SAND2011-3726C

Blackbox identity testing for bounded top fanin depth-3 circuits: the field doesn't matter

C. Seshadhri (Sandia National Labs, Livermore)

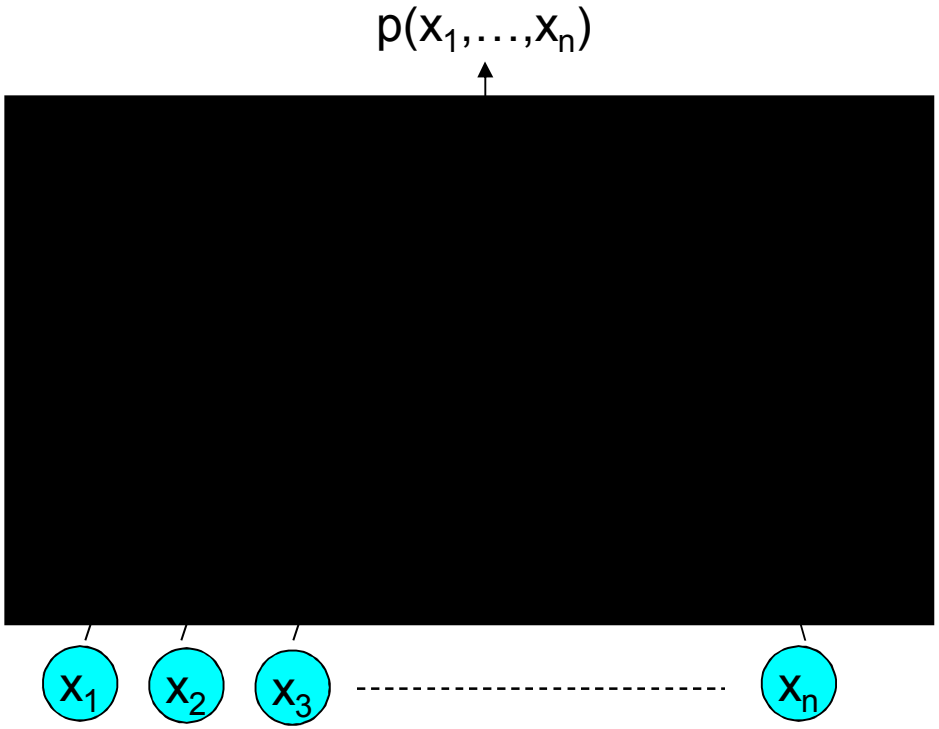
Joint work with

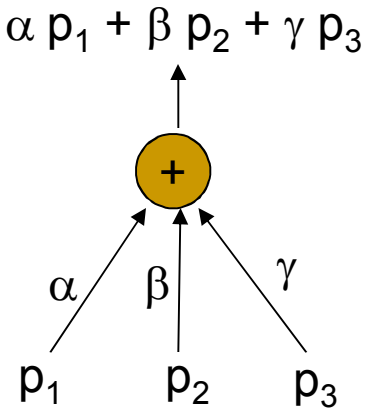
Nitin Saxena (Hausdorff Center for Mathematics)

The problem of PIT

- Polynomial identity testing: given a polynomial $p(x_1,x_2,\ldots,x_n)$ over F, is it identically zero?
  - *All* coefficients of $p(x_1,\ldots,x_n)$ are zero.
  - $(x+y)^2 - x^2 - y^2 - 2xy$ is identically zero.
  - So is: $(a^2+b^2+c^2+d^2)(A^2+B^2+C^2+D^2)$
    $- (aA+bB+cC+dD)^2 - (aB-bA+cD-dC)^2$
    $- (aC-bD-cA+dB)^2 - (aD-dA+bC-cB)^2$
  - $x(x-1)$ is NOT identically zero over F_2.

Circuits: Blackbox or not





We want algorithm whose running time is polynomial in size of the circuit (that includes # var, degree)

- Non blackbox: can analyze structure of C
- Blackbox: cannot C
  - Feed values and see what you get

A simple, randomized test

If output is 0, we guess it is identity.

Otherwise, we know it isn't.

- [Schwartz80, Zippel79] This is a randomized blackbox poly-time algorithm.
- Big big open problem: Find a deterministic polynomial time algorithm.
  - We would really like a black box algorithm
  - Base field Q is often of special interest

Why?

- Oh come on, it's an interesting mathematical problem. Do you need a reason?
- [IK04, Agr05] Derandomization implies circuit lower bounds

- [AKS] $(x + a)^n = x^n + a \pmod n$
- [L, MVV] Bipartite matching in NC?...
- Many more

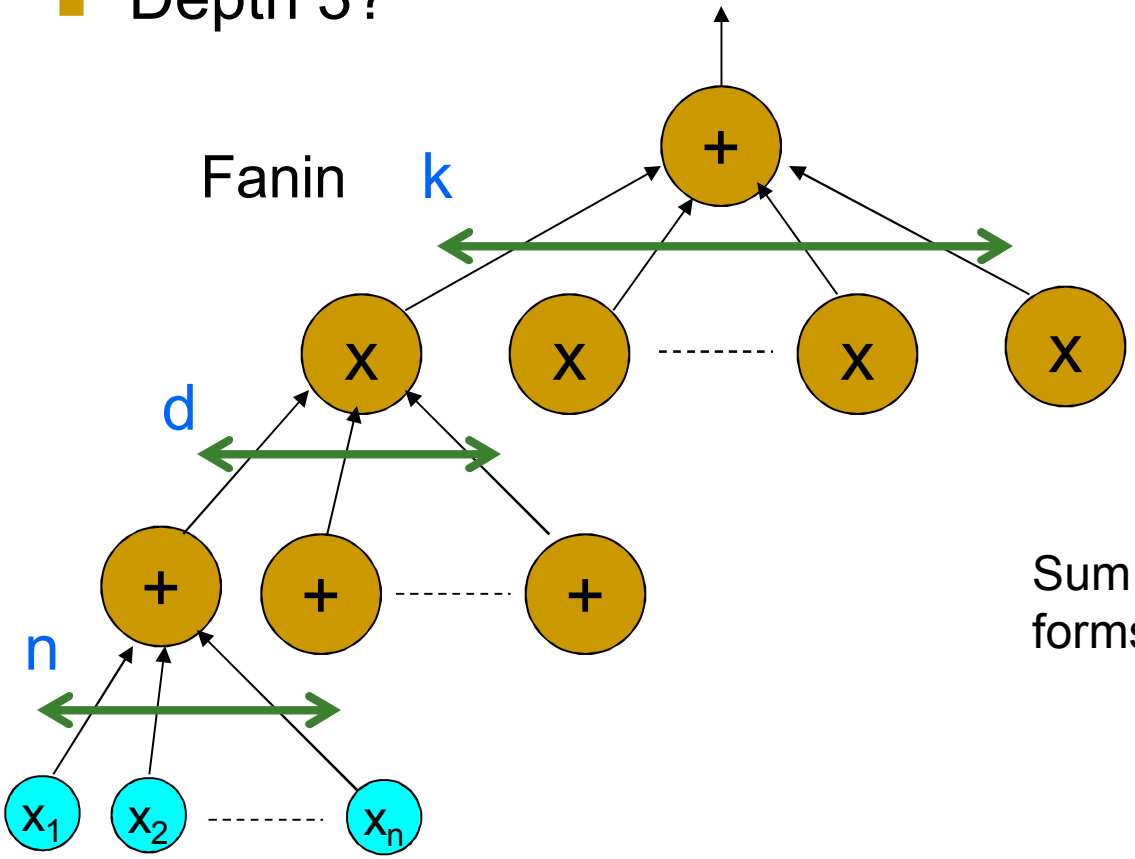
What do we do?

George Pólya

If you can't solve a problem, then there is an easier problem you *can* solve. Find it.

Get shallow results

- Let's restrict the depth and see what we get
- Depth 2? Non-blackbox trivial!
  - [GK, BOT,…,KS] Polytime with blackbox
- Depth 3?



$$C \equiv \sum_{i=1}^{k} \prod_{j=1}^{d} L_{ij} = \sum_{i=1}^{k} T_i$$

Sum of products of kd linear forms in n variables

Some examples

- Over Q

$$(x+z)(y+z) - xy - z(x+y+z) = 0$$

$$x_1x_2x_3(2y + x_1 + x_2 + x_3) - (y + x_1)(y + x_2)(y + x_3)(y + x_1 + x_2 + x_3)$$
$$+y(y + x_1 + x_2)(y + x_2 + x_3)(y + x_1 + x_3) = 0$$

- Over F_2 [Kayal Saxena 05]

$$\prod_{\sum_i b_i = 0} (b_1x_1 + b_2x_2 + b_3x_3) + \prod_{\sum_i b_i = 1} (y + b_1x_1 + b_2x_2 + b_3x_3)$$
$$+ \prod_{\sum_i b_i = 0} (y + b_1x_1 + b_2x_2 + b_3x_3) = 0$$

Some good news

- [Agrawal Vinay 08] Chasm at Depth 4!
- If you can solve blackbox PIT for depth 4, then you've solved it for all depths.

Our results

- A new black-box algorithm for depth-3 PIT
  - Parameters n, d, k (think of k as constant)

$$C \equiv \sum_{i=1}^{k} \prod_{j=1}^{d} L_{ij}$$

- For any field F: in time poly(nd^k), we can generate a hitting set S of tuples in F^n
  - For every non-identity C, there is tuple α in S, st $C(\alpha) \neq 0$
  - Even for F = F_2 and k=3, no poly-time black box algorithm known

Previously…

- So what's the best black-box running time
  - Parameters n, d, k (think of k as constant)

$$C \equiv \sum_{i=1}^{k} \prod_{j=1}^{d} L_{ij}$$

Who	What
[Karnin Shpilka 08] [Dvir Shpilka 06]	$\mathrm{poly}(n) d^{(\log d)^k}$
[Saxena S 09]	$\mathrm{poly}(n) d^{k^3 \log d}$
[Kayal Saraf 09] (over Q)	$\mathrm{poly}(n) d^{k^k}$
[Saxena S 10] (over Q)	$\mathrm{poly}(n) d^{k^2}$

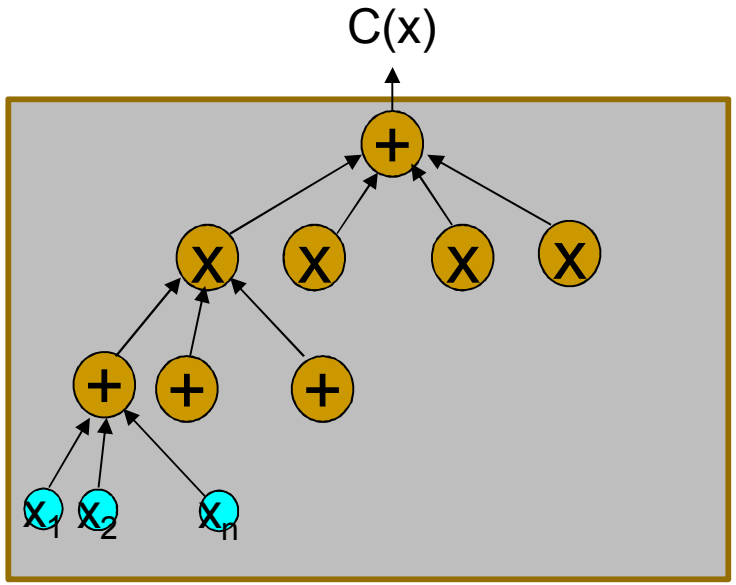
- Matches [Kayal Saxena 05] non-blackbox test of $\mathrm{poly}(n) d^k$

Proudly kitchen sink!

- [Dvir Shpilka 05] Rank: a very important concept related to depth-3
- Every previous blackbox result on depth-3 uses rank
- [Kayal Saraf 09] How rank is intimately tied to geometry
- Beautiful, but restrictive
  - Doesn't work too well for finite fields – doesn't yield poly-time algorithms over F_2

- We employ kitchen sink approach
  - Chinese-remaindering ideas from [Kayal Saxena 05] developed further in [Saxena S 10]
  - Use extractor tools from [Karnin Shpilka 08] ([Gabizon Raz 05])

Cutting down the variables



n

[y1 y2 y3 … yk] X

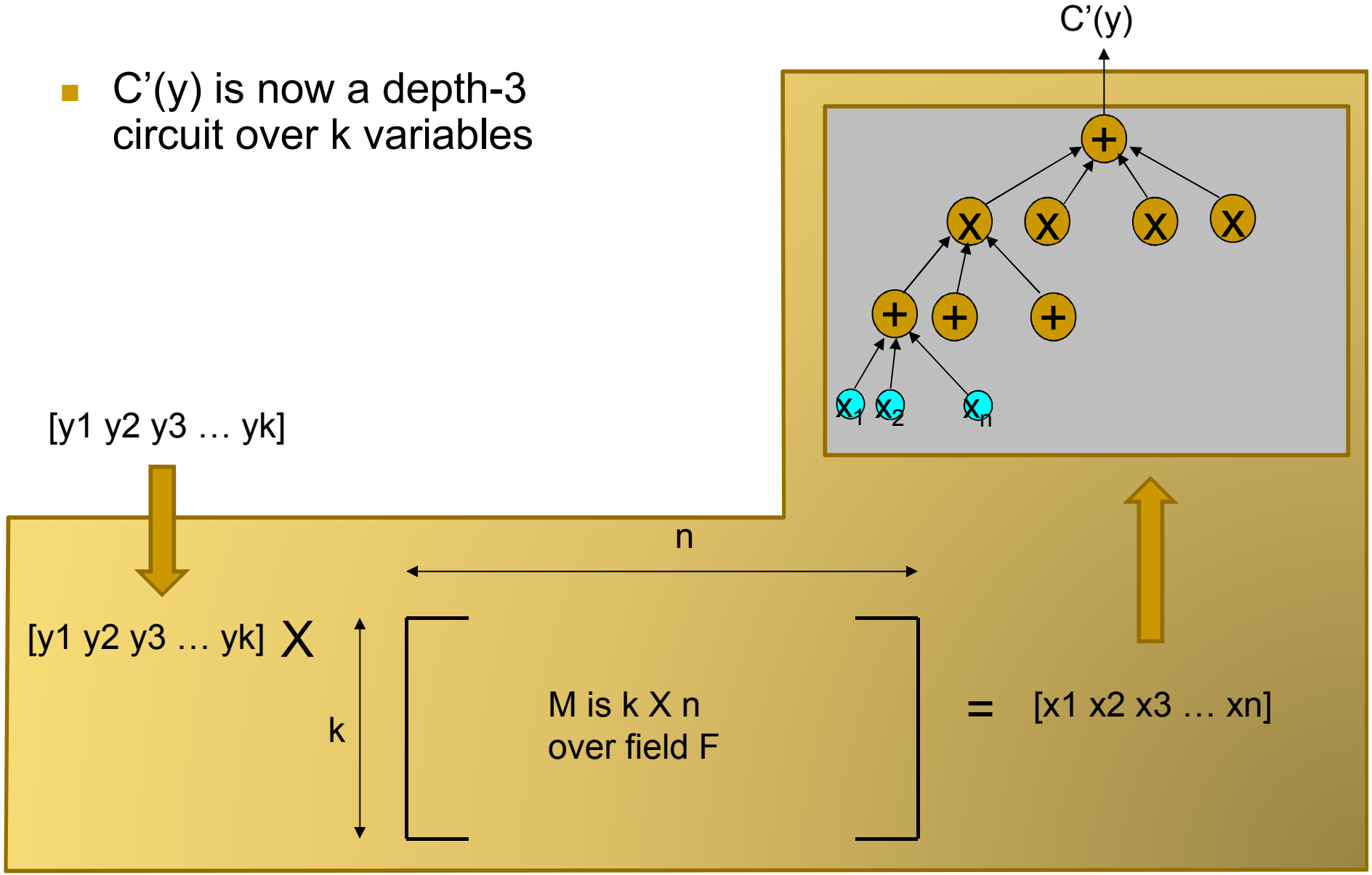
k

M is k X n
over field F

= [x1 x2 x3 … xn]

Cutting down the variables

- C’(y) is now a depth-3 circuit over k variables



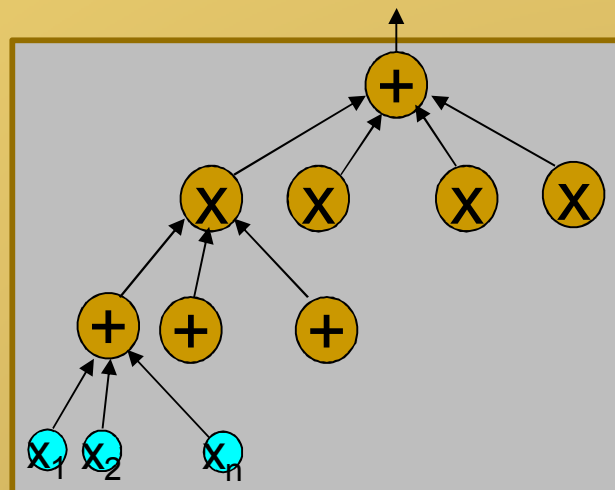
Cutting down the variables

- Theorem: Given n,d,k, in deterministic poly(ndk) time, we can construct M such that

C(x) is identity ⟺ C'(y) is identity

Technically, a set of Ms

C'(y)



[y1 y2 y3 … yk]

n

[y1 y2 y3 … yk] X

k

M is k X n
over field F

= [x1 x2 x3 … xn]

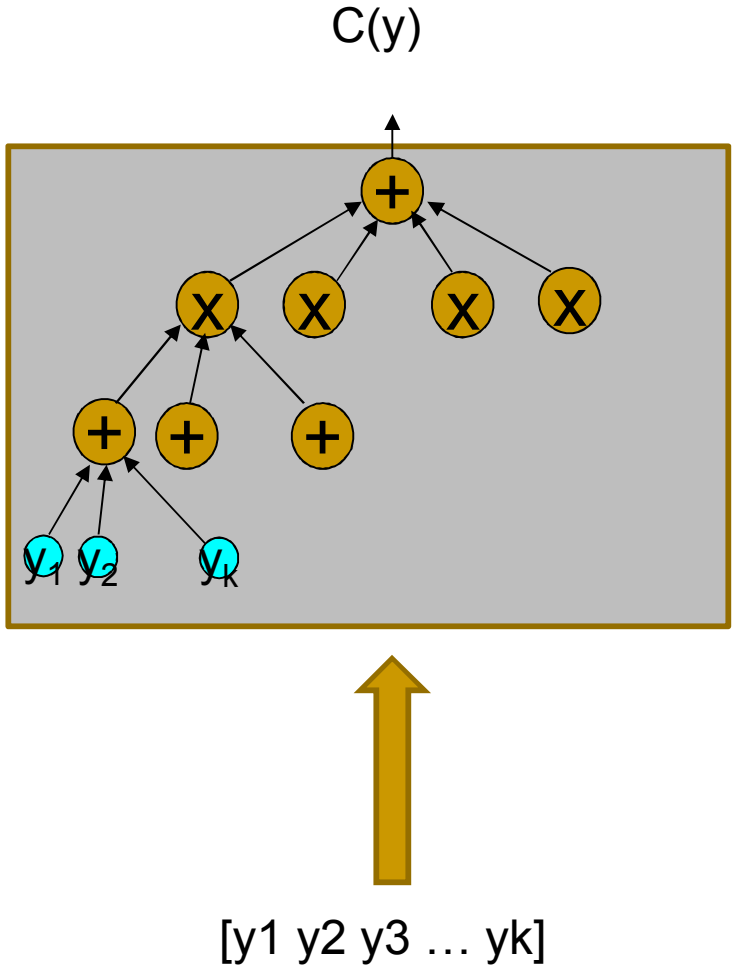
A dose of Schwartz-Zippel

- We have reduced general depth-3 blackbox PIT to PIT over depth-3 circuits with k variables

- Schwartz-Zippel Lemma: Let |T| > d be a subset of F, and f(y1,…,yk) be polynomial of degree d.

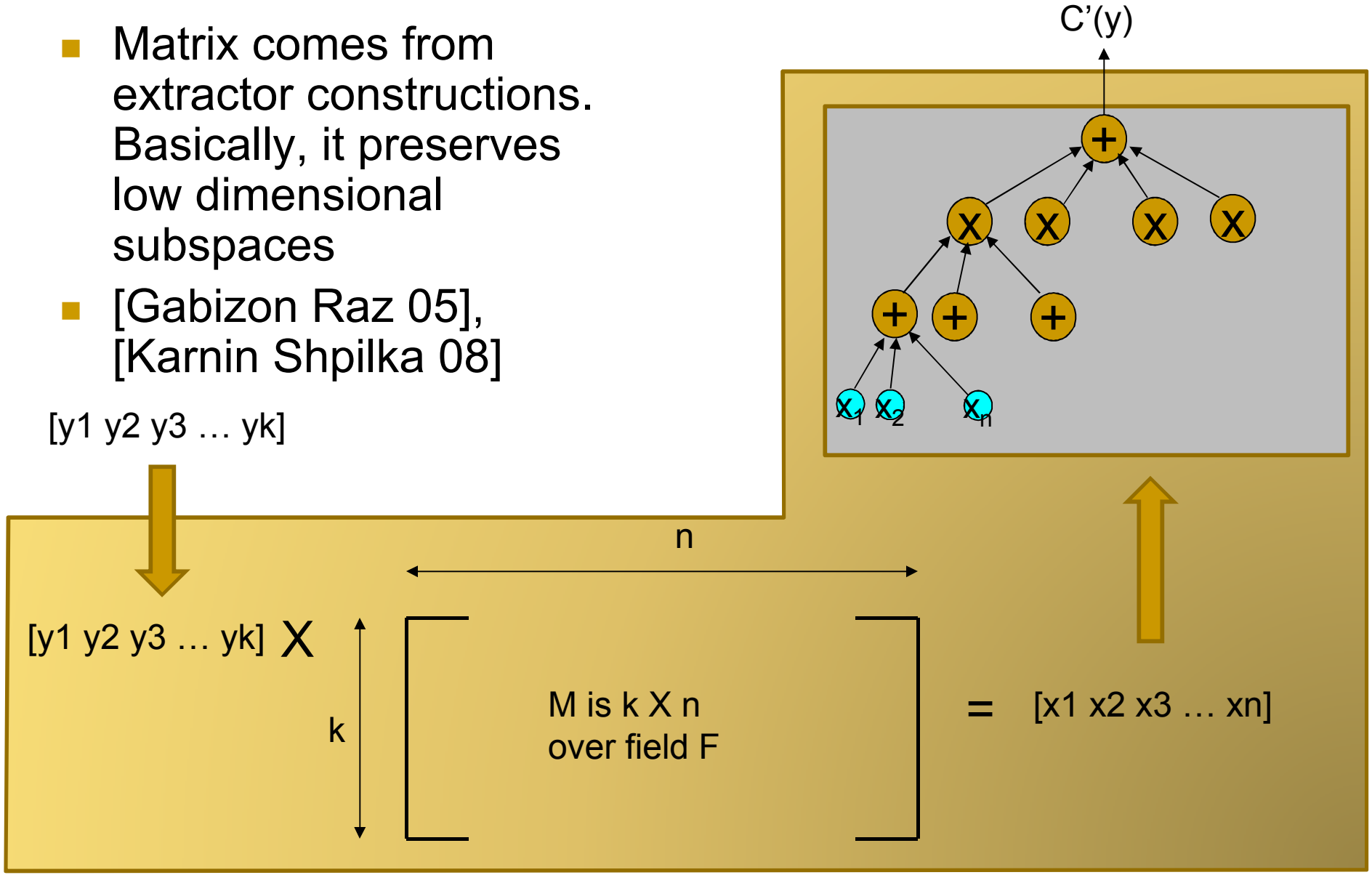
  For some y in T^k, f(y) ≠ 0

- So we get hitting set for depth-3 identities.



What's the matrix?

- Matrix comes from extractor constructions. Basically, it preserves low dimensional subspaces
- [Gabizon Raz 05], [Karnin Shpilka 08]



The Vandermonde transformation

$$C \equiv \sum_{i=1}^{k} \prod_{j=1}^{d} L_{i,j}$$

V = n × k $\left[\beta^{ij} \right]$

$$C' \equiv \sum_{i=1}^{k} \prod_{j=1}^{d} L'_{i,j}$$

$$L_{i,j} = \sum_{r=1}^{n} a_r x_r$$

[a1 a2 a3 … an] V = [b1 b2 b3 … bk]

$$b_r = \sum_{s=1}^{n} a_s \beta^{sr}$$

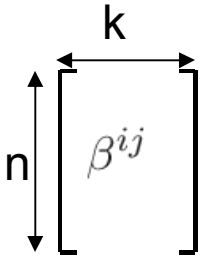
$$L'_{i,j} = \sum_{r=1}^{k} b_r y_r$$

- We are trying to convert C (n-variate) to C' (k-variate)
- [GR05] If $L_1, L_2, \ldots, L_k$ are linearly independent, then $L_1V, L_2V, \ldots, L_kV$ are linearly independent
  - The rank preserving property of this transformation
  - Preserves linear structure of subspaces of dim at most k

The Vandermonde transformation

$$C \equiv \sum_{i=1}^{k} \prod_{j=1}^{d} L_{i,j}$$

$$L_{i,j} = \sum_{r=1}^{n} a_r x_r$$

V = 

[a1 a2 a3 … an] V = [b1 b2 b3 … bk]

$$b_r = \sum_{s=1}^{n} a_s \beta^{sr}$$

$$C' \equiv \sum_{i=1}^{k} \prod_{j=1}^{d} L'_{i,j}$$

$$L'_{i,j} = \sum_{r=1}^{k} b_r y_r$$

- For setting values, observe that transpose of V is the desired M

$$x_i \rightarrow \sum_{j=1}^{k} \beta^{ij} y_j$$

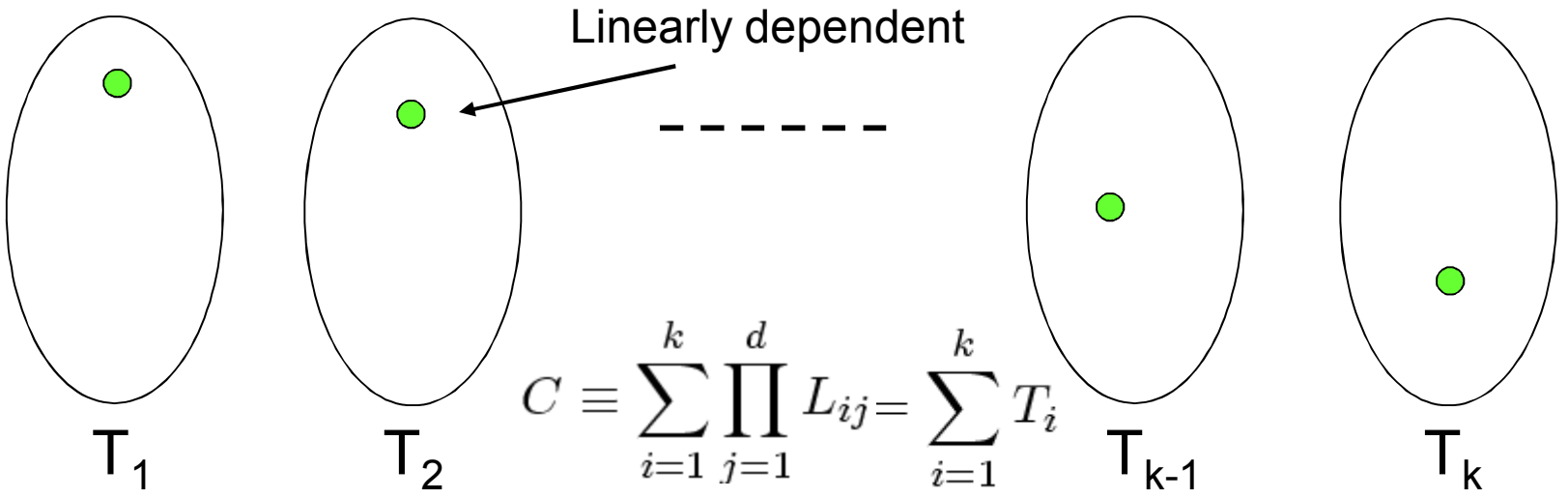
- Call this Ψ(C) = C': want to argue that Ψ(C) is identity iff C is identity

Linear dependencies

$$\prod L_i + \prod M_j + \prod N_k = 0$$

$$\prod M_j = -\prod N_k \pmod{L_1}$$

$$\longrightarrow \quad M_j = \alpha N_k + \beta L_i$$



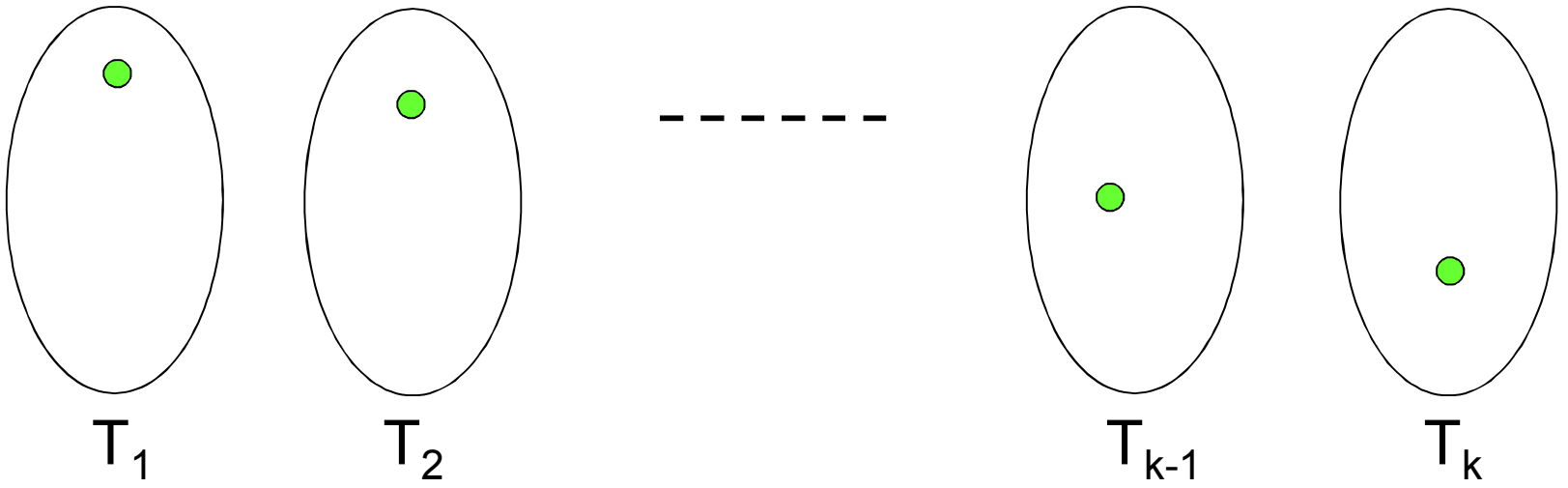
- For all L_1, L_2,…L_{k-1}, there exists L_k in linear span on (L_1, L_2,…L_{k-1}

The converse

- [Kayal Saxena 05], [Saxena S 10] (A Chinese Remainder Theorem for depth-3)

  C ≠ 0 iff there exists L_1, $L_2 \ldots L_{k-1}$, s.t. for all L_k, L_k is not in the linear span of (L_1, $L_2 \ldots$,L_{k-1})

- So there exist low rank certificates of non-identitiness



The converse

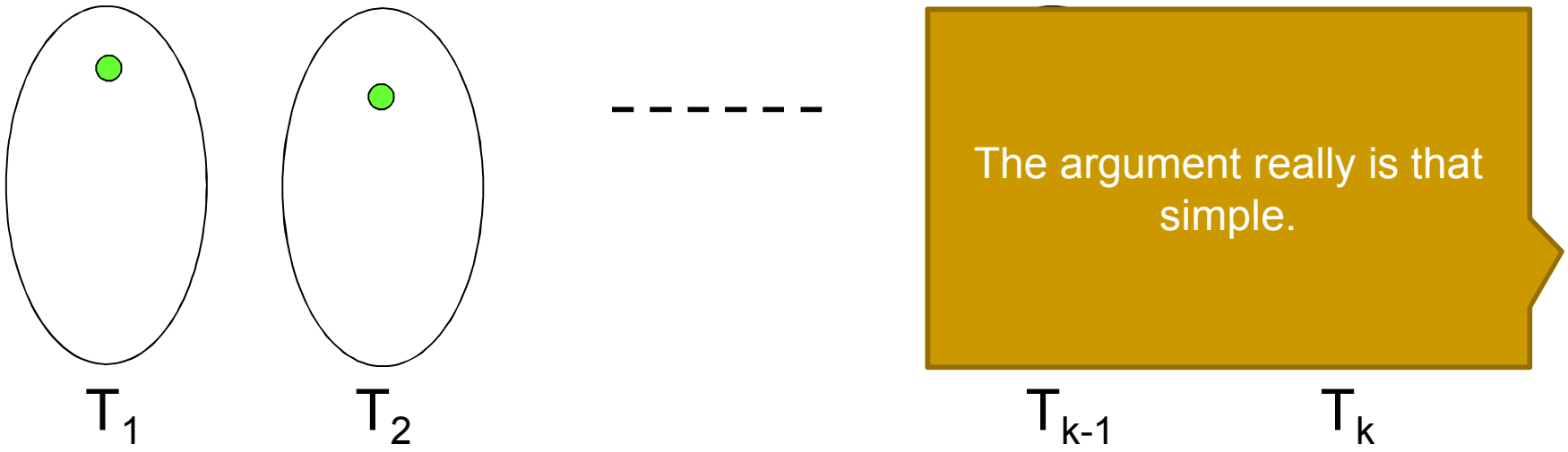
- [Kayal Saxena 05], [Saxena S 10] (A Chinese Remainder Theorem for depth-3)

  C ≠ 0 iff there exists L_1, $L_2 \ldots L_{k-1}$, s.t. for all L_k, L_k is not in the linear span of (L_1, $L_2 \ldots, L_{k-1}$)

- By rank preserving property:

  In $\Psi(C)$, there exist $\Psi(L_1)$, $\Psi(L_2) \ldots \Psi(L_{k-1})$, s.t. for all $\Psi(L_k)$, it is not in span of ($\Psi(L_1)$, $\Psi(L_2) \ldots \Psi(L_{k-1})$)

  So $\Psi(C) \neq 0$



So…

- Most of what I said is technically false, but is morally correct
  - So you get the basic idea
- That closes the gap between whitebox and blackbox testing for depth-3 PIT
- Running time is nd^k for any field
- Transformation to k-variate PIT is truly polynomial, and is an important tool
  - We can now shift focus to low degree depth-3 PIT

The road ahead

- Umm…solve identity testing
  - Surely, something intermediate…?
- Get truly polynomial (black-box or otherwise) for depth 3
  - How to remove exponential dependence on k?
- Any more results along this line of work?
  - No, I think we milked this one out

It's not o-k

- This is a very deep issue – the exponential dependence on k appears in almost all results

- Why? Because all results reduce the identity to k-variate or k-sparse.
- And then they just do a brute-force search

- We don't really understand what it means when the sum of k products cancel out (esp. when k is large)

Thank you!