Derandomization from algebraic hardness

Treading the borders

Zeyu Guo
IIT Kanpur ⟶ U. Haifa

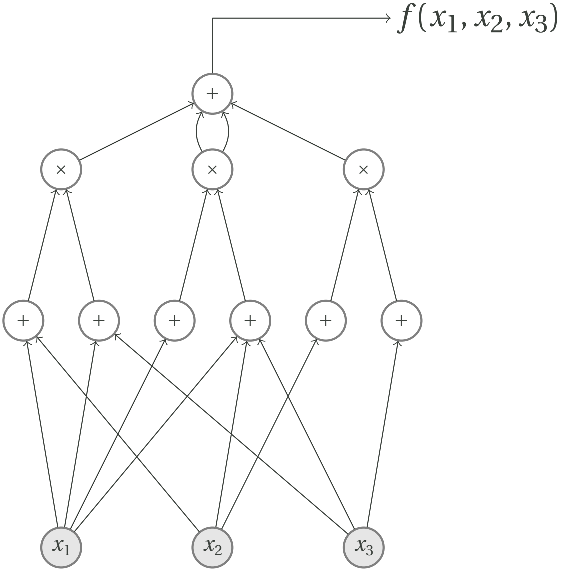
Mrinal Kumar
U. Toronto ⟶ IITB

Ramprasad Saptharishi
TIFR, Mumbai

Noam Solomon
Harvard University

IIT Bombay
June 2019

Algebraic Circuits



Aren't we all tired of this picture?

Two Important Questions

Two Important Questions

- **Lower Bounds:** Can we find an explicit family of polynomials $\{P_n\}$ that require large circuits?

Two Important Questions

- **Lower Bounds:** Can we find an explicit family of polynomials $\{P_n\}$ that require large circuits?

- **Polynomial Identity Testing:** Given a circuit C, can we check if C is computing the *zero* polynomial (deterministically)?

Two Important Questions

- **Lower Bounds:** Can we find an explicit family of polynomials $\{P_n\}$ that require large circuits?

- **Polynomial Identity Testing:** Given a circuit C, can we check if C is computing the *zero* polynomial (deterministically)?
  - **Hitting sets:** Find a set of points H such that any "small" circuit C that is computing a nonzero polynomial *must* satisfy $C(\mathbf{a}) \neq 0$ for some $\mathbf{a} \in H$.

Two Important Questions

- **Lower Bounds:** Can we find an explicit family of polynomials $\{P_n\}$ that require large circuits?

- **Polynomial Identity Testing:** Given a circuit C, can we check if C is computing the *zero* polynomial (deterministically)?

  - **Hitting sets:** Find a set of points H such that any "small" circuit C that is computing a nonzero polynomial *must* satisfy $C(\mathbf{a}) \neq 0$ for some $\mathbf{a} \in H$.

These two problems are intimately connected to each other.

A "trivial" hitting set

Lemma ([Ore, Demillo-Lipton, Schwartz, Zippel])

If $P(x_1, \ldots, x_n)$ is a nonzero polynomial of degree d, and $S \subseteq \mathbb{F}$ of size at least $d+1$, then $P(\mathbf{a}) \neq 0$ for some $\mathbf{a} \in S^n$.

A “trivial” hitting set

Lemma ([Ore, Demillo-Lipton, Schwartz, Zippel])

If $P(x_1, \ldots, x_n)$ is a nonzero polynomial of degree d, and $S \subseteq \mathbb{F}$ of size at least $d+1$, then $P(\mathbf{a}) \neq 0$ for some $\mathbf{a} \in S^n$.

We have an explicit hitting set of size $(d+1)^n$ for $\mathscr{C}(n, d, *)$.

A "trivial" hitting set

Lemma ([Ore, Demillo-Lipton, Schwartz, Zippel])

If $P(x_1, \ldots, x_n)$ is a nonzero polynomial of degree d, and $S \subseteq \mathbb{F}$ of size at least $d+1$, then $P(\mathbf{a}) \neq 0$ for some $\mathbf{a} \in S^n$.

We have an explicit hitting set of size $(d+1)^n$ for $\mathscr{C}(n, d, *)$.

Q: Are there smaller hitting sets for $\mathscr{C}(n, d, s)$?

A “trivial” hitting set

Lemma ([Ore, Demillo-Lipton, Schwartz, Zippel])

If $P(x_1, \ldots, x_n)$ is a nonzero polynomial of degree d, and $S \subseteq \mathbb{F}$ of size at least $d+1$, then $P(\mathbf{a}) \neq 0$ for some $\mathbf{a} \in S^n$.

We have an explicit hitting set of size $(d+1)^n$ for $\mathscr{C}(n, d, *)$.

Q: Are there smaller hitting sets for $\mathscr{C}(n, d, s)$?
A: Yes; almost any set of size $O(s^2)$ will work.

A “trivial” hitting set

Lemma ([Ore, Demillo-Lipton, Schwartz, Zippel])

If $P(x_1, \ldots, x_n)$ is a nonzero polynomial of degree d, and $S \subseteq \mathbb{F}$ of size at least $d+1$, then $P(\mathbf{a}) \neq 0$ for some $\mathbf{a} \in S^n$.

We have an explicit hitting set of size $(d+1)^n$ for $\mathscr{C}(n, d, *)$.

Q: Are there smaller hitting sets for $\mathscr{C}(n, d, s)$?
A: Yes; almost any set of size $O(s^2)$ will work.

Q: Can you give just one explicit example?

A "trivial" hitting set

Lemma ([Ore, Demillo-Lipton, Schwartz, Zippel])

If $P(x_1, \ldots, x_n)$ is a nonzero polynomial of degree d, and $S \subseteq \mathbb{F}$ of size at least $d+1$, then $P(\mathbf{a}) \neq 0$ for some $\mathbf{a} \in S^n$.

We have an explicit hitting set of size $(d+1)^n$ for $\mathscr{C}(n, d, *)$.

Q: Are there smaller hitting sets for $\mathscr{C}(n, d, s)$?
A: Yes; almost any set of size $O(s^2)$ will work.

Q: Can you give just one explicit example?
A: Umm...

Pseudorandom objects

"How difficult could it be to find hay in a haystack?"

— Howard Karloff

Pseudorandom objects

"How difficult could it be to find hay in a haystack?"

— Howard Karloff

- You care a lot about hay.

Pseudorandom objects

"How difficult could it be to find hay in a haystack?"

— Howard Karloff

- You care a lot about hay.
- Almost everything in a haystack is hay.

Pseudorandom objects

"How difficult could it be to find hay in a haystack?"

— Howard Karloff

- You care a lot about hay.
- Almost everything in a haystack is hay.
- Find hay.

  (Why do we still keep finding needles all the time?)

Pseudorandom objects

"How difficult could it be to find hay in a haystack?"
— Howard Karloff

- You care a lot about hard polynomials.
- Almost every polynomial is a hard polynomial.
- Find a hard polynomial.
  (Why do we still keep finding needles all the time?)

Pseudorandom objects

"How difficult could it be to find hay in a haystack?"

— Howard Karloff

- You care a lot about hitting sets.
- Almost every set of poly-size is a hitting set.
- Find a hitting set.
  (Why do we still keep finding needles all the time?)

Pseudorandom objects

"How difficult could it be to find hay in a haystack?"

— Howard Karloff

- You care a lot about hay.
- Almost everything in a haystack is hay.
- Find hay.
  (Why do we still keep finding needles all the time?)

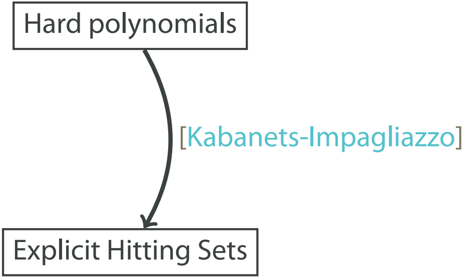
Question: Can we use one pseudorandom object to build another?

Lower bounds and hitting sets

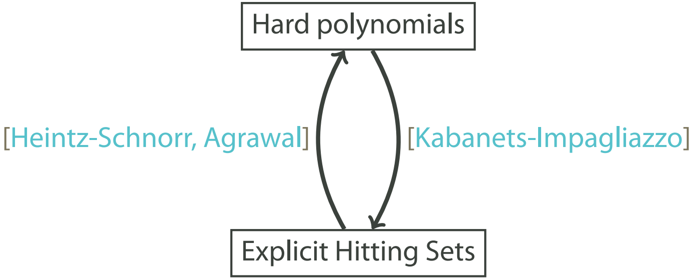
Hard polynomials

Explicit Hitting Sets

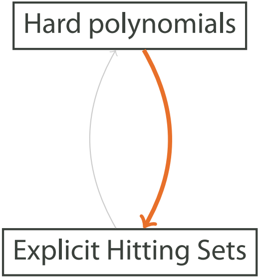
Lower bounds and hitting sets



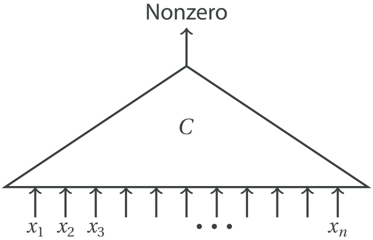
Lower bounds and hitting sets



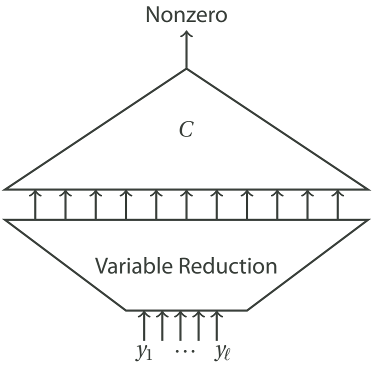
Lower bounds → hitting sets



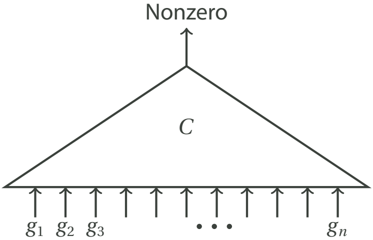
How are hitting sets constructed?



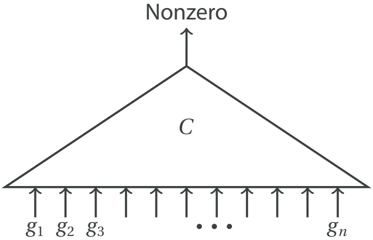
How are hitting sets constructed?



How are hitting sets constructed?



How are hitting sets constructed?

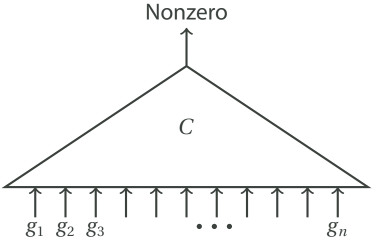


Definition (Generator)

A map $\mathcal{G} = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_\ell]^n$ is a hitting-set generator for a class $\mathscr{C}$ if

$$\forall\, C \in \mathscr{C} \quad , \quad C \neq 0 \Longleftrightarrow C \circ \mathcal{G} \neq 0.$$

How are hitting sets constructed?



Definition (Generator)

A map $\mathscr{G} = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_\ell]^n$ is a hitting-set generator for a class $\mathscr{C}$ if

$$\forall\, C \in \mathscr{C} \quad , \quad C \neq 0 \iff C \circ \mathscr{G} \neq 0.$$

The degree of the generator is $\max_i(\deg g_i)$. The stretch is $\ell \to n$.

How are hitting sets constructed?

Definition (Generator)

A map $\mathscr{G} = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_\ell]^n$ is a hitting-set generator for a class $\mathscr{C}$ if

$$\forall\, C \in \mathscr{C} \quad , \quad C \neq 0 \Longleftrightarrow C \circ \mathscr{G} \neq 0.$$

The degree of the generator is $\max_i(\deg g_i)$. The stretch is $\ell \to n$.

How are hitting sets constructed?

Definition (Generator)

A map $\mathcal{G} = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_\ell]^n$ is a hitting-set generator for a class $\mathcal{C}$ if

$$\forall\, C \in \mathcal{C} \quad , \quad C \neq 0 \iff C \circ \mathcal{G} \neq 0.$$

The degree of the generator is $\max_i(\deg g_i)$. The stretch is $\ell \to n$.

Lemma

Let $\mathcal{G} = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_\ell]^n$ be an explicit hitting-set generator for $\mathcal{C}(n, D, s)$ of degree d. Then, we have

- *An explicit hitting set H of size $(dD+1)^\ell$*

Generators assuming hardness

	Hardness assumption	Hitting set size

Generators assuming hardness

	Hardness assumption	Hitting set size
[Kabanets-Impagliazzo]	$\{p_n\}$ requires $n^{\omega(1)}$ size $\{p_n\}$ requires $2^{n^{\Omega(1)}}$ size $\{p_n\}$ requires $2^{\Omega(n)}$ size	$2^{s^{\varepsilon}}, \forall\, \varepsilon > 0$ $2^{\text{poly}\log s}$ $s^{O(\log s)}$

Generators assuming hardness

	Hardness assumption	Hitting set size
[Kabanets-Impagliazzo]	$\{p_n\}$ requires $n^{\omega(1)}$ size $\{p_n\}$ requires $2^{n^{\Omega(1)}}$ size $\{p_n\}$ requires $2^{\Omega(n)}$ size	$2^{s^\varepsilon}, \forall\, \varepsilon > 0$ $2^{\mathrm{poly}\log s}$ $s^{O(\log s)}$
[Kumar-S-Tengse]	$\{p_{k,d}\}_d$ requires $d^{\Omega(1)}$ size	$s^{\exp(\exp(\log^* s))}$

Generators assuming hardness

	Hardness assumption	Hitting set size
[Kabanets-Impagliazzo]	$\{p_n\}$ requires $n^{\omega(1)}$ size $\{p_n\}$ requires $2^{n^{\Omega(1)}}$ size $\{p_n\}$ requires $2^{\Omega(n)}$ size	$2^{s^{\varepsilon}}, \forall\, \varepsilon > 0$ $2^{\text{poly}\log s}$ $s^{O(\log s)}$
[Kumar-S-Tengse]	$\{p_{k,d}\}_d$ requires $d^{\Omega(1)}$ size	$s^{\exp(\exp(\log^* s))}$
	???	$s^{O(1)}$

Generators assuming hardness

	Hardness assumption	Hitting set size
[Kabanets-Impagliazzo]	$\{p_n\}$ requires $n^{\omega(1)}$ size $\{p_n\}$ requires $2^{n^{\Omega(1)}}$ size $\{p_n\}$ requires $2^{\Omega(n)}$ size	$2^{s^{\varepsilon}}, \forall \varepsilon > 0$ $2^{\mathrm{poly}\log s}$ $s^{O(\log s)}$
[Kumar-S-Tengse]	$\{p_{k,d}\}_d$ requires $d^{\Omega(1)}$ size	$s^{\exp(\exp(\log^* s))}$
This work	$\{p_{k,d}\}_d$ requires $d^{3+\varepsilon}$ size	$s^{O(1)}$

Generators assuming hardness

	Hardness assumption	Hitting set size
[Kabanets-Impagliazzo]	$\{p_n\}$ requires $n^{\omega(1)}$ size $\{p_n\}$ requires $2^{n^{\Omega(1)}}$ size $\{p_n\}$ requires $2^{\Omega(n)}$ size	$2^{s^{\varepsilon}}, \forall\, \varepsilon > 0$ $2^{\mathrm{poly}\log s}$ $s^{O(\log s)}$
[Kumar-S-Tengse]	$\{p_{k,d}\}_d$ requires $d^{\Omega(1)}$ size	$s^{\exp(\exp(\log^* s))}$
This work	$\{p_{k,d}\}_d$ requires $d^{3+\varepsilon}$ size $\{p_{k,d}\}_d$ requires $d^{1+\varepsilon}$ $\overline{\text{size}}$	$s^{O(1)}$ $s^{O(1)}$

Our results

Main Theorem

Theorem ([Guo-Kumar-S-Solomon])

For any k-variate polynomial P of degree d,

Main Theorem

Theorem ([Guo-Kumar-S-Solomon])

For any k-variate polynomial P of degree d, there is an explicit map

$$\mathcal{G}_P = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_k, z_1, \ldots, z_k]^n$$

such that

Main Theorem

Theorem ([Guo-Kumar-S-Solomon])

For any k-variate polynomial P of degree d, there is an explicit map

$$\mathcal{G}_P = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_k, z_1, \ldots, z_k]^n$$

such that

- $\deg(\mathcal{G}_P) = d$ *and* $\mathcal{G}_P$ *is* $d^{O(k)}$*-explicit,*

Main Theorem

Theorem ([Guo-Kumar-S-Solomon])
For any k-variate polynomial P of degree d, there is an explicit map

$$\mathcal{G}_P = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_k, z_1, \ldots, z_k]^n$$

such that

- $\deg(\mathcal{G}_P) = d$ *and* $\mathcal{G}_P$ *is* $d^{O(k)}$*-explicit,*
- *For any nonzero circuit* $C \in \mathscr{C}(n, D, s)$*,*

$$\text{if } C \circ \mathcal{G}_P = 0 \quad , \quad \text{then } \operatorname{size}(P) \ll d^k$$

Main Theorem

Theorem ([Guo-Kumar-S-Solomon])

For any k-variate polynomial P of degree d, there is an explicit map

$$\mathcal{G}_P = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_k, z_1, \ldots, z_k]^n$$

such that

- $\deg(\mathcal{G}_P) = d$ *and* $\mathcal{G}_P$ *is* $d^{O(k)}$*-explicit,*
- *For any nonzero circuit* $C \in \mathscr{C}(n, D, s)$,

$$\textit{if } C \circ \mathcal{G}_P = 0 \quad , \quad \textit{then } \operatorname{size}(P) \le n^{10k} \cdot s \cdot d^3 \cdot D$$

Main Theorem

Theorem ([Guo-Kumar-S-Solomon])

For any k-variate polynomial P of degree d, there is an explicit map

$$\mathcal{G}_P = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_k, z_1, \ldots, z_k]^n$$

such that

- $\deg(\mathcal{G}_P) = d$ *and* $\mathcal{G}_P$ *is* $d^{O(k)}$*-explicit,*
- *For any nonzero circuit* $C \in \mathscr{C}(n, D, s)$*,*

$$\text{if } C \circ \mathcal{G}_P = 0 \quad , \quad \text{then } \operatorname{size}(P) \leq n^{10k} \cdot s \cdot d^3 \cdot D \ll d^k$$

(Think of $d = n^{1000}$)

Main Theorem

Theorem ([Guo-Kumar-S-Solomon])
For any k-variate polynomial P of degree d, there is an explicit map

$$\mathcal{G}_P = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_k, z_1, \ldots, z_k]^n$$

such that

- $\deg(\mathcal{G}_P) = d$ *and* $\mathcal{G}_P$ *is* $d^{O(k)}$*-explicit,*
- *For any nonzero circuit* $C \in \mathscr{C}(n, D, s)$,

$$\text{if } C \circ \mathcal{G}_P = 0 \quad , \quad \text{then } \operatorname{size}(P) \leq n^{10k} \cdot s \cdot d^3 \cdot D \ll d^k$$

(*Think of* $d = n^{1000}$)

In other words, if P *is hard enough, then* $\mathcal{G}_P$ *is a hitting-set generator for* $\mathscr{C}(n, D, s)$.

Main Theorem

Theorem ([Guo-Kumar-S-Solomon])
For any k-variate polynomial P of degree d, there is an explicit map

$$\mathcal{G}_P = (g_1, \ldots, g_n) \in \mathbb{F}[y_1, \ldots, y_k, z_1, \ldots, z_k]^n$$

such that

- $\deg(\mathcal{G}_P) = d$ *and* $\mathcal{G}_P$ *is* $d^{O(k)}$*-explicit,*
- *For any nonzero circuit* $C \in \mathscr{C}(n, D, s)$,

$$\text{if } C \circ \mathcal{G}_P = 0 \quad , \quad \text{then } \overline{\text{size}}(P) \leq n^{10k} \cdot s \cdot d \cdot D \ll d^k$$

(Think of $d = n^{1000}$)

In other words, if P is hard enough, then $\mathcal{G}_P$ is a hitting-set generator for $\mathscr{C}(n, D, s)$.

Some consequences

Corollary

Let k be a large enough constant and $\varepsilon > 0$. Suppose $\{P_{k,d}\}_d$ is an explicit family of polynomials with $\deg P_{k,d} = d$ such that $\{P_{k,d}\}_d$ requires size $d^{3+\varepsilon}$ (or $\overline{\text{size}}$ $d^{1+\varepsilon}$).

Then, there is an explicit hitting set for $\mathscr{C}(s, s, s)$ of size $\mathrm{poly}(s)$.

Some consequences

Corollary

Let k be a large enough constant and $\varepsilon > 0$. Suppose $\{P_{k,d}\}_d$ is an explicit family of polynomials with $\deg P_{k,d} = d$ such that $\{P_{k,d}\}_d$ requires size $d^{3+\varepsilon}$ (or $\overline{\text{size}}$ $d^{1+\varepsilon}$).

Then, there is an explicit hitting set for $\mathscr{C}(s,s,s)$ of size $\text{poly}(s)$.

Proof.

Set $d \geq s^{(10k+2)/\varepsilon}$ and $P = P_{k,d}$.

□

Some consequences

Corollary

Let k be a large enough constant and $\varepsilon > 0$. Suppose $\{P_{k,d}\}_d$ is an explicit family of polynomials with $\deg P_{k,d} = d$ such that $\{P_{k,d}\}_d$ requires size $d^{3+\varepsilon}$ (or $\overline{\text{size}}$ $d^{1+\varepsilon}$).

Then, there is an explicit hitting set for $\mathscr{C}(s, s, s)$ of size $\mathrm{poly}(s)$.

Proof.

Set $d \geq s^{(10k+2)/\varepsilon}$ and $P = P_{k,d}$.
If $0 \neq C \in \mathscr{C}(s, s, s)$ such that $C \circ \mathscr{G}_P = 0$,

□

Some consequences

Corollary

Let k be a large enough constant and $\varepsilon > 0$. Suppose $\{P_{k,d}\}_d$ is an explicit family of polynomials with $\deg P_{k,d} = d$ such that $\{P_{k,d}\}_d$ requires size $d^{3+\varepsilon}$ (or $\overline{\text{size}}$ $d^{1+\varepsilon}$).

Then, there is an explicit hitting set for $\mathscr{C}(s,s,s)$ of size $\text{poly}(s)$.

Proof.

Set $d \geq s^{(10k+2)/\varepsilon}$ and $P = P_{k,d}$.
If $0 \neq C \in \mathscr{C}(s,s,s)$ such that $C \circ \mathscr{G}_P = 0$, then

$$\text{size}(P) \leq s^{10k} \cdot s^2 \cdot d^3$$

□

Some consequences

Corollary

Let k be a large enough constant and $\varepsilon > 0$. Suppose $\{P_{k,d}\}_d$ is an explicit family of polynomials with $\deg P_{k,d} = d$ such that $\{P_{k,d}\}_d$ requires size $d^{3+\varepsilon}$ (or $\overline{\text{size}}$ $d^{1+\varepsilon}$).

Then, there is an explicit hitting set for $\mathscr{C}(s,s,s)$ of size $\operatorname{poly}(s)$.

Proof.

Set $d \geq s^{(10k+2)/\varepsilon}$ and $P = P_{k,d}$.

If $0 \neq C \in \mathscr{C}(s,s,s)$ such that $C \circ \mathscr{G}_P = 0$, then

$$\begin{aligned}\operatorname{size}(P) &\leq s^{10k} \cdot s^2 \cdot d^3 \\ &\leq d^{3+\varepsilon}\end{aligned}$$

□

Some consequences

Corollary

Let k be a large enough constant and $\varepsilon > 0$. Suppose $\{P_{k,d}\}_d$ is an explicit family of polynomials with $\deg P_{k,d} = d$ such that $\{P_{k,d}\}_d$ requires size $d^{3+\varepsilon}$ (or $\overline{\text{size}}$ $d^{1+\varepsilon}$).

Then, there is an explicit hitting set for $\mathscr{C}(s,s,s)$ of size $\text{poly}(s)$.

Proof.

Set $d \geq s^{(10k+2)/\varepsilon}$ and $P = P_{k,d}$.
If $0 \neq C \in \mathscr{C}(s,s,s)$ such that $C \circ \mathscr{G}_P = 0$, then

$$\begin{aligned}\text{size}(P) &\leq s^{10k} \cdot s^2 \cdot d^3 \\ &\leq d^{3+\varepsilon} \text{ which is impossible.}\end{aligned}$$

□

Some consequences

Corollary

Let k be a large enough constant and $\varepsilon > 0$. Suppose $\{P_{k,d}\}_d$ is an explicit family of polynomials with $\deg P_{k,d} = d$ such that $\{P_{k,d}\}_d$ requires size $d^{3+\varepsilon}$ (or $\overline{\text{size}}$ $d^{1+\varepsilon}$).

Then, there is an explicit hitting set for $\mathscr{C}(s,s,s)$ of size $\text{poly}(s)$.

Proof.

Set $d \geq s^{(10k+2)/\varepsilon}$ and $P = P_{k,d}$.
If $0 \neq C \in \mathscr{C}(s,s,s)$ such that $C \circ \mathscr{G}_P = 0$, then

$$\begin{aligned}\text{size}(P) &\leq s^{10k} \cdot s^2 \cdot d^3 \\ &\leq d^{3+\varepsilon} \text{ which is impossible.}\end{aligned}$$

Hence $C \circ \mathscr{G}_P$ is a nonzero $2k$-variate polynomial of degree at most ds. □

Some consequences

Corollary

Let k be a large enough constant and $\varepsilon > 0$. Suppose $\{P_{k,d}\}_d$ is an explicit family of polynomials with $\deg P_{k,d} = d$ such that $\{P_{k,d}\}_d$ requires size $d^{3+\varepsilon}$ (or size $d^{1+\varepsilon}$).

Then, there is an explicit hitting set for $\mathscr{C}(s,s,s)$ of size $\mathrm{poly}(s)$.

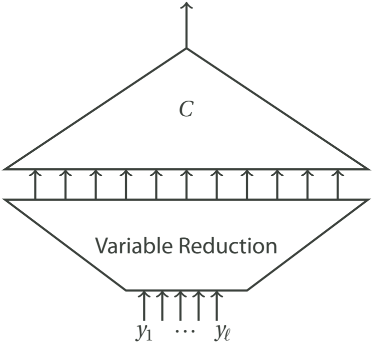
Proof.

Set $d \geq s^{(10k+2)/\varepsilon}$ and $P = P_{k,d}$.
If $0 \neq C \in \mathscr{C}(s,s,s)$ such that $C \circ \mathscr{G}_P = 0$, then

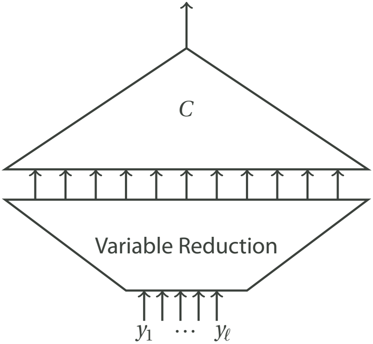
$$\mathrm{size}(P) \leq s^{10k} \cdot s^2 \cdot d^3$$
$$\leq d^{3+\varepsilon} \text{ which is impossible.}$$

Hence $C \circ \mathscr{G}_P$ is a nonzero $2k$-variate polynomial of degree at most ds. Hence, we have a hitting set of size $(ds)^{2k} = s^{O(k^2/\varepsilon)}$. □

Revisiting variable reductions

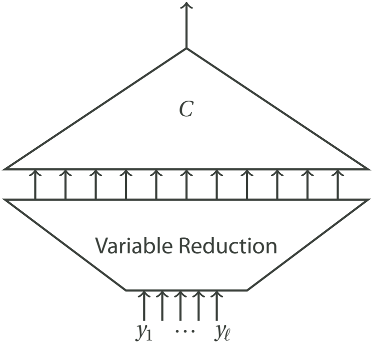


Revisiting variable reductions



Hitting-set Generator: $C \neq 0 \iff C \circ \mathscr{G} \neq 0$

Revisiting variable reductions



Hitting-set Generator: $C \neq 0 \iff C \circ \mathscr{G} \neq 0$

Dream: $\text{size}(C \circ \mathscr{G}) \approx \text{size}(C) + \text{size}(\mathscr{G})$

The Kronecker Map

The Kronecker Map

$$\mathscr{K} = \left(1, y, y^2, y^4, \ldots, y^{2^{n-1}}\right)$$

The Kronecker Map

$$\mathscr{K} = \left(1, y, y^2, y^4, \ldots, y^{2^{n-1}}\right)$$

$$x_1^{e_1} \cdots x_n^{e_n} \quad \longmapsto \quad y^{[e_1 e_2 \cdots e_n]_2}$$

The Kronecker Map

$$\mathscr{K} = \left(1, y, y^2, y^4, \ldots, y^{2^{n-1}}\right)$$

$$x_1^{e_1} \cdots x_n^{e_n} \quad \longmapsto \quad y^{[e_1 e_2 \cdots e_n]_2}$$

If P is a n-variate multilinear polynomial, then
$P \circ \mathscr{K}$ is a univariate polynomial of degree at most 2^n.

The Kronecker Map

$$\mathcal{K}_t = \left(1, y_1, y_1^2, \ldots, y_1^{2^{m-1}}, \ldots, 1, y_t, \ldots, y_t^{2^{m-1}}\right) \quad (n = t\,m)$$

$$x_1^{e_1} \cdots x_n^{e_n} \quad \longmapsto \quad y_1^{[e_1 \cdots e_m]_2} \cdots y_t^{[e_* \cdots e_n]_2}$$

If P is a n-variate multilinear polynomial, then
$P \circ \mathcal{K}$ is a t-variate polynomial of degree at most $2^{n/t}$.

The Kronecker Map

$$\mathcal{K}_t = \left(1, y_1, y_1^2, \ldots, y_1^{2^{m-1}}, \ldots, 1, y_t, \ldots, y_t^{2^{m-1}}\right) \quad (n = tm)$$

$$x_1^{e_1} \cdots x_n^{e_n} \quad \longmapsto \quad y_1^{[e_1 \cdots e_m]_2} \cdots y_t^{[e_* \cdots e_n]_2}$$

If P is a n-variate multilinear polynomial, then
$P \circ \mathcal{K}$ is a t-variate polynomial of degree at most $2^{n/t}$.

[Kabanets-Impagliazzo]: If $\{P_n\}$, multilinear, with $\text{size}(P_n) > 2^{n/1000}$, then we have $s^{O(\log s)}$-sized hitting sets.

The Kronecker Map

$$\mathscr{K}_t = \left(1, y_1, y_1^2, \ldots, y_1^{2^{m-1}}, \ldots, 1, y_t, \ldots, y_t^{2^{m-1}}\right) \quad (n = tm)$$

$$x_1^{e_1} \cdots x_n^{e_n} \quad \longmapsto \quad y_1^{[e_1 \cdots e_m]_2} \cdots y_t^{[e_* \cdots e_n]_2}$$

If P is a n-variate multilinear polynomial, then
$P \circ \mathscr{K}$ is a t-variate polynomial of degree at most $2^{n/t}$.

[Kabanets-Impagliazzo]: If $\{P_n\}$, multilinear, with $\mathrm{size}(P_n) > 2^{n/1000}$, then we have $s^{O(\log s)}$-sized hitting sets.

New: If, for some constant t, suppose $\overline{\mathrm{size}}(P_n \circ \mathscr{K}_t) \geq 2^{(1+\varepsilon)n/t}$

The Kronecker Map

$$\mathscr{K}_t = \left(1, y_1, y_1^2, \ldots, y_1^{2^{m-1}}, \ldots, 1, y_t, \ldots, y_t^{2^{m-1}}\right) \quad (n = tm)$$

$$x_1^{e_1} \cdots x_n^{e_n} \quad \mapsto \quad y_1^{[e_1 \cdots e_m]_2} \cdots y_t^{[e_* \cdots e_n]_2}$$

If P is a n-variate multilinear polynomial, then
$P \circ \mathscr{K}$ is a t-variate polynomial of degree at most $2^{n/t}$.

[Kabanets-Impagliazzo]: If $\{P_n\}$, multilinear, with $\text{size}(P_n) > 2^{n/1000}$, then we have $s^{O(\log s)}$-sized hitting sets.

New: If, for some constant t, suppose $\overline{\text{size}}(P_n \circ \mathscr{K}_t) \geq 2^{(1+\varepsilon)n/t} = d^{1+\varepsilon}$

The Kronecker Map

$$\mathscr{K}_t = \left(1, y_1, y_1^2, \ldots, y_1^{2^{m-1}}, \ldots, 1, y_t, \ldots, y_t^{2^{m-1}}\right) \quad (n = tm)$$

$$x_1^{e_1} \cdots x_n^{e_n} \quad \longmapsto \quad y_1^{[e_1 \cdots e_m]_2} \cdots y_t^{[e_* \cdots e_n]_2}$$

If P is a n-variate multilinear polynomial, then
$P \circ \mathscr{K}$ is a t-variate polynomial of degree at most $2^{n/t}$.

[Kabanets-Impagliazzo]: If $\{P_n\}$, multilinear, with $\text{size}(P_n) > 2^{n/1000}$,
then we have $s^{O(\log s)}$-sized hitting sets.

New: If, for some constant t, suppose $\overline{\text{size}}(P_n \circ \mathscr{K}_t) \geq 2^{(1+\varepsilon)n/t} = d^{1+\varepsilon}$
then we have $\text{poly}(s)$-sized hitting sets.

Consequences for bootstrapping

Theorem. [**Kumar-S-Tengse**]
Let $\varepsilon > 0$ and k (large enough) be fixed constants.

If, for all $s \geq k$, we have explicit hitting sets for $\mathscr{C}(k, s, s)$ of size

$$s^{k-\varepsilon},$$

then, we have explicit hitting sets for $\mathscr{C}(s, s, s)$ of size

$$s^{\exp(\exp(\log^* s))}$$

Consequences for bootstrapping

Corollary

Let $\varepsilon > 0$ and k (large enough) be fixed constants.

If, for all $s \geq k$, we have explicit hitting sets for $\overline{\mathscr{C}}(k, s, s)$ of size

$$s^{k-\varepsilon},$$

then, we have explicit hitting sets for $\overline{\mathscr{C}}(s, s, s)$ of size

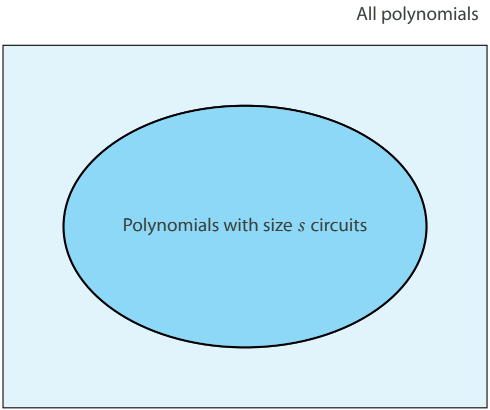
$$s^{O(1)}$$

Circuits and border are crucial for this.

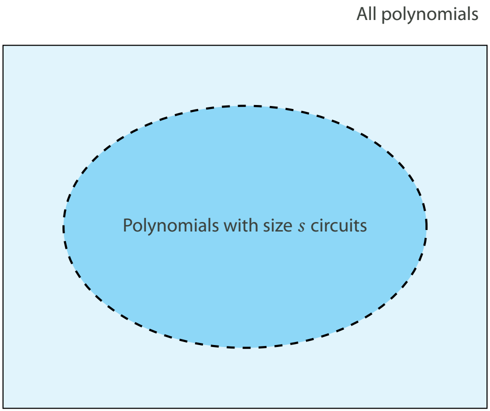
What’s all this border stuff?

The Border

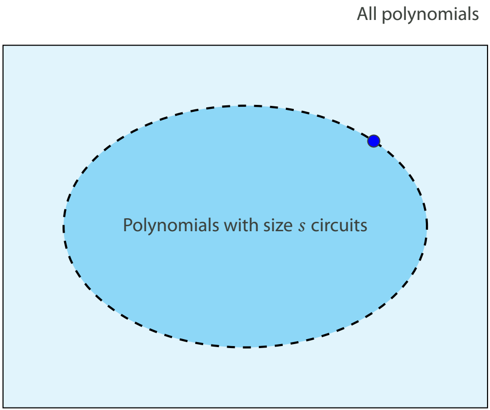
The Border



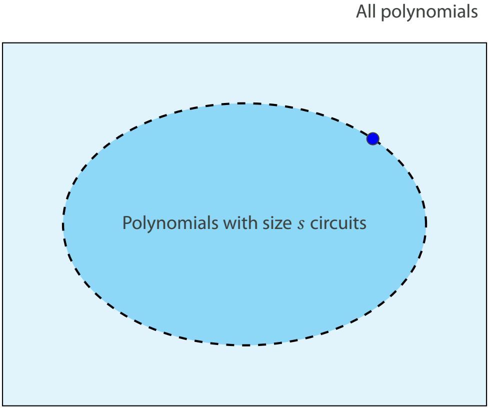
The Border



The Border



The Border



- Does not have size s circuits, but arbitrarily close to those that do.

Border computation: an example

$$\mathscr{C} = \left\{ f \ : \ f = \ell_1^d + \ell_2^d \ , \ \deg(\ell_1), \deg(\ell_2) = 1 \right\}$$

Border computation: an example

$$\mathscr{C} = \left\{ f \ : \ f = \ell_1^d + \ell_2^d \ , \ \deg(\ell_1), \deg(\ell_2) = 1 \right\}$$

Fact

If $x^{d-1} y = \ell_1^d + \cdots + \ell_s^d$, then $s \geq d$.

Border computation: an example

$$\mathscr{C} = \left\{ f \ : \ f = \ell_1^d + \ell_2^d \ , \ \deg(\ell_1), \deg(\ell_2) = 1 \right\}$$

Fact

If $x^{d-1}y = \ell_1^d + \cdots + \ell_s^d$, then $s \geq d$.
Hence, $x^{d-1}y \notin \mathscr{C}$ for any $d \geq 3$.

Border computation: an example

$$\mathscr{C} = \left\{ f \ : \ f = \ell_1^d + \ell_2^d \ , \ \deg(\ell_1), \deg(\ell_2) = 1 \right\}$$

Fact

If $x^{d-1}y = \ell_1^d + \cdots + \ell_s^d$, then $s \geq d$.
Hence, $x^{d-1}y \notin \mathscr{C}$ for any $d \geq 3$.

However,

$$C = \frac{(x + \varepsilon y)^d - x^d}{d \cdot \varepsilon}$$

Border computation: an example

$$\mathscr{C} = \left\{ f \;:\; f = \ell_1^d + \ell_2^d \;,\; \deg(\ell_1), \deg(\ell_2) = 1 \right\}$$

Fact

If $x^{d-1}y = \ell_1^d + \cdots + \ell_s^d$, then $s \geq d$.
Hence, $x^{d-1}y \notin \mathscr{C}$ for any $d \geq 3$.

However,

$$C = \frac{(x + \varepsilon y)^d - x^d}{d \cdot \varepsilon} = x^{d-1}y + O(\varepsilon)$$

Border computation: an example

$$\mathscr{C} = \left\{ f \ : \ f = \ell_1^d + \ell_2^d \ , \ \deg(\ell_1), \deg(\ell_2) = 1 \right\}$$

Fact

If $x^{d-1} y = \ell_1^d + \cdots + \ell_s^d$, then $s \geq d$.
Hence, $x^{d-1} y \notin \mathscr{C}$ for any $d \geq 3$.

However,

$$C = \frac{(x + \varepsilon y)^d - x^d}{d \cdot \varepsilon} = x^{d-1} y + O(\varepsilon) \xrightarrow{\varepsilon \to 0} x^{d-1} y$$

Border computation: an example

$$\mathscr{C} = \left\{ f \ : \ f = \ell_1^d + \ell_2^d \ , \ \deg(\ell_1), \deg(\ell_2) = 1 \right\}$$

Fact

If $x^{d-1}y = \ell_1^d + \cdots + \ell_s^d$, then $s \geq d$.
Hence, $x^{d-1}y \notin \mathscr{C}$ for any $d \geq 3$.

However,

$$C = \frac{(x+\varepsilon y)^d - x^d}{d \cdot \varepsilon} = x^{d-1}y + O(\varepsilon) \xrightarrow{\varepsilon \to 0} x^{d-1}y$$

Hence, $x^{d-1}y \in \overline{\mathscr{C}}$ but not in $\mathscr{C}$.

The one trick that we will need

The one trick that we will need

Task: Given a circuit C of size s computing a polynomial P of degree d. Compute P_d, the degree d homogeneous part of P.

The one trick that we will need

Task: Given a circuit C of size s computing a polynomial P of degree d. Compute P_d, the degree d homogeneous part of P.

Standard solution: "Homogenize" the circuit and extract the degree d part. Can be done using a circuit of size $O(sd^2)$.

The one trick that we will need

Task: Given a circuit C of size s computing a polynomial P of degree d. Compute P_d, the degree d homogeneous part of P.

Standard solution: "Homogenize" the circuit and extract the degree d part. Can be done using a circuit of size $O(sd^2)$.

Border trick:

$$C(x_1, \ldots, x_n) = P_0 + P_1 + \cdots + P_d$$

The one trick that we will need

Task: Given a circuit C of size s computing a polynomial P of degree d. Compute P_d, the degree d homogeneous part of P.

Standard solution: "Homogenize" the circuit and extract the degree d part. Can be done using a circuit of size $O(sd^2)$.

Border trick:

$$C\left(\frac{x_1}{\varepsilon}, \ldots, \frac{x_n}{\varepsilon}\right) = P_0 + \frac{P_1}{\varepsilon} + \cdots + \frac{P_d}{\varepsilon^d}$$

The one trick that we will need

Task: Given a circuit C of size s computing a polynomial P of degree d. Compute P_d, the degree d homogeneous part of P.

Standard solution: "Homogenize" the circuit and extract the degree d part. Can be done using a circuit of size $O(sd^2)$.

Border trick:

$$\varepsilon^d \cdot C\left(\frac{x_1}{\varepsilon}, \ldots, \frac{x_n}{\varepsilon}\right) = \varepsilon^d P_0 + \varepsilon^{d-1} P_1 + \cdots + P_d$$

The one trick that we will need

Task: Given a circuit C of size s computing a polynomial P of degree d. Compute P_d, the degree d homogeneous part of P.

Standard solution: "Homogenize" the circuit and extract the degree d part. Can be done using a circuit of size $O(sd^2)$.

Border trick:

$$\varepsilon^d \cdot C\left(\frac{x_1}{\varepsilon}, \ldots, \frac{x_n}{\varepsilon}\right) = \varepsilon^d P_0 + \varepsilon^{d-1} P_1 + \cdots + P_d$$
$$\xrightarrow{\varepsilon \to 0} P_d$$

The one trick that we will need

Task: Given a circuit C of size s computing a polynomial P of degree d. Compute P_d, the degree d homogeneous part of P.

Standard solution: "Homogenize" the circuit and extract the degree d part. Can be done using a circuit of size $O(sd^2)$.

Border trick:

$$\varepsilon^d \cdot C\left(\frac{x_1}{\varepsilon}, \ldots, \frac{x_n}{\varepsilon}\right) = \varepsilon^d P_0 + \varepsilon^{d-1} P_1 + \cdots + P_d$$

$$\xrightarrow{\varepsilon \to 0} P_d$$

$$\therefore \quad \overline{\text{size}}(P_d) \leq \overline{\text{size}}(P)$$

The one trick that we will need

Task: Given a circuit C of size s computing a polynomial P of degree d. Compute P_d, the degree d homogeneous part of P.

Standard solution: "Homogenize" the circuit and extract the degree d part. Can be done using a circuit of size $O(sd^2)$.

Border trick:

$$\varepsilon^d \cdot C\left(\frac{x_1}{\varepsilon}, \ldots, \frac{x_n}{\varepsilon}\right) = \varepsilon^d P_0 + \varepsilon^{d-1} P_1 + \cdots + P_d$$
$$\xrightarrow{\varepsilon \to 0} P_d$$
$$\therefore \quad \overline{\text{size}}(P_d) \leq \overline{\text{size}}(P)$$

P_d can be $\overline{\text{computed}}$ in size s as well!

\begin{proof}

Designing generators

Any sufficiently advanced

technology

is indistinguishable

from magic

Designing generators

Any sufficiently hard polynomial's evaluations

on disjoint inputs

is indistinguishable, for a small circuit,

from random inputs

Designing generators

Any sufficiently hard polynomial's evaluations

on disjoint inputs

is indistinguishable, for a small circuit,

from random inputs

$$\mathscr{G} : (\mathbf{y}_1, \ldots, \mathbf{y}_k) \mapsto (\mathbf{y}_1, \ldots, \mathbf{y}_k, P(\mathbf{y}_1), \ldots, P(\mathbf{y}_k))$$

Designing generators

Any sufficiently hard polynomial's evaluations

on **"almost disjoint"** inputs

is indistinguishable, for a small circuit,

from random inputs

$$\mathcal{G} : (\mathbf{y}_1, \ldots, \mathbf{y}_k) \mapsto (\mathbf{y}_1, \ldots, \mathbf{y}_k, P(\mathbf{y}_1), \ldots, P(\mathbf{y}_k))$$

Designing generators

Any sufficiently hard polynomial's evaluations

on **"almost disjoint"** inputs

is indistinguishable, for a small circuit,

from random inputs

[KI, NW] : $\mathcal{G} : (y_1, \ldots, y_\ell) \mapsto \left(P(\mathbf{y}|_{S_1}), \ldots, P(\mathbf{y}|_{S_n})\right)$

Designing generators

Any sufficiently hard polynomial's components

'Taylored' appropriately

is indistinguishable, for a small circuit,

from random inputs

Description of our generator

$$P(z_1, \ldots, z_k)$$

Description of our generator

$$P(\mathbf{y}+\mathbf{z}) = P(\mathbf{z}) + \sum_{i} y_i \cdot (\partial_i P)(\mathbf{z}) + \sum_{i,j} y_i y_j \cdot (\partial_{i,j} P)(\mathbf{z}) + \cdots$$

Description of our generator

$$P(\mathbf{y}+\mathbf{z}) = P(\mathbf{z}) + \sum_i y_i \cdot (\partial_i P)(\mathbf{z}) + \sum_{i,j} y_i y_j \cdot (\partial_{i,j} P)(\mathbf{z}) + \cdots$$

$$= \sum_{\mathbf{e}} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z})$$

Description of our generator

$$P(\mathbf{y}+\mathbf{z}) = P(\mathbf{z}) + \sum_i y_i \cdot (\partial_i P)(\mathbf{z}) + \sum_{i,j} y_i y_j \cdot (\partial_{i,j} P)(\mathbf{z}) + \cdots$$

$$= \sum_{\mathbf{e}} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z})$$

Definition (The generator)

For a k-variate polynomial P, define

$$\Delta_i(P) = \sum_{\mathbf{e}:|\mathbf{e}|=i} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z}).$$

The generator $\mathscr{G}_P$ is defined as

$$\mathscr{G}_P = (\Delta_0(P), \Delta_1(P), \Delta_2(P), \ldots, \Delta_n(P)) \in (\mathbb{F}[\mathbf{y}_{[k]}, \mathbf{z}_{[k]}])^{n+1}.$$

Description of our generator

$$P(\mathbf{y}+\mathbf{z}) = P(\mathbf{z}) + \sum_i y_i \cdot (\partial_i P)(\mathbf{z}) + \sum_{i,j} y_i y_j \cdot (\partial_{i,j} P)(\mathbf{z}) + \cdots$$

$$= \sum_{\mathbf{e}} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z})$$

Definition (The generator)

For a k-variate polynomial P, define

$$\Delta_i(P) = \sum_{\mathbf{e}:|\mathbf{e}|=i} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z}).$$

The generator $\mathscr{G}_P$ is defined as

$$\mathscr{G}_P = (\Delta_0(P), \Delta_1(P), \Delta_2(P), \ldots, \Delta_n(P)) \in (\mathbb{F}[\mathbf{y}_{[k]}, \mathbf{z}_{[k]}])^{n+1}.$$

Description of our generator

$$P(\mathbf{y}+\mathbf{z}) = P(\mathbf{z}) + \sum_i y_i \cdot (\partial_i P)(\mathbf{z}) + \sum_{i,j} y_i y_j \cdot (\partial_{i,j} P)(\mathbf{z}) + \cdots$$

$$= \sum_{\mathbf{e}} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z})$$

Definition (The generator)

For a k-variate polynomial P, define

$$\Delta_i(P) = \sum_{\mathbf{e}:|\mathbf{e}|=i} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z}).$$

The generator $\mathscr{G}_P$ is defined as

$$\mathscr{G}_P = (\Delta_0(P), \Delta_1(P), \Delta_2(P), \ldots, \Delta_n(P)) \in (\mathbb{F}[\mathbf{y}_{[k]}, \mathbf{z}_{[k]}])^{n+1}.$$

Description of our generator

$$P(\mathbf{y}+\mathbf{z}) = P(\mathbf{z}) + \sum_i y_i \cdot (\partial_i P)(\mathbf{z}) + \sum_{i,j} y_i y_j \cdot (\partial_{i,j} P)(\mathbf{z}) + \cdots$$

$$= \sum_{\mathbf{e}} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z})$$

Definition (The generator)

For a k-variate polynomial P, define

$$\Delta_i(P) = \sum_{\mathbf{e}:|\mathbf{e}|=i} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z}).$$

The generator $\mathscr{G}_P$ is defined as

$$\mathscr{G}_P = (\Delta_0(P), \Delta_1(P), \Delta_2(P), \ldots, \Delta_n(P)) \in (\mathbb{F}[\mathbf{y}_{[k]}, \mathbf{z}_{[k]}])^{n+1}.$$

Proof overview

Proof overview

- Assume $C \neq 0$ is a small circuit such that $C \circ \mathscr{G}_P = 0$.

Proof overview

- Assume $C \neq 0$ is a small circuit such that $C \circ \mathscr{G}_P = 0$.

- Show that we can use C, and a little more, to get a circuit that computes P.

Proof overview

- Assume $C \neq 0$ is a small circuit such that $C \circ \mathscr{G}_P = 0$.

- Show that we can use C, and a little more, to get a circuit that computes P.

Idea: *Think of* $C(\Delta_0(P), \ldots, \Delta_n(P)) = 0$ *as a* *differential equation* *and solve for* P.

Cauchy-Kowalevski Equations

$$\left(\frac{1}{2}\right) m \cdot (v(t))^2 + m \cdot g \cdot h(t) = c$$

Cauchy-Kowalevski Equations

$$\left(\frac{1}{2}\right)m \cdot (v(t))^2 + m \cdot g \cdot h(t) = c$$

Cauchy-Kowalevski Equations

$$\left(\frac{1}{2}\right) m \cdot \left(\frac{\partial h}{\partial t}\right)^2 + m \cdot g \cdot h(t) = c$$

Cauchy-Kowalevski Equations

$$Q(h(t), h^{(1)}(t)) = 0$$

Cauchy-Kowalevski Equations

$$Q(h(t), h^{(1)}(t)) = 0$$

Solve for $h(t)$ as a power series in t.

Cauchy-Kowalevski Equations

$$Q(h(t), h^{(1)}(t)) = 0$$

Solve for $h(t)$ as a power series in t.

- Start with some non-degenerate initial conditions:

$$t = a_0 \quad ; \quad h(a_0) = \beta_0 \quad ; \quad h'(a_0) = \gamma_0$$

Cauchy-Kowalevski Equations

$$Q(h(t), h^{(1)}(t)) = 0$$

Solve for $h(t)$ as a power series in t.

- Start with some non-degenerate initial conditions:

$$t = a_0 \quad ; \quad h(a_0) = \beta_0 \quad ; \quad h'(a_0) = \gamma_0$$

which is a solution modulo $(t - t_0)$.

Cauchy-Kowalevski Equations

$$Q(h(t), h^{(1)}(t)) = 0$$

Solve for $h(t)$ as a power series in t.

- Start with some non-degenerate initial conditions:

$$t = a_0 \quad ; \quad h(a_0) = \beta_0 \quad ; \quad h'(a_0) = \gamma_0$$

  which is a solution modulo $(t - t_0)$.

- Lift to a solution modulo $(t - t_0)^2$, $(t - t_0)^3$ and so on...

Cauchy-Kowalevski Equations

$$Q(h(t), h^{(1)}(t)) = 0$$

Solve for $h(t)$ as a power series in t.

- Start with some non-degenerate initial conditions:

  $$t = a_0 \quad ; \quad h(a_0) = \beta_0 \quad ; \quad h'(a_0) = \gamma_0$$

  which is a solution modulo $(t - t_0)$.

- Lift to a solution modulo $(t - t_0)^2, (t - t_0)^3$ and so on...
  Newton Iterations

Our situation

$$Q(h(t), h^{(1)}(t)) = 0$$

Solve for $h(t)$ as a power series in t.

- Start with some non-degenerate initial conditions:

$$t = a_0 \quad ; \quad h(a_0) = \beta_0 \quad ; \quad h'(a_0) = \gamma_0$$

  which is a solution modulo $(t - t_0)$.

- Lift to a solution modulo $(t - t_0)^2, (t - t_0)^3$ and so on...
  Newton Iterations

Our situation

$$C(\Delta_0(P), \ldots, \Delta_n(P)) = 0$$

Solve for $h(t)$ as a power series in t.

- Start with some non-degenerate initial conditions:

  $$t = a_0 \quad ; \quad h(a_0) = \beta_0 \quad ; \quad h'(a_0) = \gamma_0$$

  which is a solution modulo $(t - t_0)$.

- Lift to a solution modulo $(t - t_0)^2$, $(t - t_0)^3$ and so on...
  Newton Iterations

Our situation

$$C(\Delta_0(P),\ldots,\Delta_n(P))=0$$

Solve for P as a power series in $\mathbf{z}$.

- Start with some non-degenerate initial conditions:

  $$t=a_0 \quad ; \quad h(a_0)=\beta_0 \quad ; \quad h'(a_0)=\gamma_0$$

  which is a solution modulo $(t-t_0)$.

- Lift to a solution modulo $(t-t_0)^2$, $(t-t_0)^3$ and so on...
  Newton Iterations

Our situation

$$C(\Delta_0(P), \ldots, \Delta_n(P)) = 0$$

Solve for P as a power series in $\mathbf{z}$.

- Start with some non-degenerate initial conditions:

$$C \circ \mathcal{G}_P = 0$$
$$(\partial_n C) \circ \mathcal{G}_P \neq 0.$$

- Lift to a solution modulo $(t - t_0)^2$, $(t - t_0)^3$ and so on...
  Newton Iterations

Our situation

$$C(\Delta_0(P), \ldots, \Delta_n(P)) = 0$$

Solve for P as a power series in $\mathbf{z}$.

- Start with some non-degenerate initial conditions:

$$\begin{aligned} C \circ \mathcal{G}_P &= 0 \\ (\partial_n C) \circ \mathcal{G}_P &\neq 0. \end{aligned}$$

- Compute the homogeneous parts of P, one by one, via Newton Iteration

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$\begin{aligned} C' \circ \mathscr{G}_P &= 0 \\ (\partial_n C') \circ \mathscr{G}_P &\neq 0. \end{aligned}$$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$
\begin{aligned}
C' \circ \mathscr{G}_P &= 0 \\
(\partial_n C') \circ \mathscr{G}_P &\neq 0.
\end{aligned}
$$

$$C(x_0, \ldots, x_{n-1}, x_n) \neq 0$$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$\begin{aligned} C' \circ \mathscr{G}_P &= 0 \\ (\partial_n C') \circ \mathscr{G}_P &\neq 0. \end{aligned}$$

$$C(g_0, \ldots, g_{n-1}, g_n) = 0$$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

$$\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \stackrel{?}{=} 0$$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) = 0$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) = 0$

$$C(x_0, \ldots, x_{n-1}, a) \neq 0$$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) = 0$

$$C(x_0, \ldots, x_{n-1}, a) \neq 0$$
$$C(g_0, \ldots, g_{n-1}, a) = 0$$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$
\begin{aligned}
C' \circ \mathscr{G}_P &= 0 \\
(\partial_n C') \circ \mathscr{G}_P &\neq 0.
\end{aligned}
$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) = 0$

$$
\begin{aligned}
C(x_0, \ldots, x_{n-1}, a) &\neq 0 \\
C(g_0, \ldots, g_{n-1}, a) &= 0
\end{aligned}
$$

Contradicts minimality!

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

$$C(g_0, \ldots, g_{n-1}, g_n) = 0$$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$
\begin{aligned}
C' \circ \mathscr{G}_P &= 0 \\
(\partial_n C') \circ \mathscr{G}_P &\neq 0.
\end{aligned}
$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

$C(g_0, \ldots, g_{n-1}, g_n) = 0$

$(x_n - g_n)$ divides $\tilde{C}$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$\begin{aligned} C' \circ \mathscr{G}_P &= 0 \\ (\partial_n C') \circ \mathscr{G}_P &\neq 0. \end{aligned}$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

$$\begin{aligned} C(g_0, \ldots, g_{n-1}, g_n) &= 0 \\ (\partial_n C)(g_0, \ldots, g_n) &= 0 \end{aligned}$$

$(x_n - g_n)^2$ divides $\tilde{C}$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$
\begin{aligned}
C' \circ \mathscr{G}_P &= 0 \\
(\partial_n C') \circ \mathscr{G}_P &\neq 0.
\end{aligned}
$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

$$
\begin{aligned}
C(g_0, \ldots, g_{n-1}, g_n) &= 0 \\
(\partial_n C)(g_0, \ldots, g_n) &= 0 \\
(\partial_n^2 C)(g_0, \ldots, g_n) &= 0
\end{aligned}
$$

$(x_n - g_n)^3$ divides $\tilde{C}$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

$$C(g_0, \ldots, g_{n-1}, g_n) = 0$$
$$(\partial_n C)(g_0, \ldots, g_n) = 0$$
$$(\partial_n^2 C)(g_0, \ldots, g_n) = 0$$
$$\vdots$$
$$(\partial_n^r C)(g_0, \ldots, g_n) = 0$$

$(x_n - g_n)^{r+1}$ divides $\tilde{C}$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

$$C(g_0, \ldots, g_{n-1}, g_n) = 0$$
$$(\partial_n C)(g_0, \ldots, g_n) = 0$$
$$(\partial_n^2 C)(g_0, \ldots, g_n) = 0$$
$$\vdots$$
$$(\partial_n^r C)(g_0, \ldots, g_n) = 0$$

$(x_n - g_n)^{r+1}$ divides $\tilde{C}$

$(x_n - g_n)^t$ cannot divide $\tilde{C}$
if $t > \deg \tilde{C}$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$\begin{aligned} C' \circ \mathscr{G}_P &= 0 \\ (\partial_n C') \circ \mathscr{G}_P &\neq 0. \end{aligned}$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

$$\begin{aligned} (\partial_n^r C)(g_0, \ldots, g_{n-1}, g_n) = 0 \\ (\partial_n^{r+1} C)(g_0, \ldots, g_{n-1}, g_n) \neq 0 \end{aligned}$$

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

$$(\partial_n^r C)(g_0, \ldots, g_{n-1}, g_n) = 0$$
$$(\partial_n^{r+1} C)(g_0, \ldots, g_{n-1}, g_n) \neq 0$$

$C' = (\partial_n^r C)$ is what we want.

Setting-up the initial conditions

(Assuming that $\mathscr{G}_P$ is *not* a generator)

Goal: Find a circuit C' of small size such that

$$C' \circ \mathscr{G}_P = 0$$
$$(\partial_n C') \circ \mathscr{G}_P \neq 0.$$

If $\tilde{C}(x_n) = C(g_0, \ldots, g_{n-1}, x_n) \neq 0$

$$(\partial_n^r C)(g_0, \ldots, g_{n-1}, g_n) = 0$$
$$(\partial_n^{r+1} C)(g_0, \ldots, g_{n-1}, g_n) \neq 0$$

$C' = (\partial_n^r C)$ is what we want.

And, $\mathrm{size}(C') \leq \mathrm{size}(C) \cdot \deg(C)$

Some basic properties

$$\Delta_i(P) = \sum_{\mathbf{e}:|\mathbf{e}|=i} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z}).$$

Some basic properties

$$\Delta_i(P) = \sum_{\mathbf{e}:|\mathbf{e}|=i} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z}).$$

Additivity:

$$\Delta_i(P+Q) = \Delta_i(P) + \Delta_i(Q)$$

Some basic properties

$$\Delta_i(P) = \sum_{\mathbf{e}:|\mathbf{e}|=i} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z}).$$

Additivity:

$$\Delta_i(P+Q) = \Delta_i(P) + \Delta_i(Q)$$

‘Homogeneity’:

$$P(\mathbf{z}) = Q(\mathbf{z}) \bmod \langle \mathbf{z} \rangle^t$$
$$\implies \Delta_i(P) = \Delta_i(Q) \bmod \langle \mathbf{z} \rangle^{t-i}$$

Some basic properties

$$\Delta_i(P) = \sum_{\mathbf{e}:|\mathbf{e}|=i} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot (\partial_{\mathbf{e}} P)(\mathbf{z}).$$

Additivity:

$$\Delta_i(P+Q) = \Delta_i(P) + \Delta_i(Q)$$

'Homogeneity':

$$P(\mathbf{z}) = Q(\mathbf{z}) \bmod \langle \mathbf{z} \rangle^t$$

$$\implies \Delta_i(P) = \Delta_i(Q) \bmod \langle \mathbf{z} \rangle^{t-i}$$

$$P = P_0 + \cdots + P_d$$

$$\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t$$

The Reconstruction Step

$$
\begin{aligned}
C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{z}) &= 0 \\
(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{z}) &\neq 0
\end{aligned}
$$

The Reconstruction Step

$$
\begin{aligned}
C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) &= 0 \\
(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) &\neq 0
\end{aligned}
$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

Else, replace $\langle z_1, \ldots, z_\ell \rangle$ with $\langle z_1 - \alpha_1, \ldots, z_k - \alpha_k \rangle$ in what follows

The Reconstruction Step

$$
\begin{aligned}
C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) &= 0 \\
(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) &\neq 0
\end{aligned}
$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = \underbrace{P_0 + \cdots + P_n}_{\text{Bruteforce in } n^{O(k)} \text{ size}} + P_{n+1} + \cdots + P_d$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = \underbrace{P_0 + \cdots + P_n}_{\text{Bruteforce in } n^{O(k)} \text{ size}} + \underbrace{P_{n+1} + \cdots + P_d}_{\text{Compute, via Newton iterations, one by one}}$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$C'\left(g_0, \ldots, g_{n-1}, g_n\right) = 0$$

The Reconstruction Step

$$C' \circ \mathcal{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathcal{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$C'(\Delta_0(P), \ldots, \Delta_{n-1}(P), \Delta_n(P)) = 0$$

The Reconstruction Step

$$\begin{aligned} C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) &= 0 \\ (\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) &\neq 0 \end{aligned}$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$C'(\Delta_0(P), \ldots, \Delta_{n-1}(P), \Delta_n(P)) = 0 \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$\begin{aligned} C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) &= 0 \\ (\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) &\neq 0 \end{aligned}$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$(\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t)$$

$$C'(\Delta_0(P), \ldots, \Delta_{n-1}(P), \Delta_n(P)) = 0 \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$(\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t)$$

$$C'\big(\Delta_0(P_{\leq n}), \ldots, \Delta_{n-1}(P_{\leq n}), \Delta_n(P_{\leq n+1})\big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$(\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t)$$

$$C'\Big(\Delta_0(P_{\leq n}), \ldots, \Delta_{n-1}(P_{\leq n}), \Delta_n(P_{\leq n}) + \Delta_n(P_{n+1})\Big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$(\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t)$$

$$C'\Big(\Delta_0(P_{\leq n}), \ldots, \Delta_{n-1}(P_{\leq n}), \Delta_n(P_{\leq n}) + \Delta_n(P_{n+1})\Big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$(\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t)$$

$$C'\Big(\Delta_0(P_{\leq n}), \ldots, \Delta_{n-1}(P_{\leq n}), \Delta_n(P_{\leq n}) + \Delta_n(P_{n+1})\Big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$C'(R_0, \ldots, R_{n-1}, R_n + A) = 0 \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$(\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t)$$

$$C'\left(\Delta_0(P_{\leq n}), \ldots, \Delta_{n-1}(P_{\leq n}), \Delta_n(P_{\leq n}) + \Delta_n(P_{n+1})\right) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$C'(R_0, \ldots, R_{n-1}, R_n + A) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$= C'(R_0, \ldots, R_{n-1}, R_n) + A \cdot ((\partial_n C')(R_0, \ldots, R_n)) = 0 \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$(\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t)$$

$$C'\big(\Delta_0(P_{\leq n}), \ldots, \Delta_{n-1}(P_{\leq n}), \Delta_n(P_{\leq n}) + \Delta_n(P_{n+1})\big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$C'(R_0, \ldots, R_{n-1}, R_n + A) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$= C'(R_0, \ldots, R_{n-1}, R_n) + A \cdot \big((\partial_n C')(R_0, \ldots, R_n)(\mathbf{y}, \mathbf{0})\big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$(\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t)$$

$$C'\Big(\Delta_0(P_{\leq n}), \ldots, \Delta_{n-1}(P_{\leq n}), \Delta_n(P_{\leq n}) + \Delta_n(P_{n+1})\Big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$C'(R_0, \ldots, R_{n-1}, R_n + A) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$= C'(R_0, \ldots, R_{n-1}, R_n) + A \cdot \Big((\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0})\Big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$(\Delta_i(P) = \Delta_i(P_{\leq t+i-1}) \bmod \langle \mathbf{z} \rangle^t)$$

$$C'\Big(\Delta_0(P_{\leq n}), \ldots, \Delta_{n-1}(P_{\leq n}), \Delta_n(P_{\leq n}) + \Delta_n(P_{n+1})\Big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$C'(R_0, \ldots, R_{n-1}, R_n + A) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$= C'(R_0, \ldots, R_{n-1}, R_n) + A \cdot \Big((\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0})\Big) = 0 \bmod \langle \mathbf{z} \rangle^2$$

$$\therefore A = \left(\frac{C'(R_0, \ldots, R_n)}{(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0})} \right) \bmod (\mathbf{z})^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P\left(\mathbf{y}, \mathbf{0}\right) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P\left(\mathbf{y}, \mathbf{0}\right) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$\Delta_n(P_{n+1}) = \left(\frac{C'\left(\Delta_0(P_{\leq n}), \ldots, \Delta_n(P_{\leq n})\right)}{(\partial_n C') \circ \mathscr{G}_P\left(\mathbf{y}, \mathbf{0}\right)} \right) \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$\Delta_n(P_{n+1})(\mathbf{a}, \mathbf{z}) = \left(\frac{C'\big(\Delta_0(P_{\leq n}), \ldots, \Delta_n(P_{\leq n})\big)(\mathbf{a}, \mathbf{z})}{(\partial_n C') \circ \mathscr{G}_P(\mathbf{a}, \mathbf{0})} \right) \bmod \langle \mathbf{z} \rangle^2$$

The Reconstruction Step

$$C' \circ \mathscr{G}_P\,(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P\,(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$\Delta_n(P_{n+1})(\mathbf{a}, \mathbf{z}) = \left(\frac{C'\big(\Delta_0(P_{\leq n}), \ldots, \Delta_n(P_{\leq n})\big)(\mathbf{a}, \mathbf{z})}{(\partial_n C') \circ \mathscr{G}_P\,(\mathbf{a}, \mathbf{0})} \right) \bmod \langle \mathbf{z} \rangle^2$$

By trying many $\mathbf{a}$'s, we can obtain all of $\partial^{=n}(P_{n+1})$

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$\Delta_n(P_{n+1})(\mathbf{a}, \mathbf{z}) = \left(\frac{C'\big(\Delta_0(P_{\leq n}), \ldots, \Delta_n(P_{\leq n})\big)(\mathbf{a}, \mathbf{z})}{(\partial_n C') \circ \mathscr{G}_P(\mathbf{a}, \mathbf{0})} \right) \bmod \langle \mathbf{z} \rangle^2$$

By trying many $\mathbf{a}$'s, we can obtain all of $\partial^{=n}(P_{n+1})$ and hence P_{n+1} itself

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$\Delta_n(P_{n+1})(\mathbf{a}, \mathbf{z}) = \left(\frac{C'\big(\Delta_0(P_{\leq n}), \ldots, \Delta_n(P_{\leq n})\big)(\mathbf{a}, \mathbf{z})}{(\partial_n C') \circ \mathscr{G}_P(\mathbf{a}, \mathbf{0})} \right) \bmod \langle \mathbf{z} \rangle^2$$

By trying many $\mathbf{a}$'s, we can obtain all of $\partial^{=n}(P_{n+1})$
and hence P_{n+1} itself

(Euler formula: $d \cdot f = \sum x_i \partial_i f$, if f homogeneous of degree d)

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$\Delta_n(P_{n+1})(\mathbf{a}, \mathbf{z}) = \left(\frac{C'\big(\Delta_0(P_{\leq n}), \ldots, \Delta_n(P_{\leq n})\big)(\mathbf{a}, \mathbf{z})}{(\partial_n C') \circ \mathscr{G}_P(\mathbf{a}, \mathbf{0})} \right) \bmod \langle \mathbf{z} \rangle^2$$

By trying many $\mathbf{a}$'s, we can obtain all of $\partial^{=n}(P_{n+1})$
and hence P_{n+1} itself

The Reconstruction Step

$$C' \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathscr{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$\Delta_n(P_{n+1})(\mathbf{a}, \mathbf{z}) = \left(\frac{C'\big(\Delta_0(P_{\leq n}), \ldots, \Delta_n(P_{\leq n})\big)(\mathbf{a}, \mathbf{z})}{(\partial_n C') \circ \mathscr{G}_P(\mathbf{a}, \mathbf{0})} \right) \bmod \langle \mathbf{z} \rangle^2$$

By trying many $\mathbf{a}$'s, we can obtain all of $\partial^{=n}(P_{n+1})$
and hence P_{n+1} itself
modulo higher order junk

The Reconstruction Step

$$C' \circ \mathcal{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathcal{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

$$\boxed{\Delta_n(P_{n+1})(\mathbf{a}, \mathbf{z}) = \left(\frac{C'\left(\Delta_0(P_{\leq n}), \ldots, \Delta_n(P_{\leq n})\right)(\mathbf{a}, \mathbf{z})}{(\partial_n C') \circ \mathcal{G}_P(\mathbf{a}, \mathbf{0})} \right) \bmod \langle \mathbf{z} \rangle^2}$$

By trying many $\mathbf{a}$'s, we can obtain all of $\partial^{=n}(P_{n+1})$
and hence P_{n+1} itself
~~modulo higher order junk~~

Border tricks!

Or careful homogenisation

The Reconstruction Step

$$C' \circ \mathcal{G}_P(\mathbf{y}, \mathbf{0}) = 0$$
$$(\partial_n C') \circ \mathcal{G}_P(\mathbf{y}, \mathbf{0}) \neq 0$$

$$P = P_0 + \cdots + P_n + P_{n+1} + \cdots + P_d$$

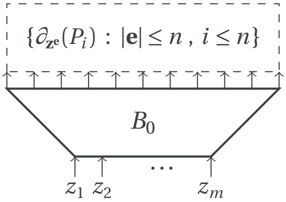
$$\boxed{\Delta_n(P_{n+j+1})(\mathbf{a}, \mathbf{z}) = \left(\frac{C'\big(\Delta_0(P_{\leq n+j}), \ldots, \Delta_n(P_{\leq n+j})\big)(\mathbf{a}, \mathbf{z})}{(\partial_n C') \circ \mathcal{G}_P(\mathbf{a}, \mathbf{0})} \right) \bmod \langle \mathbf{z} \rangle^{j+2}}$$

By trying many $\mathbf{a}$'s, we can obtain all of $\partial^{=n}(P_{n+j+1})$
and hence P_{n+j+1} itself
~~modulo higher order junk~~

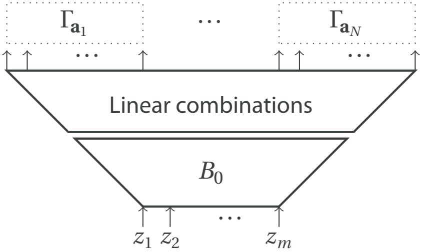
Border tricks!

Or careful homogenisation

Reconstruction Step: Pictorially

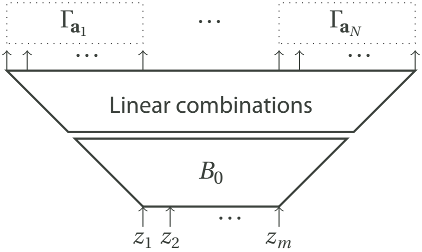


Reconstruction Step: Pictorially

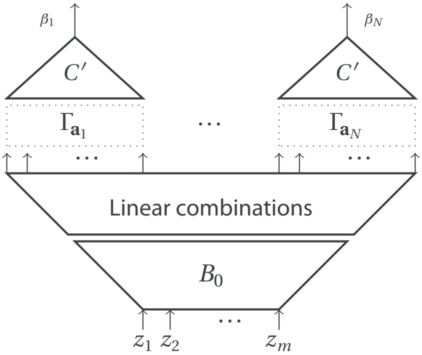


Reconstruction Step: Pictorially

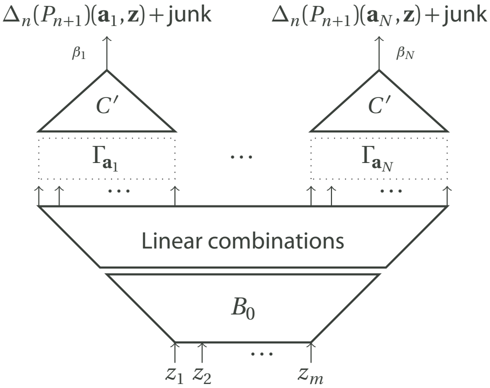
$$\Gamma_{\mathbf{a}} = \big(\Delta_0(P_{\leq n})(\mathbf{a},\mathbf{z})\,,\ \ldots\ ,\ \Delta_n(P_{\leq n})(\mathbf{a},\mathbf{z})\big)$$



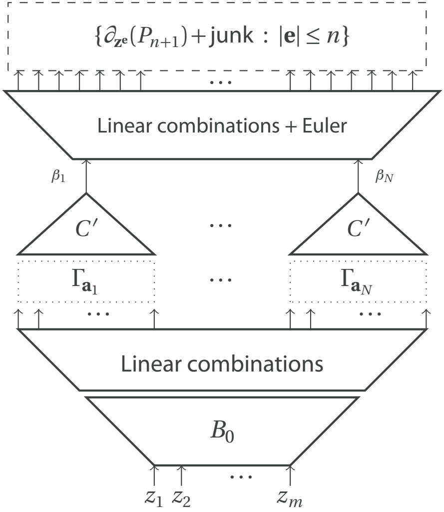
Reconstruction Step: Pictorially



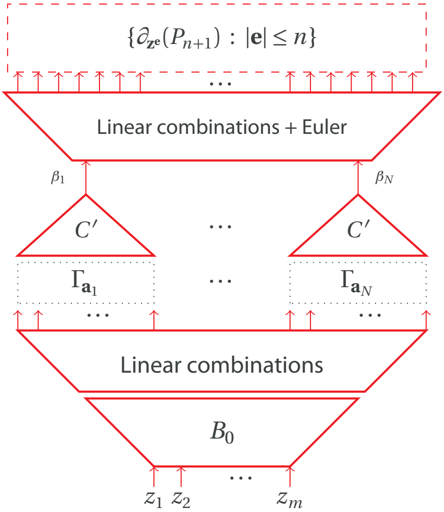
Reconstruction Step: Pictorially



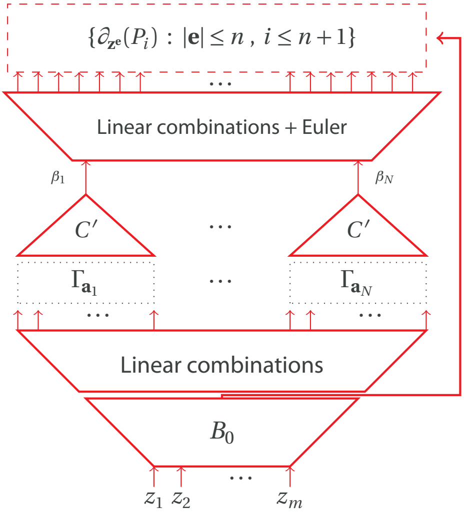
Reconstruction Step: Pictorially



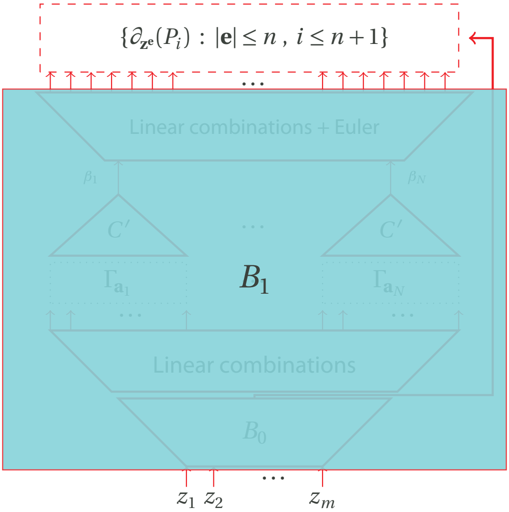
Reconstruction Step: Pictorially



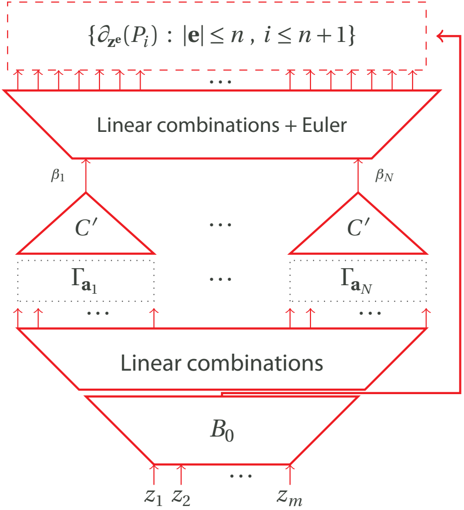
Reconstruction Step: Pictorially



Reconstruction Step: Pictorially

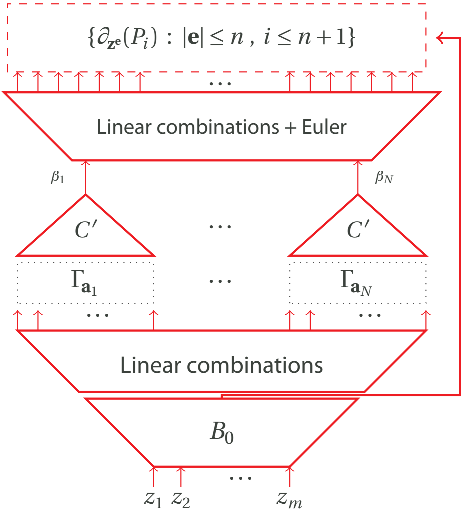


Reconstruction Step: Pictorially



$$s_0 = n^{O(k)}$$

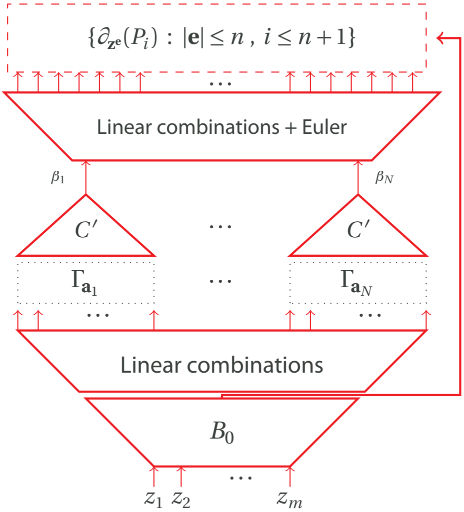
Reconstruction Step: Pictorially



$n^{O(k)}$

$s_0 = n^{O(k)}$

Reconstruction Step: Pictorially

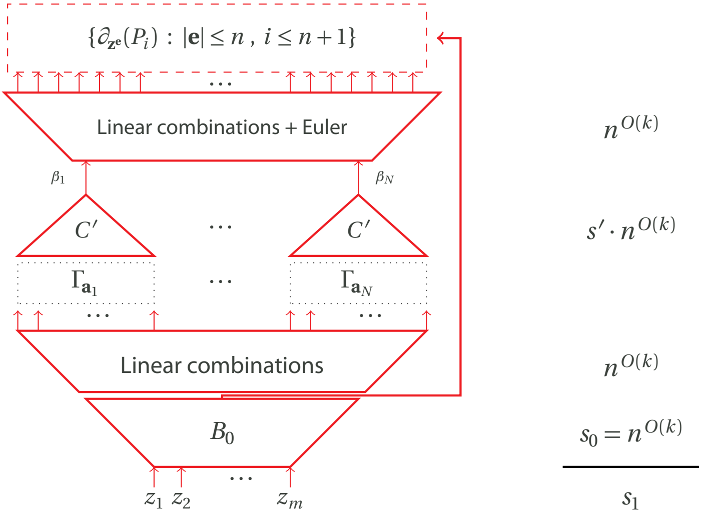


$s' \cdot n^{O(k)}$

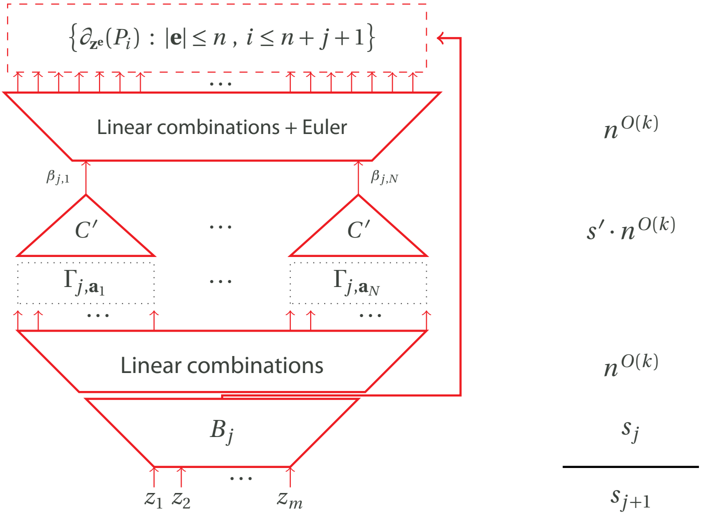
$n^{O(k)}$

$s_0 = n^{O(k)}$

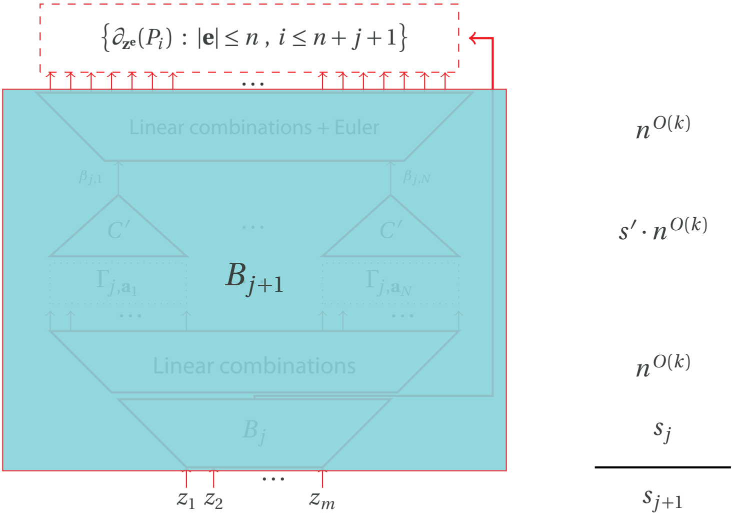
Reconstruction Step: Pictorially



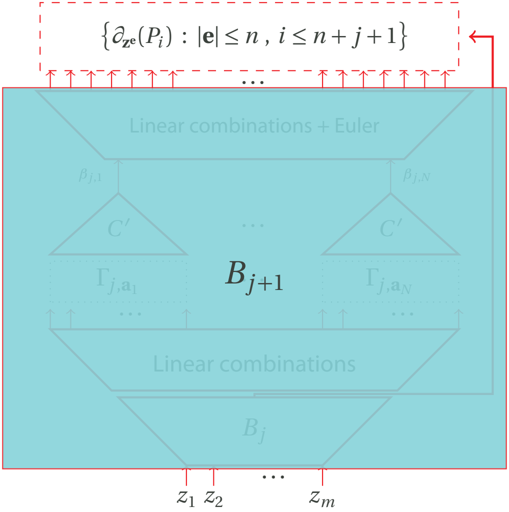
Reconstruction Step: Pictorially



Reconstruction Step: Pictorially



Reconstruction Step: Pictorially



$n^{O(k)}$

$s' \cdot n^{O(k)}$

$n^{O(k)}$

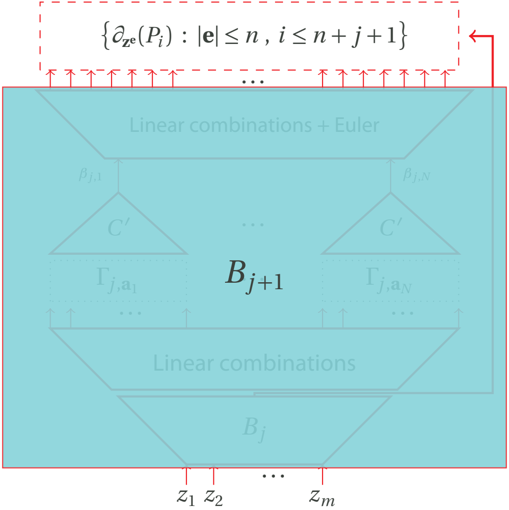
s_j

---

s_{j+1}

$$s_d \quad \leq \quad s' \cdot n^{O(k)} \cdot d$$

Reconstruction Step: Pictorially



$n^{O(k)}$

$s' \cdot n^{O(k)}$

$n^{O(k)}$

s_j

s_{j+1}

$$s_d \quad \leq \quad s \cdot D \cdot n^{O(k)} \cdot d$$

\end{proof}

Conclusion

Conclusion

Summary:

- With suitable hardness, we can get poly-sized hitting sets.

Conclusion

Summary:

- With suitable hardness, we can get poly-sized hitting sets.
- With the border, we can bootstrap from barely non-trivial hitting sets.

Conclusion

Summary:

- With suitable hardness, we can get poly-sized hitting sets.
- With the border, we can bootstrap from barely non-trivial hitting sets.

Open Problems:

- Current proof requires characteristic zero fields. Ought to work for all fields.

Conclusion

Summary:

- With suitable hardness, we can get poly-sized hitting sets.
- With the border, we can bootstrap from barely non-trivial hitting sets.

Open Problems:

- Current proof requires characteristic zero fields. Ought to work for all fields.
- The hardness depends on the degree of the circuit we are fooling. Ought to fool all small size circuits irrespective of degree (using the border).

Conclusion

Summary:

- With suitable hardness, we can get poly-sized hitting sets.
- With the border, we can bootstrap from barely non-trivial hitting sets.

Open Problems:

- Current proof requires characteristic zero fields. Ought to work for all fields.
- The hardness depends on the degree of the circuit we are fooling. Ought to fool all small size circuits irrespective of degree (using the border).

\end{document}