawal as my advisor. My stay at IIT Kanpur could not have been possible if not for his resourcefulness. He has helped me immensely throughout the last several years (right from helping me move my luggage on my first day) and I feel truly indebted to him. His tenacity to focus only on fundamental problems has been a constant source of inspiration. I hope that I can display a fraction of it during the years to come.

My interest in complexity theory stemmed from the wonderful courses I had taken over the years, and my interaction with the faculty at CMI and IMSc. I'd like to thank Madhavan, Kumar, KV, Samir, Arvind, Meena and Suresh for all the courses and discussions we've had. I'd specially like to thank Arvind for his wonderful courses on complexity and computational algebra that sparked my interest in this field.

I'd also like to thank the faculty at IIT Kanpur with whom I've interacted during my stay here. I've throughly enjoyed interacting with Somenath, Piyush, Sumit, Satyadev, Shashank and Surendar (and his classes were truly amazing!).

It was wonderful to work with all my collaborators, and I learnt a great deal from each of them. Chandan's enthusiasm and sincerity is contagious and a big thanks to him for putting up with me for all these years. I learnt an enormous amount of mathematics through all the collaborations with Nitin. Neeraj's phenomenal clarity and focus has moved me deeply, and I hope to emulate his drive for concrete big problems in future. Many thanks to Piyush, Anindya, Ankit and Pritish as well.

I had the opportunity to make several academic visits during the term. I had an amazing time at TIFR interacting with Prahladh and Jaikumar. I wish there is a day when I could give a talk like Jaikumar. I'd also like to thank Madhu for hosting me at MIT. Thanks to Kurt Mehlhorn for the wonderful visit at MPI Saarbrücken. And a big thanks to Satya and Neeraj for the internship at MSR. The internship was the best academic visit I've ever had, and working with Neeraj was wondefully illuminating.

Neeraj and Satya helped me in more ways than one, and I shall always be grateful for their unconditional support.

Life in IIT and CMI wouldn't have been half as much fun if not for the company I had there. Many thanks to Srikanth, Nutan, Suresh, Chandan, Deepanjan, Sagarmoy, Suman, Mousumi, and Saurabh. I'd specially like to thank Srikanth and Chandan for their lasting influence. Srikanth's unparalleled enthusiasm for knowledge doesn't take too long to catch on, and I've learnt so much interacting with him. Chandan has been subjected to several hours of torture by me, but ever-so-patiently answers every question irrespective of how dumb it is. His astounding clarity in exposition is something I hope to be capable of someday. I also feel compelled to thank Deepanjan and Suman for all the *Spordha* sessions.

Many thanks to my uncle, Vinay, for ... everything. He was the reason I took to mathematics and theoretical computer science in the first place, and has been an idol since my childhood. My sincere thanks to him and Subha chitthi for everything that I am today.

A big thank you to Susmita for sharing all my joy and pain, and life would have been empty without her. Most of all, thanks to my parents, my sister Kamakshi, my grandfather and grandmother for the wonderful home I grew up in. My humble *namaskarams* to my grandfather specially, who would have been very proud to see this, and it is to him that I dedicate this thesis.

To my grandfather

Contents

Introduction

1

Over the last few decades, research in theoretical computer science has become sophisticated. We have acquired a variety of tools from several fields, and used them to better our understanding of computation. Several constructs from mathematics have now become standard tools in theoretical computer science research, and the distinction (if there was any) between mathematics and theoretical computer science has become fuzzy.

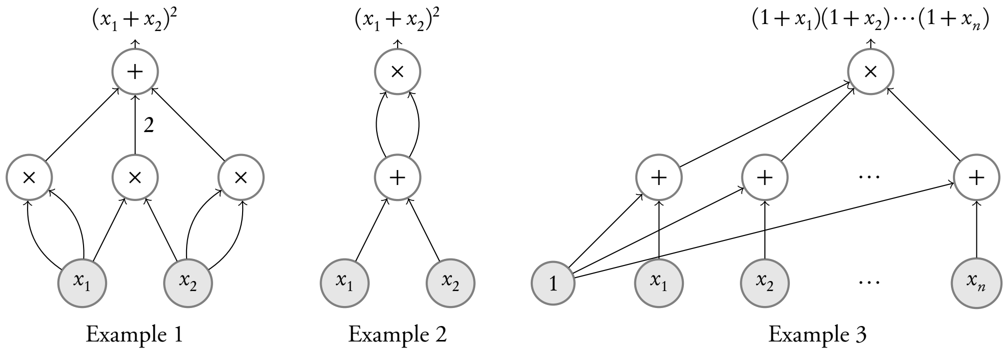
Possibly the most basic of mathematical objects are *polynomials* and we would like to understand the hardness of various natural problems concerning them. The most robust way to measure the hardness of the polynomial is via the *number of operations* required to compute it, and this notion formalized via *arithmetic circuits*.

Definition 1.1 (Arithmetic Circuits and formulas). *An* arithmetic circuit *is a directed acyclic graph with one sink (which is called the output gate or root). Each of the source vertices (which are called input nodes) are either labelled by a variable x_i or an element from an underlying field $\mathbb{F}$. Each of the internal nodes are labelled either by $+$ or $\times$ to indicate if it is an addition or multiplication gate respectively. Sometimes edges may carry weights that are elements from the field and this amounts to scaling the polynomial computed at the incoming node by that field element.*

An arithmetic circuit is a formula *if every internal node has out-degree* 1*. The* size *of the circuit/formula is the number of nodes in the graph, and the* depth *is the length of the longest path from the leaf to the root.*

Arithmetic circuits provide a very compact way of representing polynomials, and this can be seen in Example 3 which has 2^n terms on expanding fully. For any polynomial p, the size of the smallest arithmetic circuit computing it can be thought of as the *arithmetic complexity* of p. We would like to know if there are explicit examples of polynomials that are *very hard to compute*, or have very large arithmetic complexity.

Figure 1.1: Arithmetic Circuits



1.1 Lower Bounds

Amongst the multitude of polynomials, these two are of the highest significance — the *determinant* and *permanent* polynomial (denoted by Det_n and Perm_n respectively).

$$\mathsf{Det}_n = \sum_{\sigma\in S_n} \mathrm{sign}(\sigma)\cdot x_{1\sigma(1)}\dots x_{n\sigma(n)}$$

$$\mathsf{Perm}_n = \sum_{\sigma\in S_n} x_{1\sigma(1)}\dots x_{n\sigma(n)}$$

Although these two polynomials look similar, they seem to be very different computationally. The polynomial Det_n is known to be computable by arithmetic circuits of polynomial size (in n) but Perm_n is believed to have exponential complexity. Valiant [Val79] defined the classes VP and VNP as algebraic analogues of the boolean complexity classes P and NP. He showed that Det_n was *complete* for VP, and Perm_n was *complete* for VNP. Further, he showed that $\mathsf{VP} \neq \mathsf{VNP}$ must necessarily be resolved before showing $\mathsf{P} \neq \mathsf{NP}$. Separating the complexity of the determinant and permanent is also referred to as the *determinant vs permanent conjecture*, and is perhaps the most important question in arithmetic complexity theory.

> *"The determinant of this conjecture would be permanently famous."*
>
> – Neeraj Kayal

The current state of the art for lowerbounds for general circuits is rather bleak. Baur and Strassen [BS83] showed that any circuit computing the polynomial $x_1^r + \cdots + x_n^r$ must have size $\Omega(n \log r)$. This is the *only* super-linear lowerbound known for general circuits. For *arithmetic formulae*, Kalorkoti [Kal85a] showed that any *arithmetic formula* computing Det_n or Perm_n must have size $\Omega(n^3)$.

With some additional restrictions on the circuit/formula, further lowerbounds are known. Jerrum and Snir [JS82] showed that any *monotone circuit* computing Det_n or Perm_n requires $2^{\Omega(n)}$ size. Raz [Raz09] showed that any *multilinear circuit* computing Det_n or Perm_n requires $n^{\Omega(\log n)}$ size.

For constant depth circuits, the situation is more respectable. Grigoriev and Karpinski [GK98] showed that any depth-3 circuit computing Det_n over finite fields must have size $2^{\Omega(n)}$. But over characteristic zero fields, we only have an $\Omega(n^4/\log n)$ lower bound, due to Shpilka and Wigderson [SW01], for Det_n and Perm_n. For *homogeneous* depth-3 circuits, Nisan and Wigderson [NW97] gave a $2^{\Omega(n)}$ lowerbound for Det_n and Perm_n.

All the above results do not separate the complexity of Perm_n from Det_n. In the context of separation, Mignon and Ressayre [MR04] showed that the *determinental complexity* of the Perm_n is $\Omega(n^2)$. More formally, if one wishes to write Perm_n as the determinant of some $m \times m$ matrix with affine function entries, then $m = \Omega(n^2)$.

1.2 Polynomial Identity Testing

There are several algorithmic questions that can be asked about arithmetic circuits, and the simplest (to state) amongst them is *polynomial identity testing (PIT)* — given an arithmetic circuit C as input, test if the polynomial computed by C is identically zero.

Though a circuit of size s can potentially compute a polynomial of degree 2^s, the input circuit for PIT is usually promised to have *low degree*, i.e. degree polynomially bounded in the number of variables. This problem is sometimes referred to as *low degree PIT*, but throughout this thesis we will restrict ourselves to only low degree PIT and refer to it as simply 'PIT'. The circuit is usually assumed to be layered with alternating layers of $+$ and $\times$ gates which are referred to as Σ and Π layers respectively.

There are two types of algorithms for PIT that are studied — *blackbox* and *non-*

blackbox. A *blackbox PIT* is an algorithm that tests if a given circuit computes the zero polynomial by *only* evaluating the circuit on points, and not inspecting the internal structure of the circuit. Hence all that a blackbox PIT can do is evaluate the circuit on a small list of points which is guaranteed to have a property that every non-zero circuit produces at least one non-zero evaluation in the list. Such lists are also called *hitting sets*, the blackbox PITs are also called *hitting set generators*. Algorithms that use the internal structure of the input circuit are called *non-blackbox* algorithms.

PIT is an essential ingredient in several important results in complexity theory and algorithms. It plays a pivotal role in the primality test [AKS04], graph matching algorithms [Lov79, MVV87], the PCP theorem [ALM+98], interactive protocols [LFKN90, Sha90] etc. and there are several more. Further, it has a surprising connection to arithmetic and boolean circuit lower bounds. Kabanets and Impagliazzo [KI03] showed that a subexponential algorithm for PIT would imply that either Perm_n does not have polynomial sized arithmetic circuits, or $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$. A similar conclusion was also established for constant depth circuits by Dvir, Shpilka and Yehudayoff [DSY08]. Heintz-Shnorr [HS80] and Agrawal [Agr05] showed that a polynomial time blackbox algorithm for PIT on general circuits implies an exponential lowerbound for a polynomial computable in PSPACE. Under additional structural assumptions on the blackbox PIT, Agrawal [Agr11] showed that such a strong blackbox PIT would separate VP and VNP.

PIT admits a very simple randomized polynomial time algorithm — evaluate the polynomial on a random point. This is also called the Schwartz-Zippel test [Sch80, Zip79, DL78]. There has been a plethora of results following this test to reduce the number of random bits [CK97, LV98, KS01, AB99].

The Schwartz-Zippel test essentially states that a random evaluation of a non-zero polynomial is non-zero with high probability. Specifically, it does not really use the fact that the polynomial is computable by a small arithmetic circuit. A natural question to ask if we can use this structure to construct deterministic polynomial time algorithms for PIT.

For general circuits, we do not yet have a subexponential time deterministic algorithm for PIT. However, there has been a lot of progress on several restricted models of circuits using a variety of techniques.

1. *Depth-*2 *circuits (or* $\Sigma\Pi$ *circuits)*:

   These circuits have a trivial non-blackbox algorithm to test if it is identically zero as any such circuit describes a polynomial explicitly as a sum of monomials (such polynomials are also called *sparse polynomials*). We also have blackbox PITs for $\Sigma\Pi$ circuits [KS01, AB99] and we shall see this in more detail in Section 2.1.

2. *Bounded fanin depth-*3 *circuits (or* $\Sigma\Pi\Sigma(k)$ *circuits)*:

   These are circuits computing a polynomial of the form $C = T_1 + \cdots + T_k$ where each T_i is a product of linear functions $\ell_{i1} \ldots \ell_{id}$. Kayal and Saxena [KS07] gave a deterministic non-blackbox algorithm running in time $\mathsf{poly}(n, d^k)$. The main idea in their approach was chinese remaindering over local rings (and we will see this in more detail in Chapter 3).

   After a series of results on *rank estimates* for $\Sigma\Pi\Sigma(k)$ circuits[DS05, SS09, KS09b, SS10a], Saxena and Seshadhri[SS11] gave a $\mathsf{poly}(n, d^k)$ blackbox algorithm for $\Sigma\Pi\Sigma(k)$ circuits using a combination of the chinese remaindering ideas and *rank preserving homomorphisms* from Dvir, Gabizon and Wigderson[DGW09].

3. *Depth-*3 *diagonal circuits*:

   This is a special class of depth-3 circuits computing a polynomial of the form $C = \ell_1^d + \cdots \ell_s^d$, a sum of *powers* of linear functions. A more general definition of *semidiagonal circuits* (Definition 3.1) would be studied in Chapter 3.

   There are two different non-blackbox algorithms known for this class, one by Kayal [Kay10] using the partial derivative method, and one by Saxena [Sax08] using the *dual representation*. Very recently, Agrawal, Saha and Saxena [ASS12], and independently Forbes and Shpilka [FS12] gave two different $n^{O(\log n)}$-time blackbox PIT for diagonal circuits.

4. *Depth-*4*, bounded transcendence degree circuits*:

   This class of circuit shall be dealt with in more detail in Chapter 4 but an illustrative example is a circuit that depends on k sparse polynomials, i.e. $C =$

$f(g_1, \cdots, g_k)$ where f is a low-degree polynomial. Beecken, Mittman and Saxena [BMS11a] gave an $\mathsf{poly}(s^k)$ blackbox PIT for this class of circuits of fields of zero or large characteristic. The main ingredient in their proof was *algebraic independence* and the *Jacobian* (which would be seen in more detail in Chapter 4).

5. *Depth-4, bounded fan-in, multilinear circuits (or $\Sigma\Pi\Sigma\Pi^{[mult]}(k)$)*:

   A multilinear circuit is one where every gate computes a multilinear polynomial (degree of every variable is bounded by 1). Saraf and Volkovich [SV11] gave a deterministic blackbox algorithm to check if a $\Sigma\Pi\Sigma\Pi^{[\text{mult}]}$ circuit of top fanin k is identically zero running in $\mathsf{poly}(s^{k^3})$ time (where s is the size of the circuit). The main ideas used in their approach was to study the partial derivatives of the circuit and obtain *sparsity bounds* on the polynomials computed by the children of the root.

6. *Read-k, multilinear formula*:

   A *read-k* formula is one where every input variable occurs in at most k of the leaf nodes. Anderson, van Melkebeek and Volkovich[AvMV11] extended the ideas in the PIT for $\Sigma\Pi\Sigma\Pi^{[\text{mult}]}(k)$ circuits to give PITs for read-k multilinear circuits. They give an $n^{k^{O(k)}}$-time non-blackbox PIT, and a $n^{k^{O(k)}\log n}$-time blackbox PIT for this model. In the case when the depth is constant, their blackbox PIT runs in polynomial time.

The surveys by Saxena[Sax09], and by Shpilka and Yehudayoff [SY10] give wonderful expositions of the progress made so far in arithmetic circuit complexity.

An intriguing pattern in almost all lowerbounds and PITs stated so far is that progress seems to stop at depth-4. This *chasm at depth-4* was 'explained' by Agrawal and Vinay [AV08] by showing that depth-4 circuits are essentially equivalent to general circuits. They showed that any sub-exponential sized circuit computing a polynomial of low degree can be *depth-reduced* to a sub-exponential sized depth-4 circuit computing the same polynomial. The contrapositive of this theorem is that any exponential lower bound for depth-4 circuits automatically translates to an exponential lower bound for *any* circuit. Agrawal and Vinay [AV08] further showed that a polynomial time blackbox PIT for depth-4 circuits translates to an $n^{O(\log n)}$-time blackbox PIT for any general circuit. In essence, they showed that $\Sigma\Pi\Sigma\Pi$ circuits are almost as hard as general circuits.

1.3 Contributions of this thesis

With the result of Agrawal and Vinay [AV08, Koi10], the current state of the art for PITs and lowerbounds is not as bleak as it appeared. There has been considerable progress made for constant depth circuits and separating VP and VNP may not be far from our reach. However, one annoying quirk is that each of the special cases have used a specialized technique tailored towards that model (see Table 1.1). It is prima facie not clear if the techniques of one model can be used to PIT on another.

Table 1.1: PIT Techniques

Model	PIT	Main Idea
Sparse polynomials ($\Sigma\Pi$ circuits)	$\mathsf{poly}(n,s)$, blackbox	Univariate substitution
Diagonal Circuits $\ell_1^d + \cdots + \ell_s^d$	$\mathsf{poly}(n,s)$ non-blackbox $\mathsf{poly}(s^{\log s})$ blackbox	Partial derivative / dual representation
Bounded fan-in depth-3 ($\Sigma\Pi\Sigma(k)$ circuits)	$\mathsf{poly}(n,d^k)$ blackbox	Chinese remaindering over local rings
Bounded trdeg depth-4	$\mathsf{poly}(s^k)$ blackbox ($k =$ trdeg bound)	Alg. independence, Jacobian
Bounded fan-in depth-4, multilinear ($\Sigma\Pi\Sigma\Pi^{[\mathrm{mult}]}(k)$ circuits)	$\mathsf{poly}(s^{k^3})$ blackbox	Sparsity bounds
Read-k multilinear, depth D	$s^{k^{O(k)}+k\log n}$ non-blackbox $s^{k^{k^2}+O(kD)}$ blackbox	Fragmentation of partial derivatives

What is the central core that makes PIT on these models easy? Is there a unified technique to understand the different PITs? Can we solve PITs on models that are a combination of the above "easy" models? Do the lowerbound techniques of depth-3 circuits lift to depth-4? These are some of the main questions addressed in this thesis.

1.3.1 Composition of identity tests

The first question this thesis addresses is the following — suppose we know how to perform identity tests efficiently on two classes of circuits $\mathscr{C}_1$ and $\mathscr{C}_2$, how easy is it to solve PIT on the class of circuits $\mathscr{C}_1 + \mathscr{C}_2$? The class $\mathscr{C}_1 + \mathscr{C}_2$ is made up of circuits C of the form $C_1 + C_2$, where $C_1 \in \mathscr{C}_1$ and $C_2 \in \mathscr{C}_2$. Depending on the classes $\mathscr{C}_1$ and $\mathscr{C}_2$, this question can be quite non-trivial to answer. For instance, suppose we are given sparse polynomials $f, g_1, \ldots, g_t$, explicitly as sums of monomials, and asked to check if $f = \prod_{i=1}^t g_i$. Surely, it is easy to check if f or $\prod_{i=1}^t g_i$ is zero. But, it is not clear how to perform the test $f - \prod_{i=1}^t g_i \stackrel{?}{=} 0$. (This problem has also been declared open in a work by [vzG83] on sparse multivariate polynomial factoring.) The test $f - \prod_{i=1}^t g_i \stackrel{?}{=} 0$ is one of the most basic cases of depth-4 PIT that is still open.

Two of the non-trivial classes of depth-3 circuits for which efficient PIT algorithms are known are the classes of bounded top fan-in [KS07] and diagonal circuits [Kay10, Sax08]. The question is - Is it possible to glue together the seemingly disparate methods of [KS07] and [Kay10, Sax08] and give a PIT algorithm for the composition of bounded top fan-in depth-3 and diagonal circuits, or bounded top fan-in depth-3 and sparse polynomials (depth-2 circuits)? In Chapter 3, we answer this question in the affirmative. Our technique also applies to a special case of the depth-4 problem $f - \prod_{i=1}^t g_i \stackrel{?}{=} 0$ where each of the g_i's are sums of univariates.

Main ideas

The key ingredient in the proof is an algorithm to identity test and compute the *leading monomial* for a generalization of diagonal circuits and sparse polynomials called *semidiagonal circuits* (Definition 3.1).

With this in hand, the composition of PITs for $\Sigma\Pi\Sigma(k)$ and sparse/diagonal circuit proceeds by a careful analysis of the Kayal-Saxena test [KS07] on the $\Sigma\Pi\Sigma(k)$ part of the input, and studying the evolution of the sparse/diagonal part. The PIT for semidiagonal circuits would help us keep the transformation of the sparse/diagonal part in control.

To solve the problem of checking $f - \prod g_i \stackrel{?}{=} 0$ where g_i's are sums of univariates, we show that checking divisibility of a given polynomial f by a sum of univariates reduces to a semidiagonal PIT. An additional result about the irreducibility of sums of univariates allow us to use chinese remaindering to verify the given factorization.

1.3.2 A unified technique for PIT

The next question this thesis addresses is a quest for unification of the various techniques for PIT — Is there a unified approach that explains some (if not all) of the blackbox PITs on the various models? In Chapter 4 we answer this question with a "Yes!", and the unified approach is via *algebraic independence* and the *Jacobian* introduced by Beecken, Mittman and Saxena [BMS11a].

We show that the Jacobian is powerful enough to give one unified approach to give blackbox PITs for bounded fan-in depth-3, bounded fan-in depth-4 multilinear circuits , bounded read bounded depth multilinear circuits, and bounded transcendence degree depth-4 circuits of course, but with a caveat that our approach only works over fields of zero or large characteristic.

In the process of finding a universal technique, we strengthen the earlier results significantly thus giving the first blackbox PIT for these generalized models. We construct blackbox PITs for not only bounded fan-in depth-3 circuits, but also for circuits of the form $C(T_1, \cdots, T_m)$ where C is *any* polynomial of low degree and T_i's are products of linear functions with bounded transcendence degree. Further, we remove the multilinear restriction completely from the constant-depth constant-read models. The notion of 'read' is also replaced by a general notion of '*occur*' , which additionally generalizes PIT on $\Sigma\Pi\Sigma\Pi^{[\text{mult}]}(k)$ circuits as well.

The strong connection between PIT and lowerbounds that was alluded to earlier translates to this unification as well. We present lowerbounds for the Det_n and Perm_n for almost all the models we construct PITs for, again via the Jacobian.

Main ideas

The driving question is to find out the key property that makes PIT the earlier models easy, and one candidate is that some parameter for each of the models is being bounded (depth does not qualify as such a parameter, as depth-4 is nearly as hard as the general case [AV08]). The main contribution is to transform this boundedness into the transcendence degree and structure of the Jacobian.

1.3.3 Towards lower bounds for depth-4 circuits

The final question addressed in this thesis is to push towards lowerbounds for Perm_n. The result of Agrawal and Vinay [AV08] states that we only need to prove a strong enough lowerbound for the class of depth-4 circuits. Koiran [Koi10] strenghtened their result and showed that it suffices to prove a lower bound of $\exp(\omega(\sqrt{n}\log^2 n))$ for *homogeneous* depth-4 circuit with the bottom multiplication gates having $O(\sqrt{n})$ fan-in to obtain super-polynomial lower bounds for general circuits. This fact of depth-4 circuits almost capturing the general case is often referred to as the "chasm at depth-4".

As mentioned in Section 1.1, there are lower bounds known for Det_n and Perm_n for depth-3 circuits and the main idea used is to study the rank of the *partial derivative space*. Unfortunately, this technique does not scale even to the case when we have a *homogeneous* circuit of the form $C = T_1 + \cdots + T_s$ with each $T_i = Q_{i1} \ldots Q_{id}$, a product of quadratics. Is there a technique that helps us address such circuits? How close can we get to the chasm?

In Chapter 5 we present a lower bound of $\exp(\Omega(\sqrt{n}))$ for homogeneous depth-4 circuits where the fan-in of the bottom multiplication gates having $O(\sqrt{n})$ fan-in that compute Det_n or Perm_n. In the light of results of Agrawal-Vinay [AV08] and Koiran [Koi10] that a $\exp\left(\omega\left(\sqrt{n}\log^2 n\right)\right)$ lower bound would yield super-polynomial circuit lower bounds, this gets *very* close to the chasm.

Main ideas

The key technique in this result is to study the rank of *shifted partial derivatives*, or *low degree combinations* of the partial derivatives. We show that that any homogeneous depth-4 circuit with bounded bottom fan-in is *weak* with respect to the rank of the shifted partial derivatives. We then lowerbound the rank of the shifted partial derivatives of Det_n, Perm_n by reducing it to a counting problem which we then solve.

An interesting artifact of the proof is a vague possibility of separating the complexity of Det_n and Perm_n. The rank of the shifted partial derivatives is closely related to an algebraic geometric construct called the *Hilbert polynomial*. In the process of obtaining a lower bound for the rank of Det_n and Perm_n, it turns out that certain algebraic geometric properties of determinantal minors show that the rank of shifted partial derivatives of Perm_n is *provably larger* than that of Det_n. It might be possible to show that rank of Perm_n is *significantly larger* than that of the Det_n and might yield some non-trivial separation in their complexity. This measure of the rank of the shifted partial derivatives is one of the few measures that seem to visibly distinguish the complexity of the determinant and the permanent.

1.4 Structure of the thesis

Chapter 2 discusses some of the preliminaries required for the following chapters. We shall also discuss some standard PIT/lowerbound techniques that would be useful for several results in the thesis. Chapter 3 deals with the question of composing identity tests, and applying that to a special case of the "sparse factorization verification" problem. Chapter 4 presents the unified approach to blackbox PITs via the Jacobian. Chapter 5 is devoted to the *shifted partial derivative technique* and presents an exponential lower bound for bounded bottom fan-in depth-4 circuits. Chapter 6 closes with some concluding remarks, open problems and future directions.

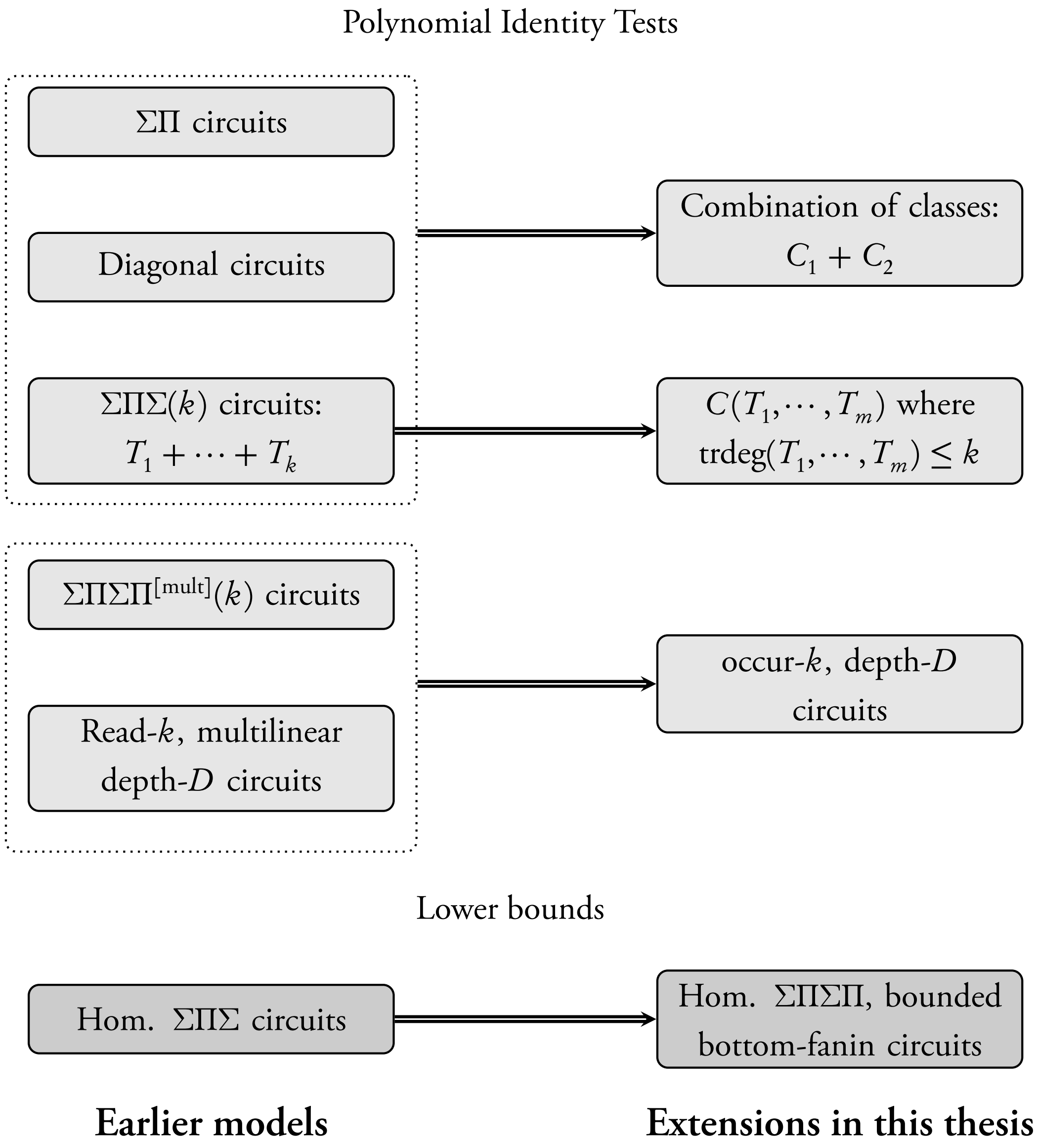

Polynomial Identity Tests
$\Sigma\Pi$ circuits
Diagonal circuits
$\Sigma\Pi\Sigma(k)$ circuits: $T_1 + \cdots + T_k$
Combination of classes: $C_1 + C_2$
$C(T_1, \cdots, T_m)$ where $\operatorname{trdeg}(T_1, \cdots, T_m) \leq k$
$\Sigma\Pi\Sigma\Pi^{[\mathrm{mult}]}(k)$ circuits
Read-k, multilinear depth-D circuits
occur-k, depth-D circuits
Lower bounds
Hom. $\Sigma\Pi\Sigma$ circuits
Hom. $\Sigma\Pi\Sigma\Pi$, bounded bottom-fanin circuits
Earlier models
Extensions in this thesis

Figure 1.2: Contributions of this thesis

2 Preliminaries

This chapter shall be devoted to the notational preliminaries, and some basics that would be required for the remaining chapters.

2.1 Notation

- For any integer n, we shall use $[n]$ to denote the set $\{1,\ldots,n\}$.
- We shall reserve the use of bold letters to denote indexed sets, for example $\mathbf{T}_n = \{T_1,\ldots,T_n\}$ or $\mathbf{f}_r = \{f_1,\ldots,f_r\}$, and we shall drop the subscript if the size of the set is clear from context.
- For $\mathbf{i} = \{i_1,\ldots,i_n\}$ and $\mathbf{x} = \{x_1,\ldots,x_n\}$, we shall use $\mathbf{x}^{\mathbf{i}}$ to be the monomials $x_1^{i_1}\ldots x_n^{i_n}$. Similarly, $\partial^{\mathbf{i}}(f)$ shall denote the partial derivative:

$$\partial^{\mathbf{i}} f \quad \overset{\text{def}}{=} \quad \frac{\partial^{i_1}}{\partial x_1^{i_1}}\left(\frac{\partial^{i_2}}{\partial x_2^{i_2}}\left(\cdots\left(\frac{\partial^{i_n} f}{\partial x_n^{i_n}}\right)\cdots\right)\right)$$

- $\partial^{=k}(f)$ shall denote the set $\left\{\partial^{\mathbf{i}}(f) \;:\; i_1+\cdots+i_n = k\right\}$, and $\mathbf{x}^{=\ell}$ shall denote the set $\left\{\mathbf{x}^{\mathbf{i}} \;:\; i_1+\cdots+i_n = \ell\right\}$. Also, $\partial^{\leq k}(f)$ shall denote $\left\{\partial^{\mathbf{i}}(f) \;:\; i_1+\cdots+i_n \leq k\right\}$, and $\mathbf{x}^{\leq \ell}$ shall denote $\left\{\mathbf{x}^{\mathbf{i}} \;:\; i_1+\cdots+i_n \leq \ell\right\}$.
- For a fixed monomial ordering on the monomials, $\mathsf{LM}(f)$ shall denote the leading monomial of f under this ordering, and $\mathsf{LC}(f)$ shall denote the coefficient of the leading monomial. And for any monomial $\mathbf{x}^{\mathbf{i}}$, we shall use $[\mathbf{x}^{\mathbf{i}}](f)$ to denote the coefficient of $\mathbf{x}^{\mathbf{i}}$ in the polynomial f.
- $\partial_i f$ shall denote the partial derivative $\frac{\partial f}{\partial x_i}$
- For any polynomial f, the term $f_{(x_i=\alpha_i)}$ denotes the polynomial obtained by setting $x_i = \alpha_i$ in f.

- For a set of polynomials $\{f_1, \ldots, f_r\}$, we shall use $\langle f_1, \ldots, f_r \rangle$ to denote the ideal generated by them, and $\langle f_1, \ldots, f_r \rangle_{\leq \ell}$ to denote the set of all ℓ-degree combinations of them. That is,

$$\langle f_1, \ldots, f_r \rangle = \{p_1 f_1 + \cdots + p_r f_r \ : \ p_i \in \mathbb{F}[\mathbf{x}]\}$$
$$\langle f_1, \ldots, f_r \rangle_{\leq \ell} = \{p_1 f_1 + \cdots + p_r f_r \ : \ p_i \in \mathbb{F}[\mathbf{x}]\,, \deg(p_i) \leq \ell\}$$

Polynomials and arithmetic circuits

- For a polynomial f, the *sparsity* of f shall denote the number of monomials in f. If the sparsity of f is polynomially bounded in the number of variables, we shall say that the polynomial f is *sparse*.

- A circuit/formula is said to be homogeneous if every gate computes a homogeneous polynomial. A circuit/formula is said to be *multilinear* if every gate computes a multilinear polynomial.

- Normally, the top gate of the circuit is assumed to be a $+$ gate (unless otherwise stated). $\Sigma\Pi$ shall denote the class of polynomial sized depth-2 circuits (which are sparse polynomials). $\Sigma\Pi\Sigma$ denotes of depth-3 circuits, and $\Sigma\Pi\Sigma\Pi$ denotes the class of depth-4 circuits. Further, we shall add the term hom or ml to denote homogeneity or multilinearity respectively (for example, $\Sigma\Pi\Sigma\Pi^{[\text{hom}]}$ denotes the class of homogeneous depth-4 circuits, and $\Sigma\Pi\Sigma\Pi^{[\text{hom,ml}]}$ denotes the class of homogeneous multilinear depth-4 circuits).

2.2 Basic tools for PIT and lower bounds

This section shall help build the some basic tools that would be required in the chapters to follow.

2.2.1 Homogenization

In the context of PIT, the input circuit is normally assumed to be homogeneous. The reason is because of a standard trick of *homogenizing* a polynomial, and also homogenizing any circuit computing a homogeneous polynomial without too much blow-up in size.

For a polynomial $f(x_1,\ldots,x_n)$ of degree d, the *homogenized version* of f is the polynomial

$$f^{[\text{hom}]}(x_1,\ldots,x_n,z) \quad \stackrel{\text{def}}{=} \quad z^d \cdot f\left(\frac{x_1}{z},\ldots,\frac{x_n}{z}\right)$$

Of course, $f = 0$ if and only if $f^{[\text{hom}]} = 0$.

Given a circuit C of size s that computes a homogeneous polynomial of degree d, the homogenization of C is defined as follows:

- For every gate $g \in C$, define gates $\left\{g^{(0)},\ldots,g^{(d)}\right\}$ (where each $g^{(i)}$ shall compute the i-th homogeneous part of the polynomial computed by $g \in C$).
- If g is a $+$ gate with children h_1, h_2, then

  $$g^{(i)} \quad \stackrel{\text{def}}{=} \quad h_1^{(i)} + h_2^{(i)} \quad \text{for all } 0 \leq i \leq d$$

  If g is a $\times$ gate with children h_1, h_2, then

  $$g^{(i)} \quad \stackrel{\text{def}}{=} \quad \sum_{j \leq i}\left(h_1^{(j)} \times h_2^{(i-j)}\right) \quad \text{for all } 0 \leq i \leq d$$

Clearly, the size of the new circuit is only larger by a factor of $O(d^2)$.

2.2.2 The Schwartz-Zippel Lemma

PIT has a natural randomized algorithm, which usually attributed to Schwartz [Sch80] and Zippel [Zip79] (though it was also observed by DeMillo and Lipton [DL78]).

Lemma 2.1 (Schwartz-Zippel Lemma [Sch80, Zip79, DL78]). *Suppose f is a non-zero degree d polynomial over n variables. Then for any set $S \subseteq \mathbb{F}$,*

$$\Pr_{a_i \in S}\left[f(a_1,\ldots,a_n) = 0\right] \quad \leq \quad \frac{d}{|S|}$$

In other words, a random evaluation of a non-zero polynomial is non-zero with high probability (if the set S is large enough). This also means that any n-variate degree d polynomial has a hitting set of size $(d+1)^n$.

Corollary 2.2. *There is a hitting set generator for the class of n-variate degree d polynomials running in time $\mathsf{poly}(d^n)$.*

Proof. Let f be any non-zero n-variate degree d polynomial. By choosing any set $S \subseteq \mathbb{F}$ of size greater than d, Lemma 2.1 asserts a random evaluation using elements of S is non-zero with probability greater than zero. Hence, there is some point in S^n on which f evaluates to a non-zero value. □

Almost all PITs proceed by starting with the input polynomial f and constructing a *variable reduction* to obtain $\tilde{f}$ on fewer variables such that $f = 0$ if and only if $\tilde{f} = 0$. Then, an application of Corollary 2.2 on $\tilde{f}$ would give the hitting set for $\tilde{f}$, and hence for f.

2.2.3 Blackbox PIT for sparse polynomials

Another recurrent tool that shall be used in the subsequent chapters is a hitting set for sparse polynomials. The hitting set for sparse polynomials is usually attributed to Klivans and Spielman[KS01], but the following description is by Agrawal and Biswas[AB99].

Let f be a non-zero n-variate polynomial of degree less than d. Suppose we want to convert f into a univariate polynomial in t by mapping distinct monomials in $\mathbf{x}$ to distinct powers of t, then one natural map is

$$\Delta : x_i \quad \mapsto \quad t^{d^i}$$

An obvious problem is that the resulting polynomial has exponential degree. Agrawal and Biswas [AB99] suggested *folding* the above map by considering

$$\Delta_p : x_i \quad \mapsto \quad t^{d^i \bmod p}$$

for a suitable choice of p. If t^a and t^b are two distinct monomials of $\Delta(f)$, we shall say a choice of p is *bad* for the pair (a, b) if $\Delta_p(f)$ maps these two monomials to the same monomial.

Note that p is bad for a pair (a, b) if and only if p divides $(a - b)$. If f has s monomials to begin with, then there are at most s^2 pairs (a, b). Since $a - b < d^{n+1}$, the number of prime factors of $(a - b)$ is at most $(n + 1)\log d$. Hence overall, there are at most $s^2(n + 1)\log d$ bad primes p. By the prime number theorem, there are $O(r/\log r)$ primes between 1 and r, and hence there are more than $s^2(n + 1)\log d$

primes within the first $(s^2(n+1)\log d)^2$ choices of p. Therefore, $\Delta_p(f) \neq 0$ for some $p \leq (s^2(n+1)\log d)^2$. Since $\Delta_p(f)$ has degree at most $p \cdot d$, this yields this hitting set. The following lemma summarizes this discussion (with a small generalization).

Lemma 2.3. *Let $f_1, \ldots, f_r$ be non-zero n-variate polynomials of degree less than d of sparsity at most s each. Let $P = r \cdot s^4(n+1)^2 \log^2 d$ and $S \subseteq \mathbb{F}$ be any set of size greater than $d \cdot P$. Then, one of the elements of the following set is a point on which each f_i is non-zero:*

$$\left\{(\alpha, \alpha^{d \bmod p}, \ldots, \alpha^{d^{n-1} \bmod p}) \ : \ \alpha \in S \ , \ p \leq P\right\}$$

Proof. We shall say p is *bad* for f_i if $\Delta(f_i)$ contains two non-zero monomials t^a, t^b with $p \mid (a-b)$. As in the above discussion, the number of p's that are bad for a single f_i is at most $(s^2(n+1)\log d)^2$. Hence, the number of p's that are bad for some f_i is at most $r \cdot (s^2(n+1)\log d)^2$. Hence, for some $p \leq P = r(s^2(n+1)\log d)^2$, we have that $\Delta_p(f_i) \neq 0$ for every i and hence $F(t) = \Delta_p(\prod_{i=1}^r f_i) \neq 0$.
Since $F(t)$ is a non-zero univariate of degree at most $d \cdot P$, it has at most $d \cdot P$ roots. Hence, for any set $S \subseteq \mathbb{F}$ of size greater than $d \cdot P$, there is some $\alpha \in S$ such that

$$F(\alpha) \quad = \quad \prod_{i=1}^{r} f_i(\alpha, \alpha^{d \bmod p}, \ldots, \alpha^{d^{n-1} \bmod p}) \quad \neq \quad 0$$

which is what we wanted to show. □

Composing identity tests, and sparse factorization 3

3.1 Introduction

This chapter addresses a question on 'composition of identity tests' — if we know PITs for two classes $\mathscr{C}_1$ and $\mathscr{C}_2$, can we construct PITs for circuits of the form $C_1 + C_2$ where $C_1 \in \mathscr{C}_1$ and $C_2 \in \mathscr{C}_2$? PIT on the class $\mathscr{C}_1 \times \mathscr{C}_2$ is trivial, as the product is zero if and only if one of them is zero.

As mentioned in Chapter 1, an open problem posed by von zur Gathen [vzG83] can be thought of as such a composition problem: Given polynomial $f, g_1, \cdots, g_t$ explicitly as a sum of monomials, check if

$$f - g_1 \cdots g_t \quad \stackrel{?}{=} \quad 0.$$

Sparse polynomials ($\Sigma\Pi$ circuits) are one of the simplest class of arithmetic circuits for which PITs are known, and following it are bounded fan-in depth-3 circuits ($\Sigma\Pi\Sigma(k)$) and diagonal circuits. What can we say about the composition of these classes? Can the seemingly different methods employed for each of these classes be used to give PITs on composed circuits? This chapter answers this question in the affirmative. This also applies to a special case of checking if $f - \prod_{i=1}^{t} g_i \stackrel{?}{=} 0$ where each g_i's are sums of univariates.

3.1.1 Contribution of this chapter

This chapter presents deterministic polynomial time algorithms for two problems on identity testing – one is on a class of depth-3 circuits, while the other is on a class of depth-4 circuits. As mentioned in Section 3.1, both these classes of circuits would be examples of composition of subclasses over which we already know how to perform PIT.

The first problem is a common generalization of the problems studied in [KS07] and [Sax08]. We shall need the following definition of a *semidiagonal* circuit.

Definition 3.1. (Semidiagonal circuit) *Let C be a depth-3 circuit, i.e. a sum of products of linear polynomials. The circuit is said to be an r-*semidiagonal circuit *if each product gate in C computes a polynomial of the form $m \cdot \prod_{i=1}^{b} \ell_i^{e_i}$, where m is a monomial, ℓ_i is a linear polynomials in the input variables and $\prod(1+e_i) \leq r$.*

*Also, a term of the form $(m \cdot \prod_{i=1}^{b} \ell_i^{e_i})$ shall be called a r-*semidiagonal term.

Remark. A general $\Sigma\Pi\Sigma$ circuit of degree d is trivially a r-semidiagonal circuit for $r = 2^d$ but we shall be more interested in circuits where r is polynomially bounded by the circuit size. We shall refer to such circuits as simply *semidiagonal circuits* (dropping the parameter r).

Problem 3.1. *Given a depth-3 circuit C_1 with bounded top fan-in and given a semidiagonal circuit C_2, test if the output of the circuit $C_1 + C_2$ is identically zero.*

The second problem is a special case of checking the validity of a given factorization of a sparse multivariate polynomial (thus, a case of PIT on depth-4 top fan-in 2).

Problem 3.2. *Given $t+1$ polynomials $f, g_1, \ldots, g_t$ explicitly as sum of monomials, where every g_i is a sum of univariate polynomials, check if $f = \prod_{i=1}^{t} g_i$.*

It is possible that though f is sparse, some of its factors are *not* sparse (an example is provided in Section 3.4). So, multiplying the g_i's in a brute force fashion is not a feasible option. In this chapter, we shall present the following:

Theorem 3.2. *Problem 3.1 and 3.2 can be solved in deterministic polynomial time.*

3.1.2 Overview of the approach

The main tool in solving both Problem 3.1 and 3.2 is a polynomial identity test for semidiagonal circuits.

PIT for semidiagonal circuits

There are two known polynomial time non-blackbox PITs for semidiagonal circuits – one approach by Saxena [Sax08] using *duality*, and another by Kayal [Kay10] using the *partial derivative method*. The two approaches are quite different, but both of them face a hurdle when the underlying field is of low characteristic. Saxena [Sax08]'s *duality* can be made to work over low-characteristic fields by moving to appropriate higher algebras, which sometimes turns out to be rather cumbersome. Kayal's [Kay10] partial derivative method doesn't work directly since derivatives of non-zero polynomials could become zero in low characteristic fields (e.g. $x^p + y^p$). In this chapter, we present a modification of Kayal's [Kay10] partial derivative method that works purely on evaluations. Further, the algorithm can be easily augmented to present an algorithm to compute the leading monomial (and coefficient) of a given semidiagonal circuit. This additional augmentation would turn out to be crucial in solving Problem 3.1.

The approaches to solve Problem 3.1 and 3.2 shall be described now.

Solving Problem 3.1

Let p and f be the polynomials computed by a $\Sigma\Pi\Sigma(k)$ circuit and a semidiagonal circuit respectively, and we wish to check if $p + f = 0$. The general idea is to apply the Kayal-Saxena test [KS07] on the polynomial p, and track the evolution of f in the process.

Let $p = T_1 + \cdots + T_k$ where each T_i is a product of linear forms. The original Kayal-Saxena test first chooses a T_j such that $\mathsf{LM}\left(T_j\right) \succeq \mathsf{LM}(p)$ (recall that $\mathsf{LM}(f)$ refers to the leading monomial of f), and then proceeds to check if $p = 0 \bmod T_j$. The purpose of this choice of T_j is to ensure that $p = 0 \bmod T_j$ if and only if $p = \alpha T_j$ for some $\alpha \in \mathbb{F}$. To apply the same test in our setting, we need to ensure that $\mathsf{LM}\left(T_j\right) \succeq \mathsf{LM}(p + f)$, and hence we would require to compute $\mathsf{LM}(f)$ as well. Fortunately, the leading monomial of a semidiagonal circuit can be computed efficiently (Theorem 3.6) and this lets us proceed further. The next few step of the Kayal-Saxena test employs some invertible maps to transform p, and this in our setting would modify f as well. By a slightly different choice of a suitable invertible map, we can maintain the semidiagonal structure of f. Hence, we can eventually reduce the problem of testing if $p + f = 0$ to a semidiagonal PIT.

Solving Problem 3.2

The approach to check if $f = \prod_{i=1}^{t} g_i$ is quite intuitive — reduce the problem to checking divisibility and then use chinese remaindering. But the two issues here are: how to check g_i divides f, and that g_i's need not be coprime. So in general we need to check if g^d divides f for some $d \geq 1$. We shall see that such divisibility checks reduce to identity testing of a (slightly general) form of semidiagonal circuits. Finally, for chinese remaindering to work, we need to say something about the coprimality of g_i and g_j. Towards this, an irreducibility result on polynomials of the form $f(x) + g(y) + h(z)$ would help us conclude the proof (in Section 3.4).

3.2 PIT for semidiagonal circuits

The main result of this section would be a deterministic polynomial time algorithm to solve a stronger problem of computing $\mathbb{F}$-*linear dependencies* for semidiagonal circuits.

Definition 3.3. *($\mathbb{F}$-linear dependencies) Let* $\mathbf{f}(\mathbf{X}) = \left(f_1(\mathbf{X}), \ldots, f_m(\mathbf{X})\right) \in (\mathbb{F}[\mathbf{X}])^m$ *be a vector of polynomials. The set of* $\mathbb{F}$-linear dependencies *of these polynomials, denoted by* $\mathbf{f}^{\perp}$*, is defined as*

$$\mathbf{f}^{\perp} \quad = \quad \left\{(a_1, \ldots, a_m) \in \mathbb{F}^m \ : \ \sum a_i f_i = 0\right\}$$

Since $\mathbf{f}^{\perp}$ forms a vector space, we can talk about computing a basis of this vector space in polynomial time. The computational problem of obtaining a basis of $\mathbf{f}^{\perp}$ given $\mathbf{f}$ is referred to as POLYDEP($\mathbf{f}$) (studied in great detail in [Kay10] where it was first introduced).

Just like polynomial identity testing, POLYDEP also admits a randomized polynomial time algorithm when the polynomials in the input vector is presented as circuits. It is also known that PIT testing reduces to POLYDEP via turing reductions but is unclear they are equivalent. Kayal [Kay10] showed that POLYDEP of semidiagonal terms can be computed in deterministic polynomial time over fields of zero or large characteristic, by exploiting the structure of partial derivatives. In this section, using slight modification, we will adapt the algorithm of Kayal [Kay10] to compute POLYDEP over arbitrary fields (or finite dimensional algebras over fields) of large enough size.

To proceed with the algorithm, we shall need the following two simple lemmas.

Lemma 3.4. *Let* $\mathbf{f}(\mathbf{X}) \stackrel{\text{def}}{=} (f_1(\mathbf{X}), \ldots, f_m(\mathbf{X})) \in (\mathbb{F}[\mathbf{X}])^m$ *be a vector of polynomials each of whose* x_1*-degree is bounded by* d*. Then, for any* $(d+1)$ *distinct scalars* $\alpha_1, \ldots, \alpha_{d+1} \in \mathbb{F}$,

$$\mathbf{f}^{\perp} \quad = \quad \bigcap_{i=1}^{d+1} \left(\mathbf{f}_{(x_1=\alpha_i)}\right)^{\perp}$$

where by $\mathbf{f}_{(x_1=\alpha_i)}$ *shall denote the vector of polynomials obtained by substituting* $x_1 = \alpha_i$ *in each of its coordinates.*

Proof. Of course, any vector $(a_1, \cdots, a_m)$ satisfying $\sum a_i f_i = 0$ would also satisfy the equation $\sum a_i f_{(x_1=\alpha)} = 0$ for any α. Hence, it is clear that $\mathbf{f}^{\perp}$ is contained in the RHS. As for the other direction, consider any $(a_1, \ldots, a_m)$ that is not contained in $\mathbf{f}^{\perp}$. The linear combination $\sum a_i f_i$ can be thought of as a non-zero polynomial in x_1 with coefficients as polynomials in the other variables. Since the degree is bounded by d, there can be at most d roots to this polynomial. Hence $(a_1, \ldots, a_m)$ is not in $\left(\mathbf{f}_{(x_1=\alpha_i)}\right)^{\perp}$ for some $1 \leq i \leq d+1$. □

Lemma 3.5. *[Kay10] Let* $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$*. Suppose* $h_1, \ldots, h_t \in \mathbb{F}[x_1, \ldots, x_n]$ *such that for every* i*, there is some* $\mathbf{b_i} \stackrel{\text{def}}{=} (b_{i1}, \ldots, b_{it}) \in \mathbb{F}^t$ *such that*

$$f_i \quad = \quad b_{i1}h_1 + \cdots + b_{it}h_t$$

Given a basis for $\mathbf{h}^{\perp}$ *and the vectors* $\mathbf{b}_i$*, we can compute* $\mathbf{f}^{\perp}$ *in deterministic polynomial time.*

Proof. Given a basis for $\mathbf{h}^{\perp}$, we can compute a basis $\{h_1, \ldots, h_r\}$ that are linearly independent and rewrite every other h_i as linear combination of $\{h_1, \ldots, h_r\}$. Therefore, each f_i can be rewritten in this basis as well, i.e.,

$$f_i \quad = \quad c_{i1}h_1 + \ldots c_{ir}h_r \quad \text{where } c_{ij} \in \mathbb{F}.$$

It follows that $\mathbf{f}^{\perp}$ is just $\left\{(c_{i1}, \ldots, c_{ir}) \;:\; i = 1, \ldots, m\right\}^{\perp}$ as $\{h_1, \ldots, h_r\}$ are linearly independent. □

We are now ready to state and prove the main theorem of this section.

Theorem 3.6 (Semidiagonal PIT). *Given an* r*-semidiagonal circuit* C *(over* n *variables and degree* d*) of the form*

$$C \quad = \quad \sum_{i=1}^{s} \alpha_i \cdot m_i \prod_{j=1}^{b} \ell_{ij}^{e_{ij}} \quad , \quad \alpha_i \in \mathbb{F}$$

we can test in deterministic $\mathsf{poly}(s,n,d,r)$ *time if* C *is identically zero. Further, if* C *is not identically zero, we can compute the leading monomial and coefficient induced by the lexicographic ordering on the variables.*

Proof. We shall in fact present a polynomial time algorithm for PolyDep on a given set of semidiagonal terms. It is clear that checking if C is identically zero is equivalent to checking if $(\alpha_1,\dots,\alpha_s)$ is contained in $\left\{m_i \prod_{j=1}^b \ell_{ij}^{e_j}\right\}^\perp$. It would be useful to perform a one-time 'saturation' of the set of semidiagonal terms to the following set S defined as

$$S \overset{\text{def}}{=} \left\{ m_i \prod_{j=1}^{b} \ell_{ij}^{e'_{ij}} \;:\; i=1,\dots,s \text{ and } 1 \le e'_{ij} \le e_{ij} \right\}.$$

Note that $|S| \le \max_i \left(\prod_{j=1}^b (1+e_{ij})\right) \cdot s$ and this is bounded by $s \cdot r$. The reason for this saturation is because this gives us a handle on how terms evolve under partial evaluations. For any $f \in S$ and $\alpha \in \mathbb{F}$, we can write $f_{(x_1=\alpha)}$ as a linear combination of the polynomials in

$$S_1 = \left\{ (m_i)_{(x_1=1)} \left(\prod_{j=1}^{b} \ell_{ij}^{e'_{ij}} \right)_{(x_1=0)} \;:\; i=1,\dots,s \text{ and } 0 \le e'_{ij} \le e_{ij} \right\}.$$

Further, these linear combinations can be efficiently computed given α and C by simply expanding using the binomial expansion. Observe that the size of S_1 is at most the size of S, but every element of S_1 is semidiagonal term over $(n-1)$ variables. This leads to a natural recursive algorithm for PolyDep(S) using Lemma 3.4 and Lemma 3.5.

1. If $n=0$, the elements of S are just scalars and the problem is trivial.

2. Otherwise, pick distinct $\alpha_1,\dots,\alpha_{d+1} \in \mathbb{F}$. For each $f \in S$ and $i=1,\dots,(d+1)$, write $f_{(x_1=\alpha_i)}$ as a linear combination of elements in S_1.

3. Recursively compute PolyDep(S_1), a basis for the set of dependencies of S_1.

4. From PolyDep(S_1) (using Lemma 3.5), compute a basis of dependencies for

$$V_i = \left\{ f_{(x_1=\alpha_i)} \;:\; f \in S \right\}^\perp \quad \text{for each } i=1,\dots,(d+1).$$

5. Return a basis for
$$S^{\perp} \quad = \quad \bigcap_{i=1}^{d+1} V_i \qquad \text{(by Lemma 3.4)}$$

The correctness of the algorithm is clear from Lemma 3.4 and Lemma 3.5. As for the time complexity analysis, notice that every step besides Step 3 can be computed in $\mathsf{poly}(n,d,|S|)$ time. And Step 3 is a recursive call to PolyDep on polynomials over $(n-1)$ variables, and the size of S_1 is no larger than the size of S. Hence, if $T(n,d,|S|)$ denotes the time complexity of PolyDep, we have:

$$\begin{aligned} & T(n,d,|S|) &=& \ T(n-1,d,|S|) + \mathsf{poly}(n,d,|S|) \\ \Longrightarrow \quad & T(n,d,|S|) &=& \ \mathsf{poly}(n,d,|S|) \quad = \quad \mathsf{poly}(n,d,r,s) \end{aligned}$$

Computing the leading monomial: The coefficient of any degree d univariate polynomial $f(x_1)$ can be interpolated $(d+1)$ evaluations. Formally, if $[x_1^i]f$ denotes the coefficient of x_1^i in f, for every $\alpha_1,\ldots,\alpha_{d+1} \in \mathbb{F}$, there exists $\beta_1,\ldots,\beta_{d+1} \in F$ such that

$$[x_1^i]f \quad = \quad \beta_1 f_{(x_1=\alpha_1)} + \cdots + \beta_{d+1} f_{(x_1=\alpha_{d+1})}$$

In the case when f is a semidiagonal circuit (interpretting it as a univariate in x_1 with coefficients in the remaining variables), the RHS is a linear combination of semidiagonal terms. Hence, we have a natural algorithm to compute the leading monomial of f under the lexicographic order.

1. If $n = 0$, then f is a scalar and the problem is trivial.

2. Otherwise, compute the largest i for which
$$[x_1^i]f \quad = \quad \beta_1 f_{(x_1=\alpha_1)} + \cdots + \beta_{d+1} f_{(x_1=\alpha_{d+1})} \quad \neq \quad 0$$
If no such i exists, return Zero. Else, recursively compute the leading monomial and coefficient of $[x_1^i]f$ and return $\mathsf{LM}(f) = x_1^i \cdot \mathsf{LM}\left([x_1^i]f\right)$ and $\mathsf{LC}(f) = \mathsf{LC}\left([x_1^i]f\right)$.

The correctness of the algorithm is obvious and the time complexity is $\mathsf{poly}(n,d,r,s)$ as earlier. □

Remark. The above proof works for a class of polynomials slightly more general than semidiagonal circuits with each term be a product of few sums of univariates rather than linear functions. These are polynomials of the form $f = T_1 + \cdots + T_s$ with each summand being of the form $m \cdot \prod_{i=1}^{b} g_i^{e_i}$ for some monomial m and $g_i = u_{i1}(x_1) + \cdots + u_{in}(x_n)$. The algorithms are exactly the same and shall avoid stating the more general theorem and proof. This generalization, however, would be required in Section 3.4 to solve Problem 3.2.

3.3 Solving Problem 3.1

This section would be devoted to the solution of Problem 3.1. To recall the problem statement:

> Given a $\Sigma\Pi\Sigma(k)$ circuit computing a polynomial p and a semidiagonal circuit computing a polynomial f, check if $p + f = 0$.

Assume, without any loss of generality, that p and f are homogeneous polynomials having the same degree d. Let $p = \sum_{i=1}^{k} T_i$ where each T_i is a product of d linear forms over $\mathbb{F}$, and let f be a sum of s semidiagonal terms. Let $X = \{x_1, \ldots, x_n\}$ be the underlying set of variables. The algorithm builds upon the Kayal-Saxena test [KS07], which tests if a given $\Sigma\Pi\Sigma(k)$ circuit is zero. To put things in context, let us review their algorithm first.

3.3.1 Reviewing the Kayal-Saxena test

Suppose $p = T_1 + \cdots + T_k$, where each T_i is a product of linear forms. Fix a monomial order induced by, say, $x_1 \succ \cdots \succ x_n$. Let us assume without loss of generality that $\mathsf{LM}(T_1) \succeq \mathsf{LM}\left(T_i\right)$ for every $i \leq k$. If $p = 0 \bmod T_1$, then $p = \alpha T_1$ for some $\alpha \in \mathbb{F}$. In this case, it is easy to check if $\alpha = 0$ by just checking if the contribution of $\mathsf{LM}(T_1)$ in $T_1 + \cdots + T_k$ is zero or not.

To check if $p = 0 \bmod T_1$, Kayal and Saxena [KS07] employ chinese remaindering. If $T_1 = \ell_{11}^{e_1} \ldots \ell_{1d}^{e_d}$, then it suffices to check if $p = 0 \bmod \ell_{1i}^{e_i}$ for every $1 \leq i \leq d$. How does one check, say, if $p = 0 \bmod x_1^{e_1}$? Kayal and Saxena do this by thinking of $p \bmod x_1^{e_1}$ as a polynomial over $\mathscr{R}[x_2, \ldots, x_n]$ where $\mathscr{R} = \mathbb{F}[x_1]/(x_1^{e_1})$, which is a *local ring*.

They show that the essential ideas and chinese remaindering continues to work over such local rings. Before we proceed to the algorithm of Kayal and Saxena, we shall spend some time understanding the basic properties of local rings that we shall require.

3.3.2 A brief introduction to local rings

Definition 3.7. *A commutative ring $\mathcal{R}$ over a field $\mathbb{F}$ is said to be a* local ring *if every non-invertible element $a \in \mathcal{R}$ is nilpotent i.e. there is some positive integer k such that $a^k = 0$.*

Rings like $\mathbb{F}[x]/(x^5), \mathbb{F}[x,y]/(x^4,(y+x)^3)$ are examples of local rings (these would be the sort of rings that arise form the Kayal-Saxena test).

Local rings have lots of very interesting properties, making it behave very close to a field. The set of all nilpotent elements of a local ring $\mathcal{R}$ form a maximal ideal of $\mathcal{R}$, and is in fact the unique maximal ideal of $\mathcal{R}$.

Proposition 3.8 (cf. [AM69]). *Every element $a \in \mathcal{R}$ can be uniquely expressed as $\alpha + \tau$ where $\alpha \in \mathbb{F}$ and τ is a nilpotent in $\mathcal{R}$.*

The fact that the only non-invertible elements in $\mathcal{R}$ are nilpotent is intuitively why a local ring behaves "almost like" a field (wherein the only non-invertible element is zero). In fact, the above proposition yields a natural homomorphism $\varphi : \mathcal{R} \to \mathbb{F}$ that sends $(\alpha + \tau)$ to α. This map φ lifts naturally to polynomials over $\mathcal{R}$ as well, enables a transition from the ring to the field while preserving a lot of algebraic properties. One particular example is a version of *chinese remaindering*.

Lemma 3.9 ([KS07] Chinese Remaindering over Local Rings). *Let $\mathcal{R}$ be a local ring over $\mathbb{F}$ and $p, g, h \in \mathcal{R}[x_1, \ldots, x_n]$ be multivariate polynomials such that $\varphi(g)$ and $\varphi(h)$ are coprime over $\mathbb{F}$. If $p = 0 \bmod g$ and $p = 0 \bmod h$ then $p = 0 \bmod gh$.*

This is exactly what would be required to check if $p = 0 \bmod T_1$ even over local rings. With this sketch in mind, let us proceed to Kayal-Saxena test.

3.3.3 Reviewing the Kayal-Saxena test

The intermediate recursive steps of the Kayal-Saxena test would deal with local rings of the form $\mathcal{R} = \mathbb{F}[x_1, \cdots, x_c]/(\ell_1^{e_1}, \ldots, \ell_c^{e_c})$ where $\ell_1, \ldots, \ell_c$ are linear forms in the variables $x_1, \dots, x_c$. Note that $\dim_{\mathbb{F}} \mathcal{R} \leq d^c$ where d is an upper-bound on the e_i's. We shall refer to the rest of the variables $\{x_{c+1}, \dots, x_n\}$ as the *free variables*. Note that any $\mathbb{F}$-linear combination of $x_1, \dots, x_n$, when thought of as an element of $\mathcal{R}[x_{c+1}, \dots, x_n]$, can be expressed as a sum $\ell + \tau$ where ℓ is a polynomial over the free variables and τ is a nilpotent in $\mathcal{R}$. The following Algorithm **KS-Test** would take as input the description of $\mathcal{R}$ specified by the relations $\left\{\ell_1^{e_1}, \dots, \ell_c^{e_c}\right\}$, products of linear functions $\{T_1, \cdots, T_{k'}\}$ and test if $\tilde{p} = T_1 + \cdots + T_{k'}$ is zero or not. To begin with, $\mathcal{R} = \mathbb{F}$, $c = 0$, $k' = k$ and $\tilde{p} = p$. The invariant that shall be maintained is that $c + k' \leq k$.

The description of the following algorithm is the same as in [KS07], except a slight modification in Step 3.1 which is somewhat necessary for our purpose (explained in remark after Algorithm **KS-Test**).

Algorithm KS-Test$(\mathcal{R}, \{T_1, \dots, T_{k'}\})$:

Step 1: (Rearranging terms) Order the terms T_i so that $\mathsf{LM}(T_1) \succeq \mathsf{LM}(T_i)$, for all $2 \leq i \leq k'$.

Step 2: (Base Cases) If $k' = 1$ check if $T_1 = 0$. If so, return YES, and NO otherwise.

If $\mathsf{LM}(T_1) \in \mathcal{R}$, then each of the T_i's are just elements of $\mathcal{R}$. Add the elements and return YES if zero, and NO otherwise.

Step 3: (Verifying that $\tilde{p} = 0 \bmod T_1$) By suitably grouping the factors of T_1, it can be expressed as $T_1 = \alpha_1 \cdot S_1 \dots S_r$ where $\alpha_1 \in \mathcal{R}$, and each S_j is of the form

$$S_j = (\ell_j + \tau_1) \cdot (\ell_j + \tau_2) \dots (\ell_j + \tau_{t_j}),$$

where each τ_i is a nilpotent in $\mathcal{R}$ and ℓ_j is a non-zero linear form over the free variables. Further, ℓ_i and ℓ_j are coprime linear forms (over $\mathbb{F}$) when $i \neq j$. Check if $\tilde{p} = 0 \bmod S_j$, for every $1 \leq j \leq r$, in the following way.

> **Step 3.1:** (Building the new ring) Suppose $\ell_j = c_u x_u + \ell'_j$ where x_u is a free variable, $0 \neq c_u \in \mathbb{F}$ and ℓ'_j is independent of x_u. Define an invertible linear transformation on the free variables that maps x_u to $c_u^{-1}(x_u - \ell'_j)$ and leaves every other variable unchanged. In other words, σ is defined such

that $\sigma(\ell_j) = x_u$. Hence, $\sigma(S_j) = (x_u + \tau_1)\ldots(x_u + \tau_{t_j})$. Define the ring

$$\mathscr{R}' = \mathscr{R}[x_u]/(\sigma(S_j))$$

Step 3.2: (Recursively verify if $\sigma(\tilde{p}) = 0 \bmod \sigma(S_j)$) Note that $\sigma(T_1) = 0 \bmod \sigma(S_j)$ as S_j divides T_1.

Recursively call **KS-Test**$(\mathscr{R}', \{\sigma(T_2), \ldots, \sigma(T_{k'})\})$. If the recursive call returns NO then output NO and exit, otherwise declare $\tilde{p} = 0 \bmod S_j$.

Declare $\tilde{p} = 0 \bmod T_1$, if $\tilde{p} = 0 \bmod S_j$ for all $1 \leq j \leq r$.

Step 4: Check if $[\mathsf{LM}(T_1)]\tilde{p}$ is zero by using the fact that

$$[\mathsf{LM}(T_1)]\tilde{p} \quad = \sum_{\mathsf{LM}(T_i) = \mathsf{LM}(T_1)} \mathsf{LC}(T_i)$$

Return YES if zero, and NO otherwise.

Remark on Step 3.1: In [KS07], the linear transformation σ is described as a map that takes ℓ_j to some fixed variable x_1 and transforms the remaining variables $x_2, \ldots, x_n$ accordingly so that σ is invertible. In the application, we will also need the property that σ maps only one variable to a general linear form, whereas any other variables remain unchanged. We will need this attribute of σ, in Section 3.3.4, to ensure that a the semidiagonal structure of the given polynomial is preserved at every intermediate stage of the algorithm.

Correctness of Algorithm KS-Test

By Step 3, we are guaranteed that $\mathsf{LM}(T_1) \succeq \mathsf{LM}(\tilde{p})$. Therefore, if $\tilde{p} = 0 \bmod T_1$, then $\tilde{p} = 0$ if and only if $[\mathsf{LM}(T_1)]\tilde{p} = 0$. This is verified in Step 4.

It remains to show the correctness of Step 3. In order to check if $\tilde{p} = 0 \bmod T_1$, the algorithm finds out if $\tilde{p} = 0 \bmod S_j$ for every $1 \leq j \leq r$. That fact that this is a sufficient condition is implied by Lemma 3.9. Since $\varphi(S_j) = \ell_j^{t_j}$ (recall that φ is the map that sends all nilpotents of $\mathscr{R}$ to zero) and ℓ_i, ℓ_j are coprime for $i \neq j$, the correctness of Step 3 follows from chinese remaindering. Finally notice that $\tilde{p} = 0 \bmod S_j$ if and only if $\sigma(\tilde{p}) = 0 \bmod \sigma(S_j)$ as σ is an invertible linear transformation. The check $\sigma(\tilde{p}) = 0 \bmod \sigma(S_j)$ is done recursively in Step 3.2.

Complexity of Algorithm KS-Test

At the start, Algorithm **KS-Test** is called on polynomial p. So, at every intermediate level $\deg(S_j) \leq \deg(T_1) \leq d$. Therefore, $\dim_{\mathbb{F}}(\mathscr{R}') \leq d \cdot \dim_{\mathbb{F}}(\mathscr{R})$. Time spent by Algorithm **KS-Test** is bounded by $\mathsf{poly}(n, k', d, \dim_{\mathbb{F}}(\mathscr{R}))$ in Steps 1, 2, 3.1 and 4. Moreover, time spent in Step 3.2 is at most d times a smaller problem (with top fan-in reduced by 1) while dimension of the underlying local ring gets raised by a factor at most d. Unfolding the recursion, we get the time complexity of Algorithm **KS-Test** on input p to be $\mathsf{poly}(n, d^k)$.

3.3.4 Adapting Algorithm KS-Test to solve Problem 3.1

We now wish to identity test a polynomial of the form $p + f$ where p is computed by a $\Sigma\Pi\Sigma(k)$ and f is an r-semidiagonal circuit (cf. Definition 3.1). Though $p + f$ is a $\Sigma\Pi\Sigma$ circuit, it is a $\Sigma\Pi\Sigma$ circuit of unbounded top fan-in and hence we can not apply the Kayal-Saxena test directly. We shall apply the test on p, which is a fanin-k depth-3 circuit, and track the evolution of f in the process. We shall see that the semidiagonal structure of f is preserved throughout the execution. To begin with, f an r-semidiagonal circuit.

Just as in the Kayal-Saxena test, an intermediate level of the recursion would involve a local ring $\mathscr{R} = \mathbb{F}[x_1, \ldots, x_c]/(\ell_1^{e_1}, \ldots, \ell_c^{e_c})$. As before, we shall refer to the variables $\{x_{c+1}, \ldots, x_n\}$ as the *free variables*. But now, we have to test if a polynomial of the form

$$q = \sum_{i=1}^{k'} T_i + \sum_{t=1}^{s'} \omega_t$$

is zero where each T_i is a product of linear forms and each ω_t is an r-semidiagonal term. Each of the T_i's, like earlier, can be expressed as $T_i = \alpha_i \cdot \prod_{j=1}^{d} (\ell_{ij} + \tau_{ij})$ for some $\alpha_i \in \mathscr{R}$, linear forms ℓ_{ij} over the free variables, and nilpotents $\tau_{ij} \in \mathscr{R}$. And each ω_t is an r'-semidiagonal term that can be written as

$$\omega_t = \beta_t \cdot m_t \prod_{i=1}^{b} (\ell_{it} + \tau_{it})^{e_{it}} \tag{3.1}$$

where m_t is a monomial in the free variables and $\beta_t \in \mathscr{R}$. Let $\tilde{p}$ denote the part $T_1 + \cdots + T_{k'}$, and let $\tilde{f}$ denote the part $\sum_{t=1}^{s'} \omega_t$. The polynomials $\tilde{p}$ and $\tilde{f}$ are the evolutions of p and f in the course of the algorithm. At the beginning $\tilde{p} = p$ and $\tilde{f} = f$, and the algorithm shall maintain the invariant that $c + k' \leq k$ and $r' \leq d^c \cdot r$, which would ensure that $\tilde{f}$ stays semidiagonal.

In this modification, we would need to compute the leading monomial and coefficient of $\tilde{f}$ as well. We know from Theorem 3.6 that we can compute the leading monomial and coefficient of a semidiagonal polynomial over $\mathbb{F}$, but here $\tilde{f}$ is a semidiagonal polynomial over a local ring $\mathscr{R}$. The same approach, with minor modifications, can be used to compute the leading monomial and coefficient of $\tilde{f}$ as well, but we shall defer that to later. For now let us assume that we know how to compute $\mathsf{LM}\left(\tilde{f}\right)$ and $\mathsf{LC}\left(\tilde{f}\right)$. Below is the modified algorithm, with the changes from Algorithm **KS-Test** marked.

Algorithm KS-TEST-MODIFIED$\left(\mathscr{R}, \{T_1, \ldots, T_{k'}\}, \tilde{f}\right)$:

Step 1: (Rearranging terms) Order the terms T_i so that $\mathsf{LM}(T_1) \succeq \mathsf{LM}(T_i)$, for all $2 \leq i \leq k'$.

Step 1.1: If $\mathsf{LM}(\tilde{f}) \succ \mathsf{LM}(T_1)$, return NO.

Step 2: (Base Cases) If $\tilde{f} = 0$, return **KS-Test**$(\mathscr{R}, \{T_1, \ldots, T_{k'}\})$. If $k' = 0$ check if $\tilde{f} = 0$. If so, return YES, and NO otherewise.

If $\mathsf{LM}(T_1) \in \mathscr{R}$, then each of the T_i's and $\tilde{f}$ are just elements of $\mathscr{R}$. Add the elements and return YES if zero, and NO otherwise.

Step 3: (Verifying that $\tilde{p} + \tilde{f} = 0 \bmod T_1$) By suitably grouping factors of T_1, it can be written as $T_1 = \alpha_1 \cdot S_1 \ldots S_m$ where $\alpha_1 \in \mathscr{R}$, and each S_j is of the form

$$S_j = (\ell_j + \tau_1) \cdot (\ell_j + \tau_2) \ldots (\ell_j + \tau_{t_j}),$$

where each τ_i is a nilpotent in $\mathscr{R}$ and ℓ_j is a non-zero linear form over the free variables. Further, ℓ_i and ℓ_j are coprime linear polynomials (over $\mathbb{F}$) when $i \neq j$. Check if $\tilde{p} + \tilde{f} = 0 \bmod S_j$, for every $1 \leq j \leq m$, in the following way.

Step 3.1: (Building the new ring) Suppose $\ell_j = c_u x_u + \ell'_j$ where x_u is a free variable, $0 \neq c_u \in \mathbb{F}$ and ℓ'_j is independent of x_u. Define an invertible linear transformation on the free variables that maps x_u to $c_u^{-1}(x_u - \ell'_j)$ and leaves every other variable unchanged. In other words, σ is defined such that $\sigma(\ell_j) = x_u$. Hence, $\sigma(S_j) = (x_u + \tau_1)\ldots(x_u + \tau_{t_j})$. Define the ring

$$\mathscr{R}' = \mathscr{R}[x_u]/(\sigma(S_j))$$

Step 3.2: (Recursively verify if $\sigma(\tilde{p} + \tilde{f}) = 0 \bmod \sigma(S_j)$) Notice that we have $\sigma(T_1) = 0 \bmod \sigma(S_j)$ as S_j divides T_1.

Recursively call **KS-TEST-MODIFIED**$\left(\mathscr{R}', \left\{\sigma(T_2), \ldots, \sigma(T'_{k'})\right\}, \sigma(\tilde{f})\right)$. Return NO if the call returns NO, otherwise declare $p + \tilde{f} = 0 \bmod S_j$.

Declare $\tilde{p} + \tilde{f} = 0 \bmod T_1$, if $\tilde{p} + \tilde{f} = 0 \bmod S_j$ for all $1 \leq j \leq m$.

Step 4: Check if $[\mathsf{LM}(T_1)](\tilde{p} + \tilde{f}) = 0$ using the fact that

$$[\mathsf{LM}(T_1)](\tilde{p} + \tilde{f}) \quad = \quad \begin{cases} \displaystyle\sum_{\mathsf{LM}(T_i) = \mathsf{LM}(T_1)} \mathsf{LC}(T_i) & \text{if } \mathsf{LM}(\tilde{f}) \prec \mathsf{LM}(T_1) \\ \mathsf{LC}(\tilde{f}) \quad + \displaystyle\sum_{\mathsf{LM}(T_i) = \mathsf{LM}(T_1)} \mathsf{LC}(T_i) & \text{otherwise} \end{cases}$$

Return YES if zero, and NO otherwise.

Correctness of Algorithm KS-Test-Modified

Recall that in Step 1 of Algorithm **KS-Test**, we rearranged terms to have $\mathsf{LM}(T_1) \succeq \mathsf{LM}(T_i)$ for all $2 \leq i \leq k'$. The purpose of this step was to ensure that $\mathsf{LM}(T_1) \succeq \mathsf{LM}(\tilde{p})$, so that $\tilde{p} = 0 \bmod T_1$ implies that $\tilde{p} = \alpha \cdot T_1$ for some $\alpha \in \mathscr{R}$. Since we are now dealing with $\tilde{p} + \tilde{f}$, we need to account for the contribution of $\tilde{f}$ as well. Note that if $\mathsf{LM}(\tilde{f}) \succ \mathsf{LM}(T_i)$ for all $1 \leq i \leq k'$ then surely $\tilde{p} + \tilde{f} \neq 0$. Otherwise, T_1 (after reordering) satisfies $\mathsf{LM}(T_1) \succeq \mathsf{LM}(\tilde{p})$ and we can proceed to Step 2. This is precisely what is checked in Step 1.1 of the modified algorithm. Therefore if $\tilde{p} + \tilde{f} = 0 \bmod T_1$, then $\tilde{p} + \tilde{f} = \alpha \cdot T_1$, and this is checked in Step 3. Note that Step 1.1 ensures that, in Step 3, we just have to check if $\tilde{p} + \tilde{f} = 0 \bmod T_1$ rather than $\tilde{p} + \tilde{f} = 0 \bmod \tilde{f}$ (which, presumably, is a much harder task).

Step 2 of the modified algorithm also handles the base case when $k = 0$, where we have to check if $\tilde{f} = 0$ in $\mathcal{R}$.

Step 3 remains the same as before, and the only property that needs to be ensured is that $\sigma(\tilde{f})$ continues to stay semidiagonal. Note that the choice of σ ensures that at most one new free variable is mapped to a linear form (Step 3.2 only replaces x_u, and other variables remain unchanged). Therefore, $\sigma(\tilde{f})$ is a sum of the terms $\sigma(\omega_t)$'s, and each $\sigma(\omega_t)$ has at most one power of a linear form more than ω_t. In other words, if $\tilde{f}$ is r'-semidiagonal, then $\sigma(\tilde{f})$ is dr'-semidiagonal. Again Lemma 3.9 ensures that $\tilde{p} + \tilde{f} = 0 \bmod S_j$ for all $1 \leq j \leq m$ implies $\tilde{p} + \tilde{f} = 0 \bmod T_1$.

Finally, in Step 4, we need to confirm if $[\mathsf{LM}(T_1)](\tilde{p}+\tilde{f}) = 0$. Since we have ensured in Step 1.1 that $\mathsf{LM}(\tilde{f}) \preceq \mathsf{LM}(T_1)$, Step 4 of the modified algorithm additional accounts for the contribution of $\tilde{f}$ to the sum depending on whether $\mathsf{LM}(\tilde{f}) = \mathsf{LM}(T_1)$ or not.

Also note that to begin with $k' = k$, $c = 0$, and $r' = r$. And in each recursive stage of the algorithm, c increases by at most one, r' increases by a factor of d and k' decreases by 1. Hence, we always have that $c + k' \leq k$ and $r' \leq rd^c \leq d^k$.
We now shall see how the leading monomial and coefficient of $\tilde{f}$ can be computed.

Computing $\mathsf{LM}\left(\tilde{f}\right)$ - From Equation 3.1

$$\begin{aligned} \tilde{f} &= \sum_{t=1}^{s'} \omega_t \\ \text{where each} \quad \omega_t &= \beta_t \cdot m_t \cdot \prod_{i=1}^{b'} (\ell_{it} + \tau_{it})^{e_{it}} \end{aligned}$$

By using the binomial expansion on $(\ell_{it} + \tau_{it})^{e_{it}}$, we can express $\tilde{f}$ as an $\mathcal{R}$-linear combination of semidiagonal terms (reusing symbols β_t's):

$$\tilde{f} = \sum_{\substack{t \leq s' \\ e'_{it} \leq e_{it}}} \beta_t \cdot m_t \cdot \prod_{i=1}^{b'} \ell_{it}^{e'_{it}} \tag{3.2}$$

Note that each ℓ_{it} is a linear form over the base field $\mathbb{F}$, and hence the above expression is a $\mathcal{R}$-linear combination of semidiagonal terms over $\mathbb{F}$. Also, the number of summands in the above expression of $\tilde{f}$ is at most $r's'$ since $\tilde{f}$ is r'-semidiagonal.

Let $\{v_1, \ldots, v_{\dim_{\mathbb{F}} \mathscr{R}}\}$ be an $\mathbb{F}$-basis of $\mathscr{R}$ and let each β_i be expressed in this basis as

$$\beta_i = \sum_j b_{ij} v_j \quad \text{where } b_{ij} \in \mathbb{F}.$$

Then, Equation (3.2) can be split in terms of these basis vectors as follows:

$$\begin{aligned} \text{If} \quad \tilde{f}_j &\stackrel{\text{def}}{=} \sum_{\substack{t \leq s' \\ e'_{it} \leq e'_{it}}} b_{tj} \cdot m_t \prod_{i=1}^{b'} \ell_{it}^{e_{it}} \quad \text{for } j = 1, \ldots, \dim_{\mathbb{F}} \mathscr{R} \\ \text{then} \quad \tilde{f} &= \sum_{j=1}^{\dim_{\mathbb{F}} \mathscr{R}} v_j \cdot \tilde{f}_j \end{aligned}$$

Thus to compute the leading monomial (or coefficient) of $\tilde{f}$, it suffices to compute the leading monomial (or coefficient) of each of the $\tilde{f}_j$'s which are semidiagonal circuits over $\mathbb{F}$. Hence, the leading monomial and coefficient of $\tilde{f}$ can be computed in deterministic $\mathsf{poly}(n, d, r', s', \dim_{\mathbb{F}} \mathscr{R})$ time by $(\dim_{\mathbb{F}}(\mathscr{R}))$-many applications of Theorem 3.6.

Using an analysis similar to the complexity analysis of Algorithm **KS-Test** (presented in Section 3.3.3) and observing that $\dim_{\mathbb{F}} \mathscr{R} \leq d^k$, we see that Algorithm **KS-Test-Modified** takes time $\mathsf{poly}(n, r, s, d^k)$. This solves Problem 3.1 in deterministic polynomial time as promised. Summarizing this as a theorem:

Theorem 3.10. *Given a $\Sigma\Pi\Sigma(k)$ circuit computing a polynomial p, and an r-semidiagonal circuit computing a polynomial f, there is a deterministic $\mathsf{poly}(n, r, s, d^k)$-time algorithm to check if $p + f = 0$.* □

3.4 Solving Problem 3.2

This section shall address Problem 3.2, a special case of "sparse factorization verification":

> Given polynomials $f, g_1, \cdots, g_t$ explicitly as a sum of monomial with each g_i being a sum of univariates, check if $f - g_1 \ldots g_t = 0$.

The naïve approach of multiplying all g_i's is infeasible because of intermediate swelling in the number of monomials (as sparse polynomials could have factors of very large sparsity). Consider the following examples: the polynomial $\prod_{i=1}^{n}(x_i^d - 1)$ has $s = 2^n$ monomials, and its factor $\prod_{i=1}^{n}(x_i^{d-1} + x_i^{d-2} + \ldots + 1)$ has $d^n = s^{\log d}$ monomials. Over the finite field $\mathbb{F}_p$, the polynomial $(\ell_1^p - \ell_2^p)$ has $2n$ monomials (for any choice of linear forms ℓ_1, ℓ_2) and has a factor $(\ell_1^p - \ell_2^p)/(\ell_1 - \ell_2)$ which has exponentially many monomials for a generic choice of ℓ_1, ℓ_2. Thus the naïve approach may lead to one of these large factors intermediately.

Notice that, when g_i's are linear polynomials, Problem 3.2 becomes a special case of Problem 3.1 and can therefore be solved in deterministic polynomial time. However, the approach using chinese remaindering given in Section 3.3 does not seem to generalize directly to the case when, instead of linear functions, g_i's are sums of univariates. This case is handled in this section.

3.4.1 Checking divisibility by (a power of) a sum of univariates

Given $g_1, \ldots, g_t$, group together the polynomials g_i's that are just $\mathbb{F}$-multiples of each other. After this is done, we need to check if f is equal to a product of the form $a \cdot g_1^{d_1} \ldots g_t^{d_t}$ (reusing symbol t) for some $a \in \mathbb{F}$, where no two g_i's are scalar multiples of each other. Suppose g_i and g_j are coprime for $i \neq j$ (this assumption is justified later in Section 3.4.2 by essentially proving them irreducible). Then, Problem 3.2 gets reduced to the problem of checking divisibility followed by a comparison of the leading monomials of f and $a \cdot g_1^{d_1} \ldots g_t^{d_t}$. The latter is easy as we have f and g_i's explicitly. Checking divisibility, however, is more technical and we do that in this section. We once again use Theorem 3.6, but on a slightly more general form of semidiagonal polynomials.

Theorem 3.11. *Checking divisibility of a sparse polynomial f by g^d, where g is a sum of univariates, can be done in deterministic polynomial time.*

Proof. Let $g = \sum_{i=1}^{n} u_i(x_i)$, where u_i is a univariate in x_i. Assume without loss of generality that $u_1 \neq 0$ and let $\deg_{x_1} u_1 = e$. By replacing the partial sum $\sum_{i=2}^{n} u_i(x_i)$ in g by a new variable y, let $h(x_1, y) = (u_1(x_1) + y)^d$ be the bivariate thus obtained. Since h^d is a sparse polynomial that is monic in x_1, for every $k \geq ed$ we can employ long division and uniquely express $x_1^k = q_k \cdot h^d + r_k$ where $\deg_{x_1} r_k < ed$ and $\deg_y r_k \leq k$. Further, since each r_k is a bivariate, the number of monomials in them is bounded by

$ed(k+1)$. Using this, f can be written as $q \cdot h^d + r$ where $\deg_{x_1} r < ed$ and the number of monomials in $r(x_1, \ldots, x_n, y)$ is bounded by $s \cdot ed(d+1)$ where the s is the number of monomials in f. Substituting $y = \sum_{i=2}^{n} u_i(x_i)$, we get

$$f(x_1, \ldots, x_n) \quad = \quad \tilde{q} \cdot g^d + \tilde{r}\,(x_1, \ldots, x_n)$$

where $\tilde{r}\,(x_1, \ldots, x_n) = r\left(x_1, \ldots, x_n, \sum_{i=2}^{n} u_i(x_i)\right)$. Since $\deg_{x_1} \tilde{r} < ed = \deg_{x_1} g^d$ it follows that $f = 0 \bmod g^d$ if and only if $\tilde{r} = 0$.

Polynomial $\tilde{r}$ is of the form of a sum of products, where each product term looks like $m \cdot (\sum_{i=2}^{n} u_i(x_i))^j$ for some monomial m and $j \leq \deg f$. This form is similar to that of a semidiagonal circuit (Definition 3.1), except that $\sum_{i=2}^{n} u_i(x_i)$ is a sum of univariates instead of a linear form. Nevertheless, Theorem 3.6 would still allow us to check if this is zero in deterministic polynomial time (explained in remark following Theorem 3.6).

The polynomial $\tilde{r}$ can be constructed in polynomial time, and identity testing $\tilde{r}$ also takes polynomial time by Theorem 3.6. □

3.4.2 Irreducibility of a sum of univariates

As mentioned in Section 3.4.1, we would like to employ chinese remaindering to solve Problem 3.2. In this section, we show that any sum of univariates that depends on at least *three* variables non-trivially is either irreducible or a p-th power of some polynomial (in characteristic p).

Theorem 3.12. *(Irreducibility) Let g be a polynomial over a field $\mathbb{F}$ that is a sum of univariates. Suppose g depends non-trivially on at least three variables. Then either g is irreducible, or g is a p-th power of some polynomial where $p = char(\mathbb{F})$.*

Remark. Such a statement is false when g is a sum of just two univariates. For eg., the real polynomial $g := x_1^4 + x_2^4$ has factors $(x_1^2 + x_2^2 \pm x_1 x_2 \sqrt{2})$ (which is not even a sum of univariates!).

We need the following observations (with g being a sum of univariates as in Theorem 3.12).

Observation 3.13. *Let $g = u \cdot v$ be a non-trivial factorization. Then both u and v are monic in every variable that g depends on. In particular, they depend non-trivially on every variable that g depends on.*

Proof. If u is not monic in, say, x_1 then fix an ordering amongst the variables under which x_1 is the highest. Then the leading monomial of $g = u \cdot v$ is a mixed term, and that is not possible as g is a sum of univariates. □

Observation 3.14. *If g is not a p-th power of any polynomial then it is square-free.*

Proof. Suppose not, then $g = u^2 v$ for some polynomials u and v. If g is not a p-th power, there must exist a variable x_i such that $0 \neq \partial_i g = 2uv\partial_i u + u^2 \partial_i v$. Since u divides the RHS, u must be a univariate as $\partial_i g$ is a univariate in x_i. But this forces g to be a univariate as u is also a factor of g. □

Proof of Theorem 3.12. Suppose that g is not a p-th power of any polynomial. Then there exists a variable, say x_1, such that $\partial_1 g \neq 0$. Suppose $g = u \cdot v$; this means $\partial_i g = (\partial_i u)v + (\partial_i v)u$.

There are two variables other than x_1, say x_2 and x_3, that g depends non-trivially on. If one of these partial derivatives is zero, say $\partial_2 g = 0$, then $0 = (\partial_2 u)v + (\partial_2 v)u$. Since both u and v are monic in x_1 (by Observation 3.13), $\deg_{x_1}(\partial_2 u)$ is strictly less than $\deg_{x_1} u$ and similarly for v. As we also know that u and v do not share any common factor (since g is square-free, by Observation 3.14), the above equation forces $\partial_2 u = \partial_2 v = 0$. This implies that every occurrence of x_2 in g, u and v has an exponent that is a multiple of $p = \text{char}(\mathbb{F})$. Hence,

$$\begin{array}{rrcl} & g' & = & u' \cdot v' \\ \text{where} & g(x_1, x_2, \ldots, x_n) & =: & g'(x_1, x_2^p, \ldots, x_n) \\ & u(x_1, x_2, \ldots, x_n) & =: & u'(x_1, x_2^p, \ldots, x_n) \\ & v(x_1, x_2, \ldots, x_n) & =: & v'(x_1, x_2^p, \ldots, x_n). \end{array}$$

and we can induct on this “smaller” equation $g' = u'v'$.

Suppose that both $\partial_2 g$ and $\partial_3 g$ are non-zero. Denote by $h_{(x_i=\alpha)}$, the polynomial h evaluated at $x_i = \alpha$, where $\alpha \in \overline{\mathbb{F}}$ (the algebraic closure of $\mathbb{F}$).

Claim 3.15. *There exists an $\alpha \in \overline{\mathbb{F}}$ such that $u_{(x_1=\alpha)}, v_{(x_1=\alpha)} \neq 0$ and they share a non-trivial common factor.*

For now, let us assume the claim is true and complete the proof. Since $\partial_2 g$ and $\partial_3 g$ do not depend on x_1, evaluating x_1 to α doesn't change the LHS.

$$\begin{aligned} (\partial_2 g)_{(x_1=\alpha)} = \quad \partial_2 g &= (\partial_2 u)_{(x_1=\alpha)}\, v_{(x_1=\alpha)} + (\partial_2 v)_{(x_1=\alpha)}\, u_{(x_1=\alpha)}. \\ \text{Similarly,} \qquad \partial_3 g &= (\partial_3 u)_{(x_1=\alpha)}\, v_{(x_1=\alpha)} + (\partial_3 v)_{(x_1=\alpha)}\, u_{(x_1=\alpha)}. \end{aligned}$$

Since both $\partial_2 g$ and $\partial_3 g$ are non-zero, then the last two equations imply that the gcd of $u_{(x_1=\alpha)}$ and $v_{(x_1=\alpha)}$ (which is non-trivial, by Claim 3.15) divides the LHS of both the equations. But this leads to a contradiction as the LHS of the two equations are univariates in x_2 and x_3 respectively. □

Proof of Claim 3.15.

Suppose $g_1 = \partial_1 g$, $u_1 = \partial_1 u$ and $v_1 = \partial_1 v$, and let $w = \gcd(g_1, u_1, v_1)$ which is a univariate in x_1 as g_1 is a univariate. Consider the following equation

$$0 \neq \frac{g_1}{w} = \left(\frac{u_1}{w}\right) v + \left(\frac{v_1}{w}\right) u.$$

Note that neither of u_1 and v_1 is zero. This is because, if $u_1 = 0$ then $g_1 = v_1 \cdot u$. This then forces u to be a univariate in x_1, which is not possible as u is also a factor of g.

Since x_1-degree of u_1 is less than that of g_1, the univariate g_1/w has degree in x_1 at least one. Let $\alpha \in \bar{\mathbb{F}}$ be a root of g_1/w. Substituting $x_1 = \alpha$ in the above expression, we get an equation of the form

$$\tilde{u} \cdot v_{(x_1=\alpha)} + \tilde{v} \cdot u_{(x_1=\alpha)} = 0, \tag{3.3}$$

where $\tilde{u} = (u_1/w)_{(x_1=\alpha)}$ and $\tilde{v} = (v_1/w)_{(x_1=\alpha)}$. Since u and v are factors of g which depends non-trivially on other variables, it follows that $u_{(x_1=\alpha)}$ and $v_{(x_1=\alpha)}$ are non-zero. Further, $\tilde{u}$ and $\tilde{v}$ cannot both be zero as $\frac{g}{w}, \frac{u}{w}$ and $\frac{v}{w}$ do not share any common factor (in particular $x_1 - \alpha$). Hence, the above equation is a non-trivial equation.

Since u is monic in x_2 (by Observation 3.13), degree of x_2 in u_1 is strictly less than that in u. Which means, degree of x_2 in $\tilde{u}$ is also strictly less than degree of x_2 in $u_{(x_1=\alpha)}$. Therefore, by treating the terms $\tilde{u}, \tilde{v}, u_{(x_1=\alpha)}, v_{(x_1=\alpha)}$ as polynomials in x_2 over the function field $\bar{\mathbb{F}}(x_3, \ldots, x_n)$, we can conclude from Equation 3.3 that $u_{(x_1=\alpha)}$ and $v_{(x_1=\alpha)}$ must share a nontrivial factor. □

3.4.3 Finishing the argument

Theorem 3.11 and 3.12 essentially give the solution to Problem 3.2. Although, Theorem 3.12 justifies our assumption of g_i's being essentially coprime when g_i depends on three or more variables, we need to slightly change our strategy for g_i's that are bivariates or univariates. In this case, we first take pairwise gcd of the bivariates (similarly, pairwise gcd of the univariates) and factorize accordingly till all the bivariates (similarly, the univariates) are mutually coprime. Taking gcd of two bivariate (monic) polynomials takes only polynomial time using Euclidean gcd algorithm (by long division). Once coprimeness is ensured, we can directly check if a bivariate $g_i^{d_i}$ divides f by expressing f as $f = \sum_j f_j m_j$, where f_j's are bivariate polynomials depending on the same two variables as g_i, and m_j's are monomials in the remaining variables. Then,

$$g_i^{d_i} \mid f \quad \text{if and only if} \quad g_i^{d_i} \mid f_j \text{ for all } j.$$

Once again, just like gcd, bivariate divisibility is a polynomial time computation (simply by long division). Finally, we can use chinese remaindering to complete the argument in a similar fashion as in Section 3.4.1. We now summarize this as a theorem.

Theorem 3.16. *Let $f, g_1, \ldots, g_t$ be n-variate polynomials given explicitly as a sum of at most s monomials, each of whose degrees are bounded by d and each g_i being a sum of univariates. Then, we can check if $f = g_1 \ldots g_t$ in deterministic time* $\mathsf{poly}(n, d, t, s)$. □

Identity testing via algebraic independence 4

4.1 Introduction

During the past decade, the quest for derandomization of PIT has yielded several results on restricted models of circuits. But, fortunately, the search has been made more focussed by a result [AV08, VSBR83] which states that a polynomial time *blackbox* derandomization of identity testing for depth-4 circuits (via a certain pseudo-random generator) implies a quasi-polynomial time derandomization of PIT for *poly-degree*[1] circuits.

With depth-4 as the final frontier, the results that have been achieved so far include polynomial time hitting-set generators for the following models:

- depth-2 ($\Sigma\Pi$) circuits (or *sparse* polynomials) [KS01, AB99],
- depth-3 ($\Sigma\Pi\Sigma$) circuits with constant top fanin [SS11],
- depth-4 ($\Sigma\Pi\Sigma\Pi$) multilinear circuits with constant top fanin [SV11],
- constant-depth constant-read multilinear formulas [AvMV11],
- circuits *generated* by sparse polynomials with low *transcendence degree* [BMS11a].

To our knowledge, these are the only instances for which polynomial time hitting-set generators are known. The result on depth-3 bounded top fanin circuits is based upon the Chinese Remaindering technique of [KS07] and the ideal-theoretic framework studied in [SS10b]. Their work followed after a sequence of developments in rank bound estimates [DS05, KS08, SS09, KS09b, SS10b], some using results from incidence geometry (although, this result [SS11] in particular is not rank based). On

[1] Circuits computing polynomials with degree bounded by a polynomial function in the size of the circuit.

the other hand, the results on constant-depth multilinear formulas [AvMV11, SV11] is obtained by building upon and extending the techniques of other earlier results [KMSV10, SV09, SV08] on 'read-once' models. At a high level, this involved a study of the structure of multilinear formulas under the application of partial derivatives with respect to a carefully chosen set of variables and invoking depth-3 rank bounds (cf. [SY10] for details). More recently, a third technique has emerged in [BMS11a] which is based upon the concept of *algebraic independence* of polynomials. They showed that for any given circuit C of polynomial degree and sparse polynomials $f_1, \ldots, f_m$ with constant *transcendence degree*, a hitting-set generator for $C(f_1, \ldots, f_m)$ can be constructed in polynomial time.

4.1.1 Contribution of this chapter

With these diverse techniques floating around the study of hitting-set generators, one wonders: Could there be one single tool that is sufficiently powerful to capture all these models? Is there any unique feature underlying these seemingly different models that can lend itself to the conception of such a unifying tool? The answer to both these questions, as we shall see in this chapter, is *yes*. The key to this lies in studying the properties of the *Jacobian*, a mathematical object lying at the very core of algebraic independence. And as for the 'unique feature', notice that in the above four models some *parameter* of the circuit is *bounded* – be it bounded top fanin, bounded read of variables, or bounded *transcendence degree*. At an intuitive level, it seems to us that it is this 'bounded parameter'-ness of the circuit that makes the Jacobian perform at its best.

In the process of finding a universal technique, we strengthen the earlier results significantly thus giving the first blackbox PIT for these generalized models. Besides $\Sigma\Pi\Sigma(k)$ circuits, we construct blackbox PITs for circuits of the form $C(T_1, \cdots, T_m)$ where C is *any* polynomial of polynomial degree and T_i's are products of linear functions with bounded transcendence degree. Further, we remove the multilinear restriction completely from the constant-depth constant-read models. The notion of 'read' is also replaced by a general notion of '*occur*' , which additionally generalizes PIT on multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuits as well.

At this point, one is faced with a natural question: how effective is this new tool in proving lower bounds? The intimate connection between efficient algorithms and lower bounds has recurrently appeared in various contexts [Wil11, Rag08, Uma03, PSZ00, IW97]. For arithmetic circuits, this link is provably tight [KI03, Agr05, AV08]: Derandomizing identity testing is *equivalent* to proving circuit lower bounds. This suggests that one might have to look for techniques that are powerful enough to handle the dual worlds of algorithm design and lower bounds with equal effectiveness. For example the *partial derivative technique* has been used to prove lower bounds and identity testing (albeit non-blackbox) on restricted models (survey [CKW11]); the *τ-conjecture* is another such example [GKPS11]. In this chapter, we shall see that the Jacobian is yet another tool using which we can prove exponential lower bounds for the determinant/permament on the *same* depth-3 and depth-4 models for which we give efficient PIT algorithms. In particular, this includes depth-4 constant-occur formulas, depth-4 circuits with constant transcendence degree of the underlying sparse polynomials (which significantly generalizes the lower bound result in [GKPS11]), and depth-3 circuits with constant transcendence degree of the polynomials computed by the product gates.

To state the results more formally, we would need the following definition.

Definition 4.1. *A set of polynomials* $\mathbf{f} = \{f_1, \cdots, f_m\} \subset \mathbb{F}[x_1, \cdots, x_n]$ *is* algebraically independent *over* $\mathbb{F}$ *if there is no nonzero polynomial* $H \in \mathbb{F}[y_1, \cdots, y_m]$ *such that* $H(f_1, \cdots, f_m)$ *is identically zero.*

A maximal subset of $\mathbf{f}$ *that is algebraically independent is a* transcendence basis *of* $\mathbf{f}$ *and the size of such a basis is the* transcendence degree[2] *of* $\mathbf{f}$ *(denoted* $\mathrm{trdeg}_{\mathbb{F}}\mathbf{f}$*).*

The first result is the generalization of PIT on $\Sigma\Pi\Sigma(k)$ circuits.

Theorem 4.2. *Let* C *be a poly-degree circuit of size* s *and each of* $T_1, \ldots, T_m$ *be a product of* d *linear polynomials in* $\mathbb{F}[x_1, \ldots, x_n]$ *such that* $\mathrm{trdeg}_{\mathbb{F}}\{T_1, \ldots, T_m\} \leq r$. *A hitting-set for* $C(T_1, \ldots, T_m)$ *can be constructed in time polynomial in* n *and* $(sd)^r$, *assuming* $\mathrm{char}(\mathbb{F}) = 0$ *or larger than* d^r.

[2] Since algebraic independence satisfies the matroid property cf. [Oxl92], transcendence degree is well-defined.

If C is a single $+$ gate, we get a hitting-set generator for depth-3 circuits with constant transcendence degree of the polynomials computed by the product gates (there is *no* restriction on top fanin).

The second result uses the following generalization of *read-k* formulas (where every variable appears in at most k leaf nodes of the formula) to *occur-k* formulas. Two reasons behind this generalization are: Firstly, to accommodate the power of exponentiation — if we take the e-th power of a read-k formula using a product gate, the 'read' of the resulting formula goes up to ek. We would like to avoid this superfluous blow up in read. Secondly, a read-k formula has size $O(kn)$, which severely hinders its power of computation - for instance, determinant and permanent cannot even be expressed in this model when k is a constant [Kal85b]. This calls for the following definition.

Definition 4.3. *An* occur-k formula *is a rooted tree with internal gates labelled by* $+$ *and* $\hat{\times}$*. A* $\hat{\times}$ *gate, on inputs* $g_1, \ldots, g_m$ *with incoming edges labelled* $e_1, \ldots, e_m \in \mathbb{N}$*, computes* $g_1^{e_1} \cdots g_m^{e_m}$*. At the leaves of this tree are depth-2 formulas computing sparse polynomials (*leaf nodes*), where every variable occurs in at most* k *of these sparse polynomials.*

We shall define the size and depth of circuits slightly differently. Since the point of a $\hat{\times}$ gate was to simulate powering along with multiplication, we shall define the size of such a $\hat{\times}$ gate as $(e_1 + \cdots + e_m)$ (akin to the number of wires feeding into a $\times$ gate that simulates this product). The size of each $+$ gate, as usual, is counted as 1. The size of each leaf node is the size of the corresponding depth-2 circuit. With these conventions, the size of the circuit is the sum of sizes of each of its gates (and leaves). The depth of the circuit is the number of layers of $+$ and $\hat{\times}$ gates plus 2 (to account for the depth-2 circuits at the leaves).

Thus, occur-k is more relaxed than the traditional read-k as it packs the "power of powering" (to borrow from [GKPS11]), and the leaves are sparse polynomials (at most kn many) whose dependence on its variables is arbitrary. E.g. $(x_1^3 x_2 + x_1^2 x_3^2 + x_1 x_4)^e$ is *not* read-1 but is trivially depth-3 occur-1. This relaxation is also similar to "sparse-substituted variants" in the earlier results on read-k formulas.

Theorem 4.4. *A hitting-set for a depth-D occur-k formula of size* s *can be constructed in time polynomial in* s^R*, where* $R = (2k)^{4D \cdot 2^D}$ *(assuming* $\mathrm{char}(\mathbb{F}) = 0$ *or* $> s^R$*).*

A tighter analysis for depth-4 occur-k formulas yields a better time complexity. Note that a depth-4 occur-k formula allows unbounded top fanin. Also, it can be easily seen to subsume $\Sigma\Pi\Sigma\Pi(k)$ multilinear circuits studied by [SV11, KMSV10].

Theorem 4.5. *A hitting-set for a depth-4 occur-k formula of size s can be constructed in time polynomial in s^{k^2} (assuming $\mathrm{char}(\mathbb{F}) = 0$ or $> s^{2k}$).*

For constant-depth, the above theorems not only remove the restriction of multilinearity (and relax read-k to occur-k), but further improve upon the time complexity of [AvMV11] and [SV11]. The hitting-set generator of [AvMV11] works in time $n^{k^{O(k^2)}+O(kD)}$, which is super-exponential when $k = \Omega(s^{\varepsilon/2D\cdot 2^D})$ for any positive $\varepsilon < 1$ and a constant D, whereas the generator in Theorem 4.4 runs in subexponential time for the same choice of parameters. The running time of [SV11] is $s^{O(k^3)}$, which is slightly worse than that of Theorem 4.5. However, the hitting-set generator of [AvMV11] is quasi-polynomial sized even for non-constant depth whereas Theorem 4.4 does not give anything non-trivial for $O(\log n)$ depth.

Since any polynomial has an exponential-sized depth-2, occur-1 formula (just the sparse representation), proving lower bounds on this model is an interesting proposition in its own right.

Theorem 4.6. *Any depth-4 occur-k formula that computes Det_n or Perm_n must have size $s = 2^{\Omega\left(n/k^2\right)}$ over any field of characteristic zero.*

Our next result is an exponential lower bound on the model for which hitting-set was developed in [BMS11a] (but no lower bound was shown). It is also an improvement over the result obtained in [GKPS11] which holds only for more restricted depth-4 circuits over reals.

Theorem 4.7. *Let C be any circuit. Let $f_1,\ldots,f_m$ be sparse polynomials (of any degree) with sparsity bounded by s and their* trdeg *bounded by r. If $C(f_1,\ldots,f_m)$ computes Det_n or Perm_n, then $s = 2^{\Omega(n/r)}$ over any field of characteristic zero.*

The next result is on the model for which hitting-set is given by Theorem 4.2.

Theorem 4.8. *Let C be any circuit and $T_1,\ldots,T_m$ be products of linear polynomials. If $C(T_1,\ldots,\, T_m)$ computes Det_n or Perm_n then $\mathrm{trdeg}_{\mathbb{F}}\{T_1,\ldots,T_m\} = \Omega(n)$ over any field of characteristic zero.*

We shall now see the main ideas used in all the above results.

4.1.2 The main ideas

To a set of polynomials $\{T_1,\ldots,T_m\}$ we associate a polynomial, called the Jacobian $J(T_1,\ldots,T_r)$ (assuming this to be a transcendence basis of the T_i's), that captures the algebraic independence of $T_1,\ldots,T_r$.

If we could find an r-variate linear map Ψ such that $\Psi(T_1),\ldots,\Psi(T_r)$ algebraically independent, then it can be shown that for *any* C: $C(T_1,\ldots,T_m)=0$ if and only if $C(\Psi(T_1),\ldots,\Psi(T_m))=0$. Turns out that to find such a map Ψ, it suffices to find an r-variate linear map Φ such that $\Phi\circ J(T_1,\ldots,T_r)\neq 0$. For generic T_i's, the Jacobian is usually a difficult polynomial to work with, and so is finding Φ. However, for the special models in this chapter we shall be able to design Φ. The construction of Φ ultimately provides a hitting-set for $C(T_1,\ldots,T_m)$, as we reduce to a situation where r is constant.

The initial idea for lower bounds is similar. Suppose $\mathsf{Det}_n = C(T_1,\ldots,T_m)$. Then, by algebraic dependence, $J(\mathsf{Det}_n,T_1,\ldots,T_r)=0$. The proofs then exploit the nature of this identity for the special models. This part requires proving several combinatorial properties of the determinant/permanent.

4.2 Algebraic independence and the Jacobian

This section shall develop the main tools required for the variou results. The first definition we need is that of the Jacobian.

Definition 4.9 (Jacobian). *The* Jacobian *of a set of n-variate polynomials* $\mathbf{f}=\{f_1,\cdots,f_m\}$ *is the matrix* $\mathcal{J}_{\mathbf{x}}(\mathbf{f})=(\partial_{x_j}f_i)_{m\times n}$*, where* $\partial_{x_j}f_i=\partial f_i/\partial x_j$*. Let* $S\subseteq\mathbf{x}=\{x_1,\ldots,x_n\}$ *and* $|S|=m$*. Then* $J_S(\mathbf{f})$ *denotes the minor of* $\mathcal{J}_{\mathbf{x}}(\mathbf{f})$ *formed by the columns corresponding to the variables in* S*.*

The Jacobian of a set of polynomials has a very useful "linearizing effect". The *linear* rank (over the function field) of the Jacobian *exactly* captures their *algebraic rank* i.e. transcendence degree.

Fact 4.10 (Jacobian criterion). *Let* $\mathbf{f}\subset\mathbb{F}[\mathbf{x}]$ *be a finite set of polynomials of degree at most* d*, and* $\mathrm{trdeg}_{\mathbb{F}}\mathbf{f}\leq r$*. If* $\mathrm{char}(\mathbb{F})=0$ *or* $\mathrm{char}(\mathbb{F})>d^r$*, then* $\mathrm{trdeg}_{\mathbb{F}}\mathbf{f}=\mathrm{rank}_{\mathbb{F}(\mathbf{x})}\mathcal{J}_{\mathbf{x}}(\mathbf{f})$.

The proof of this fact may be seen in [BMS11b].

The next definition we need is that of a *faithful homomorphism*.

Definition 4.11 (Faithful homomorphism). *A homomorphism* $\Phi : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[\mathbf{y}]$ *(*$\mathbf{y}$ *is another set of variables) is said to be* faithful *to a finite set of polynomials* $\{f_1, \ldots, f_m\} \subset \mathbb{F}[\mathbf{x}]$ *if*

$$\mathrm{trdeg}_{\mathbb{F}} \{f_1, \ldots, f_m\} = \mathrm{trdeg}_{\mathbb{F}} \{\Phi(f_1), \ldots, \Phi(f_m)\}.$$

The intuition is that $\mathrm{trdeg}_{\mathbb{F}} \{\mathbf{f}\}$ measures the number of "effective variables" present amongst them. All the PITs in this chapter would try to construct a faithful homomorphism to $\mathbb{F}[y_1, \ldots, y_r]$ where $r \approx \mathrm{trdeg}_{\mathbb{F}} \{\mathbf{f}\}$.

The following result asserts that faithful homomorphisms for $\mathbf{f}$ do not change nonzeroness any polynomial combination of $\mathbf{f}$.

Theorem 4.12 (Faithful preserves nonzeroness). *Let* $\mathbf{f} = \{f_1, \cdots, f_m\} \subset \mathbb{F}[\mathbf{x}]$ *and* Φ *be a homomorphism faithful to* $\mathbf{f}$. *For any polynomial* $C \in \mathbb{F}[y_1, \cdots, y_m]$, $C(\mathbf{f}) = 0 \Leftrightarrow C(\Phi(\mathbf{f})) = 0$.

Before we see the proof, it would be useful to illustrate why this result is surprising. Let us say $\{f_1, \ldots, f_r\}$ is one maximal algebraically independent and Φ is a map that ensures that $\{\Phi(f_1), \ldots, \Phi(f_r)\}$ continue to remain algebraically independent. However, the circuit C could involve only $f_{s+1}, \ldots, f_m$ and it is far from obvious why just this suffices to preserve relations between them. But the fact that $\{f_1, \ldots, f_s\}$ is a maximal algebraically independent set forces all relations amongst the f_i's to be preserved exactly by Φ.

The first proof of was by Beecken, Mittmann and Saxena [BMS11b] and used a deep theorem from algebraic geometry called *Krull's hauptidealsatz*. The following is an alternate elementary proof of the above theorem using only basic field theory.

Proof of Theorem 4.12. Since Φ is faithful to $\mathbf{f}$, there is a transcendence basis (say, $f_1, \ldots, f_s$) of $\mathbf{f}$ such that $\Phi(f_1), \ldots, \Phi(f_s)$ is a transcendence basis of $\Phi(\mathbf{f})$. The function field $\mathbb{K} = \mathbb{F}(\mathbf{f})$ essentially consists of elements that are polynomials in $f_{s+1}, \ldots, f_m$ with coefficients from $\mathbb{F}(f_1, \ldots, f_s)$. Treating $C(\mathbf{f})$ as a nonzero element of $\mathbb{K}$, there is an inverse $Q \in \mathbb{K}$ such that $Q \cdot C = 1$. Since Q is a polynomial in $f_{s+1}, \ldots, f_m$ with coefficients from $\mathbb{F}(f_1, \ldots, f_s)$, by clearing off the denominators of these coefficients in Q, we get an equation $\tilde{Q} \cdot C = P(f_1, \ldots, f_s)$, where $\tilde{Q}$ is a nonzero polynomial in $\mathbf{f}$ and P is a nonzero polynomial in $f_1, \ldots, f_s$. Applying Φ to both sides of the equation, we conclude that $C(\Phi(\mathbf{f})) = \Phi(C(\mathbf{f})) \neq 0$, otherwise $P(\Phi(f_1), \ldots, \Phi(f_s)) = \Phi(P(f_1, \ldots, f_s)) = 0$ which is not possible as $\Phi(f_1), \ldots, \Phi(f_s)$ are algebraically independent and P is a nontrivial polynomial. □

A recipe for constructing faithful maps

Given a set of polynomials $\{f_1, \ldots, f_m\}$, the goal is to construct a map Φ such that $\{\Phi(f_1), \ldots, \Phi(f_m)\}$ continue to stay algebraically independent. By the Jacobian Criterion (Fact 4.13), we want to ensure that the rank of $\mathscr{J}(\Phi(f_1), \ldots, \Phi(f_m))$ is equal to the rank of $\mathscr{J}(f_1, \ldots, f_m)$. The first step forward would be to understand how the Jacobian evolves under such a homomorphism.

Fact 4.13 (Chain rule). *For any finite set of polynomials $\{f_1, \ldots, f_m\} \subset \mathbb{F}[\mathbf{x}]$ and a homomorphism $\Phi : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[\mathbf{y}]$, we have*

$$\mathscr{J}_{\mathbf{y}}(\Phi(f_1), \ldots, \Phi(f_m)) = \Phi\left(\mathscr{J}_{\mathbf{x}}(f_1, \ldots, f_m)\right) \cdot \mathscr{J}_{\mathbf{y}}(\Phi(x_1), \ldots, \Phi(x_n)) \qquad (4.1)$$

The proof of this fact follows directly from the chain-rule for differentiation.

The matrix $\Phi(\mathscr{J}_{\mathbf{x}}(f_1, \ldots, f_m))$ is an $m \times n$ matrix, and $\mathscr{J}_{\mathbf{y}}(\Phi(x_1), \ldots, \Phi(x_n))$ is an $n \times r$ matrix. Suppose we ensure that $\Phi(\mathscr{J}_{\mathbf{x}}(\mathbf{f}))$ has the same rank as $\mathscr{J}_{\mathbf{x}}(\mathbf{f})$, we additionally need to ensure that post-multiplication by $\mathscr{J}_{\mathbf{y}}(\Phi(\mathbf{x}))$ does not change the rank. Similar questions have been studied earlier in the context of *rank extractors* by Gabizon and Raz [GR05], and it is known that post-multiplication by a (suitable) Vandermonde matrix preserves the rank.

Lemma 4.14 ([GR05]). *Let A be a $m \times n$ matrix with entries in a field $\mathbb{F}$, and let t be an indeterminate. Then,* $\operatorname{rank}_{\mathbb{F}(t)}\left(A \cdot (t^{ij})_{i \in [n], j \in [m]}\right) = \operatorname{rank}_{\mathbb{F}} A$.

The next theorem shall simulate this by augmenting the map Φ slightly to make the matrix on the right of Equation (4.1) a Vandermonde matrix (essentially).

Lemma 4.15 (Composition Lemma). *Let $\mathbf{f} \subset \mathbb{F}[\mathbf{x}]$ be a finite set of polynomials of degree at most d,* $\operatorname{trdeg}_{\mathbb{F}} \mathbf{f} \leq r$, *and* $\operatorname{char}(\mathbb{F}) = 0$ *or* $> d^r$. *Let $\Phi : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[\mathbf{z}]$ be a homomorphism* *such that* $\operatorname{rank}_{\mathbb{F}(\mathbf{x})}\mathscr{J}_{\mathbf{x}}(\mathbf{f}) = \operatorname{rank}_{\mathbb{F}(\mathbf{z})}\Phi(\mathscr{J}_{\mathbf{x}}(\mathbf{f}))$. *Then, the map* $\Psi : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[\mathbf{z}, t, y_1, \ldots, y_r]$ *such that*

$$\Psi(x_i) \quad \mapsto \quad \left(\sum_{j=1}^{r} y_j t^{ij}\right) + \Phi(x_i) \quad \textit{for } 1 \leq i \leq n$$

is a homomorphism faithful to $\mathbf{f}$.

Proof. Without loss of generality, let $\operatorname{trdeg}_{\mathbb{F}}(\mathbf{f}) = r$, which (by Jacobian criterion) is the rank of $\mathscr{J}_{\mathbf{x}}(\mathbf{f})$. We intend to show that the matrix $\mathscr{J}_{\mathbf{y}}(\Psi(\mathbf{f}))$ is of rank r, which would imply (by Jacobian criterion) that $\operatorname{trdeg}_{\mathbb{F}(t,\mathbf{z})} \Psi(\mathbf{f}) = r$. Consider the projection $\mathscr{J}'$ of $\mathscr{J}_{\mathbf{y}}(\Psi(\mathbf{f}))$ obtained by setting $y_1 = \cdots = y_r = 0$.

$$\begin{aligned} \mathscr{J}' = \left[\mathscr{J}_{\mathbf{y}}(\Psi(\mathbf{f}))\right]_{\mathbf{y}=0} &= \left[\Psi(\mathscr{J}_{\mathbf{x}}(\mathbf{f})) \cdot \mathscr{J}_{\mathbf{y}}(\Psi(\mathbf{x}))\right]_{\mathbf{y}=0} \quad \text{(By chain rule)} \\ &= \Phi(\mathscr{J}_{\mathbf{x}}(\mathbf{f})) \cdot \mathscr{J}_{\mathbf{y}}(\Psi(\mathbf{x})) \end{aligned}$$

Observe that the matrix $\mathscr{J}_{\mathbf{y}}(\Psi(\mathbf{x}))$ is exactly the Vandermonde matrix that is present in Lemma 4.14. Also, $\Phi(\mathscr{J}_{\mathbf{x}}(\mathbf{f}))$ has entries in $\mathbb{F}(\mathbf{z})$, and by assumption has the same rank as $\mathscr{J}_{\mathbf{x}}(\mathbf{f})$. Hence, by Lemma 4.14,

$$\operatorname{rank}_{\mathbb{F}(t,\mathbf{z})}\mathscr{J}' = \operatorname{rank}_{\mathbb{F}(t,\mathbf{z})}\left(\Phi(\mathscr{J}_{\mathbf{x}}(\mathbf{f})) \cdot \mathscr{J}_{\mathbf{y}}(\Psi(\mathbf{x}))\right) = \operatorname{rank}_{\mathbb{F}(\mathbf{z})}\Phi(\mathscr{J}_{\mathbf{x}}(\mathbf{f})) = r.$$

And since $\mathscr{J}'$ is just a projection of $\mathscr{J}_{\mathbf{y}}(\Psi(\mathbf{f}))$, the rank of the latter must also be r. Hence, Ψ is indeed faithful. □

The different results in this chapter would be constructions of such a map Φ that preserves the rank of the Jacobian, which by the above lemma can be augments to give a faithful map.

4.3 Depth-3 circuits of bounded transcendence degree

Let C be any circuit of polynomial degree and $D = C(T_1, \cdots, T_m)$, where each T_i is of the form $\prod_{j=1}^{d} \ell_{ij}$, every ℓ_{ij} is a linear polynomial in $\mathbb{F}[x_1, \ldots, x_n]$. Denote by $\mathbf{T}$ the set $\{T_1, \ldots, T_m\}$ and by $L(T_i)$ the multiset of linear polynomials that constitute T_i. This section shall present a hitting set for such circuits where $\operatorname{trdeg}_{\mathbb{F}}\{\mathbf{T}\}$ is bounded by a constant. Note that $\Sigma\Pi\Sigma(k)$ circuit is a special case of such a circuit where the circuit $D = C(T_1, \ldots, T_k)$ where C is a $+$ gate.

Suppose $\mathrm{trdeg}_{\mathbb{F}}\{\mathbf{T}\} = k \leq r$ and $\mathbf{T}_k = \{T_1, \ldots, T_k\}$ be a transcendence basis of $\mathbf{T}$. Since $\mathscr{J}_{\mathbf{x}}(\mathbf{T}_k)$ has full rank ($\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > d^r$), without loss of generality assume that the columns corresponding to $\mathbf{x}_k = \{x_1, \cdots, x_k\}$ form a nonzero $k \times k$ minor of $\mathscr{J}_{\mathbf{x}}(\mathbf{T}_k)$. By Lemma 4.15, if we construct a $\Phi : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[\mathbf{z}]$ that keeps $J_{\mathbf{x}_k}(\mathbf{T}_k)$ nonzero then Φ can easily be extended to a homomorphism $\Psi : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[\mathbf{z}, t, y_1, \ldots, y_r]$ that is faithful to $\mathbf{T}$. And hence, by Theorem 4.12, it would follow that $\Psi(D) = 0$ if and only if $D = 0$.

4.3.1 Preserving non-zeroness of $J_{\mathbf{x}_k}(\mathbf{T}_k)$

Let us first understand the structure of $J_{\mathbf{x}_k}(\mathbf{T}_k)$. If $T_i = \prod_{j=1}^{d} \ell_{ij}$ then

$$\partial_x T_i = T_i \cdot \left(\sum_{j=1}^{d} \frac{\partial_x \ell_{ij}}{\ell_{ij}} \right)$$

By expanding, using this additive structure of $\partial_x T_i$ and the linearity of determinant with respect to rows, the determinant $J_{\mathbf{x}_k}(\mathbf{T}_k)$ takes the following form,

$$J_{\mathbf{x}_k}(\mathbf{T}_k) = \sum_{\ell_1 \in L(T_1), \ldots, \ell_k \in L(T_k)} \frac{T_1 \cdots T_k}{\ell_1 \cdots \ell_k} \cdot J_{\mathbf{x}_k}(\ell_1, \cdots, \ell_k). \tag{4.2}$$

We shall call a set of linear polynomials *independent* if the correponding homogenous linear parts (i.e. the constant-free parts) are $\mathbb{F}$-linearly independent. The term $J_{\mathbf{x}_k}(\ell_1, \cdots, \ell_k)$ ensures that the above sum is only over those $\{\ell_1, \cdots, \ell_k\}$ that are independent linear polynomials (otherwise $J_{\mathbf{x}_k}(\ell_1, \ldots, \ell_k)$ is zero). The sum in equation (4.2) has the form of a depth-3 circuit. We shall call it H_0, and we intend to construct a low-variate Φ such that $\Phi(H_0) \neq 0$. We show that this is achieved by a Φ that preserves the independence of a 'small' set of linear polynomials. In other words, we would like to construct a *certificate* of non-zeroness of H_0.

Certificate of H_0: We can assume that the terms $J_{\mathbf{x}_k}(\ell_1, \cdots, \ell_k)$ in equation (4.2) are non-zero field constants. Let $\mathscr{L}(H_0)$ be the set of all linear polynomials occurring in the denominator terms "$\ell_1 \cdots \ell_k$" of all the summands in sum (4.2). By adjusting the field constants at the numerators, we can assume that no two linear polynomials in $\mathscr{L}(H_0)$ are constant multiple of each other. This means, the depth-3 circuit H_0 has the form

$$H_0 \quad = \quad T \cdot \sum_L \frac{\alpha_L}{\ell_1 \cdots \ell_k}$$

where $T := \prod_{i=1}^{k} T_k$, each α_L is a nonzero field constant and the sum runs over sets $L = \{\ell_1, \cdots, \ell_k\}$ of k independent linear polynomials in $\mathscr{L}(H_0)$.

Let us define the *content* of a depth-3 circuit, $G = \sum_i P_i$ where P_i is a product of linear polynomials, as $\mathrm{cont}(G) := \mathrm{gcd}_i\{P_i\}$. Also, let the *simple part* of G be defined as $\mathrm{sim}(G) := G/\mathrm{cont}(G)$. Hence $\mathrm{cont}(H_0) = \mathrm{gcd}_L\{T/\ell_1 \cdots \ell_k\}$ and

$$\mathrm{sim}(H_0) = F_0 \cdot \sum_L \frac{\alpha_L}{\ell_1 \cdots \ell_k}, \text{ where } F_0 = \frac{T}{\mathrm{cont}(H_0)}, \tag{4.3}$$

Note that, since $\ell \in \mathscr{L}(H_0)$ if and only if $\ell \| F_0$, we see that F_0 is simply the product of the linear polynomials in $\mathscr{L}(H_0)$ and hence $\deg(F_0) = |\mathscr{L}(H_0)|$. Therefore, for any $\ell \in \mathscr{L}(H_0)$, the terms in $\mathrm{sim}(H_0)$ that survive modulo ℓ_1 are those with ℓ_1 in the denominator "$\ell_1 \cdots \ell_k$" of the above expression. Hence,

$$H_1 := \mathrm{sim}(H_0) \bmod \ell_1 \;=\; \frac{F_0}{\ell_1} \cdot \sum_{\ell_2, \cdots, \ell_k} \frac{\alpha_L}{\ell_2 \cdots \ell_k}$$

We can treat H_1 as a depth-3 circuit in one less variable: if $\ell_1 = c_1 x_1 + \sum_{i=2}^{n} c_i x_i$ where c_i's $\in \mathbb{F}$ and $c_1 \neq 0$, then going modulo ℓ_1 is equivalent to replacing x_1 by $-\sum_{i=2}^{n} c_i x_i / c_1$ in $\mathrm{sim}(H_0)$. Hence, H_1 becomes a depth-3 circuit in $\mathbb{F}[x_2, \ldots, x_n]$. Therefore, it makes perfect sense to talk about $\mathrm{cont}(H_1)$ and $\mathrm{sim}(H_1)$. Observe that $\ell_2, \cdots, \ell_k$ remain independent linear polynomials modulo ℓ_1, and so H_1 is a depth-3 circuit of the 'same nature' as H_0 but with one less linear polynomials in the denominators. Also, the linear polynomials in $\mathscr{L}(H_1)$ is a subset of the linear polynomials in $\mathscr{L}(H_0)$ modulo ℓ_1.

Extending the above argument, we can define the following sequence of circuits: $H_i := \mathrm{sim}(H_{i-1}) \bmod \tilde{\ell}_i$, $(1 \leq i \leq k)$ where $\tilde{\ell}_i \in \mathscr{L}(H_{i-1})$. Further, $\mathscr{L}(H_i)$ is a subset of $\mathscr{L}(H_{i-1})$ modulo $\tilde{\ell}_i$, which implies that essentially there are independent linear polynomials, say $\ell_1, \ldots, \ell_k$, in $\mathscr{L}(H_0)$ such that $\tilde{\ell}_i = \ell_i \bmod (\ell_1, \ldots, \ell_{i-1})$ and therefore $H_i = \mathrm{sim}(H_{i-1}) \bmod (\ell_1, \ldots, \ell_i)$.

Lemma 4.16 (Certifying path). *Suppose $H_0 \neq 0$. Then there exists independent linear polynomials $\{\ell_1, \cdots, \ell_k\} \subseteq \mathscr{L}(H_0)$ such that $H_i \neq 0 \bmod (\ell_1, \ldots, \ell_i)$, $\forall i \in [k]$, and H_k is a nonzero product of linear polynomials in $\mathscr{L}(H_0)$ modulo $(\ell_1, \cdots, \ell_k)$.*

Proof. The proof is by induction on k and follows the above sketch. The degree of the nonzero polynomial $\mathrm{sim}(H_0)$ is $|\mathscr{L}(H_0)|-k$. By Chinese remaindering, there exists an $\ell_1 \in \mathscr{L}(H_0)$ such that $H_1 := \mathrm{sim}(H_0) \bmod \ell_1 \neq 0$. In the base case ($k=1$), it is easy to see that H_1 is a nonzero product of linear polynomials modulo ℓ_1. For any larger k, the depth-3 polynomial H_1 has exactly the same form as H_0 but with $k-1$ independent linear polynomials in the denominators. Thus we can induct on this smaller value $k-1$, keeping in mind that $\mathscr{L}(H_i) \subset \mathscr{L}(H_0)$ modulo $(\ell_1, \ldots, \ell_i)$. □

A set $\{\ell_1, \ldots, \ell_k\}$, satisfying Lemma 4.16, shall be called a *certifying path* of H_0. Fix a certifying path $\{\ell_1, \cdots, \ell_k\}$. We shall now see that any map that preserves the linear independence of the certifying path also preserves the non-zeroness of H_0.

Theorem 4.17 (Preserving the certificate). *Let $\{\ell_1, \ldots, \ell_k\}$ be a certifying path for H_0 and $\Phi : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[z_1, \ldots, z_{k+1}]$ be a linear map such that*

$$\forall \ell \in \mathscr{L}(H_0): \quad \mathrm{rank}\,\{\ell, \ell_1, \ldots, \ell_k\} = \mathrm{rank}\,\{\Phi(\ell), \Phi(\ell_1), \ldots, \Phi(\ell_k)\}$$

Then $\Phi(H_0) \neq 0$.

Proof. We shall denote by $\mathscr{I}_j$ the ideal generated by the linear forms $\left\{\ell_1, \ldots, \ell_j\right\}$. The proof would proceed by backward induction on k: Assuming $\Phi(H_i) \neq 0 \bmod \Phi(\mathscr{I}_i)$, we shall see that $\Phi(H_{i-1}) \neq 0 \bmod \Phi(\mathscr{I}_{i-1})$ for $k \geq i \geq 1$.

The base case: By Lemma 4.16, H_k is a *nonzero* product of linear polynomials in $\mathscr{L}(H_0)$ modulo $\mathscr{I}_k$, so the definition Φ, we have that $\Phi(H_k) \neq 0 \mod \Phi(\mathscr{I}_k)$.

The inductive step: By construction, $H_{i-1} = \mathrm{cont}(H_{i-1}) \cdot \mathrm{sim}(H_{i-1}) = \mathrm{cont}(H_{i-1}) \cdot [q_i \ell_i + H_i] \mod \mathscr{I}_{i-1}$, for some polynomial q_i. By applying Φ throughout, this implies that $\Phi(H_{i-1}) = \Phi(\mathrm{cont}(H_{i-1})) \cdot [\Phi(q_i)\Phi(\ell_i) + \Phi(H_i)] \bmod \Phi(\mathscr{I}_{i-1})$.

By the definition of a certifying path, $H_{i-1} \neq 0 \bmod \mathscr{I}_{i-1}$ which then implies that $\mathrm{cont}(H_{i-1}) \neq 0 \bmod \mathscr{I}_{i-1}$. This implies that none of the linear polynomials in $\mathrm{cont}(H_{i-1})$ are present in $\mathscr{I}_{i-1}$, and Φ continues to preserve this by definition. Hence we infer that $\Phi(\mathrm{cont}(H_{i-1})) \neq 0 \bmod \Phi(\mathscr{I}_{i-1})$. Also, if $[\Phi(q_i)\Phi(\ell_i) + \Phi(H_i)] = 0 \bmod \Phi(\mathscr{I}_{i-1})$, then $\Phi(H_i) = 0 \bmod \Phi(\mathscr{I}_i)$ contradicting the induction hypothesis. Hence we have that $\Phi(H_{i-1}) \neq 0 \bmod \Phi(\mathscr{I}_{i-1})$. □

Constructing Φ: Such a map Φ is almost immediate from Lemma 4.14. For any $\ell \in \mathscr{L}(H_0)$, let A_ℓ be the $(k+1) \times n$ matrix whose rows are the coefficients of the linear forms $\{\ell, \ell_1, \ldots, \ell_k\}$. If Φ_t is defined to be the map that sends x_i to $\sum_{j=1}^{k+1} z_j t^{ij}$, the images of $\{\ell, \ell_1, \ldots, \ell_k\}$ under Φ_t can be read off from the rows of the matrix product $A'_\ell := A_\ell \cdot (t^{ij})_{i\in[n], j\in[k+1]}$. Lemma 4.14 asserts that the rank of A'_ℓ is equal the the rank of A_ℓ. As A'_ℓ is an $(k+1)\times(k+1)$ matrix, any non-zero minor of A'_ℓ is a non-zero polynomial in t of degree at most $n(k+1)^2$ and hence has at most $n(k+1)^2$ roots in $\mathbb{F}$. Thus $\text{rank}_{\mathbb{F}(t)}(A'_\ell) = \text{rank}_{\mathbb{F}}((A'_\ell)_{(t=\alpha)})$ for all but $n(k+1)^2$ many $\alpha \in \mathbb{F}$. By running over all $\ell \in \mathscr{L}(H_0)$, for all but $n(k+1)^2 \cdot |\mathscr{L}(H_0)|$ values of $\alpha \in \mathbb{F}$, we infer that Φ_α satisfies

$$\text{rank}\{\ell, \ell_1, \ldots, \ell_k\} \quad = \quad \text{rank}\{\Phi(\ell), \Phi(\ell_1), \ldots, \Phi(\ell_k)\} \quad \forall \ell \in \mathscr{L}(H_0)$$

We shall refer to one such Φ_α by simply Φ. We can now finish the proof by appealing to the Lemma 4.15.

Theorem 4.2 (restated). Let C be a poly-degree circuit of size s and each of $T_1, \ldots, T_m$ be a product of d linear polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ such that $\text{trdeg}_{\mathbb{F}}\{T_1, \ldots, T_m\} \le r$. A hitting-set for $C(T_1, \ldots, T_m)$ can be constructed in time polynomial in n and $(sd)^r$, assuming $\mathit{char}(\mathbb{F}) = 0$ or larger than d^r.

Proof. As $r \ge k$, we can assume that the map Φ is map from $\mathbb{F}[\mathbf{x}]$ to $\mathbb{F}[z_1, \ldots, z_{r+1}]$ as above. Therefore, by Lemma 4.15, the map $\Psi : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[y_1, \ldots, y_r, t, z_1, \ldots, z_{r+1}]$ defined as

$$\Psi : x_i \quad \mapsto \quad \Phi(x_i) + \sum_{j=1}^{k} y_j t^{ij}$$

ensures that $D = 0$ if and only if $\Psi(D) = 0$. Since C is a poly-degree circuit of size s, $\Psi(C(T_1, \ldots, T_m))$ is a polynomial of degree $nrds^{O(1)}$ in the variables $\mathbf{y}, \mathbf{z}$ and t. Using Corollary 2.2, we can construct a hitting-set for $\Psi(D)$ in time polynomial in $n(sd)^r$. Since construction of Ψ takes time $\text{poly}(ndr)$, the total time taken is $\text{poly}(n, (sd)^r)$. □

4.4 Constant-depth constant-occur formulas

This section shall focus on constant-depth constant-occur formulas. Recall that an occur-k formula (Definition 4.3) is a tree comprising of $+$ and $\hat{\times}$ gates, with leaves consisting of sparse plynomials such that each variable occurs in at most k of the sparse polynomials.

The top fan-in of an occur-k formula could be unbounded. However, for the purpose of identity testing, we can assume that the top fan-in is bounded as well by a simple trick.

Observation 4.18 (Top fan-in reduction). *Let C be a non-constant polynomial. Then, there is an i such that $\tilde{C} := C(x_1,\cdots,x_{i-1},x_i+1,x_{i+1},\cdots,x_n)-C(x_1,\cdots,x_n)\neq 0$, assuming* $\operatorname{char}(\mathbb{F}) > \deg(C)$.

Proof. If C depends non-trivially on x_i and $d = \deg_{x_i}(C)$, it is easy to see that the coefficient of x_i^{d-1} in $\tilde{C}$ is non-zero. □

If C has a $+$ gate on top then $C(\mathbf{x}) = \sum_{i=1}^m T_i$, where T_i's are computed by $\hat{\times}$ gates at the next level. Since x_i occurs in at most k of the T_i's, $\tilde{C}$ has top fanin at most $2k$. If C has a $\hat{\times}$ gate on top then $\tilde{C}$ has a $+$ gate on top with fanin 2 and $\operatorname{depth}(\tilde{C}) = D+1$. Therefore, $\tilde{C}$ belongs to the class of depth-$(D+1)$ occur-$2k$ formulas of size at most (s^2+s), and a $+$ gate on top with fanin bounded by $2k$. Suppose $\tilde{\mathscr{H}}$ is a hitting-set for $\tilde{C}$, we can form a new set $\mathscr{H} \supset \tilde{\mathscr{H}}$ by including points $(\alpha_1+1,\alpha_2,\ldots,\alpha_n),(\alpha_1,\alpha_2+1,\ldots,\alpha_n),\ldots,(\alpha_1,\ldots,\alpha_{n-1},\alpha_n+1)$ in $\mathscr{H}$ for every $(\alpha_1,\alpha_2,\ldots,\alpha_n)\in\tilde{\mathscr{H}}$. It is easy to see that $\mathscr{H}$ is a hitting-set for C and $\operatorname{size}(\mathscr{H}) = n\cdot\operatorname{size}(\tilde{\mathscr{H}})$. Therefore, it is sufficient if we construct $\tilde{\mathscr{H}}$. By reusing symbols, assume that C is a depth-D occur-k formula of size s with a $+$ gate on top having top fanin at most k.

Let $C(\mathbf{x}) = \sum_{i=1}^k T_i$. The goal is to construct a Ψ that is faithful to $\mathbf{T} = \{T_1,\ldots,T_k\}$. Let $\mathbf{T}_r = \{T_1,\ldots,T_r\}$ be a transcendence basis of $\mathbf{T}$. Since $\mathscr{J}_{\mathbf{x}}(\mathbf{T}_r)$ has full rank (if $\operatorname{char}(\mathbb{F}) = 0$ or $> s^{Dr}$), assume that the columns corresponding to $\mathbf{x}_r = \{x_1,\ldots,x_r\}$ form a nonzero minor of $\mathscr{J}_{\mathbf{x}}(\mathbf{T}_r)$. By Lemma 4.15, it suffices to construct a Φ that keeps $J_{\mathbf{x}_r}(\mathbf{T}_r)\neq 0$.

Proof idea - Suppose $C = T_1+\cdots+T_k$ where each $T_i = \prod P_{ij}^{e_j}$. Since C is an occur-k formula, it follows that the variables $x_1,\ldots,x_r$ occurs in at most kr of the $P_{i,j}$'s, say $P_{i,1},\ldots P_{i,kr}$. Hence,

$$\partial_j T_i \quad = \quad \left(\prod_{\ell=kr+1}^{d} P_{i,\ell}^{e_{i,\ell}}\right)\cdot\left(\partial_j \prod_{\ell=1}^{kr} P_{i,\ell}^{e_{i,\ell}}\right) \qquad \text{for every } 1\leq i,j\leq r$$

Therefore, we have that

$$J_{\mathbf{x}_r}(T_1,\ldots,T_r) \quad = \quad \left(\prod_{i=1}^{r}\prod_{\ell=kr+1}^{d} P_{i,\ell}^{e_{i,\ell}}\right) J_{\mathbf{x}_r}(T'_1,\ldots,T'_r)$$

where $T'_i = \prod_{\ell=1}^{kr} P_{i,\ell}^{e_{i,\ell}}$ for $1 \leq i \leq r$. We wish to construct a map Φ to preserve the non-zeroness of the above expression. The Jacobian term on the RHS, notice that $J_{\mathbf{x}_r}(T'_1,\ldots,T'_r)$ is a polynomial in $P_{i,\ell}$ and $\partial_j P_{i,\ell}$, for $1 \leq i,j \leq r$ and $1 \leq \ell \leq kr$ (and the exponents $e_{i,\ell}$'s are rather irrelevant besides contributing to the degree). So, if Φ is faithful to the set $\mathscr{P} := \{P_{i,\ell}, \partial_j P_{i,\ell} : 1 \leq i,j \leq r, 1 \leq \ell \leq kr\}$ and the singleton sets $\{P_{i,\ell}\}$ for $1 \leq i \leq r$, $kr+1 \leq \ell \leq d$, then $\Phi(J_{\mathbf{x}_r}(\mathbf{T}_r)) \neq 0$.

Observe that the polynomials in $\mathscr{P}$ and the singleton sets are (zeroth and first order) derivatives of the gates at lower levels, and further these sets involve (the derivatives of) *disjoint* groups of polynomials. This disjointness ensures that the number of such sets is at most s.

Thus, we have reduced the problem of constructing a faithful map Φ for $\mathbf{T}$ to the problem of constructing a map Φ' that is faithful to at most s many sets each containing derivatives of gates at lower levels. By applying the argument recursively, we eventually reach the level of the sparse polynomials (the leaf nodes) where a faithful map can be constructed using Lemma 2.3.

As an illustrative example, we shall restrict to depth-4 occur-k formulas first. The general case proceeds exactly along these lines.

4.4.1 Restriction to the case of depth-4

Theorem 4.5 (restated). A hitting-set for a depth-4 occur-k formula of size s can be constructed in time polynomial in s^{k^2} (assuming *char*$(\mathbb{F}) = 0$ or $> s^{2k}$).

Proof. Let $C = \sum_{i=1}^{k} T_i$ be a depth-4 occur-k formula, where $T_i = \prod_{j=1}^{d} P_{ij}^{e_{ij}}$ and P_{ij}'s are sparse polynomials. Observation 4.18 justifies the assumption that top fanin is k. Once again, assuming $\mathbf{T}_r$ to be a transcendence basis of $\mathbf{T}$, we need to design a Φ such that $\Phi(J_{\mathbf{x}_r}(\mathbf{T}_r)) \neq 0$.

Let us count the number of P_{ij}'s that depend on the variables $\mathbf{x}_r$, the remaining $P_{ij}^{e_{ij}}$'s can be taken out common from every row of $\mathscr{J}_{\mathbf{x}_r}(\mathbf{T}_r)$ while computing its determinant. Let $c_{i\ell}$ be the number of P_{ij}'s present in T_i that depend on x_ℓ, and $c_i := \sum_\ell c_{i\ell}$

is the number of P_{ij}'s in T_i that depend on $\mathbf{x}_r$. The total number of sparse polynomials depending on $\mathbf{x}_r$ is therefore $\sum_{1\leq i,\ell\leq r} c_{i\ell}$. Since we have an occur-k formula, $\sum_i c_{i\ell} \leq k$ and hence $\sum_{i,\ell} c_{i\ell} \leq rk \leq k^2$.

For an $\mathbf{x}_r$-dependent P_{ij}, we can also take $P_{ij}^{e_{ij}-1}$ common from the i-th row of $\mathcal{J}_{\mathbf{x}_r}(\mathbf{T}_r)$. The sparsity of every entry of the i-th row of the residual matrix M is bounded by $c_i s^{c_i}$, where s is the size of C. Thus, $\det(M)$ has sparsity at most $r!\cdot\prod_{i=1}^{r} c_i s^{c_i} = s^{O(k^2)}$, which implies that $J_{\mathbf{x}_r}(\mathbf{T}_r)$ is a product of at most $s+1$ powers of sparse polynomials, each of whose sparsity is bounded by $s^{O(k^2)}$ and degree bounded by sk. Hence, by Lemma 2.3, we have our desired Φ that preserves the non-zeroness of $J_{\mathbf{x}_r}(\mathbf{T}_r)$. Lemma 4.15 and Corollary 2.2 gives the hitting set. □

4.4.2 Generalizing to larger depth

The ideas of the depth-4 case carry over to higher depth as well. We shall need some notation to proceed further. For any *multiset* of variables S, let $\Delta_S f$ denote the partial derivative of f with respect to the variables in S (including repetitions, as S is a multiset). Let $\mathrm{var}(S)$ denote the set of distinct variables in S.

One of the crucial properties used in the earlier proof was that each row of the Jacobian minor shared a large gcd which could be pulled out. The following lemma formalizes that notion for larger depth.

Lemma 4.19 (Content removal). *Let G be any gate in an occur-k formula and $S_1, \cdots, S_w$ be multisets of variables. Then there exists another occur-k formula G' for which, the vector of polynomials $\left(\Delta_{S_1} G, \cdots, \Delta_{S_w} G\right) = V_G \cdot \left(\Delta_{S_1} G', \cdots, \Delta_{S_w} G'\right)$ such that*

1. *If G is a $+$ gate then G' is also a $+$ gate whose children consist of at most $k\cdot\left|\cup_{i=1}^{w} \mathrm{var}(S_i)\right|$ of the children of G, and $V_G = 1$.*
2. *If G is a $\hat{\times}$ gate, then G' is also a $\hat{\times}$ gate whose children consist of at most $k\cdot\left|\cup_{i=1}^{w} \mathrm{var}(S_i)\right|$ of the children of G, and $V_G = G/G'$.*

Further, the gates constituting G' and V_G are disjoint.

Proof. 1. Suppose $G = H_1 + \cdots + H_m$. Then at most $k\cdot|\cup \mathrm{var}(S_i)|$ of its children depend on the variables present in $\cup\mathrm{var}(S_i)$; let G' be the sum of these children. Then, $\Delta_{S_i} G = \Delta_{S_i} G'$ as the other gates are independent of the variables in $\cup S_i$.

2. Suppose $G = H_1^{e_1} \cdots H_m^{e_m}$. Since G is a gate in an occur-k formula, at most $k \cdot |\cup \mathrm{var}(S_i)|$ of the H_i's depend on the variables in $\cup S_i$; call these $H_1, \cdots, H_t$. Let $G' := H_1^{e_1} \cdots H_t^{e_t}$ and $V_G := G/G'$. Then, $\Delta_{S_i} G = V_G \cdot \Delta_{S_i} G'$ as claimed. □

We shall say that a map is faithful to a collection of sets if it is faithful to every set in the collection. Also, we shall say that a gate is at level ℓ if its distance to the root is ℓ. Going by the 'proof idea', suppose at the ℓ-th level of the recursion we want to construct a Ψ_ℓ that is faithful to a collection of (at most) s sets of polynomials, each set containing at most r_ℓ partial derivatives (of order up to c_ℓ) of the gates at level ℓ. Moreover, the sets involve derivatives of disjoint groups of gates. By Lemma 4.15, it suffices to find a map Φ_ℓ that preserves the non-zeroness of a jacobian minor of each of the sets of polynomials. To begin with: $\ell = 2$ and we wish to construct a Φ_2 that preserves the non-zeroness of a jacobian minor of just one set $\mathbf{T} := \{T_1, \ldots, T_k\}$, so $r_2 \leq k$ and $c_2 = 0$. The next lemma captures the evolution of the recursion.

Lemma 4.20 (Evolution via factoring). *Let $\mathcal{U}$ be a set of r_ℓ derivatives (of orders up to c_ℓ) of a set of gates $\mathcal{G}_\ell$ at level ℓ, and $\mathcal{U}'$ be a transcendence basis of $\mathcal{U}$. Any $|\mathcal{U}'| \times |\mathcal{U}'|$ minor of $\mathcal{J}_{\mathbf{x}}(\mathcal{U}')$ is of the form $\prod_i V_i^{e_i}$, where V_i's are polynomials in at most $r_{\ell+1} := (c_\ell + 1) \cdot 2^{c_\ell + 1} k \cdot {r_\ell}^2$ many derivatives (of order up to $c_{\ell+1} := c_\ell + 1$) of disjoint groups of children of $\mathcal{G}_\ell$.*

Proof. Let $G \in \mathcal{G}_\ell$ be a gate at level ℓ and $\{U_1, \ldots, U_{t_G}\} \subset \mathcal{U}'$ be the set of all the derivatives of G present in $\mathcal{U}'$. Fix any $|\mathcal{U}'| \times |\mathcal{U}'|$ sub-matrix M of $\mathcal{J}_{\mathbf{x}}(\mathcal{U}')$. Consider the t_G rows of M that contain the derivatives of $U_1, \ldots, U_{t_G}$. These rows together contain a total of $w := t_G \cdot |\mathcal{U}'|$ elements that are derivatives of G of order up to $(c_\ell + 1)$. Let us view all the elements of these t_G rows as a single vector $\left(\Delta_{S_1} G, \cdots, \Delta_{S_w} G\right)$ and apply Lemma 4.19 to express it as $V_G \cdot \left(\Delta_{S_1} G', \cdots, \Delta_{S_w} G'\right)$. To bound $\left|\cup_{i=1}^{w} S_i\right|$, observe that each of the t_G rows could potentially be derivatives of G with respect to disjoint sets of c_ℓ variables, and the different columns of the Jacobian minor additionally takes derivatives of $|\mathcal{U}'|$ variables. Hence, $\left|\cup_{i=1}^{w} \mathrm{var}(S_i)\right| \leq t_G \cdot c_\ell + |\mathcal{U}'| \leq t_G \cdot c_\ell + r_\ell$. So, in $\det(M)$ we can take V_G common from each of these t_G rows such that the elements present inside the determinant are of the form $\Delta_{S_i} G'$, where G' has at most $k(t_G c_\ell + r_\ell)$ children.

Since $|S_i| \leq c_\ell + 1$, at most $k(c_\ell + 1)$ children of G' depend on $\mathrm{var}(S_i)$. If G' is a $+$ gate, then $\Delta_{S_i} G'$ is the sum of the derivatives of at most $k(c_\ell + 1)$ of its children (that depend on $\mathrm{var}(S_i)$). If G' is a $\hat{\times}$ gate computing $H_1^{e_1}\cdots H_t^{e_t}$ (where $t \leq k(e_G c_\ell + r_\ell)$), then $\Delta_{S_i} G'$ is a polynomial combination of the H_i's and $\left\{\Delta_T H_j\right\}_{\emptyset \neq T \subseteq S_i}$ for each H_j depending on $\mathrm{var}(S_i)$. Hence in either case, $\Delta_{S_i} G'$ is a polynomial in the children of G' and at most $(2^{c_\ell+1}-1)\cdot k(c_\ell+1)$ of their derivatives (of order between one and $(c_\ell+1)$).

Summing over all the w elements $\Delta_{S_i} G'$, the elements of the t_G rows of M are polynomials in at most $k(t_G c_\ell + r_\ell) + w\cdot(2^{c_\ell+1}-1)k(c_\ell+1) = k(t_G c_\ell + r_\ell) + t_G \cdot |\mathscr{U}'|\cdot(2^{c_\ell+1}-1)k(c_\ell+1)$ derivatives of the children of G'. Going over all $G \in \mathscr{G}_\ell$, $\det(M)$ can be expressed as a product $\prod_{G\in\mathscr{G}_\ell} V_G^{t_G}$ and a polynomial V in at most $k(r_\ell c_\ell + r_\ell^2) + r_\ell^2 \cdot (2^{c_\ell+1}-1)k(c_\ell+1) \leq r_{\ell+1}$ derivatives (of order up to $c_\ell + 1$) of a group of gates in level $\ell+1$. Further, the groups of gates whose derivatives constitute the V_G's and V are mutually disjoint (by Lemma 4.19). □

To begin with, let $C_2 = \{\{T_1,\ldots,T_k\}\}$ and we would like to construct a map Φ_2 that preserves the non-zeroness of a non-zero Jacobian minor of the T_i's. Applying Lemma 4.20 (with $\mathscr{U} = \{T_1,\ldots,T_m\}$), any Jacobian minor of a set $\mathscr{U}$ can be written as a product of V_i's that are polynomials in a set of the derivatives at lower levels. Let us denote this set of derivatives by $\mathrm{Elem}(V_i)$, and define $\mathscr{C}_{\ell+1}$ as the collection of sets $\mathrm{Elem}(V_i)$ as $\mathscr{U}$ varies over all the sets in the collection $\mathscr{C}_\ell$. It follows from the lemma that the groups of gates whose derivatives form the different $\mathrm{Elem}(V_i)$'s are disjoint and therefore $\left|\mathscr{C}_{\ell+1}\right| \leq s$. Using Lemma 4.15 & 4.20, we can lift a map $\Phi_{\ell+1}$ to construct Φ_ℓ via the following corollary.

Corollary 4.21. *If $\Phi_{\ell+1}$ is faithful to $\mathscr{C}_{\ell+1}$ then $\Phi_\ell : x_i \mapsto \left(\sum_{j=1}^{r_\ell} y_{j,\ell}\cdot(t_\ell)^{ij}\right) + \Phi_{\ell+1}(x_i)$ is faithful to $\mathscr{C}_\ell$, where $\left\{y_{1,\ell},\cdots,y_{r_\ell,\ell},t_\ell\right\}$ is a fresh set of variables.*

Finally, we can obtain our hitting set by repeating the above process until $\mathscr{C}_\ell$ consists only of sparse polynomials.

Theorem 4.4 (restated). A hitting-set for a depth-D occur-k formula of size s can be constructed in time polynomial in s^R, where $R = (2k)^{4D\cdot 2^D}$ (assuming $\mathit{char}(\mathbb{F}) = 0$ or $> s^R$).

Proof. Unfolding the above recursion, we eventually reach the level of the sparse polynomials at depth $D-2$ and are required to construct a map Ψ_{D-2} that is faithful to a collection $\mathscr{C}_{D-2}$ of at most s sets of derivatives of sparse polynomials, each set containing at most r_{D-2} elements. Using the relation between $r_{\ell+1}$ and r_ℓ from Lemma 4.20, it is easy to bound $\sum r_\ell$ by $R = (2k)^{4D \cdot 2^D}$.

Let $\mathscr{U} \in \mathscr{C}_{D-2}$ with transcendence basis $\mathscr{U}'$. Any $|\mathscr{U}'| \times |\mathscr{U}'|$ minor of $\mathscr{J}_{\mathbf{x}}(\mathscr{U}')$ is a sparse polynomial with sparsity bounded by s^R and degree bounded by sR. Using Corollary 2.3, the nonzeroness of this determinant is maintained by one of the maps $\Delta_p : x_i \mapsto u^{(sR+1)^i \bmod p}$ as p varies from 1 to a fixed $\mathsf{poly}(s^R)$. Since $\left|\mathscr{C}_{D-2}\right| \leq s$, one of the maps Δ_p preserves the rank of the Jacobian of all $\mathscr{U}$ in $\mathscr{C}_{D-2}$ — fix such a Δ_p. Lemma 4.15 implies that $\Phi_{D-2} : x_i \mapsto \sum_{j=1}^{r_{D-2}} y_{j,D-2} t_{D-2}^{ij} + \Delta_p(x_i)$ is faithful to $\mathscr{C}_{D-2}$. Using Corollary 4.21, we can lift this to a map Φ_2 defined as

$$\Phi_2 \quad : \quad x_i \quad \mapsto \quad \sum_{\ell=2}^{D-2} \left(\sum_{j=1}^{r_\ell} y_{j,\ell} t_\ell^{ij} \right) \quad + \quad \Delta_p(x_i)$$

that is faithful to $\{T_1, \ldots, T_k\}$. The map Φ_2 reduces the number of variables to $O(R)$ and hence an application of Corollary 2.2 leads to a hitting-set generator with running time $\mathrm{poly}(s^R)$. For the Jacobian criterion to work we need $\mathrm{char}(\mathbb{F}) = 0$ or $> s^R$. □

4.5 Related lower bounds

As mentioned earlier, polynomial identity tests are intimately related to lower bounds. We shall now see a few lower bounds for the determinant/permanent for the models studied in this chapter. The following two lemmas are at the heart of the approach to proving lower bounds. Let $\mathbf{x} := \{x_{ij} : 1 \leq i, j \leq n\}$ and $\mathbf{T} := \{T_1, \ldots, T_m\}$, where T_i's are polynomials in $\mathbb{F}[\mathbf{x}]$. Though all the following results work for Perm_n as well, we shall state and prove them for the Det_n for simplicity.

Lemma 4.22. *Suppose* $\mathsf{Det}_n = C(T_1, \ldots, T_m)$*, where* C *is any circuit and suppose* $\mathbf{T}_r = \{T_1, \ldots, T_r\}$ *be a transcendence basis of* $\mathbf{T}_r$ *with* $r < n$*. Then, there exist a set of* $r+1$ *variables* $\mathbf{x}_{r+1} \subset \mathbf{x}$ *and an equation* $\sum_{i=1}^{r+1} c_i f_i \cdot M_i = 0$ *such that* M_i*'s are distinct first order principal minors of* M*,* f_i*'s are distinct* $r \times r$ *minors of* $\mathscr{J}_{\mathbf{x}_{r+1}}(\mathbf{T}_r)$*, not all* f_i*'s are zero, and* $c_i \in \mathbb{F}^*$.

Lemma 4.23. *If* $M_1, \cdots, M_t$ *are distinct first order principal minors of* M *and* $\sum_{i=1}^{t} f_i \cdot M_i = 0$ *(not all* f_i*'s are zero) then the total sparsity of the* f_i*'s is at least* $2^{n/2-t}$.

We shall defer the proofs of these technical lemmas to the end of the section and proceed to see how they imply the lower bounds.

4.5.1 Lower bound on depth-4 occur-k formulas

Theorem 4.6 (restated). Any depth-4 occur-k formula that computes Det_n must have size $s = 2^{\Omega\left(n/k^2\right)}$ over any field of characteristic zero.

Proof. Let C be a depth-4 occur-k formula of size s that computes Det_n. Since Det_n is irreducible we can assume a top $+$ gate in C. Then $\tilde{C} := C(x_{11}+1, x_{12}, \ldots, x_{nn}) - C(\mathbf{x})$ is a depth-4 occur-$2k$ formula of size at most $2s^2$ and top fanin bounded by $2k$ (similar argument as at the beginning of Section 4.4). Moreover, $\tilde{C}$ computes the minor of M with respect to x_{11} which is essentially Det_{n-1}. By reusing symbols, assume that C is a depth-4 occur-k formula with top fanin bounded by k, and C computes Det_n.

Let $C = \sum_{i=1}^{k} T_i = \mathrm{Det}_n$, where $T_i = \prod_{j=1}^{d} P_{ij}^{e_{ij}}$, P_{ij}'s are sparse polynomials. Let $\mathbf{T}_r$ be a transcendence basis of $\mathbf{T} = \{T_1, \ldots, T_k\}$. By Lemma 4.22, we have an equation $\sum_{i=1}^{r+1} c_i f_i \cdot M_i = 0$ such that f_i's are distinct $r \times r$ minors of $\mathcal{J}_{\mathbf{x}_{r+1}}(\mathbf{T}_r)$ for some set of $r+1$ variables $\mathbf{x}_{r+1}$. Arguing in the same way as in the proof of Theorem 4.5 (in Section 4.4.1), we can throw away certain common terms from the minors f_i's and get another equation $\sum_{i=1}^{r+1} g_i M_i = 0$, where the sparsity of each g_i is $s^{O(k^2)}$. If we apply Lemma 4.23 on this equation, we get our desired result. □

It is also natural to ask for similar lower bounds for occur-k depth-D formulas, and such a property can be shown under a conjecture about determinant of determinants. The conjecture is a little cumbersome to explain and omitted here. The interested reader can see the formulation in the published version of the contents [ASSS12].

4.5.2 Lower bound on circuits generated by $\Sigma\Pi$ polynomials

Theorem 4.7 (restated). Let C be any circuit. Let $f_1, \ldots, f_m$ be sparse polynomials (of any degree) with sparsity bounded by s and their trdeg bounded by r. If $C(f_1, \ldots, f_m)$ computes Det_n, then $s = 2^{\Omega(n/r)}$ over any field of characteristic zero.

Proof. In Lemma 4.22, take the T_i's to be sparse polynomials with sparsity bounded by s. Then, in the equation $\sum_{i=1}^{r+1} c_i f_i \cdot M_i = 0$, each f_i has sparsity $s^{O(r)}$. Finally, apply Lemma 4.23 to obtain the desired lower bound. □

4.5.3 Lower bound on circuits generated by $\Pi\Sigma$ polynomials

Theorem 4.8 (restated). Let C be any circuit and $T_1, \ldots, T_m$ be products of linear polynomials. If $C(T_1, \ldots, T_m)$ computes Det_n then $\mathrm{trdeg}_{\mathbb{F}}\{T_1, \ldots, T_m\} = \Omega(n)$ over any field of characteristic zero.

Proof of Theorem 4.8. Let $\mathbf{T} = \{T_1, \cdots, T_m\}$ be products of linear polynomials such that $C(T_1, \cdots, T_m) = \mathsf{Det}_n$ with $\mathbf{T}_k = \{T_1, \cdots, T_k\}$ being a transcendence basis. By Lemma 4.22, we get $\sum_{i=1}^{k+1} c_i f_i M_i = 0$ where the f_i's are $k \times k$ minors of $\mathcal{J}_{\mathbf{x}_{k+1}}(\mathbf{T}_k)$ and wlog $f_1 \neq 0$. Like in Section 4.3, we can rewrite this equation in the form $H_0 := T \cdot \sum_L \alpha_L(\mathbf{M}_{k+1})/\ell_1 \cdots \ell_k = 0$ where $\alpha_L(\mathbf{M}_{k+1}) := \sum_{i=1}^{k+1} \alpha_{L,i} M_i$ is an $\mathbb{F}$-linear combination of $\mathbf{M}_{k+1} := \{M_1, \cdots, M_{k+1}\}$. Observe that H_0 is a sum of products of linear polynomials, with 'coefficients' being $\mathbb{F}$-linear combinations of $\mathbf{M}_{k+1}$. And since $f_1 \neq 0$, the 'coefficient' of M_1 in H_0 is a nonzero depth-3 circuit.

The idea is to apply a similar treatment as in Section 4.3 to evolve H_0. The invariant that shall be maintained is that the coefficient of M_1 (modulo some linear polynomials), which is a depth-3 circuit, would stay nonzero. This would finally yield a non-trivial linear combination $\alpha_L(\mathbf{M}_{k+1}) = 0 \bmod \ell_k$ whence we can apply the following lemma.

Lemma 4.24. *If $M_1, \cdots, M_t$ are distinct first order principal minors of M and $\sum_{i=1}^t \alpha_i M_i = 0 \bmod \ell_k$ (not all $\alpha_i = 0$) for independent linear polynomials ℓ_k, then $t + k \geq n$.*

We shall refer the proof of this lemma to the end of the section as well. Formally, define the *content* of a circuit $H = T \sum_L \alpha_L(\mathbf{M}_{k+1})/\ell_1 \cdots \ell_k$ as $\mathrm{cont}(H) := \gcd_L \left\{ \frac{T}{\ell_1 \cdots \ell_k} \right\}$, and define $\mathrm{sim}(H) := H/\mathrm{cont}(H)$. Let $\mathrm{sim}(H_0)$ have the form $F_0 \sum_L \alpha_L(\mathbf{M}_{k+1})/\ell_1 \cdots \ell_k$. The coefficient of M_1 in the above expression is a nonzero depth-3 circuit, whose degree is $|\mathcal{L}(H_0)| - k$. Therefore by Chinese remaindering, $\exists \ell_1 \in \mathcal{L}(H_0)$ such that this coefficient is nonzero modulo ℓ_1. Hence, we can define $H_1 := \mathrm{sim}(H_0) \bmod \ell_1$ which has the form $H_1 = F_0/\ell_1 \cdot \sum_{L \ni \ell_1} \alpha_L(\mathbf{M}_{k+1})/\ell_2 \cdots \ell_k = 0 \bmod \ell_1$. And like in Section 4.3, the above equation can be rewritten by replacing a variable occuring in ℓ_1 by a suitable linear combination of the rest. Thus, we may write $H_1 = F_1 \sum_L \alpha_L(\mathbf{M}_{k+1} \bmod \ell_1)/\ell_2 \cdots \ell_k = 0$, and maintaining the invariant that the coefficient of $M_1 \bmod \ell_1$ is nonzero. Repeating this argument, we eventually obtain $H_k := F_k \cdot \alpha_L(\mathbf{M}_{k+1} \bmod \ell_k) = 0$ while the coefficient of $M_1 \bmod \ell_k$ is nonzero. This implies that $\alpha_L(\mathbf{M}_{k+1}) = 0 \bmod \ell_k$

is a non-trivial equation. And Lemma 4.24 asserts that this is not possible unless $2k+1 \geq n$ or $k \geq (n-1)/2$. □

4.5.4 Proofs of the technial lemmas

All the following lemmas hinge on this basic fact about the determinant — if less than c entries of a symbolic $n \times n$ matrix is replaced by arbitrary polynomials, the determinant remains non-zero.

Fact 4.25. *Let $M = (x_{ij})_{1 \leq i,j \leq n}$ and M' be the matrix obtained by setting $c < n$ entries of M to arbitrary polynomials in $\mathbb{F}[\mathbf{x}]$. Then we have* $\det(M') \neq 0$.

Proof. We shall say an entry of M' is *corrupted* if it is one of the c entries of M that has been replaced by a polynomial. We shall prove this by carefully rearranging the rows and columns so that all the corrupted entries are above the diagonal. Then, since all entries below the diagonal are free, we may set all of them to zero and the determinant reduces to a single nonzero monomial.

Since less than n entries of M' have been altered, there exists a column that is free of any corrupted entries. By relabelling the columns if necessary, let the first column be free of any corrupted entry. By relabelling the rows if necessary, we can assume that the first row contains a corruption. This ensures that the first column is free of any corrupted entry, and the $(n-1) \times (n-1)$ matrix defined by rows and columns, 2 through n, contain less than $c-1$ corruptions. By induction, the $c-1$ corruptions may be moved above the diagonal by suitable row/column relabelling. And since the first column is untouched during the process, we now have all c corruptions above the diagonal. Now setting all entries below the diagonal to zeroes reduces the determinant to a single nonzero monomial. □

Lemma 4.22 (restated). Suppose $\mathsf{Det}_n = C(T_1, \ldots, T_m)$, where C is any circuit and let $\mathbf{T}_r = \{T_1, \ldots, T_r\}$ be a transcendence basis of $\mathbf{T}$ with $r < n$. Then, there exist a set of $r+1$ variables $\mathbf{x}_{r+1} \subset \mathbf{x}$ and an equation $\sum_{i=1}^{r+1} c_i f_i \cdot M_i = 0$ such that M_i's are distinct first order principal minors of M, f_i's are distinct $r \times r$ minors of $\mathcal{J}_{\mathbf{x}_{r+1}}(\mathbf{T}_r)$, not all f_i's are zero, and $c_i \in \mathbb{F}^*$.

Proof. Every column of a Jacobian $\mathscr{J}_{\mathbf{x}}(\cdot)$ consists of entries that are differentiated with respect to a specific variable x; we shall say that the column is *indexed* by x. Let $\mathbf{T}_r = \{T_1, \cdots, T_r\}$ be a transcendence basis of $\mathbf{T}$. Amongst the nonzero $r \times r$ minors of $\mathscr{J}_{\mathbf{x}}(\mathbf{T}_r)$, let us pick one (and call the matrix associated with the minor as N) that maximizes the number of diagonal variables $\{x_{ii} : 1 \le i \le n\}$ indexing the columns of N. Let S denote the set of variables indexing the columns of N.

Since $r < n$, there exists a diagonal variable $x_{jj} \notin S$. Consider the $(r+1) \times (r+1)$ minor of $\mathscr{J}_{\mathbf{x}}(\{\mathsf{Det}_n\} \cup \mathbf{T}_r)$ corresponding to the columns indexed by $S' := S \cup \left\{x_{jj}\right\}$ - call the associated $(r+1) \times (r+1)$ matrix $\tilde{N}$. Since, $\mathsf{Det}_n = C(\mathbf{T})$, the polynomials Det_n and $T_1, \ldots, T_r$ are algebraically dependent and hence $\det(\tilde{N}) = 0$. Expanding $\det(\tilde{N})$ along the first row of $\tilde{N}$, which contains signed first order minors (cofactors) of M, we have an equation $\sum_{i=1}^{r+1} c_i f_i M_i = 0$, where M_i's are distinct minors of M, f_i's are distinct $r \times r$ minors of $\mathscr{J}_{S'}(\mathbf{T}_r)$, and $c_i \in \mathbb{F}^*$. If M_i is the principal minor of M with respect to the variable x_{jj} then $f_i = \det(N) \neq 0$ (by construction).

It suffices to show that if M_i is a non-principal minor of M then $f_i = 0$. Consider any non-principal minor M_i in the above sum, say it is the minor of M with respect to $x_{k\ell}$. The corresponding f_i is precisely the $r \times r$ minor of $\mathscr{J}_{S'}(\mathbf{T}_r)$ with respect to the columns $S' \setminus \{x_{k\ell}\} = \left(S \setminus \{x_{k\ell}\}\right) \cup \left\{x_{jj}\right\}$. Hence, by the maximality assumption on the number of diagonal elements of M in S, $f_i = 0$. □

Lemma 4.23 (restated). If $M_1, \cdots, M_t$ are distinct first order principal minors of M and $\sum_{i=1}^{t} f_i \cdot M_i = 0$ (not all f_i's are zero) then the total sparsity of the f_i's is at least $2^{n/2-t}$.

Proof. The proof is by contradiction. The idea is to start with the equation $\sum_{i=1}^{t} f_i M_i = 0$ and apply two steps — *sparsity reduction* and *fanin reduction* — alternatively, till we arrive at a contradiction in the form of an equation $f_j \cdot M_j = 0$, where neither f_j nor M_j is zero if the total sparsity of the f_i's is less than $2^{n/2-t}$.

With an equation of the form $\sum_{i=1}^{\tau} g_i N_i = 0$, we associate four parameters τ, s, η and c. These parameters are as follows: τ is called the *fanin* of the equation, s is the total sparsity of the g_i's (we always assume that not all the g_i's are zero), every N_i is a distinct first order principal minor of a symbolic $\eta \times \eta$ matrix $N = (x_{ij})$, and c is the maximum number of entries of N that are set as constants. To begin with, $g_i = f_i$ and $N_i = M_i$ for all $1 \le i \le t$, so $\tau = t$, $\eta = n$, $N = M$ and $c = 0$.

In the 'sparsity reduction' step, we start with an equation $\sum_{i=1}^{\tau} g_i N_i = 0$, with parameters τ, s, η, c and arrive at an equation $\sum_{i=1}^{\tau'} g_i' N_i' = 0$ with parameters τ', s', η', c' such that $\tau' \leq \tau$, $s' \leq s/2$, $\eta - 1 \leq \eta' \leq \eta$, and $c' \leq c + 1$.

In the 'fanin reduction' step, we start with an equation $\sum_{i=1}^{\tau} g_i N_i = 0$, with parameters τ, s, η, c and arrive at an equation $\sum_{i=1}^{\tau'} g_i' N_i' = 0$ with parameters τ', s', η', c' such that one of the two cases happens — either $(\tau' \leq \tau - 1, s' \leq s, \eta' = \eta - 1, c' = c)$ or $(\tau' = 1, s' \leq s, \eta' = \eta, c' \leq c + \tau)$.

Naturally, starting with $\sum_{i=1}^{t} f_i M_i = 0$, the 'sparsity reduction' step can only be performed at most $\log s$ many times (since the total sparsity of the g_i's reduces by at least a factor of half every time this step is executed), whereas the 'fanin reduction' step can be performed at most $t - 1$ times (as the fanin goes down by at least one for every such step). Finally, when this process of alternating steps ends, we have an equation of the form $g_i \cdot N_i = 0$, where $g_i \neq 0$ and N_i is a principal minor of a symbolic matrix N of dimension at least $n - (\log s + t - 1)$ such that at most $(\log s + t)$ entries of N are set as constants. Now, if $\log s + t \leq n - (\log s + t)$ the N_i can never be zero (by Fact 4.25) and hence we arrive at a contradiction. Therefore, $s > 2^{n/2 - t}$. Now, the details of the sparsity reduction and the fanin reduction steps.

Suppose, we have an equation $\sum_{i=1}^{\tau} g_i N_i = 0$ as mentioned above. Without loss of generality, assume that the minor N_i is the minor of N with respect to the i^{th} diagonal element of N. Call all the variables x_{ij} in N with both $i, j > \tau$ as the *white variables*. These are the variables that are present in every minor N_i in the sum $\sum_{i=1}^{\tau} g_i N_i$. The variables x_{ij} where both $i, j \leq \tau$ are called the *black variables*, and the remaining are the *grey variables*. By assumption, c of the variables in N are set as constants.

Sparsity reduction step - Say x is a white variable that one of the g_i's depends on. Writing each g_i as a polynomial in x, there must be two distinct powers of x amongst the g_i's (for if not, then x can be taken common across all g_i's). Let x^{ℓ} be the lowest degree and x^h be the highest. Dividing the entire equation $\sum_{i=1}^{\tau} g_i N_i = 0$ by x^{ℓ}, we can further assume that $\ell = 0$. Each of the g_i's and N_i's can be expressed as, $g_i = g_{i,0} + x \cdot g_{i,1} + \cdots + x^h \cdot g_{i,h}$ and $N_i = N_{i,0} + x \cdot N_{i,1}$, where $g_{i,j}$'s and $N_{i,j}$'s are x-free. Looking at the coefficients of x^0 and x^{h+1} in the equation yields $\sum_{i=1}^{\tau} g_{i,0} \cdot N_{i,0} = 0$ and $\sum_{i=1}^{\tau} g_{i,h} \cdot N_{i,1} = 0$. Note that $N_{i,0}$'s can be thought of as principal minors of the $\eta \times \eta$ matrix N' obtained by setting $x = 0$ in N. And each of the $N_{i,1}$'s can be thought of as minors of the $(\eta-1)\times(\eta-1)$ matrix N' which is the matrix associated with the minor of N with respect to x. Since the monomials in $g_{i,0}$ and $x^h g_{i,h}$ are disjoint, either the total sparsity of the $g_{i,0}$'s or the total sparsity of the $g_{i,h}$'s is $\leq s/2$. Thus, one of the equations $\sum_{i=1}^{\tau} g_{i,0}\cdot N_{i,0} = 0$ or $\sum_{i=1}^{\tau} g_{i,h}\cdot N_{i,1} = 0$ yields an equation of the form $\sum_{i=1}^{\tau'} g_i' N_i' = 0$ with parameters τ', s', η', c' as claimed before. (In case, we choose $\sum_{i=1}^{\tau} g_{i,h}\cdot N_{i,1} = 0$ as our next equation, we also set the variables in the same columns and rows of x to constants in such a way that a $g_{i,h}$ stays nonzero. This is certainly possible over a characteristic zero field (Lemma 2.1) The sparsity reduction step is performed whenever the starting equation $\sum_{i=1}^{\tau} g_i N_i = 0$ has a white variable among the g_i's. When all the g_i's are free of white variables, we perform the *fanin reduction step*.

Fanin reduction step - When we perform this step, all the g_i's consist of black and grey variables. Pick a row R from N barring the first τ rows. Let $y_1,\cdots,y_\tau$ be the grey variables occuring in R (these are, respectively, the variables in the first τ columns of R). Starting with y_2, divide the equation $\sum_{i=1}^{\tau} g_i N_i = 0$ by the largest power of y_2 common across all monomials in the g_i's, and then set $y_2 = 0$. This process lets us assume that there exists at least one g_i which is not zero at $y_2 = 0$. On the residual equation, repeat the same process with y_3 and then with y_4 and so on till y_τ. Thus, we can assume without loss of generality that in the equation $\sum_{i=1}^{\tau} g_i N_i = 0$ there is at least one g_i that is not zero when $y_2,\ldots,y_\tau$ are set to zero. Observe that if g_1 is the only g_i that stays nonzero under the projection $y_2 = \ldots = y_\tau = 0$ then $(g_1 N_1)_{(y_2=\ldots=y_\tau=0)} = 0$, implying that $N_1 = 0$ under the same projection - this is Case 2 of the fanin reduction step mentioned earlier. Now, assume that there is a g_i other than g_1 (say, g_2) that is nonzero under the projection $y_2 = \ldots = y_\tau = 0$. Set all the remaining variables of row R to zero except y_1 - these are the white variables in R. Since the g_i's are free of white variables (or else, we would have performed the 'sparsity reduction' step), none of the g_i's is effected by this projection. However, N_1 being a minor with respect to the first diagonal element of N, vanishes completely after the projection. Any other N_i takes the form $y_1\cdot N_i'$, where N_i' is a principal minor of a $(\eta-1)\times(\eta-1)$ matrix N' which is the matrix associated with the minor of N with respect to y_1. Therefore, after the projection, the equation $\sum_{i=1}^{\tau} g_i N_i = 0$ becomes $\sum_{i=2}^{\tau} \tilde{g}_i\cdot y_1 N_i' = 0 \Rightarrow \sum_{i=2}^{\tau} \tilde{g}_i \cdot N_i' = 0$, where $\tilde{g}_i$ is the image of g_i under the above mentioned projection and further $\tilde{g}_2 \neq 0$. The $\tilde{g}_i$'s might still contain variables from the first column of N. So, as a final step, set these variables to values so that a nonzero $\tilde{g}_i$ remains nonzero after this projection (Lemma 2.1 asserts that such values exist in plenty). This gives us the desired form $\sum_{i=1}^{\tau'} g_i' N_i' = 0$ with parameters τ', s', η', c' as claimed before (Case 1 of the fanin reduction step mentioned earlier). □

Lemma 4.24 (restated). If $M_1, \cdots, M_t$ are distinct first order principal minors of M and $\sum_{i=1}^{t} \alpha_i M_i = 0 \bmod \ell_k$ (not all $\alpha_i = 0$) for independent linear polynomials ℓ_k, then $t + k \geq n$.

Proof. Assume that $t + k < n$ (with $t \geq 1$ it means $k \leq n - 2$). Since $\ell_1, \cdots, \ell_k$ are independent linear polynomials, the equation may be rewritten as $\sum_{i=1}^{t} \alpha_i M_i' = 0$ where (M_i')s are minors of the matrix M' obtained by replacing k entries of M by linear polynomials in other variables. We shall call these entries as *corrupted* entries. Without loss of generality, we shall assume that M_i' is the minor corresponding to the i-th diagonal variable and that all the α_i's are nonzero.

Claim 4.26. *Each of the first t rows and columns must have a corrupted entry.*

Pf. Suppose the first row (without loss of generality) is free of any corrupted entry. Then, setting the entire row to zero would make all $M_i' = 0$ for $i \neq 1$. But since $\sum \alpha_i M_i' = 0$, this forces M_1' to become zero under the projection as well. This leads to a contradiction as M_1' is a determinant of an $(n-1) \times (n-1)$ symbolic matrix under a projection, and this can not be zero unless $k \geq n - 1$ (by Fact 4.25). □ (Claim)

Since $n - k > t$, there must exist a set of $t - 1$ rows $\{R_1, \cdots, R_{t-1}\}$ of M that are free of any corrupted entries. For each of these rows, set the i-th variable of row R_i to 1, and every other variable in $R_1, \cdots, R_{t-1}$ to zero. These projections make $M_i' = 0$ for all $i \neq t$ (as in these minors an entire row vanishes). And since $\sum_{i=1}^{t} \alpha_i M_i' = 0$, this forces M_t' to become zero under this projection as well. But under this projection, M_t' just reduces (up to a sign) to the minor obtained from M' by removing the columns $\{1, \cdots, t\}$ and rows $\{R_1, \cdots, R_{t-1}\} \cup \{t\}$. This is a determinant of an $(n-t) \times (n-t)$ symbolic matrix, containing at most $k - t$ corrupted entries, thus $k - t \geq n - t$ (by Fact 4.25). But then $k \geq n$, which contradicts our initial assumption. □

Approaching the chasm at depth four

5

5.1 Introduction

As mentioned in Chapter 1, the main motivating question in the field of arithmetic circuit complexity is proving super-polynomial lower bounds for the permanent. The permanent, by virtue of being complete for the class VNP (an algebraic analogue of the class NP, defined in [Val79]), occupies a central position in the study of the complexity of counting problems. The best known circuit for the permanent is actually a depth three homogeneous circuit of size $O(n^2 \cdot 2^n)$ and is called the Ryser's formula. Its illustrious sibling, the determinant, is widely believed to be comparatively easy, being complete for a subclass of VP (an algebraic analogue of P, also defined in [Val79]). It is conjectured (cf. [AV08]) that any arithmetic circuit computing the $n \times n$ permanent must be of $\exp(n)$ size. Meanwhile, the arithmetic complexity of computing the determinant equals $\tilde{O}(n^{\omega})$, where ω is the exponent of matrix multiplication. Resolving the arithmetic complexity of computing the permanent and the determinant (i.e. determining the exponent of matrix multiplication) are two of the most fascinating open problems of our times.

5.1.1 Prior Work

Lower bounds have been obtained earlier for depth three arithmetic circuits (with some restrictions) and constant depth multilinear circuits. Specifically, Nisan and Wigderson [NW97] showed that any homogeneous depth three circuit computing the permanent (also the determinant) must be of exponential size. Following that, Grigoriev and Karpinski [GK98] showed that any depth three arithmetic circuit over a finite field computing the permanent (also the determinant) requires exponential size but proving lower bounds for depth three circuits over fields of characteristic zero (or even over the algebraic closure of a finite field) remains an outstanding open problem. In this direction Shpilka and Wigderson [SW01] proved quadratic lower bounds for depth three circuits over arbitrary fields (without the homogeneity restriction). Meanwhile, Raz [Raz09] showed that any multilinear formula computing the permanent (also the determinant) must be of superpolynomial size. Following this, Raz and Yehudayoff [RY08] proved exponential lower bounds for constant depth multilinear circuits.

5.1.2 The model

In this chapter, we focus our attention on depth four homogeneous circuits. From the discussion in Section 2.2.1, as computation by polynomial-sized arithmetic circuits of unbounded depth is concerned one can assume without loss of generality that the circuit is homogeneous. Specifically, if a homogeneous polynomial f of degree d can be computed by an (unbounded depth) arithmetic circuit of size s, then it can also be computed by a *homogeneous* circuit of size $O(d^2 \cdot s)$. General depth-4 circuits could use polynomials of arbitrary degree during the intermediate computations. This chapter shall study a sub-class of homogeneous depth-4 circuits whose bottom multiplication gates have fan-in bounded by a parameter t; we shall denote this class by $\Sigma\Pi\Sigma\Pi^{[\mathrm{hom}]}(t)$ circuits.

A $\Sigma\Pi\Sigma\Pi^{[\mathrm{hom}]}(t)$ circuit computes a polynomial of the form

$$C \quad = \quad \sum_{i=1}^{s} (Q_{i1} \cdot Q_{i2} \cdot \ldots \cdot Q_{id}) \tag{5.1}$$

where each Q_{ij} is homogeneous polynomial of degree bounded by t, and every summand has the same degree. Our motivation for investigating representations of the form (5.1) stems from a recent result of Agrawal and Vinay [AV08], and a subsequent strengthening by Koiran [Koi10].

Theorem 5.1. *[AV08, Koi10] If f is a degree-d N-variate polynomial computed by a polynomial size homogeneous circuit, then there is a $\Sigma\Pi\Sigma\Pi^{[\mathrm{hom}]}(\sqrt{d})$ circuit computing f of size* $\exp(O(\sqrt{d}\log^2 N))$ *computing f.*

Similarly, if f is computed by a polynomial sized homogeneous formula, then there is a $\Sigma\Pi\Sigma\Pi^{[\mathrm{hom}]}(\sqrt{d})$ circuit computing f of size $\exp(O(\sqrt{d}\log N))$.

The contrapositive of the above statement for Perm_n is that it suffices to show a $\exp(\omega(\sqrt{n}\log^2 n))$ lower bound for $\Sigma\Pi\Sigma\Pi^{[\mathrm{hom}]}(\sqrt{n})$ circuits computing the Perm_n to prove a super-polynomial circuit lower bound. Thus, a good enough lower bound for $\Sigma\Pi\Sigma\Pi^{[\text{hom}]}(\sqrt{n})$ circuits would imply super-polynomial lower bounds for Perm_n. In this chapter, we give a lower bound for the permanent (or determinant) that comes very close to the above threshold.

Theorem 5.2. *Any* $\Sigma\Pi\Sigma\Pi^{[\text{hom}]}(t)$ *that computes the polynomial* Perm_n *(or* Det_n*) must have size* $\exp\left(\Omega\left(\frac{n}{t}\right)\right)$.

Corollary 5.3. *Any* $\Sigma\Pi\Sigma\Pi^{[\text{hom}]}(\sqrt{n})$ *that computes the polyomial* Perm_n *(or* Det_n*) must have size* $\exp\left(\Omega\left(\sqrt{n}\right)\right)$.

Remark. The results of Agrawal-Vinay [AV08] and Koiran [Koi10] depth-reduce any polynomial sized circuit computing a degree n polynomial to a $\Sigma\Pi\Sigma\Pi^{[\text{hom}]}(t)$ formula with top fanin $\exp\left(\frac{n}{t}\log^2 n\right)$. The above theorem infact is a bound on the top fanin of $\Sigma\Pi\Sigma\Pi^{[\text{hom}]}(t)$ circuits computing the permanent or determinant. In Section 5.5, we shall prove a prove a generalization of Theorem 5.2 by extending the lowerbound for all circuits that are sums of arbitrary powers of $O\left(\frac{n}{t}\right)$-many degree t polynomials. Further, the proofs are completely elementary and self-contained. Also, though the above theorem gives a lower bound for both the determinant and permanent, there is a subtle difference between the two and we discuss this in Section 5.6.

5.2 Basic Idea and Outline

The key idea is to exploit the *shifted derivatives of a polynomial*, which we shall now define. Recall some notations (from Section 2.1): For an n-tuple $\mathbf{i} = (i_1, i_2, \ldots, i_n) \in \mathbb{Z}^n_{\geq 0}$, $\mathbf{x}^{\mathbf{i}}$ denotes the monomial $(x_1^{i_1} \cdot x_2^{i_2} \cdot \ldots \cdot x_n^{i_n})$ which has degree $|\mathbf{i}| \stackrel{\text{def}}{=} (i_1 + i_2 + \ldots + i_n)$. Also, $\partial^{\mathbf{i}} f$ denotes the partial derivative of f with respect to the monomial $\mathbf{x}^{\mathbf{i}}$,

$$\partial^{\mathbf{i}} f \quad \stackrel{\text{def}}{=} \quad \frac{\partial^{i_1}}{\partial x_1^{i_1}}\left(\frac{\partial^{i_2}}{\partial x_2^{i_2}}\left(\cdots\left(\frac{\partial^{i_n} f}{\partial x_n^{i_n}}\right)\cdots\right)\right).$$

For a finite subset of polynomials $S \subseteq \mathbb{F}[\mathbf{x}]$, the $\mathbb{F}$-span of S, denoted $\mathbb{F}\text{-span}(S)$, is the set of all possible $\mathbb{F}$-linear combinations of polynomials in S. i.e.

$$\mathbb{F}\text{-span}(S) \quad \stackrel{\text{def}}{=} \quad \left\{\sum_{i=1}^{|S|} \alpha_i \cdot f_i \ : \ \alpha_i \in \mathbb{F}, \quad f_i \in S\right\}.$$

With these notational preliminaries in hand, we are now ready to define the key concept.

Definition 5.4 (Shifted Derivatives). *Let $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a multivariate polynomial. The* span of the ℓ-shifted k-th order derivatives *of f, denoted $\langle \partial^{=k} f \rangle_{\leq \ell}$, is defined as*

$$\langle \partial^{=k} f \rangle_{\leq \ell} \quad \stackrel{\text{def}}{=} \quad \mathbb{F}\text{-span}\left\{ \mathbf{x}^{\mathbf{i}} \cdot (\partial^{\mathbf{j}} f) \quad : \quad \mathbf{i}, \mathbf{j} \in \mathbb{Z}^n_{\geq 0} \quad \text{with } |\mathbf{i}| \leq \ell \quad \text{and } |\mathbf{j}| = k \right\}$$

Since the set $\langle \partial^{=k} f \rangle_{\leq \ell}$ forms an $\mathbb{F}$-vector space, we shall denote by $\dim(\langle \partial^{=k} f \rangle_{\leq \ell})$ *the dimension of this space.*

Recent work in arithmetic complexity has shown how $\langle \partial^{=k} f \rangle_{\leq \ell}$ can give insights into the structure and complexity of f in ways that are sometimes surprising and unexpected. The dimension of partial derivatives employed by Nisan and Wigderson [NW97] in their lower bound proofs corresponds to looking at $\dim(\langle \partial^{=k} f \rangle_{\leq 0})$. Kayal [Kay12a] showed that $\langle \partial^{=1} f \rangle_{\leq 1}$ yields a lie algebra that can help efficiently determine if f is equivalent (via an affine change of variables) to the permanent (or determinant). For $\ell = \infty$, note that $\langle \partial^{=k} f \rangle_{\leq \ell}$ is precisely the ideal generated by the k-th order derivatives of f. Gupta, Kayal and Qiao [GKQ12] recently exploited the structure of $\langle \partial^{=1} f \rangle_{\leq \infty}$ to devise an efficient reconstruction algorithm for random arithmetic formulas. Closer to the present application, Kayal [Kay12b] showed how $\dim(\langle \partial^{=k} f \rangle_{\leq \ell})$ (for suitably chosen ℓ and k) can be used to prove an exponential lower bound for representing a polynomial as a sum of powers of bounded degree polynomials. We shall see in this chapter that for suitably chosen values of ℓ and k, we can establish a large gap between $\dim(\langle \partial^{=k} \mathsf{Perm}_n \rangle_{\leq \ell})$ and $\dim(\langle \partial^{=k} f \rangle_{\leq \ell})$ when f is computed by a small $\Sigma\Pi\Sigma\Pi^{[\text{hom}]}(t)$ circuit. This separation shall give the lower bound.

5.2.1 Outline of the chapter

We shall execute this idea as follows. In Section 5.3 we shall give an upper bound on $\dim(\langle \partial^{=k} C \rangle_{\leq \ell})$ for C being a polynomial computed by a $\Sigma\Pi\Sigma\Pi^{[\text{hom}]}(t)$ circuit, i.e. when C is of the form given in equation (5.1). In Section 5.4, we shall present a lower bound estimate for $\dim(\langle \partial^{=k} \mathsf{Perm}_n \rangle_{\leq \ell})$. We then combine these bounds to obtain a proof of our main theorem in Section 5.5. Finally, in Section 5.6, we discuss the possibility of improving the estimates for $\dim(\langle \partial^{=k} \mathsf{Perm}_n \rangle_{\leq \ell})$ obtained here.

5.3 Shifted partials of $\Sigma\Pi\Sigma\Pi^{[\mathrm{hom}]}(t)$ circuits

In this section we give an upper bound on $\dim(\langle\partial^{=k}C\rangle_{\leq\ell})$ when C is computed by a depth four circuit, i.e. C is of the form given in equation (5.1). We begin by noting that $\dim(\langle\partial^{=k}f\rangle_{\leq\ell})$ is sub-additive.

Proposition 5.5. *For all $k,\ell\geq 0$, we have* $\dim(\langle\partial^{=k}(f+g)\rangle_{\leq\ell}) \leq \dim(\langle\partial^{=k}f\rangle_{\leq\ell})+\dim(\langle\partial^{=k}g\rangle_{\leq\ell})$.

Proof. By linearity of partial derivatives, we have $\mathbf{x}^{\mathbf{i}}\cdot\partial^{\mathbf{j}}(f+g)=\mathbf{x}^{\mathbf{i}}\cdot\partial^{\mathbf{j}}f+\mathbf{x}^{\mathbf{i}}\cdot\partial^{\mathbf{j}}g$. Hence,

$$\langle\partial^{=k}(f+g)\rangle_{\leq\ell} \quad\subseteq\quad \mathbb{F}\text{-span}\left(\langle\partial^{=k}f\rangle_{\leq\ell}\cup\langle\partial^{=k}g\rangle_{\leq\ell}\right)$$

The proposition follows. □

Let C be a depth-4 circuit computing a polynomial of the form[1]

$$C \quad=\quad \sum_{i=1}^{s}Q_{i1}^{e_{i1}}\cdot Q_{i2}^{e_{i2}}\dots Q_{id}^{e_{id}} \quad \text{where } \deg(Q_{ij})\leq t.$$

By Proposition 5.5, it suffices to understand the growth of $\dim(\langle\partial^{=k}C\rangle_{\leq\ell})$ of a single term $(Q_1^{e_1}\dots Q_d^{e_d})$.

Proposition 5.6. *If $f=Q_1^{e_1}\dots Q_d^{e_d}$ where each $Q_i\in\mathbb{F}[\mathbf{x}_N]$ is a polynomial of degree bounded by t. Then, for any $\ell\geq 0$,*

$$\dim(\langle\partial^{=k}f\rangle_{\leq\ell}) \quad\leq\quad \binom{d+k-1}{k}\binom{N+(t-1)k+\ell}{N}$$

Proof. Let $\mathbf{j}\in\mathbb{Z}_{\geq 0}^{d}$ be any d-tuple satisfying $|\mathbf{j}|=k$. Using the product rule of differentiation,

$$\partial^{\mathbf{j}}\left(Q_1^{e_1}\dots Q_d^{e_d}\right) \quad=\quad \sum_{\mathbf{j}_1+\dots+\mathbf{j}_d=\mathbf{j}}\left(\partial^{\mathbf{j}_1}Q_1^{e_1}\right)\dots\left(\partial^{\mathbf{j}_d}Q_d^{e_d}\right)$$

Let j_i be the sum of the entries of the tuple $\mathbf{j}_i$. Note that since $|\mathbf{j}|=k$ we have $\sum_{i=1}^{d}j_i=k$. Hence, each term in the above sum can be written as $\left(Q_1^{e_1-j_1}\dots Q_d^{e_d-j_d}\right)\cdot\tilde{Q}$ where $\sum j_i=k$ and $\tilde{Q}$ has degree at most $(tk-k)$. Thus, every element of $\mathbf{x}^{\leq\ell}\partial^{=k}(Q_1^{e_1}\cdots Q_d^{e_d})$ can be written as a linear combination of $\left(Q_1^{e_1-j_1}\dots Q_d^{e_d-j_d}\right)\mathbf{x}^{\mathbf{r}}$ where $\sum j_i=k$ and $\mathbf{x}^{\mathbf{r}}$

[1] This is slightly more general than the form described in Equation (5.1).

is a monomial of degree at most $\ell+(t-1)k$. The total number of monomials of degree at most $\ell+(t-1)k$ over N variables is $\binom{N+(t-1)k+\ell}{N}$, and the total number of choices for $j_1+\cdots+j_d=k$ is $\binom{d+k-1}{k}$. Hence we obtain,

$$\dim(\langle\partial^{=k}(Q_1^{e_1}\cdots Q_d^{e_d})\rangle_{\leq\ell}) \quad\leq\quad \binom{d+k-1}{k}\binom{N+(t-1)k+\ell}{N}$$

□

The following corollary follows directly from the above observation via sub-additivity.

Corollary 5.7. *If $C=\sum_{i=1}^{s}\prod_{j=1}^{d}Q_{ij}^{e_{ij}}$ where each $Q_{ij}\in\mathbb{F}[\mathbf{x}_N]$ is a polynomial of degree bounded by t, then for any $k\leq d$*

$$\dim(\langle\partial^{=k}(C)\rangle_{\leq\ell}) \quad\leq\quad s\cdot\binom{d+k-1}{k}\binom{N+(t-1)k+\ell}{N}$$

Since our lower bound results are for homogeneous depth four formulae, we would require an upper bound on d used in Corollary 5.7. Suppose the degree of the polynomial computed is D. It is possible that in a $\Sigma\Pi\Sigma\Pi^{[\mathrm{hom}]}(t)$ circuit, several Q_{ij}'s could have degree much smaller than t and hence d could potentially be much larger than D/t. However, by multiplying *low degree* Q_{ij}'s together and thereby ensuring that every Q_{ij} (except perhaps one) have degree at least $t/2$, we can assume without loss of generality that $d\leq 2(D/t)+1$. Note that this process may blow up the size of the formula, nevertheless we need this only for upper bounding $\dim(\langle\partial^{=k}(C)\rangle_{\leq\ell})$, which does not change since the polynomial computed by C does not change. Thus, we have the following corollary,

Corollary 5.8. *If $C=\sum_{i=1}^{s}\prod_{j=1}^{d_i}Q_{ij}$ is a degree D polynomial, where each $Q_{ij}\in\mathbb{F}[\mathbf{x}_N]$ is a homogeneous polynomial of degree bounded by t, then for any $k\leq 2(D/t)+1$*

$$\dim(\langle\partial^{=k}(C)\rangle_{\leq\ell}) \quad\leq\quad s\cdot\binom{2(D/t)+k}{k}\binom{N+(t-1)k+\ell}{N}$$

In the next section we give a reasonable lower bound for $\dim(\langle\partial^{=k}(\mathsf{Perm}_n)\rangle_{\leq\ell})$ for suitable choice of parameters k and ℓ.

5.4 Shifted partials of the Permanent

Reducing dimension computation to counting leading monomials. In this section, we shall present a lower bound for $\dim(\langle\partial^{=k}(\mathsf{Perm}_n)\rangle_{\leq\ell})$. Let $\succ$ be any admissible monomial ordering[2]. Recall that the leading monomial of a polynomial $f \in \mathbb{F}[\mathbf{x}]$, denoted $\mathsf{LM}(f)$, is the largest monomial $\mathbf{x}^{\mathbf{i}}$ under the ordering $\succ$.

Proposition 5.9. *Let $S \subseteq \mathbb{F}[\mathbf{x}]$ be any finite set of polynomials. Then*

$$\dim(\mathbb{F}\text{-}span(S)) = \#\{\mathsf{LM}(f) \ : \ f \in \mathbb{F}\text{-}span(S)\}.$$

The proof is a simple application of Gaussian elimination. As a corollary we obtain

Corollary 5.10. *For any polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ we have*

$$\dim(\langle\partial^{=k} f\rangle_{\leq\ell}) \quad \geq \quad \#\{\mathbf{x}^{\mathbf{i}} \cdot \mathsf{LM}(\partial^{\mathbf{j}} f) \ : \ \mathbf{i},\mathbf{j} \in \mathbb{Z}_{\geq 0}^{|\mathbf{x}|}, \ |\mathbf{i}| \leq \ell \quad \text{and } |\mathbf{j}| = k\}$$

The lower bound given by this corollary is usually a severe underestimate but fortunately even this will suffice for our purpose for the case when $f = \mathsf{Perm}_n$.

Reduction to counting monomials with increasing subsequences. Let us fix $\succ$ to be the lexicographic monomial ordering induced by the following ordering on the variables: $x_{11} \succ \cdots \succ x_{1n} \succ x_{21} \succ \cdots \succ x_{nn}$. Now note that any partial derivative of Perm_n is just the corresponding permanental minor (or just 'P-minor'). Hence by the above corollary we have

$$\dim(\langle\partial^{=k}(\mathsf{Perm}_n)\rangle_{\leq\ell}) \geq \#\left\{\mathbf{x}^{\mathbf{i}} \cdot \mathsf{LM}(M) \ : \ \begin{array}{l} \mathbf{x}^{\mathbf{i}} \text{ is a monomial of degree at most } \ell \text{ and} \\ M \text{ is an } (n-k)\times(n-k) \text{ P-minor} \end{array}\right\}$$

Now note that the leading monomial under $\succ$ of any $(n-k)\times(n-k)$ P-minor M is just the product of the variables along the principal diagonal of M. Now if the variables along the principal minor of M are $(x_{i_1 j_1}, \cdots, x_{i_{n-k} j_{n-k}})$ then the indices satisfy

$$i_1 < i_2 < \ldots < i_{n-k} \quad \text{and } j_1 < j_2 < \ldots < j_{n-k}$$

This naturally leads to the following definition.

[2] For more on monomial orderings and their applications in algebraic geometry, we refer the interested reader to Chapter 2 of the text by Cox, Little and O'Shea [CLO07]

Definition 5.11 (Increasing sequence). *A sequence of variables* $(x_{i_1 j_1}, \cdots, x_{i_t j_t})$ *is called* increasing sequence *if the indices satisfy*

$$i_1 < i_2 < \ldots < i_t \quad \text{and } j_1 < j_2 < \ldots < j_t.$$

We will say that a monomial $A = \mathbf{x^j}$ contains an increasing sequence of length t *if there exists an increasing sequence* $(x_{i_1 j_1}, \cdots, x_{i_t j_t})$ *wherein every variable* $x_{i_r j_r}$ *(*$r \in [t]$*) divides* A*.*

In this terminology we would then say that the leading monomial of any $(n-k) \times (n-k)$ P-minor is exactly the product of the variables in an increasing sequence of length $(n-k)$. Consequently for any P-minor M of size $(n-k)$ we have that $\mathbf{x^i} \cdot \mathsf{LM}(M)$ contains an increasing sequence of length $(n-k)$. Conversely, every monomial of degree at most $(n-k+\ell)$ that contain an increasing sequence of length $(n-k)$ can be written as the leading monomial of $\mathbf{x^i} \cdot M$ for some monomial $\mathbf{x^i}$ of degree at most ℓ and a $(n-k) \times (n-k)$ P-minor M. Hence we have:

Corollary 5.12. $\dim(\langle \partial^{=k}(\mathsf{Perm}_n) \rangle_{\leq \ell})$ *is lower bounded by the number of distinct monomials of degree at most* $(n-k+\ell)$ *over* n^2 *variables that contain an increasing sequence of length* $(n-k)$*.*

In order to count the number of monomials of degree bounded by $n-k+\ell$ that contain an increasing sequence, we shall restrict ourselves to a very *small set* of variables to contribute the increasing sequence, and "fill-up" the remaining degree using the other variables. The "small set" that we consider here is just two diagonals – the principal diagonal and the one above it.

5.4.1 Restricting to two diagonals

We shall focus on the variables $D_{2,n} = \{x_{ii} \,:\, 1 \leq i \leq n\} \cup \left\{x_{i(i+1)} \,:\, 1 \leq i \leq n-1\right\}$. To get a lower bound on monomials containing an $(n-k)$-length increasing sequence, we shall fix an increasing sequence $Q \subseteq D_{2,n}$ and count all monomials m that contain Q as the "leading increasing sequence" i.e. amongst all possible increasing sequences contained in m, the predefined Q is highest under the lexicographic order defined earlier. The following lemma counts the number of increasing sequences contained in $D_{2,n}$.

Lemma 5.13. *The number of length r increasing sequences in contained in $D_{2,n}$ is exactly $\binom{2n-r}{r}$.*

Proof. Consider the $(2n-1)$ variables in $D_{2,n}$ in the sequence $x_{11}, x_{12}, x_{21}, \ldots, x_{nn}$. Picking an increasing sequence of length r is the same as picking r of the $(2n-1)$ variables such that no two adjacent variables (in the above order) are chosen. This can be thought as distributing the $(2n-r-1)$ variables that won't be picked such that there is at least one between any two variables that are picked, and this is exactly equal to

$$\binom{(2n-r-1-(r-1))+(r+1)-1}{(r+1)-1} = \binom{2n-r}{r}$$

□

For any variable x_{ij}, define its *companions* to be the variables to its right in the same row, or below it in the same column, i.e. $\left\{x_{ij'} \ : \ j' > j\right\} \cup \left\{x_{i'j} \ : \ i' > i\right\}$. Fix an increasing sequence $Q = \left\{x_{i_1 j_1}, \ldots, x_{i_r j_r}\right\} \subseteq D_{2,n}$. Let Q' be the set of all companions of variables in Q which are in $D_{2,n}$. The key observation is that adding elements of Q' to Q does not alter the leading increasing sequence. For any increasing sequence that uses elements of Q', replacing every $x_{i'j'} \in Q'$ by the corresponding $x_{ij} \in Q$ for which it is a companion for yields a "higher" increasing sequence. Hence adding any subset $T \subseteq Q'$ to Q does not alter the leading increasing sequence. Let us call the remaining variables $D_{2,n} \setminus (Q \cup Q')$ as *potentially forbidden variables*. Thus, any monomial that does not include any potentially forbidden variable cannot alter the leading increasing sequence.

Note that every element of $D_{2,n}$ besides x_{nn} has exactly one companion in $D_{2,n}$. Hence, if Q is a length r increasing sequence, the set of companions its Q' has cardinality at least $(r-1)$ and hence the set of forbidden variables has cardinality at most $2(n-r)$. Using this, the following bound follows almost immediately.

Corollary 5.14. *For every $n, k, \ell \geq 0$,*

$$\dim(\langle \partial^{=k} \mathsf{Perm}_n \rangle_{\leq \ell}) \geq \binom{n+k}{2k} \cdot \binom{n^2+\ell-2k}{n^2-2k}$$

Proof. For any fixed length $(n-k)$ increasing sequence Q, there are $2k$ potentially forbidden variables. Thus, any monomials of degree at most ℓ that does not include the $2k$ potentially forbidden variables can be adjoined to Q without altering the leading increasing sequence. The number of monomials on n^2-2k variables of degree at most ℓ is exactly $\binom{n^2+\ell-2k}{n^2-2k}$, and multiplying by the number of choices of Q (by Lemma 5.13) gives the required lower bound. □

5.5 Putting it all together

We shall require a few technical lemmas about the growth of binomial coefficients, factorials etc.

5.5.1 Growth of binomial estimates

Almost all bounds on the binomial coefficients follow from Stirling's approximation.

Proposition 5.15 (Stirling's Formula, cf. [Rom00]). $\ln(n!) \quad = \quad n\ln n - n + O(\ln n)$

Using Stirling's approximation for binomial coefficients eventually yield expressions involving the entropy function.

Definition 5.16 (Entropy). *The binary entropy function H_2 is defined as*

$$H_2(x) \quad = \quad -x\cdot\log_2(x) - (1-x)\cdot\log_2(1-x)$$

The natural-log version of the entropy function, denoted by H_e is defined analogously as

$$H_e(x) = -x\cdot\ln(x) - (1-x)\ln(1-x)$$

Lemma 5.17. *For any $0 < x < 1$, we have $x\ln\frac{1}{x} \leq H_e(x) \leq x\ln\frac{1}{x} + x$.*

Lemma 5.18. *For any constants $\alpha \geq \beta > 0$,*

$$\ln\binom{\alpha n}{\beta n} \quad = \quad aH_e\left(\frac{\beta}{\alpha}\right)n + O(\ln n)$$

Proof. By Stirling's approximation (Proposition 5.15),

$$\begin{aligned}\ln\frac{(\alpha n)!}{(\beta n)!((\alpha-\beta)n)!} &= (\alpha n)\ln(\alpha n) - \alpha n - (\beta n)\ln(\beta n) + \beta n\\ &\qquad -(\alpha-\beta)n\ln((\alpha-\beta)n) + (\alpha-\beta)n + O(\ln n)\\ &= n(\alpha\ln\alpha - \beta\ln\beta - (\alpha-\beta)\ln(\alpha-\beta)) + O(\ln n)\\ &= \alpha n\cdot H_e\left(\frac{\beta}{\alpha}\right) + O(\ln n)\end{aligned}$$

□

The following lemma would be useful for estimating the growth of ratio of binoimial coefficients.

Lemma 5.19. *Let $a(n),\ f(n),\ g(n) : \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$ be integer valued function such that $(f+g) = o(a)$. Then,*

$$\ln \frac{(a+f)!}{(a-g)!} \quad = \quad (f+g)\ln a \pm O\left(\frac{(f+g)^2}{a}\right)$$

Proof.

$$\frac{(a+f)!}{(a-g)!} = (a+f)(a+f-1)\ldots(a-g)$$

$$\implies \quad a^{f+g}\left(1-\frac{g}{a}\right)^{f+g} \quad \leq \quad \frac{(a+f)!}{(a-g)!} \quad \leq \quad a^{f+g}\left(1+\frac{f}{a}\right)^{f+g}$$

$$\implies \quad (f+g)\ln\left(1-\frac{g}{a}\right) \quad \leq \quad \ln\frac{(a+f)!}{(a-g)!} - (f+g)\ln a \quad \leq \quad (f+g)\ln\left(1+\frac{f}{a}\right)$$

Using the fact that $\frac{x}{1+x} \leq \ln(1+x) \leq x$ for $x > -1$, it is easy to see that both the LHS and RHS are bounded by $O\left(\frac{(f+g)^2}{a}\right)$. □

5.5.2 Proof of the main theorem

We are now ready to prove the main theorem, which is a stronger form of Theorem 5.2.

Theorem 5.20. *Let $t : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ be any increasing function such that $t(n) = o(n)$. Suppose C is a circuit of the form $C = \sum_{i=1}^{s} Q_{i1}^{e_{i1}} \cdots Q_{id}^{e_{id}}$ where each Q_{ij} is a polynomial of degree bounded by t, and $d = cn/t$ for some constant c. If C computes the polynomial Perm_n, then $s \geq \exp\left(\Omega\left(\frac{n}{t}\right)\right)$.*

Proof. From Corollary 5.7, $\dim(\langle \partial^{=k} C\rangle_{\leq \ell})$ can be upper bounded as

$$\dim(\langle \partial^{=k}(C)\rangle_{\leq \ell}) \quad \leq \quad s \cdot \binom{d+k-1}{k}\binom{n^2+\ell+(t-1)k}{n^2} \tag{5.2}$$

Also, Corollary 5.14 gives a lower bound for $\dim(\langle \partial^{=k}\mathsf{Perm}_n\rangle_{\leq \ell})$

$$\dim(\langle \partial^{=k}\mathsf{Perm}\rangle_{\leq \ell}) \quad \geq \quad \binom{n+k}{2k}\binom{n^2+\ell-2k}{n^2-2k} \tag{5.3}$$

Both these equations imply that

$$s \quad \geq \quad \frac{\binom{n+k}{2k}\binom{n^2+\ell-2k}{n^2-2k}}{\binom{d+k-1}{k}\binom{n^2+\ell+(t-1)k}{n^2}}$$

We shall set parameters as $\ell = n^2 t$ and $k = \varepsilon(n/t)$ (for an $\varepsilon > 0$ that shall be chosen shortly). The proofs of the following estimates for binomial coefficients are straightforward applications of Lemma 5.19 and Lemma 5.18, and we shall defer its proof to the end of the section.

Claim 5.21. *For the above choice of parameters:*

(a) $\ln\binom{n+k}{2k} = 2\varepsilon\left(\frac{n}{t}\right)\left(\ln\left(\frac{t}{2\varepsilon}\right)+1\right) \pm O\left(\frac{n}{t^2}\right)$

(b) $\ln\binom{cn/t+k-1}{k} = (c+\varepsilon)H_e\left(\frac{\varepsilon}{c+\varepsilon}\right)\cdot\left(\frac{n}{t}\right) \pm O(\ln n)$

(c) $\ln\frac{\binom{n^2+\ell-2k}{n^2-2k}}{\binom{n^2+\ell+(t-1)k}{n^2}} = -2\varepsilon\left(\frac{n}{t}\right)\ln t - 2\varepsilon\left(\frac{n}{t}\right)\left(\frac{t+1}{t}\right) \pm O\left(\frac{n}{t^2}\right)$

Using this, we get

$$\ln s \geq \left(2\varepsilon\ln\left(\frac{1}{2\varepsilon}\right) - (c+\varepsilon)H_e\left(\frac{\varepsilon}{c+\varepsilon}\right) - \frac{2\varepsilon}{t}\right)\left(\frac{n}{t}\right) \pm O\left(\frac{n}{t^2}\right)$$

which after an application of Lemma 5.17 yields

$$\begin{aligned}\ln s &\geq \left(2\varepsilon\ln\frac{1}{2\varepsilon} - \varepsilon\ln\left(\frac{c+\varepsilon}{\varepsilon}\right) - 3\varepsilon\right)\\ &= \left(\varepsilon\ln\frac{1}{\varepsilon} - \varepsilon\ln(4e^3(c+1))\right)\left(\frac{n}{t}\right) \pm + O\left(\frac{n}{t^2}\right)\end{aligned}$$

Choosing ε small enough gives $\ln s = O\left(\frac{n}{t}\right)$, i.e. $s \geq \exp\left(\Omega\left(\frac{n}{t}\right)\right)$ as claimed □

Remark. Though the above theorem is stated for any increasing function $t(n)$, the result also holds when t is a constant. The choice of parameters in that case would be $\ell = n^2$, $m = 3(n-k)/2$ and $k = \varepsilon n$. Using similar estimates on the binomial coefficients, it can be shown that $\log s = \Omega(n)$ by choosing a small enough $\varepsilon > 0$.

The above theorem, along with Corollary 5.7, completes the proof of Theorem 5.2 as well. □

Proof of Claim 5.21.

(a) $\binom{n+k}{2k} = \frac{(n+k)!}{(n-k)!} \cdot \frac{1}{(2k)!}$. Since $k = o(n)$, using Lemma 5.18 and Lemma 5.19 gives

$$\begin{aligned}\ln \binom{n+k}{2k} &= 2k \ln n - (2k)\ln(2k) + 2k \pm O\left(\frac{k^2}{n}\right) \\ &= 2\varepsilon\left(\frac{n}{t}\right)\left(\ln\left(\frac{t}{2\varepsilon}\right)+1\right) \pm O\left(\frac{n}{t^2}\right)\end{aligned}$$

(b) Follows directly from Lemma 5.18.

(c)

$$\frac{\binom{n^2+\ell-2k}{n^2-2k}}{\binom{n^2+\ell+(t-1)k}{n^2}} = \frac{(n^2+\ell-2k)!}{(n^2+\ell+(t-1)k)!} \cdot \frac{(n^2)!}{(n^2-2k)!} \cdot \frac{(\ell+(t-1)k)!}{(\ell)!}$$

Using the fact that $tk = o(n^2+\ell)$, Lemma 5.19 can be applied on each of these ratios to give

$$\begin{aligned}\frac{\binom{n^2+\ell-2k}{n^2-2k}}{\binom{n^2+\ell+(t-1)k}{n^2}} &= \frac{1}{(n^2+\ell)^{(t+1)k}} \cdot (n^2)^{2k} \cdot \ell^{(t-1)k} \cdot \mathsf{poly}(n) \\ &= \frac{1}{\left(1+\frac{n^2}{\ell}\right)^{(t+1)k}} \cdot \left(\frac{n^2}{\ell}\right)^{2k} \cdot \mathsf{poly}(n) \\ \implies \ln \frac{\binom{n^2+\ell-2k}{n^2-2k}}{\binom{n^2+\ell+(t-1)k}{n^2}} &= -(t+1)k \cdot \ln\left(1+\frac{1}{t}\right) - 2k \ln t \pm O(\log n) \\ &= -2\varepsilon\left(\frac{n}{t}\right) \cdot \left(\frac{t+1}{t}\right) - 2\varepsilon\left(\frac{n}{t}\right) \cdot \ln t \pm O\left(\frac{n}{t^2}\right)\end{aligned}$$

□

5.6 Discussion

The proof of Theorem 5.2 remains valid if we replace every occurrence of Perm_n by Det_n but there turns out to be a very interesting distinction between these two polynomials with respect to the dimension of their shifted partial derivatives. In the particular case of the determinant, Corollary 5.12 can be strengthened to say that the number of monomials of degree at most $n-k+\ell$ with an increasing sequence of length $(n-k)$ is not just a lower bound but is exactly equal to $\dim(\langle\partial^{=k}(\mathsf{Det}_n)\rangle_{\leq\ell})$. This follows from the following powerful result on gröbner bases of determinantal ideals which has been proved independently by Sturmfels[Stu90], Narasimhan [Nar86] and Caniglia, Guccione and Guccione [CGG90].

Theorem 5.22 ([Stu90], [Nar86], [CGG90]). *Let $\succ$ be the lexicographic ordering on monomials defined in Section 5.4. Then the set of all order $r \times r$ minors of Det_n is the reduced gröbner basis for the ideal generated by them under the monomial ordering $\succ$.*

It is known that the set of 2×2 permanental minors *do not* form a gröbner basis for the ideal they generate. Thus it is presumable that $\dim(\langle\partial^{=k}(\mathsf{Perm}_n)\rangle_{\leq\ell})$ is much larger compared to the determinant.

6 Conclusion and future directions

The main motivation of this thesis has been to advance our understanding of polynomial identity testing and lower bounds with an emphasis on a holistic approach. By providing approaches for unification, we can hopefully gain more insight into the general problem of performing PIT on general circuits/formulae, or lower bounds for the permanent.

Each of the chapters leave scope for future directions. These are some of the obvious steps to take towards advancing the current state-of-the-art.

6.1 Composing Identity tests

The two problems studied in Chapter 3 also have natural generalizations. The first relates to depth-4 fan-in 2 PIT.

Open Problem 3.1. Find a deterministic polynomial time algorithm to check if $f = \prod_{i=1}^{t} g_i^{d_i}$, where f is a sparse polynomial and the g_i's are mutually coprime, bounded degree polynomials.

One particular case of interest is when the g_i's are quadratic forms. Observe that a polynomial g^d divides f if and only if g divides f and g^{d-1} divides $\frac{\partial f}{\partial x_1}$ (assuming f depends on x_1 and $\deg(f) > \mathrm{char}(\mathbb{F})$). Since $\frac{\partial f}{\partial x_1}$ is also sparse, using this observation, the problem eventually boils down to checking if g divides h, where both g and h are sparse polynomials. Now suppose g is a quadratic form. It is known that there exists an efficiently computable linear transformation σ on the variables such that $\sigma(g) = \sum_{i=1}^{r} x_i^2$, which is a sum of univariates. The polynomial g divides h if and only if $\sigma(g)$ divides $\sigma(h)$. We have shown how to divide a sparse polynomial by a sum of univariates. But, the issue here is that $\sigma(h)$ need not be sparse - it is an image of a sparse h under an invertible σ. Is it possible to resolve this issue?

The second relates to depth-4 higher fan-in PIT.

Open Problem 3.2. Find a deterministic polynomial time algorithm to solve PIT on depth-4 circuits with bounded top fan-in k, where each of the k multiplication gates is a product of sums of univariate polynomials.

Note that, a solution for $k = 2$ easily follows from Theorem 3.12 and unique factorization. But, it is unclear how to solve this problem even for $k = 3$. The problem can also be seen as a certain generalization of bounded top fan-in depth-3 PIT [KS07] to the case of depth-4 circuits.

Recent results of Agrawal, Saha and Saxena [ASS12], and also by Shpilka and Forbes [FS12] present quasipolynomial blackbox PITs for semidiagonal circuits. In this light, it is natural to ask if the blackbox PITs for semidiagonal circuits and $\Sigma\Pi\Sigma(k)$ circuits can be composed.

Open Problem 3.3. Given *blackbox* access to a semidiagonal circuit f and a polynomial p computed by a $\Sigma\Pi\Sigma(k)$ circuit C, can we check if $p + f \overset{?}{=} 0$?

6.2 PITs via algebraic independence

Spurred by the success of Jacobian in solving the hitting-set problem for *constant-trdeg* depth-3 circuits and *constant-occur* constant-depth formulas, one is naturally inspired to investigate the strength of this approach against other 'constant parameter' models - the foremost of which is *constant top fanin* depth-4 circuits?

Open Problem 4.1. Can the Jacobian based approach be used to construct polynomial time hitting-set generator for constant top fanin depth-4 circuits (even fanin 2)?

Another problem, which is closely related to hitting-sets and lower bounds, is reconstruction of arithmetic circuits [SY10, Chapter 5]. There is a quasi-polynomial time reconstruction algorithm [KS09a], for a polynomial computed by a depth-3 constant top fanin circuit, that outputs a depth-3 circuit with quasi-polynomial top fanin.

Open Problem 4.2. Can the Jacobian based approach be used to design a polynomial time reconstruction algorithm for constant top fanin depth-3 circuits?

An answer in the affirmative would further reinforce the versatility of this tool.

6.3 Shifted partial derivatives

Though the dimension of the shifted partial derivatives aid us get very close to the chasm, it is not powerful enough to let us jump across. Perhaps some modifications of the measure would be what is required to prove super-polynomial formula/circuit lower bounds for the permanent.

Further, as mentioned in Section 5.6, the dimension of the shifted partial derivatives of the permanent is strictly larger than that for the determinant, and this begs for the question "How much larger?"

Open Problem 5.1. There exists choices for $\ell, k \geq 0$ such that $\dim(\langle \partial^{=k}(\mathsf{Perm}_n)\rangle_{\leq \ell})$ is superpolynomially larger (in n) than $\dim(\langle \partial^{=k}(\mathsf{Det}_n)\rangle_{\leq \ell})$.

There are lots of interesting problems and the fundamental question does appear to be within reach. No doubt it would require a lot more new ideas but the endeavour towards settling the "Determinant vs Permanent" problem now appears to have more clarity. It would not be an easy journey, but that never stopped us from trying and certainly will not stop us now.

The woods are lovely, dark, and deep,
But I have promises to keep,
And miles to go before I sleep,
And miles to go before I sleep.
– Robert Frost

Bibliography

[AB99] Manindra Agrawal and Somenath Biswas. Primality and Identity Testing via Chinese Remaindering. In *FOCS*, pages 202–209, 1999. [ii, 4, 5, 16, 41]

[Agr05] Manindra Agrawal. Proving Lower Bounds Via Pseudo-random Generators. In *FSTTCS*, pages 92–105, 2005. [i, 4, 43]

[Agr11] Manindra Agrawal. On the arithmetic complexity of euler function. In *CSR*, pages 43–49, 2011. [4]

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in P. *Annals of Mathematics*, 160(2):781–793, 2004. [4]

[ALM$^+$98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verification and the Hardness of Approximation Problems. *Journal of the ACM*, 45(3):501–555, 1998. [4]

[AM69] M.F. Atiyah and I.G. MacDonald. *Introduction to Commutative Algebra*. Addison-Wesley Publishing Company, 1969. [27]

[ASS12] Manindra Agrawal, Chandan Saha, and Nitin Saxena. Quasi-polynomial Hitting-set for Set-depth-Δ Formulas. *Technical Report TR12-113, (ECCC)*, 2012. [5, 82]

[ASSS12] Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: hitting-sets, lower bounds for depth-d occur-k formulas & depth-3 transcendence degree-k circuits. In *STOC*, pages 599–614, 2012. [60]

[AV08] Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *FOCS*, pages 67–75, 2008. [ii, iii, 6, 7, 9, 10, 41, 43, 67, 68, 69]

[AvMV11] Matthew Anderson, Dieter van Melkebeek, and Ilya Volkovich. Derandomizing Polynomial Identity Testing for Multilinear Constant-Read Formulae. In *IEEE Conference on Computational Complexity*, pages 273–282, 2011. [ii, 6, 41, 42, 45]

[BMS11a] Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic Independence and Blackbox Identity Testing. In *ICALP (2)*, pages 137–148, 2011. [ii, 6, 9, 41, 42, 45]

[BMS11b] Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic Independence and Blackbox Identity Testing. *CoRR*, abs/1102.2789, 2011. [47]

[BS83] Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theor. Comput. Sci.*, 22:317–330, 1983. [3]

[CGG90] L. Caniglia, J. A. Guccione, and J. J. Guccione. Ideals of generic minors. *Commutative Algebra*, 18:2633–2640, 1990. [80]

[CK97] Zhi-Zhong Chen and Ming-Yang Kao. Reducing Randomness via Irrational Numbers. In *STOC*, pages 200–209, 1997. [4]

[CKW11] Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial Derivatives in Arithmetic Complexity (and beyond). *Foundation and Trends in Theoretical Computer Science*, 2011. [43]

[CLO07] D.A. Cox, J.B. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. Undergraduate texts in mathematics. Springer, 2007. [73]

[DGW09] Zeev Dvir, Ariel Gabizon, and Avi Wigderson. Extractors And Rank Extractors For Polynomial Sources. *Computational Complexity*, 18(1):1–58, 2009. [5]

[DL78] Richard A. DeMillo and Richard J. Lipton. A Probabilistic Remark on Algebraic Program Testing. *Inf. Process. Lett.*, 7(4):193–195, 1978. [4, 15]

[DS05] Zeev Dvir and Amir Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. In *STOC*, pages 592–601, 2005. [5, 41]

[DSY08] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. In *STOC*, pages 741–748, 2008. [4]

[FS12] Michael Forbes and Amir Shpilka. Quasipolynomial-time Identity Testing of Non-Commutative and Read-Once Oblivious Algebraic Branching Programs. *Technical Report TR12-115, (ECCC)*, 2012. [5, 82]

[GK98] Dima Grigoriev and Marek Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In *STOC*, pages 577–582, 1998. [i, 3, 67]

[GKPS11] Bruno Grenet, Pascal Koiran, Natacha Portier, and Yann Strozecki. The Limited Power of Powering: Polynomial Identity Testing and a Depth-four Lower Bound for the Permanent. *Proceedings of the 31st Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, Arxiv preprint arXiv:1107.1434, 2011. [43, 44, 45]

[GKQ12] Ankit Gupta, Neeraj Kayal, and Youming Qiao. Random arithmetic formulas can be reconstructed efficiently. Technical report, Electronic Colloquium on Computational Complexity (ECCC), 2012. [70]

[GR05] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. In *FOCS*, pages 407–418, 2005. [48]

[HS80] J. Heintz and C. P. Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the twelfth annual ACM symposium on Theory of computing*, STOC '80, pages 262–272, 1980. [4]

[IW97] Russell Impagliazzo and Avi Wigderson. P = BPP if E Requires Exponential Circuits: Derandomizing the XOR Lemma. In *STOC*, pages 220–229, 1997. [43]

[JS82] Mark Jerrum and Marc Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29(3):874–897, 1982. [i, 3]

[Kal85a] Kyriakos Kalorkoti. A Lower Bound for the Formula Size of Rational Functions. *SIAM J. Comput.*, 14(3):678–687, 1985. [3]

[Kal85b] Kyriakos Kalorkoti. A Lower Bound for the Formula Size of Rational Functions. *SIAM J. Comput.*, 14(3):678–687, 1985. [44]

[Kay10] Neeraj Kayal. Algorithms for Arithmetic Circuits. *Technical Report TR10-073, (ECCC)*, 2010. [ii, 5, 8, 20, 21, 22, 23]

[Kay12a] Neeraj Kayal. Affine projections of polynomials. In *STOC*, pages 643–662, 2012. [70]

[Kay12b] Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. Technical report, Electronic Colloquium on Computational Complexity (ECCC), 2012. [70]

[KI03] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *STOC*, pages 355–364, 2003. [i, 4, 43]

[KMSV10] Zohar Shay Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. Deterministic identity testing of depth-4 multilinear circuits with bounded top fan-in. In *STOC*, pages 649–658, 2010. [42, 45]

[Koi10] Pascal Koiran. Arithmetic circuits: the chasm at depth four gets wider. *CoRR*, abs/1006.4700, 2010. [ii, iii, iv, 7, 10, 68, 69]

[KS01] Adam Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *STOC*, pages 216–223, 2001. [ii, 4, 5, 16, 41]

[KS07] Neeraj Kayal and Nitin Saxena. Polynomial Identity Testing for Depth 3 Circuits. *Computational Complexity*, 16(2):115–138, 2007. [ii, 5, 8, 20, 21, 26, 27, 28, 29, 41, 82]

[KS08] Zohar Shay Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. In *IEEE Conference on Computational Complexity*, pages 280–291, 2008. [41]

[KS09a] Zohar Shay Karnin and Amir Shpilka. Reconstruction of Generalized Depth-3 Arithmetic Circuits with Bounded Top Fan-in. In *IEEE Conference on Computational Complexity*, pages 274–285, 2009. [82]

[KS09b] Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth-3 circuits. In *FOCS*, 2009. [5, 41]

[LFKN90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic Methods for Interactive Proof Systems. In *FOCS*, pages 2–10, 1990. [4]

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *FCT*, pages 565–574, 1979. [4]

[LV98] Daniel Lewin and Salil P. Vadhan. Checking Polynomial Identities over any Field: Towards a Derandomization? In *STOC*, pages 438–447, 1998. [4]

[MR04] T. Mignon and N. Ressayre. A quadratic bound for the determinant and permanent problem. *International Mathematics Research Notices*, 79:4241–4253, 2004. [ii, 3]

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching Is as Easy as Matrix Inversion. In *STOC*, pages 345–354, 1987. [4]

[Nar86] H. Narasimhan. The irreducibility of ladder determinantal varieties. *Journal of Algebra*, 102:162–185, 1986. [80]

[NW97] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. [i, 3, 67, 70]

[Oxl92] James G. Oxley. *Matroid theory*. Oxford University Press, 1992. [43]

[PSZ00] Ramamohan Paturi, Michael E. Saks, and Francis Zane. Exponential lower bounds for depth three Boolean circuits. *Computational Complexity*, 9(1):1–15, 2000. [43]

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *STOC*, pages 245–254, 2008. [43]

[Raz09] Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2), 2009. [i, 3, 68]

[Rom00] Dan Romik. Stirling's Approximation for $n!$: The Ultimate Short Proof? *The American Mathematical Monthly*, 107(6):556–557, 2000. [76]

[RY08] R. Raz and A. Yehudayoff. Lower bounds and separations for constant depth mutilinear circuits. In *Proceedings of the 23rd IEEE Annual Conference on Computational Complexity*, pages 128–139, 2008. [68]

[Sax08] Nitin Saxena. Diagonal Circuit Identity Testing and Lower Bounds. In *ICALP (1)*, pages 60–71, 2008. [ii, 5, 8, 20, 21]

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009. [6]

[Sch80] Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM*, 27(4):701–717, 1980. [4, 15]

[Sha90] Adi Shamir. IP=PSPACE. In *FOCS*, pages 11–15, 1990. [4]

[SS09] Nitin Saxena and C. Seshadhri. An Almost Optimal Rank Bound for Depth-3 Identities. In *IEEE Conference on Computational Complexity*, pages 137–148, 2009. [5, 41]

[SS10a] Nitin Saxena and C. Seshadhri. From Sylvester-Gallai Configurations to Rank Bounds: Improved Black-box Identity Test for Depth-3 Circuits. *Technical Report TR10-013, (ECCC)*, 2010. [5]

[SS10b] Nitin Saxena and C. Seshadhri. From Sylvester-Gallai Configurations to Rank Bounds: Improved Black-Box Identity Test for Depth-3 Circuits. In *FOCS*, pages 21–29, 2010. [41]

[SS11] Nitin Saxena and C. Seshadhri. Blackbox identity testing for bounded top fanin depth-3 circuits: the field doesn't matter. In *STOC*, pages 431–440, 2011. [ii, 5, 41]

[Stu90] Bernd Sturmfels. Gröbner bases and stanley decompositions of determinantal rings. *Mathematische Zeitschrift*, 209:137–144, 1990. [80]

[SV08] Amir Shpilka and Ilya Volkovich. Read-once polynomial identity testing. In *STOC*, pages 507–516, 2008. [42]

[SV09] Amir Shpilka and Ilya Volkovich. Improved Polynomial Identity Testing for Read-Once Formulas. In *APPROX-RANDOM*, pages 700–713, 2009. [42]

[SV11] Shubhangi Saraf and Ilya Volkovich. Black-box identity testing of depth-4 multilinear circuits. In *STOC*, pages 421–430, 2011. [ii, 6, 41, 42, 45]

[SW01] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001. [3, 68]

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010. [6, 42, 82]

[Uma03] Christopher Umans. Pseudo-random generators for all hardnesses. *J. Comput. Syst. Sci.*, 67(2):419–440, 2003. [43]

[Val79] Leslie G. Valiant. Completeness Classes in Algebra. In *STOC*, pages 249–261, 1979. [i, 2, 67]

[VSBR83] Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM J. Comput.*, 12(4):641–644, 1983. [41]

[vzG83] Joachim von zur Gathen. Factoring Sparse Multivariate Polynomials. In *FOCS*, pages 172–179, 1983. [ii, 8, 19]

[Wil11] Ryan Williams. Non-uniform ACC Circuit Lower Bounds. In *IEEE Conference on Computational Complexity*, pages 115–125, 2011. [43]

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. *EUROSAM*, pages 216–226, 1979. [4, 15]