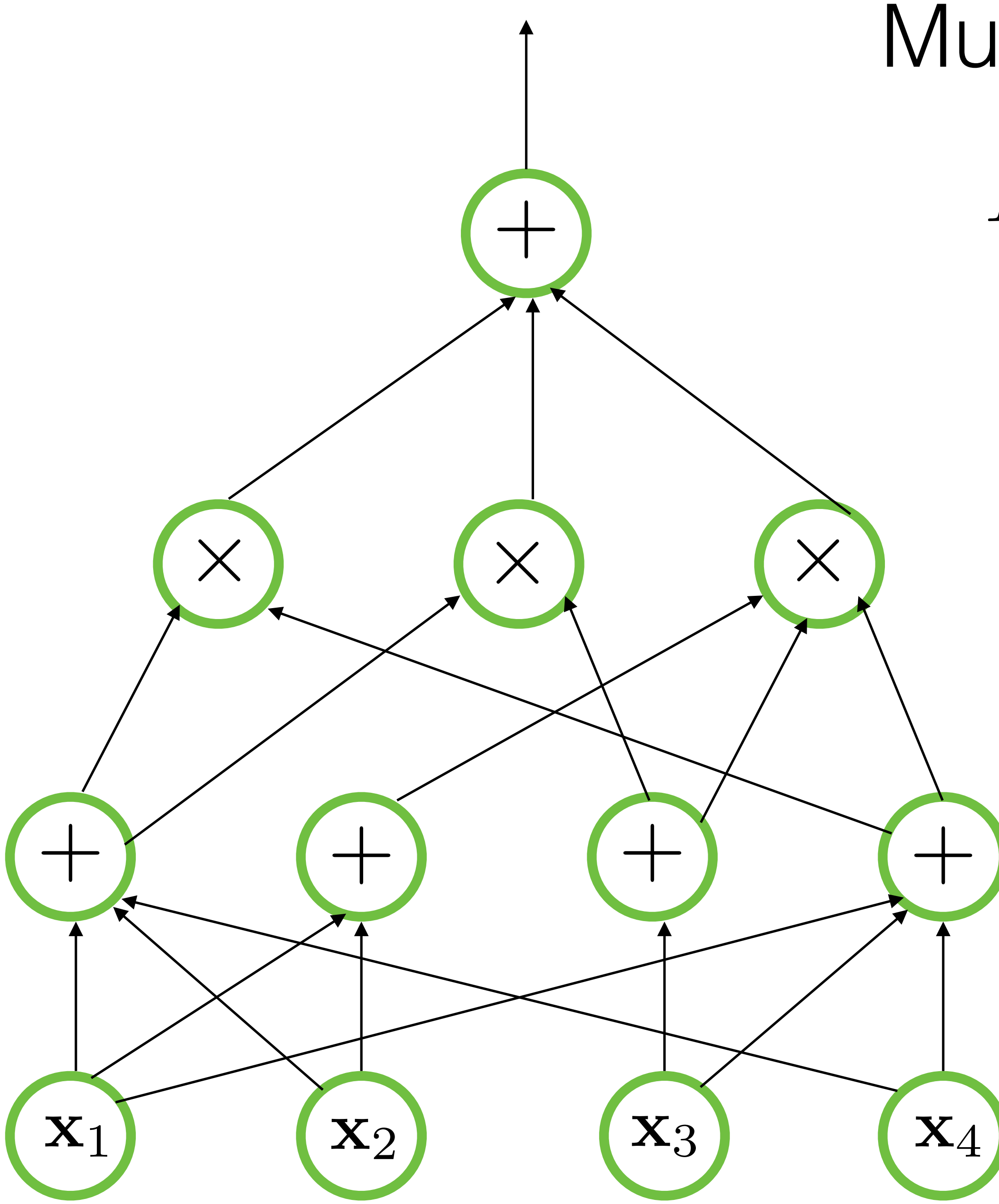
Some Closure Results for Polynomial Factorization and Applications

Chi-Ning Chou

Harvard University

Joint work with **Mrinal Kumar** and **Noam Solomon**

Arithmetic Circuits



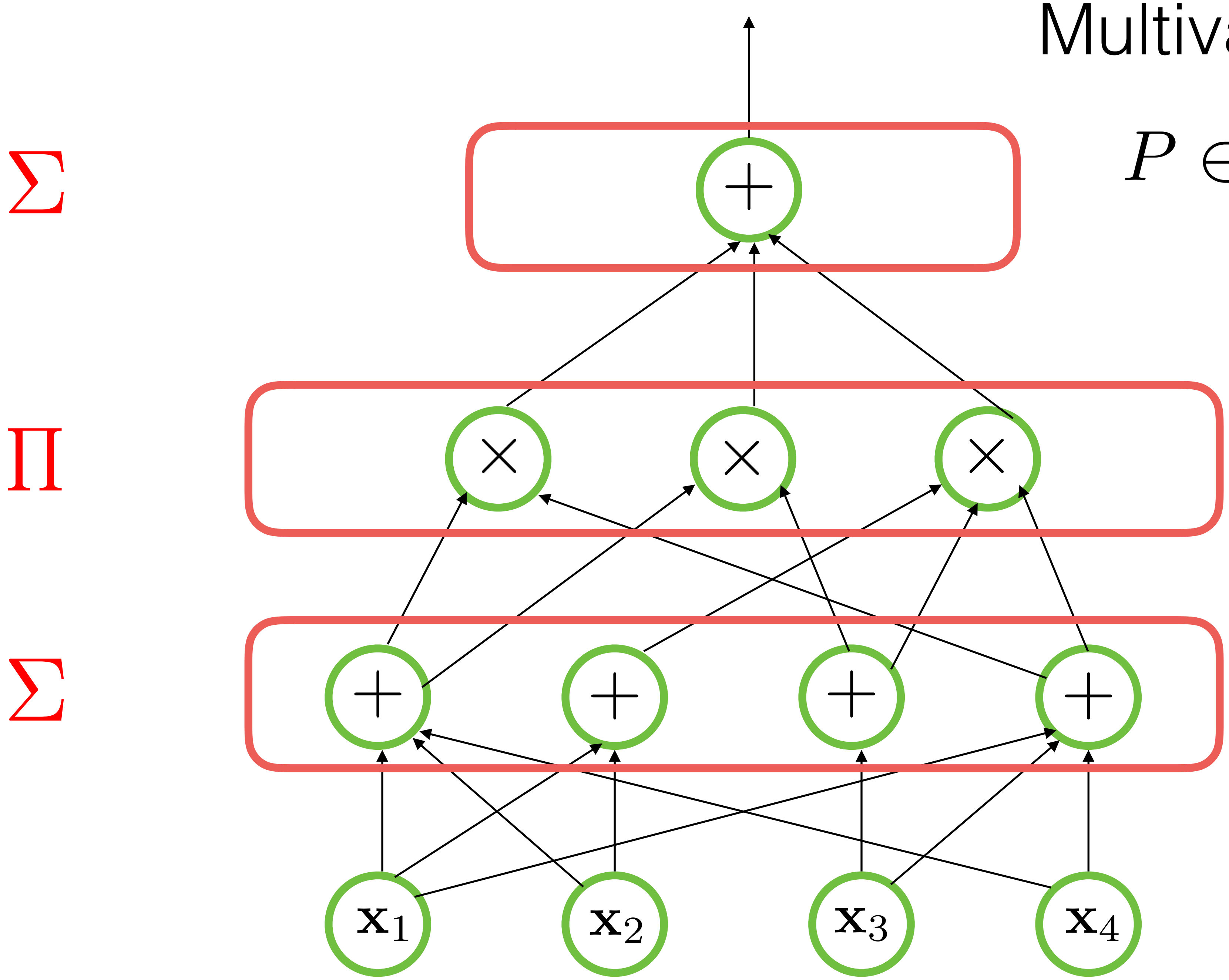
Multivariate polynomial

$$P \in \mathbb{F}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$$

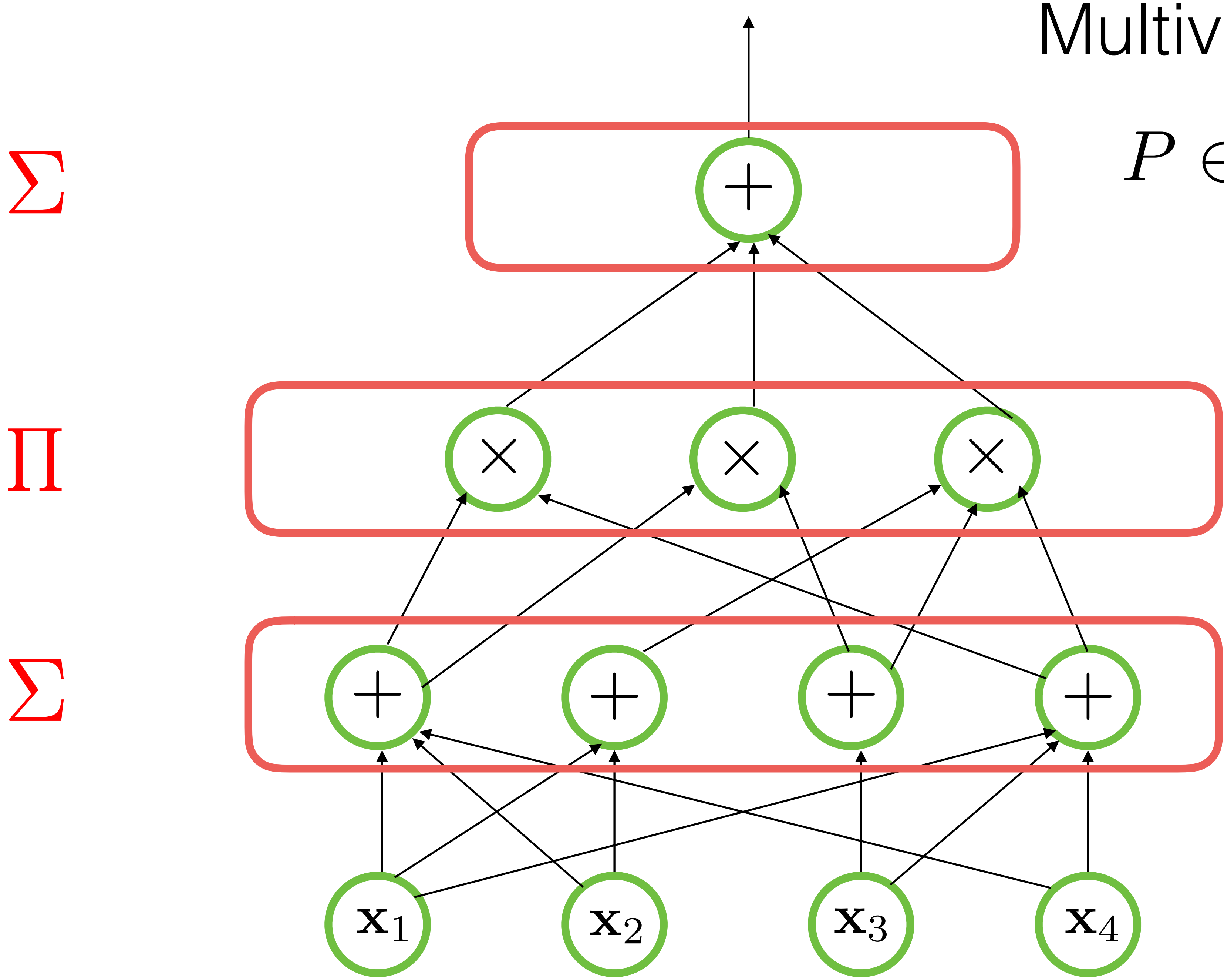
Arithmetic Circuits

Multivariate polynomial

$$P \in \mathbb{F}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$$



Arithmetic Circuits

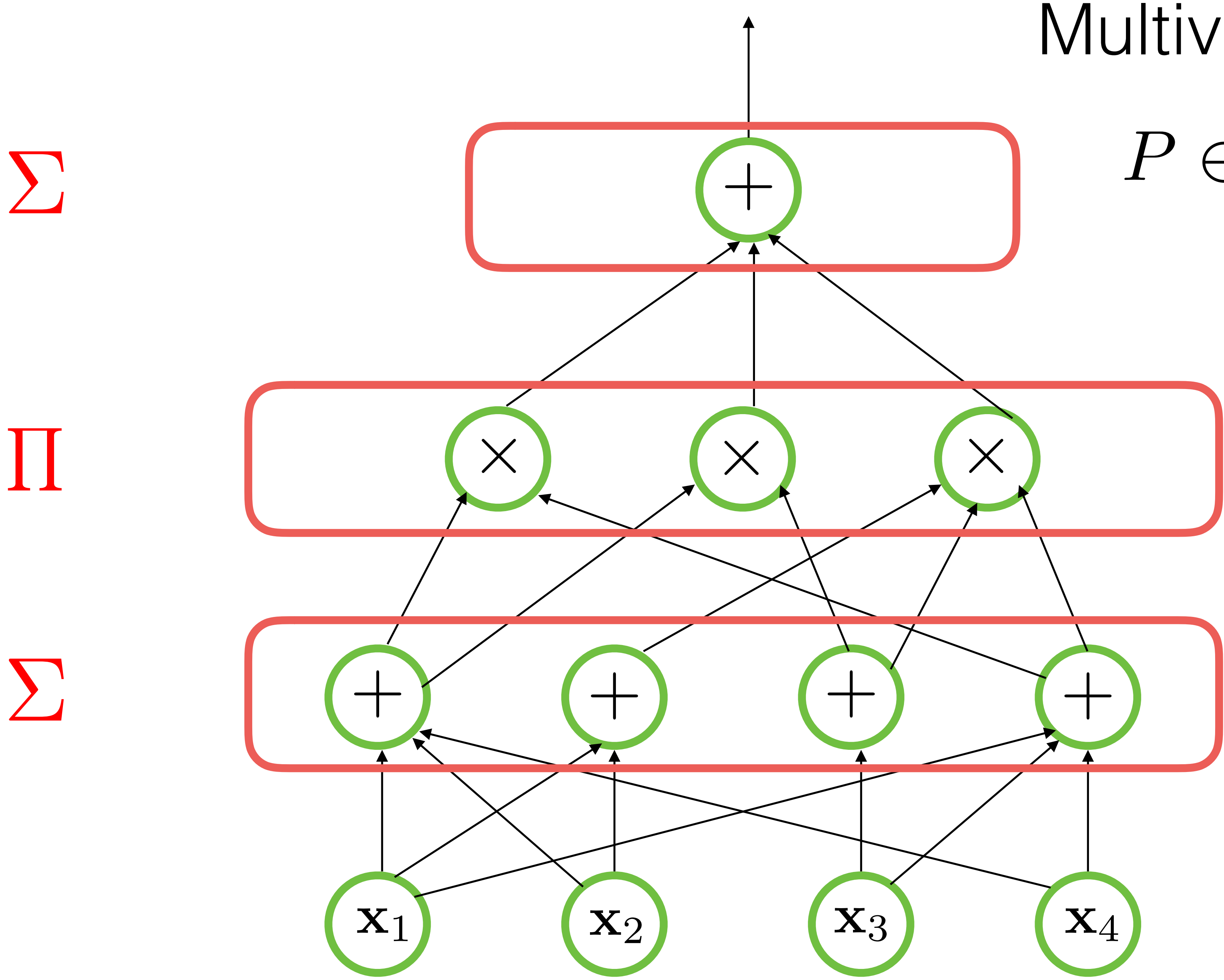


Multivariate polynomial

$$P \in \mathbb{F}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$$

Depth - 3

Arithmetic Circuits



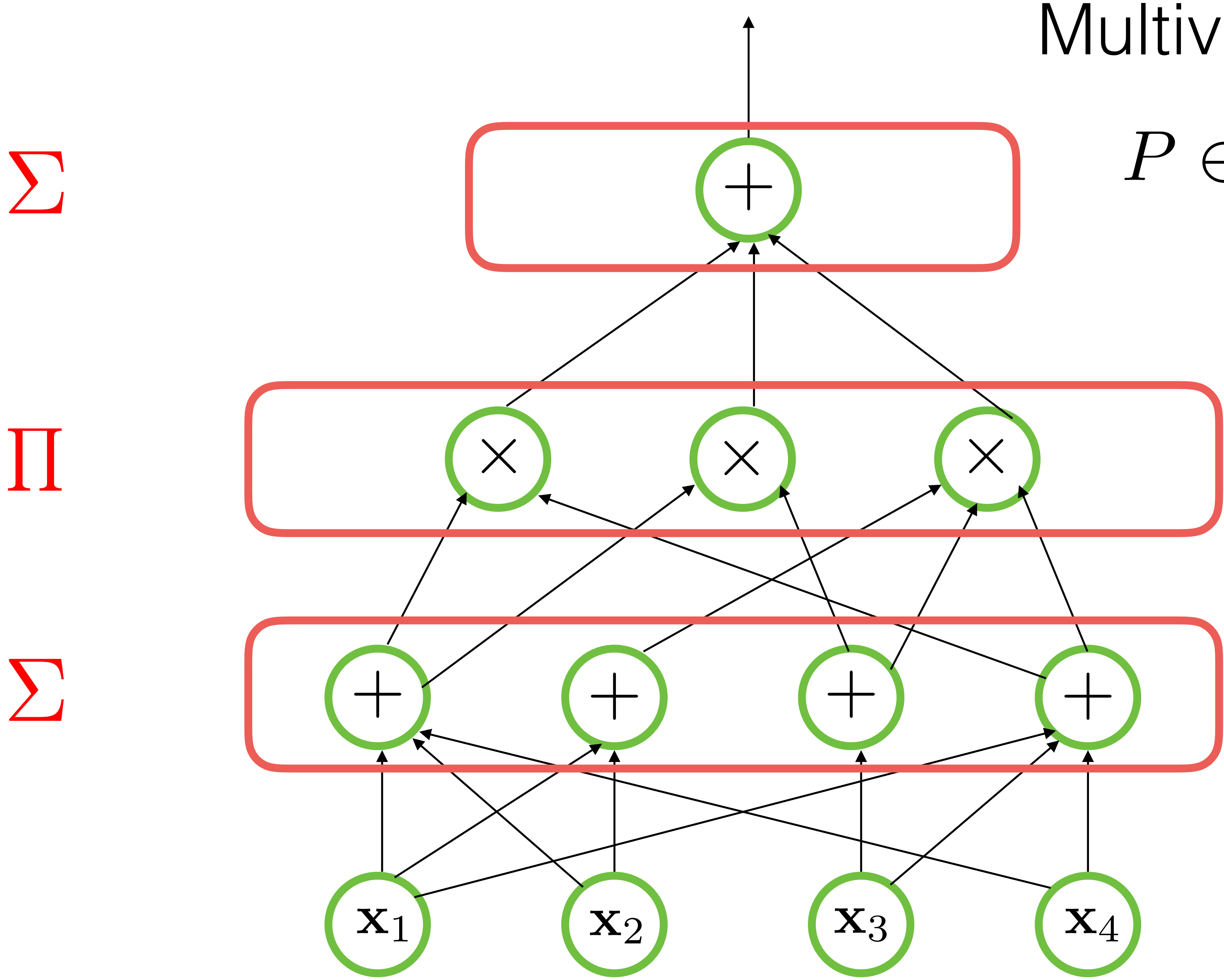
Multivariate polynomial

$P \in \mathbb{F}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$

Depth - 3

Size - 8

Arithmetic Circuits



Multivariate polynomial

$$P \in \mathbb{F}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$$

Depth - 3

Size - 8

*Assume $\mathbb{F} = \mathbb{Q}$

Algebraic Complexity Classes

Algebraic Complexity Classes

$$\mathcal{C} = \Big\{ \{f_1, f_2, \ldots\} \Big\}$$

Algebraic Complexity Classes

$$\mathcal{C} = \Big\{ \{ f_1, f_2, \ldots \} \Big\}$$

For simplicity, denote $f = f_n$.

Algebraic Complexity Classes

$$\mathcal{C} = \Big\{\{f_1, f_2, \ldots\}\Big\}$$

For simplicity, denote $f = f_n$.

- VP: Polynomials computed by poly(n) size, poly(n) degree arithmetic circuits.

Algebraic Complexity Classes

$$\mathcal{C} = \left\{ \{f_1, f_2, \ldots\} \right\}$$

For simplicity, denote $f = f_n$.

- VP: Polynomials computed by poly(n) size, poly(n) degree arithmetic circuits.
- $\text{Depth-}\Delta$: Polynomials computed by poly(n) size, poly(n) degree, and depth-Δ arithmetic circuits.

Algebraic Complexity Classes

$$\mathcal{C} = \left\{ \{f_1, f_2, \ldots\} \right\}$$

For simplicity, denote $f = f_n$.

- VP: Polynomials computed by poly(n) size, poly(n) degree arithmetic circuits.
- $\text{Depth-}\Delta$: Polynomials computed by poly(n) size, poly(n) degree, and depth-Δ arithmetic circuits.
- Many more such as VF, VBP, VNP…

Some Natural Questions

Some Natural Questions

- Lower bounds:

Some Natural Questions

- Lower bounds:

  Goal: Find an explicit f such that $f \notin \mathcal{C}$.

Some Natural Questions

- Lower bounds:

  Goal: Find an explicit f such that $f \notin \mathcal{C}$.

- Polynomial identity test (PIT):

Some Natural Questions

- Lower bounds:

  Goal: Find an explicit f such that $f \notin \mathcal{C}$.

- Polynomial identity test (PIT):

  Goal: Given $f, g \in \mathcal{C}$, does $f(x) = g(x)\ \forall x$?

Some Natural Questions

- Lower bounds:

  Goal: Find an explicit f such that $f \notin \mathcal{C}$.

- Polynomial identity test (PIT):

  Goal: Given $f, g \in \mathcal{C}$, does $f(x) = g(x)\ \forall x$?

- Polynomial factorization:

Some Natural Questions

- Lower bounds:

  Goal: Find an explicit f such that $f \notin \mathcal{C}$.

- Polynomial identity test (PIT):

  Goal: Given $f, g \in \mathcal{C}$, does $f(x) = g(x)\ \forall x$?

- Polynomial factorization:

  Goal: Given $f \in \mathcal{C}$ where $f = gh$, find g.

Some Natural Questions

- Lower bounds:

  Goal: Find an explicit f such that $f \notin \mathcal{C}$.

- Polynomial identity test (PIT):

  Goal: Given $f, g \in \mathcal{C}$, does $f(x) = g(x)\ \forall x$?

- Polynomial factorization:

  Goal: Given $f \in \mathcal{C}$ where $f = gh$, find g.

Polynomial Factorization

Polynomial Factorization

- Let $f = g^e h$

Polynomial Factorization

- Let $f = g^e h$, g is irreducible

Polynomial Factorization

- Let $f = g^e h$, g is irreducible and g, h coprime.

Polynomial Factorization

- Let $f = g^e h$, g is irreducible and g, h coprime.
- **Algorithmic problem**: Given f,output g.

Polynomial Factorization

- Let $f = g^e h$, g is irreducible and g, h coprime.
- **Algorithmic problem**: Given f,output g.
- **Closure problem**: If $f \in \mathcal{C}$, does $g \in \mathcal{C}'$?

Polynomial Factorization

- Let $f = g^e h$, g is irreducible and g, h coprime.
- **Algorithmic problem**: Given f,output g.
- **Closure problem**: If $f \in \mathcal{C}$, does $g \in \mathcal{C}'$?
- Applications:

Polynomial Factorization

- Let $f = g^e h$, g is irreducible and g, h coprime.
- **Algorithmic problem**: Given f,output g.
- **Closure problem**: If $f \in \mathcal{C}$, does $g \in \mathcal{C}'$?
- Applications:

— Decoding Reed-Solomon codes.

Polynomial Factorization

- Let $f = g^e h$, g is irreducible and g, h coprime.
- **Algorithmic problem**: Given f,output g.
- **Closure problem**: If $f \in \mathcal{C}$, does $g \in \mathcal{C}'$?
- Applications:

— Decoding Reed-Solomon codes.

— Hardness versus Randomness.

Polynomial Factorization

- Let $f = g^e h$, g is irreducible and g, h coprime.
- **Algorithmic problem**: Given f,output g.
- **Closure problem**: If $f \in \mathcal{C}$, does $g \in \mathcal{C}'$?
- Applications:

— Decoding Reed-Solomon codes.

— Hardness versus Randomness.

Polynomial Factorization (Closure Problem)

Goal: If $f \in \mathcal{C}$ then $g \in \mathcal{C}'$.

Polynomial Factorization (Closure Problem)

Goal: If $f \in \mathcal{C}$ then $g \in \mathcal{C}'$.

existential

Polynomial Factorization (Closure Problem)

Goal: If $f \in \mathcal{C}$ then $g \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP

Polynomial Factorization (Closure Problem)

Goal: If $f \in \mathcal{C}$ then $g \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09, **C**KS18]	$\bigcup_{\Delta \in \mathbb{N}} \text{Depth*-}\Delta$	$\bigcup_{\Delta \in \mathbb{N}} \text{Depth-}\Delta$

* There are some degree constraints.

Polynomial Factorization (Closure Problem)

Goal: If $f \in \mathcal{C}$ then $g \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09, **C**KS18]	$\bigcup_{\Delta \in \mathbb{N}} \text{Depth*-}\Delta$	$\bigcup_{\Delta \in \mathbb{N}} \text{Depth-}\Delta$
[DSS18, **C**KS18]	$\mathrm{VF}(n^{\log n}))$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n}))$)	$\mathrm{VF}(n^{\log n}))$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n}))$)

* There are some degree constraints.

Polynomial Factorization (Closure Problem)

Goal: If $f \in \mathcal{C}$ then $g \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09, **C**KS18]	$\bigcup_{\Delta \in \mathbb{N}} \text{Depth*-}\Delta$	$\bigcup_{\Delta \in \mathbb{N}} \text{Depth-}\Delta$
[DSS18, **C**KS18]	$\mathrm{VF}(n^{\log n}))$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n})$)	$\mathrm{VF}(n^{\log n}))$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n})$)
[**C**KS18]	VNP	VNP

* There are some degree constraints.

Polynomial Factorization (Closure Problem)

Goal: If $f \in \mathcal{C}$ then $g \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09, **C**KS18]	$\bigcup_{\Delta \in \mathbb{N}} \text{Depth*-}\Delta$	$\bigcup_{\Delta \in \mathbb{N}} \text{Depth-}\Delta$
[DSS18, **C**KS18]	$\mathrm{VF}(n^{\log n}))$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n})$)	$\mathrm{VF}(n^{\log n}))$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n})$)
[**C**KS18]	VNP	VNP

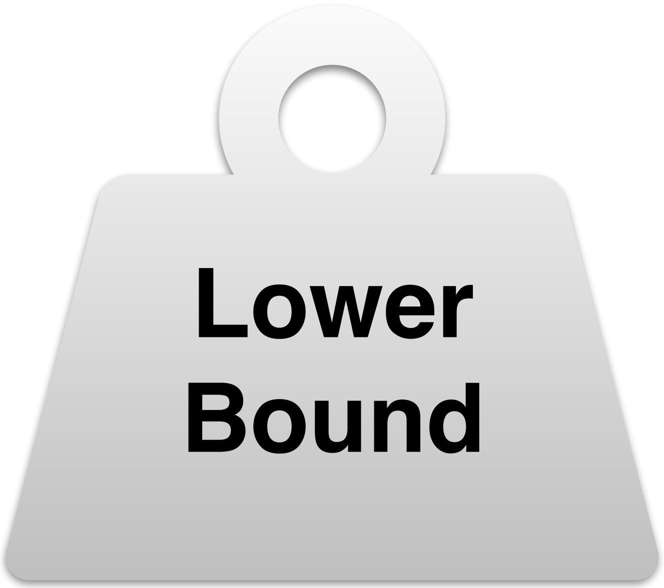
* There are some degree constraints.

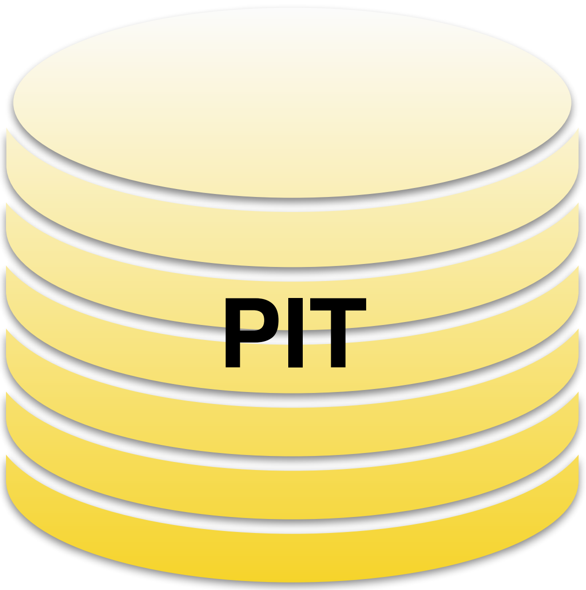
Application:

Hardness versus Randomness

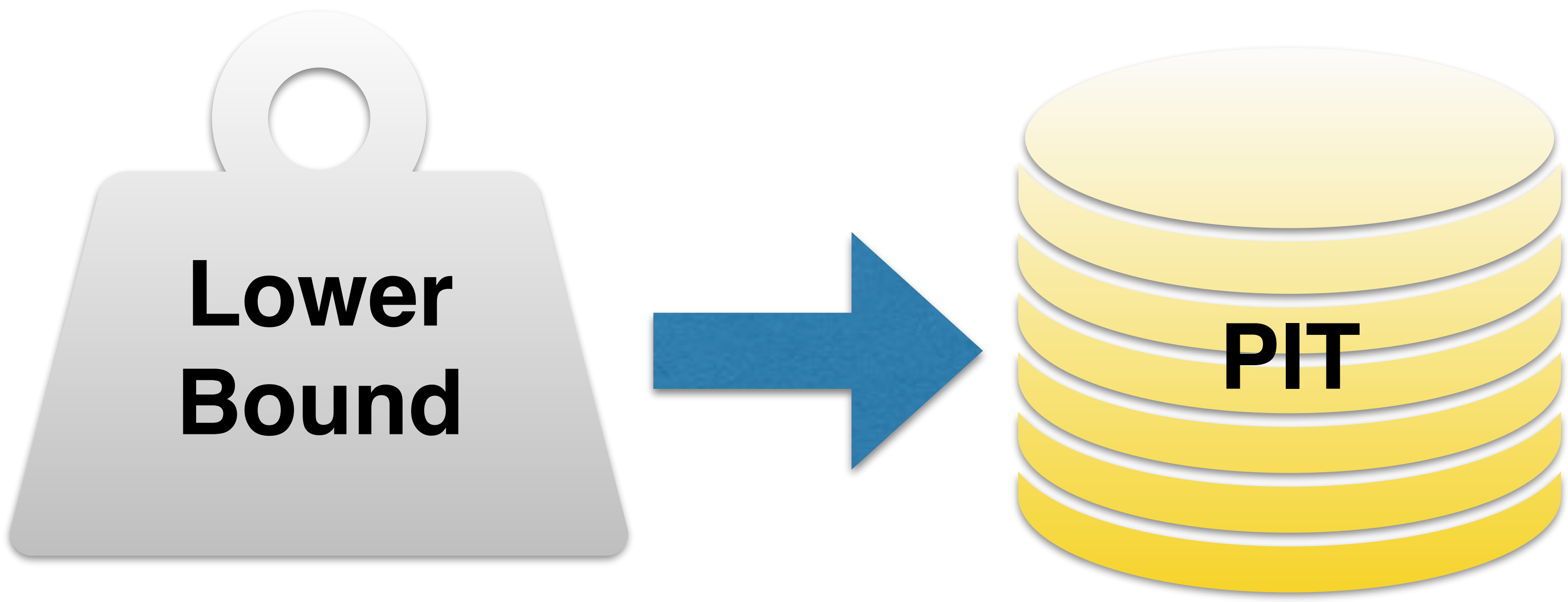
Hardness versus Randomness

Hardness versus Randomness

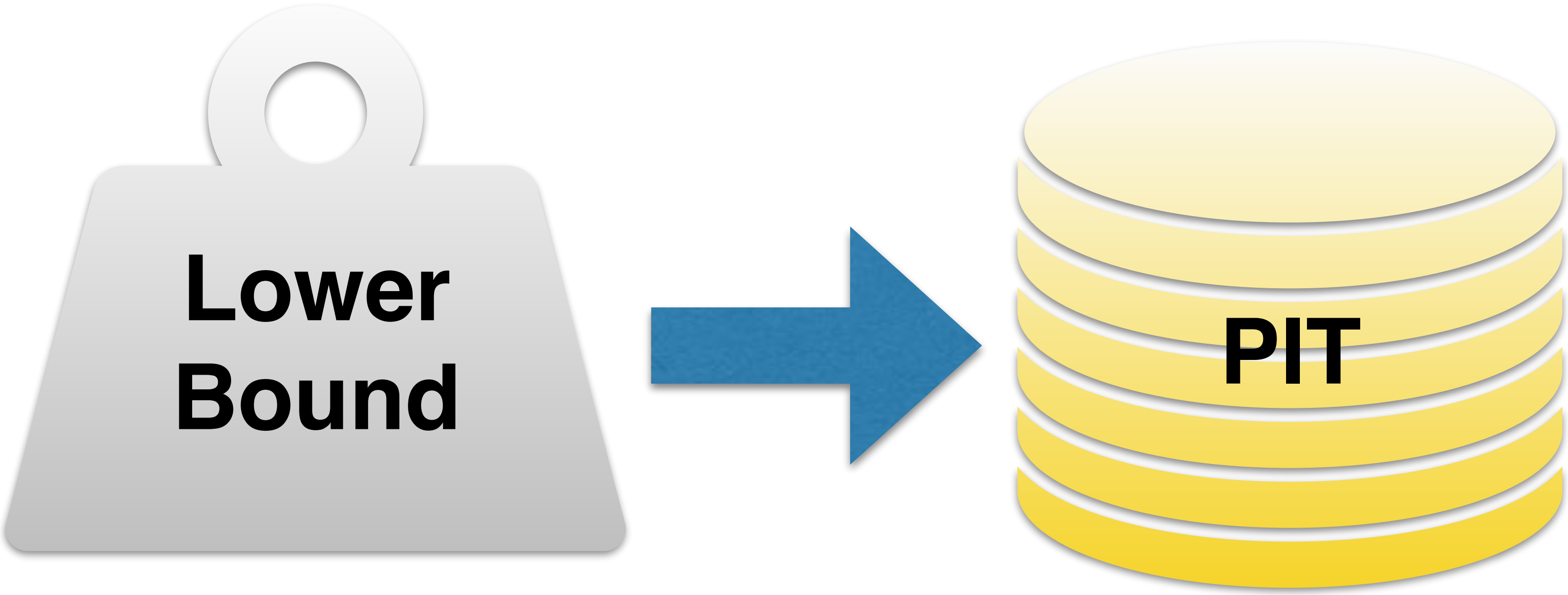




Hardness versus Randomness

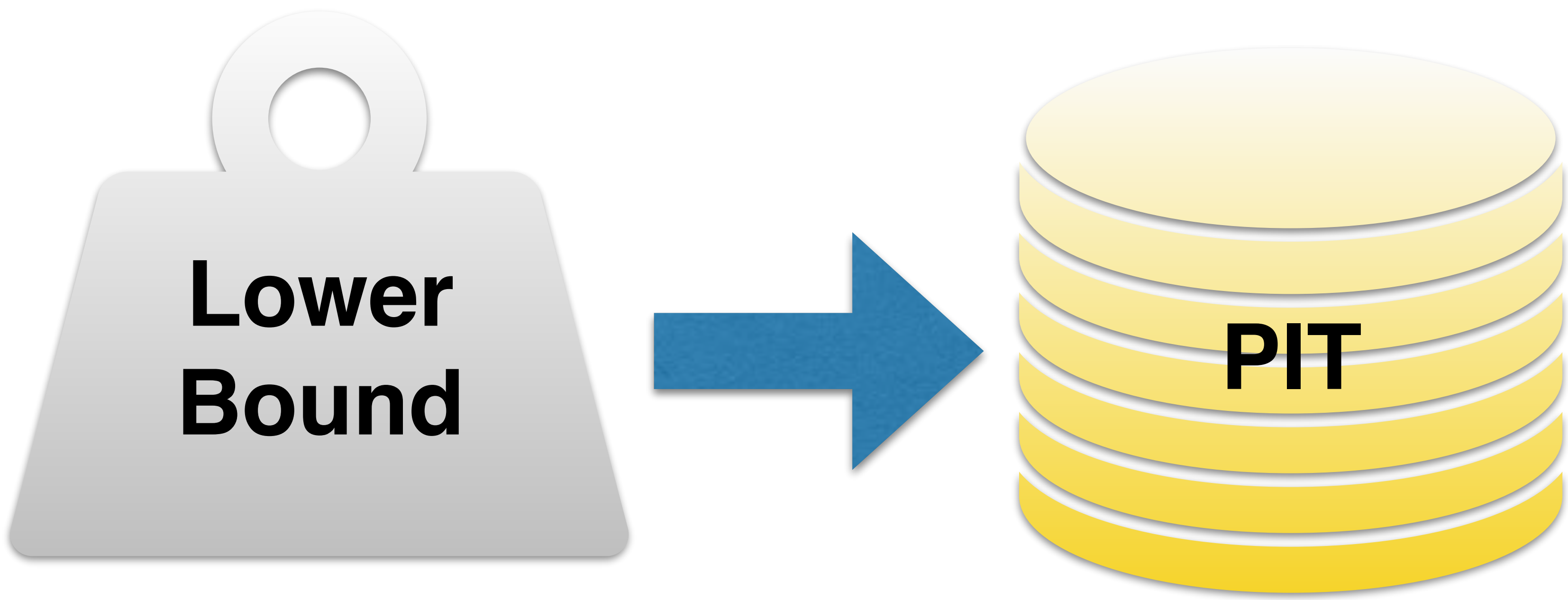


Hardness versus Randomness



- [KI04]: Permanent not in $\mathbf{VP}$ => PIT for $\mathbf{VP}$

Hardness versus Randomness



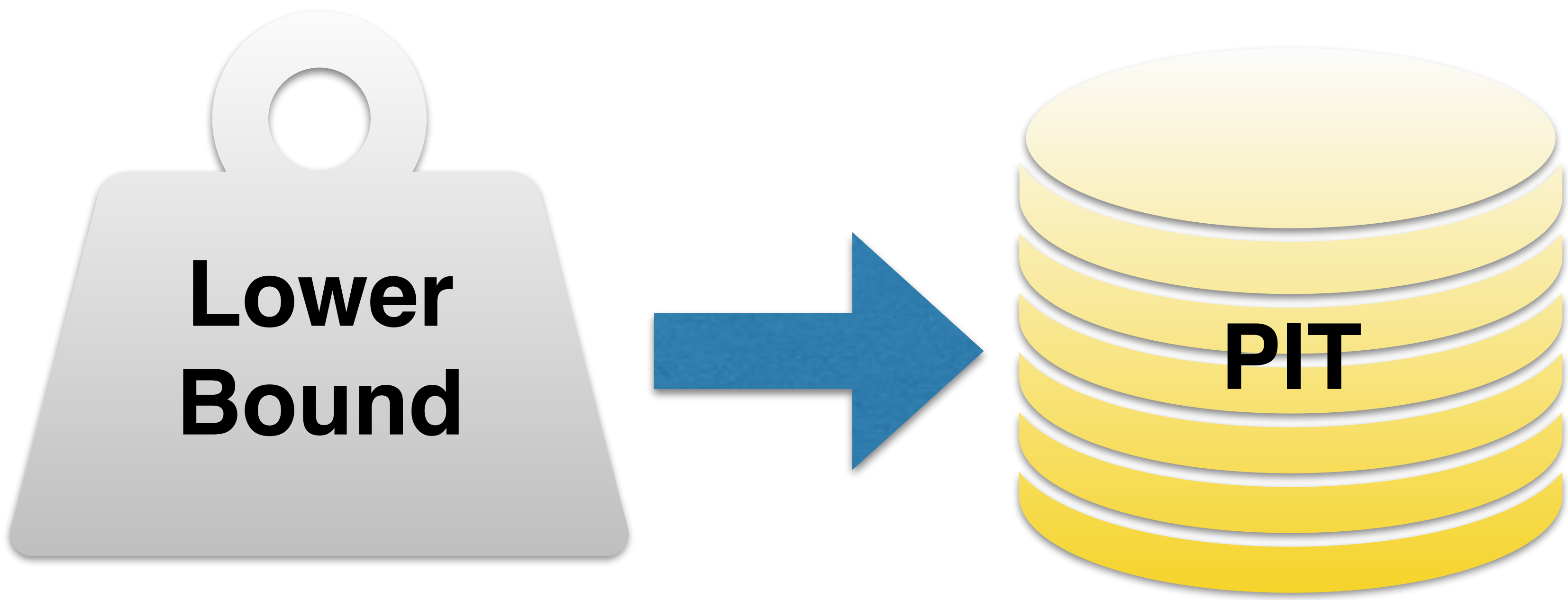
- [KI04]: Permanent not in VP => PIT for VP

multilinear

with bounded individual degree

- [DSY09]: $\omega(\mathrm{poly}(n))$ for $\text{Depth-}\Delta$ => PIT for $\text{Depth-}\Delta - 5$

Hardness versus Randomness



- [KI04]: Permanent not in VP => PIT for VP

multilinear

with bounded individual degree

- [DSY09]: $\omega(\mathrm{poly}(n))$ for $\text{Depth-}\Delta$ => PIT for $\text{Depth-}\Delta - 5$

multilinear with degree $O(\log^2 n/\log^2 \log n)$

- [**C**KS18]: $\omega(\mathrm{poly}(n))$ for $\text{Depth-}\Delta$ => PIT for $\text{Depth-}\Delta - 5$

Lower Bounds (Hardness)

Lower Bounds (Hardness)

Goal: Find an explicit $\{f_n\}$ such that $\{f_n\} \notin \mathcal{C}$.

Lower Bounds (Hardness)

Goal: Find an explicit $\{f_n\}$ such that $\{f_n\} \notin \mathcal{C}$.

- [S73, BS83] An $\Omega(n \log n)$ lower bound for general arithmetic circuits.

Lower Bounds (Hardness)

Goal: Find an explicit $\{f_n\}$ such that $\{f_n\} \notin \mathcal{C}$.

- [S73, BS83] An $\Omega(n \log n)$ lower bound for general arithmetic circuits.
- [NW97, KS14, KLSS14, FLMS14, KS17] Exponential lower bounds for **homogeneous** depth-3, depth-4, depth-5 circuits.

Lower Bounds (Hardness)

Goal: Find an explicit $\{f_n\}$ such that $\{f_n\} \notin \mathcal{C}$.

- [S73, BS83] An $\Omega(n \log n)$ lower bound for general arithmetic circuits.
- [NW97, KS14, KLSS14, FLMS14, KS17] Exponential lower bounds for **homogeneous** depth-3, depth-4, depth-5 circuits.
- [R10] $n^{1+\Omega(1)}$ lower bound for constant depth circuits.

Lower Bounds (Hardness)

Goal: Find an explicit $\{f_n\}$ such that $\{f_n\} \notin \mathcal{C}$.

- [S73, BS83] An $\Omega(n \log n)$ lower bound for general arithmetic circuits.
- [NW97, KS14, KLSS14, FLMS14, KS17] Exponential lower bounds for **homogeneous** depth-3, depth-4, depth-5 circuits.
- [R10] $n^{1+\Omega(1)}$ lower bound for constant depth circuits.
- No lower bounds for VP and $\mathrm{Depth}\text{-}\Delta$.

Lower Bounds (Hardness)

Goal: Find an explicit $\{f_n\}$ such that $\{f_n\} \notin \mathcal{C}$.

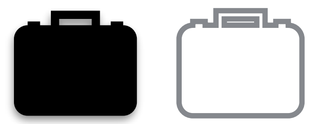
- [S73, BS83] An $\Omega(n \log n)$ lower bound for general arithmetic circuits.
- [NW97, KS14, KLSS14, FLMS14, KS17] Exponential lower bounds for **homogeneous** depth-3, depth-4, depth-5 circuits.
- [R10] $n^{1+\Omega(1)}$ lower bound for constant depth circuits.
- No lower bounds for VP and $\mathrm{Depth}\text{-}\Delta$.

PIT (Randomness)

PIT (Randomness)

Goal: Given $f \in \mathcal{C}$, determine whether $f \equiv 0$.

PIT (Randomness)



Goal: Given $f \in \mathcal{C}$, determine whether $f \equiv 0$.

PIT (Randomness)

Goal: Given $f \in \mathcal{C}$, determine whether $f \equiv 0$.

- Easy when using *randomness*: Schwartz-Zippel.

PIT (Randomness)

Goal: Given $f \in \mathcal{C}$, determine whether $f \equiv 0$.

- Easy when using *randomness*: Schwartz-Zippel.

sub-exponential time

- No non-trivial *deterministic* PIT for VP and $\mathrm{Depth}\text{-}\Delta$.

PIT (Randomness)

Goal: Given $f \in \mathcal{C}$, determine whether $f \equiv 0$.

- Easy when using *randomness*: Schwartz-Zippel.
- No non-trivial *deterministic* PIT for VP and $\mathrm{Depth}\text{-}\Delta$.

sub-exponential time

PIT (Randomness)

Goal: Given $f \in \mathcal{C}$, determine whether $f \equiv 0$.

- Easy when using *randomness*: Schwartz-Zippel.

sub-exponential time

- No non-trivial *deterministic* PIT for VP and $\mathrm{Depth}\text{-}\Delta$.

PIT = Hitting Set

PIT (Randomness)

Goal: Given $f \in \mathcal{C}$, determine whether $f \equiv 0$.

- Easy when using *randomness*: Schwartz-Zippel.

sub-exponential time

- No non-trivial *deterministic* PIT for VP and $\mathrm{Depth}\text{-}\Delta$.

PIT = Hitting Set

$\mathcal{P}$ is a hitting set for $\mathcal{C}$ if for any **non-zero** $f \in \mathcal{C}$

$$\exists \mathbf{a} \in \mathcal{P}, \; f(\mathbf{a}) \neq 0.$$

PIT (Randomness)

Goal: Explicitly construct a hitting set $\mathcal{P}$ for $\mathcal{C}$.

PIT = Hitting Set

$\mathcal{P}$ is a hitting set for $\mathcal{C}$ if for any **non-zero** $f \in \mathcal{C}$

$$\exists \mathbf{a} \in \mathcal{P}, \ f(\mathbf{a}) \neq 0.$$

PIT (Randomness)

Goal: Explicitly construct a hitting set $\mathcal{P}$ for $\mathcal{C}$.

- Running time is $\mathrm{poly}(n, |\mathcal{P}|)$.

PIT = Hitting Set

$\mathcal{P}$ is a hitting set for $\mathcal{C}$ if for any **non-zero** $f \in \mathcal{C}$

$$\exists \mathbf{a} \in \mathcal{P},\ \ f(\mathbf{a}) \neq 0.$$

PIT (Randomness)

Goal: Explicitly construct a hitting set $\mathcal{P}$ for $\mathcal{C}$.

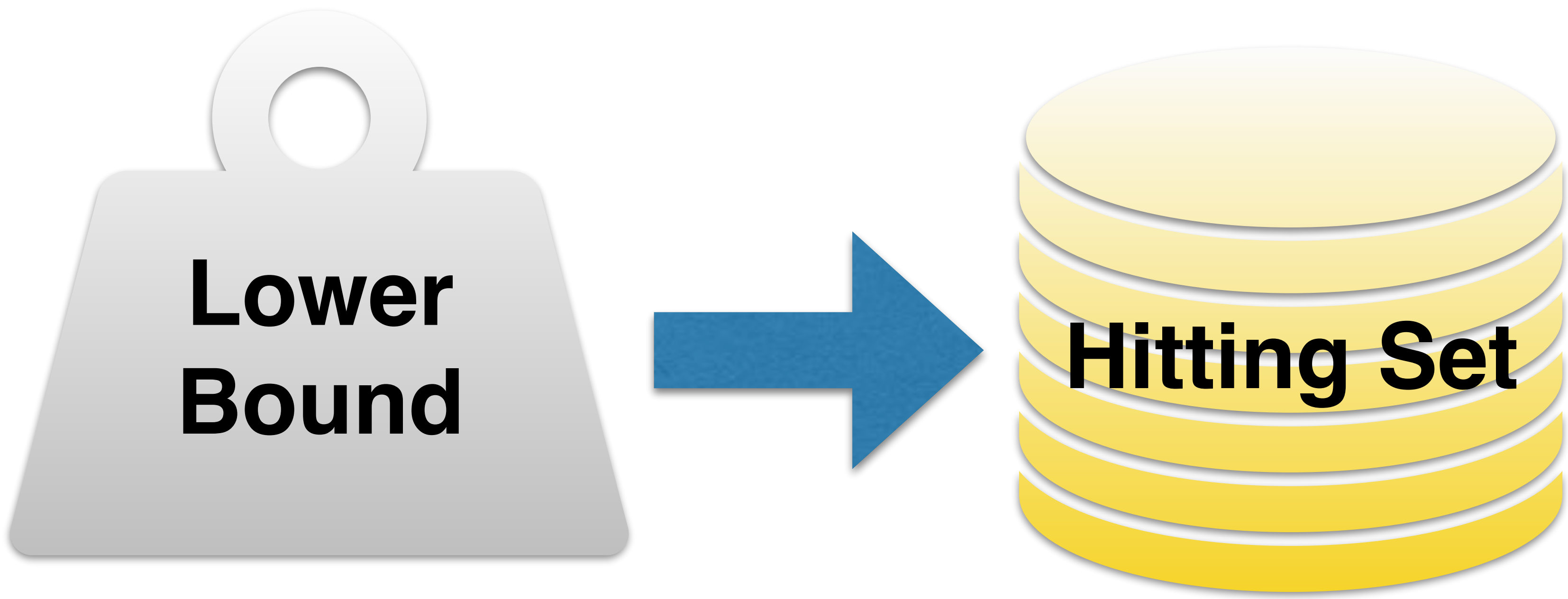
- Running time is $\mathrm{poly}(n, |\mathcal{P}|)$.
- No sub-exponential size hitting set for VP, $\mathrm{Depth}\text{-}\Delta$.

PIT = Hitting Set

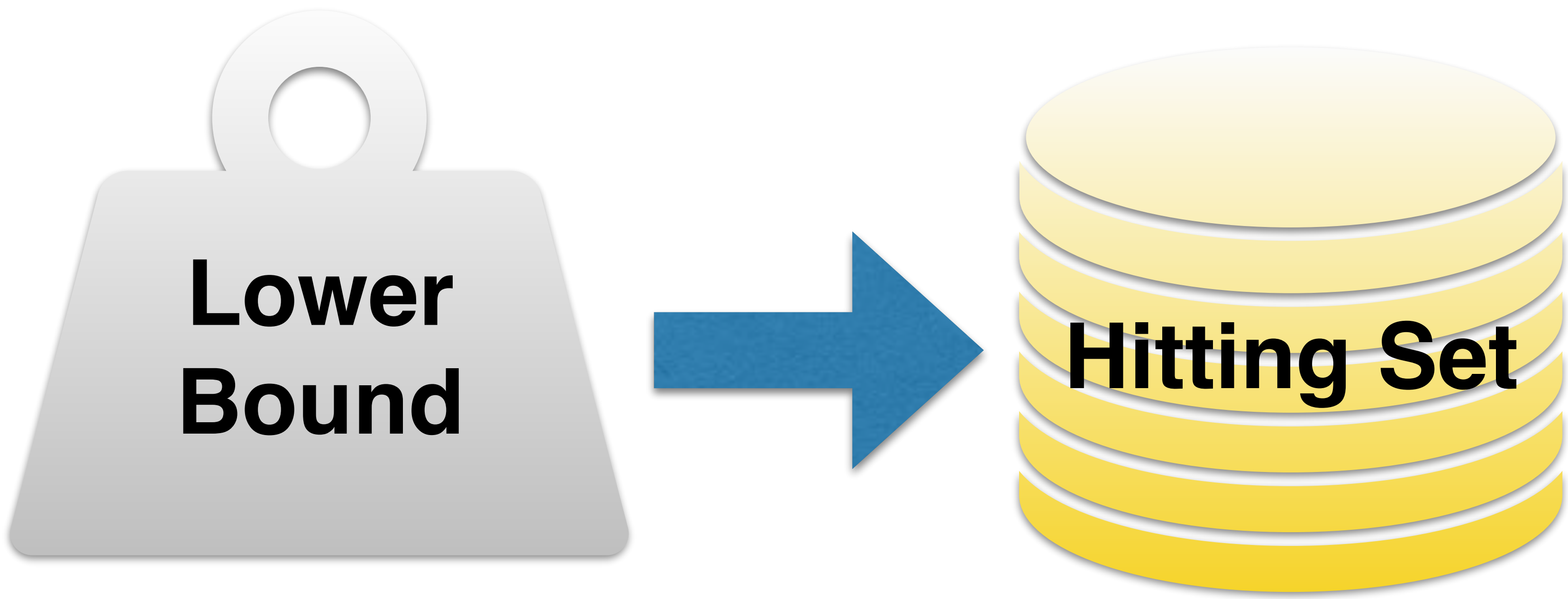
$\mathcal{P}$ is a hitting set for $\mathcal{C}$ if for any **non-zero** $f \in \mathcal{C}$

$$\exists \mathbf{a} \in \mathcal{P},\ \ f(\mathbf{a}) \neq 0.$$

Hardness versus Randomness framework [KI04, DSY09, **C**KS18]



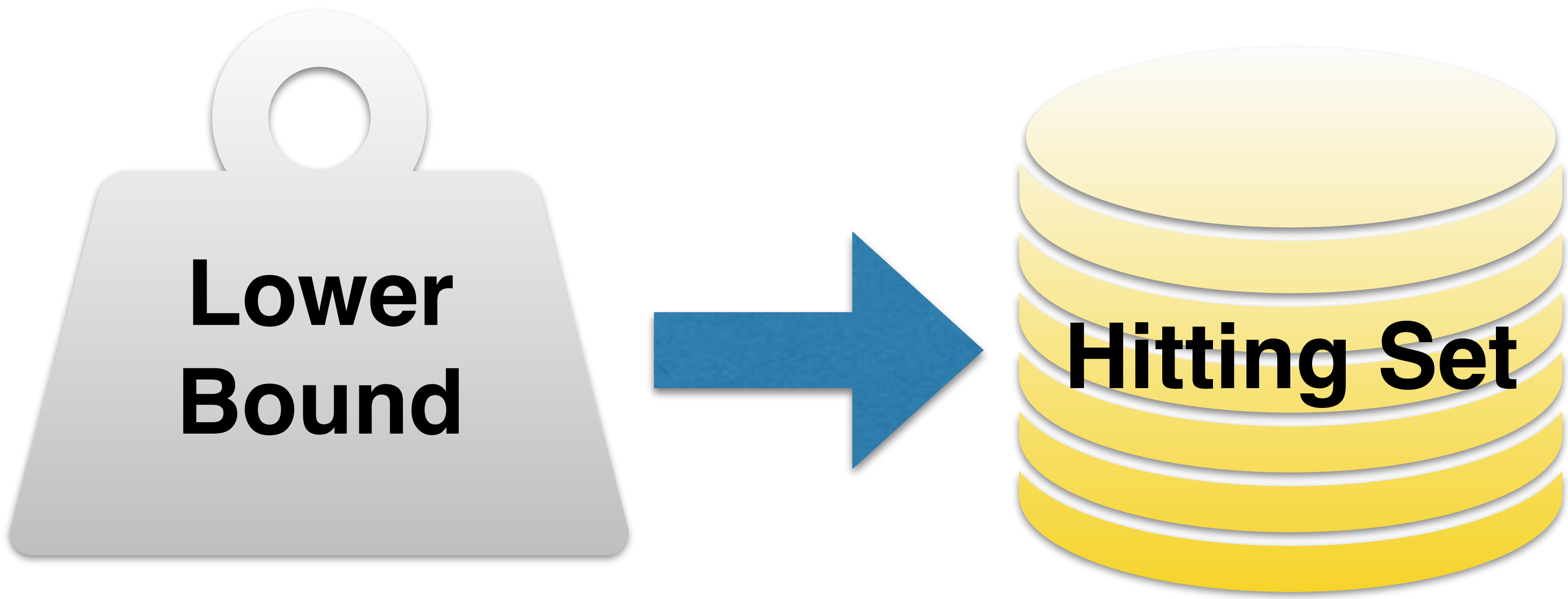
Hardness versus Randomness framework [KI04, DSY09, **C**KS18]



Nisan-Wigderson generator

Reduce #variables from $n \to \ell$

Hardness versus Randomness framework [KI04, DSY09, **C**KS18]



Nisan-Wigderson generator

Schwartz-Zippel lemma

Reduce #variables from $n \to \ell$

Brute-force to find hitting set in time $d^{O(\ell)}$

Reducing #Variables

Reducing #Variables

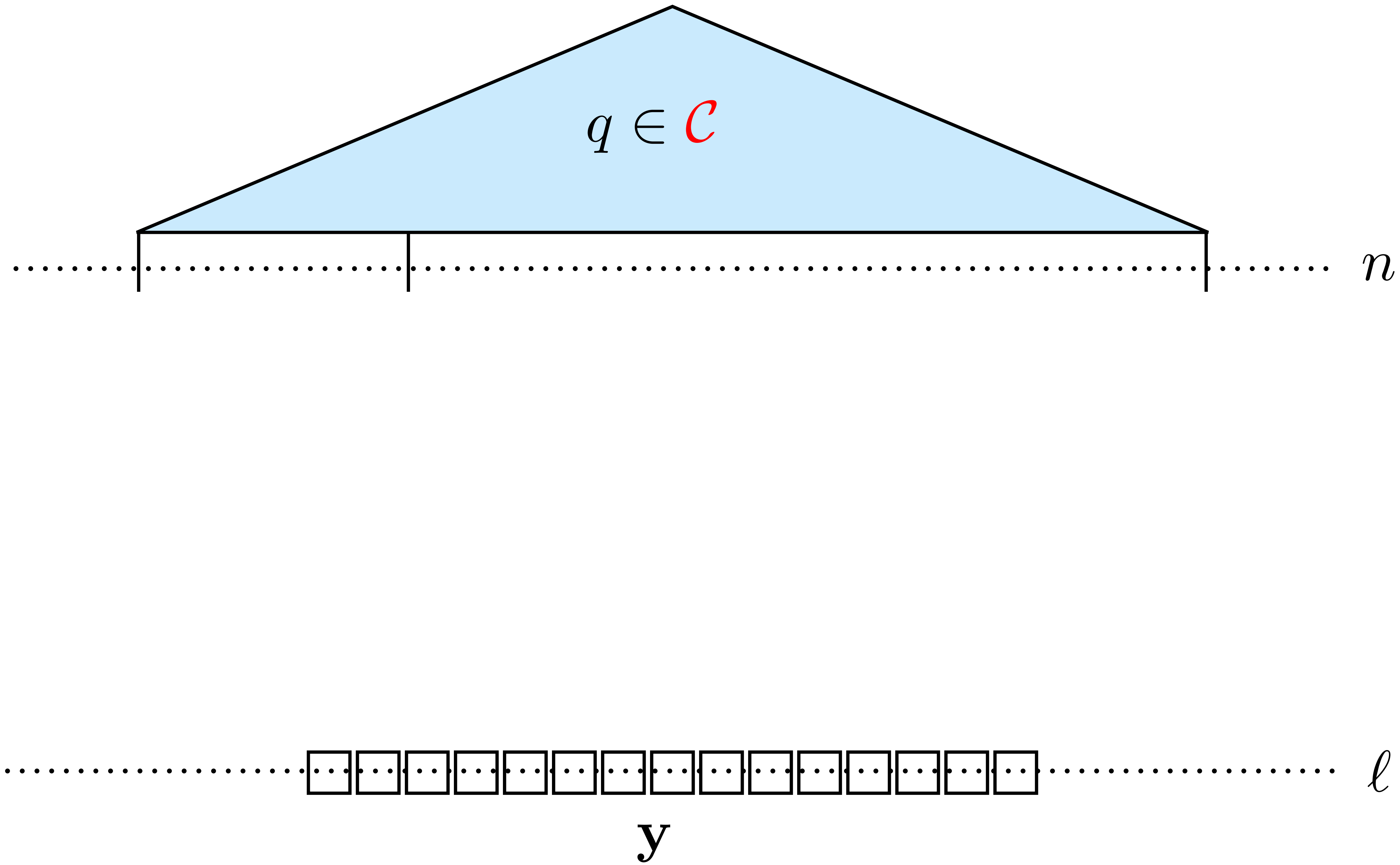
Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$

Reducing #Variables

Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$

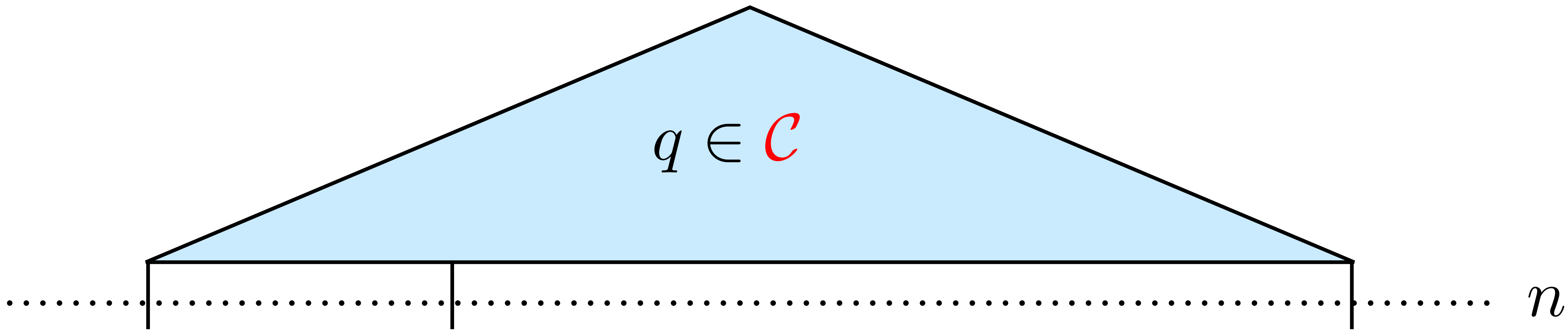
Reducing #Variables

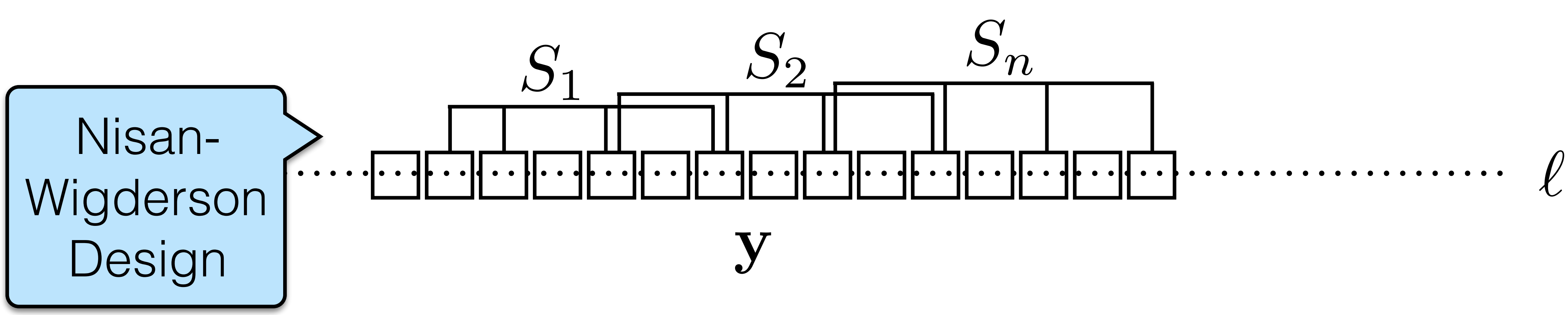
Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$



Reducing #Variables

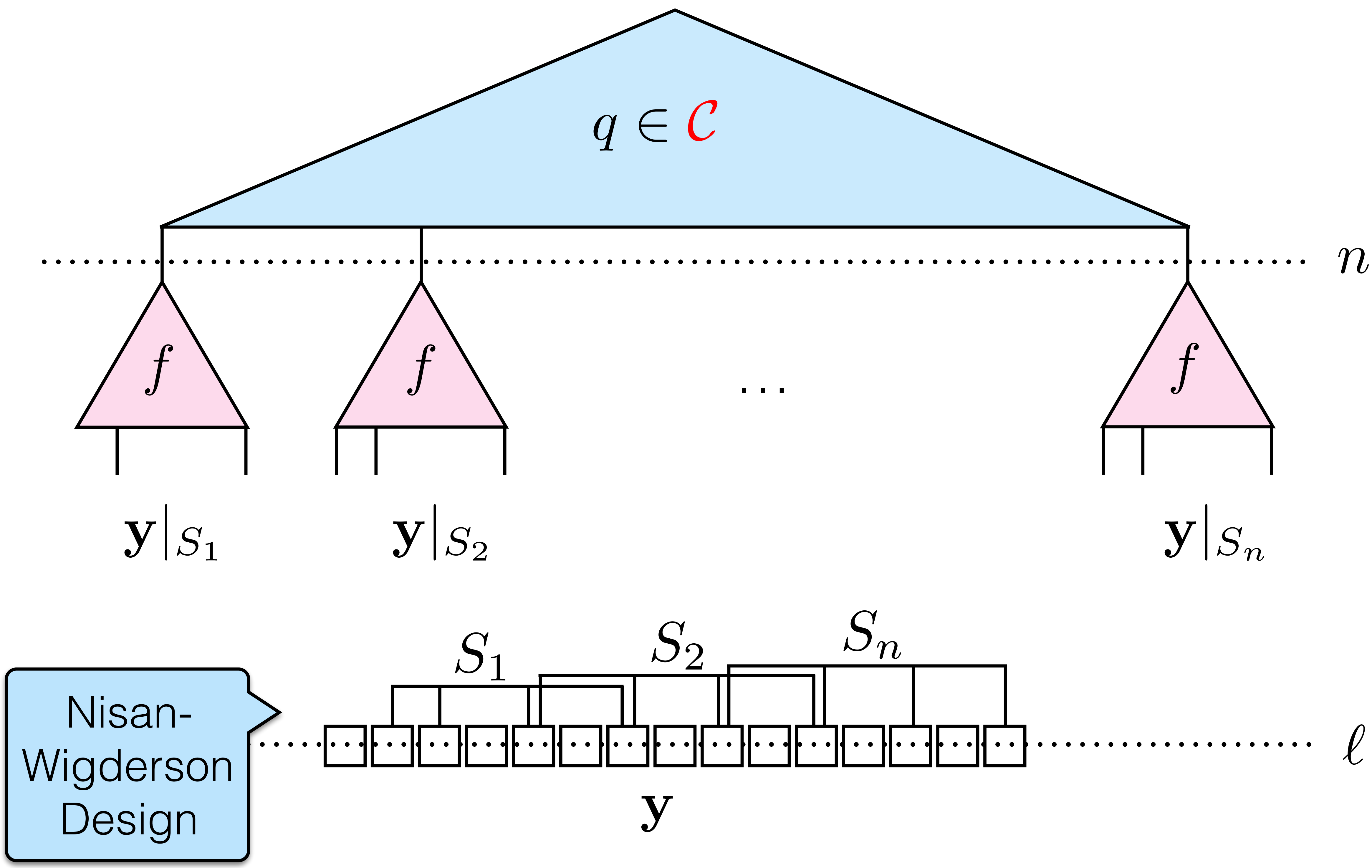
Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$





Reducing #Variables

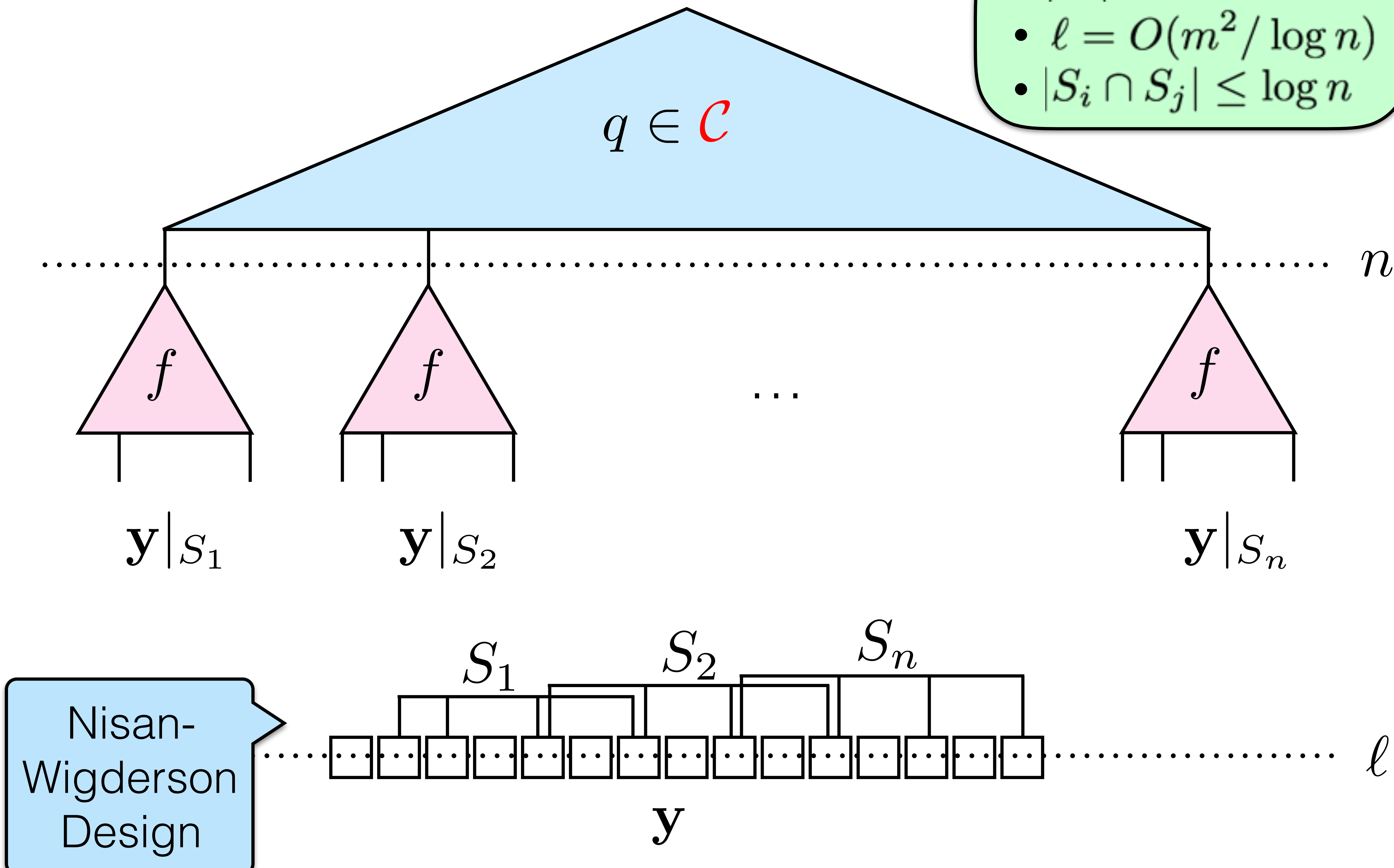
Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$



Reducing #Variables

Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$

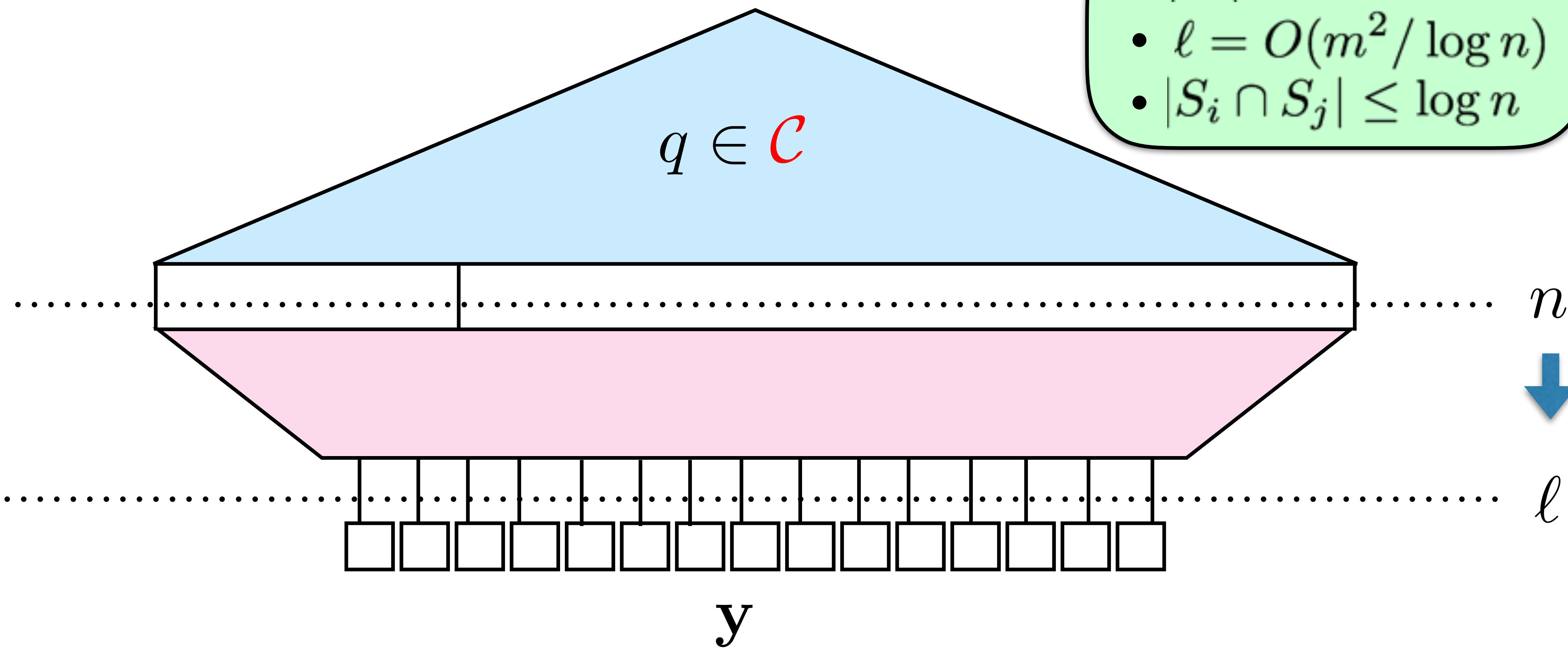
- $|S_i| = m$
- $\ell = O(m^2 / \log n)$
- $|S_i \cap S_j| \leq \log n$



Reducing #Variables

Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$

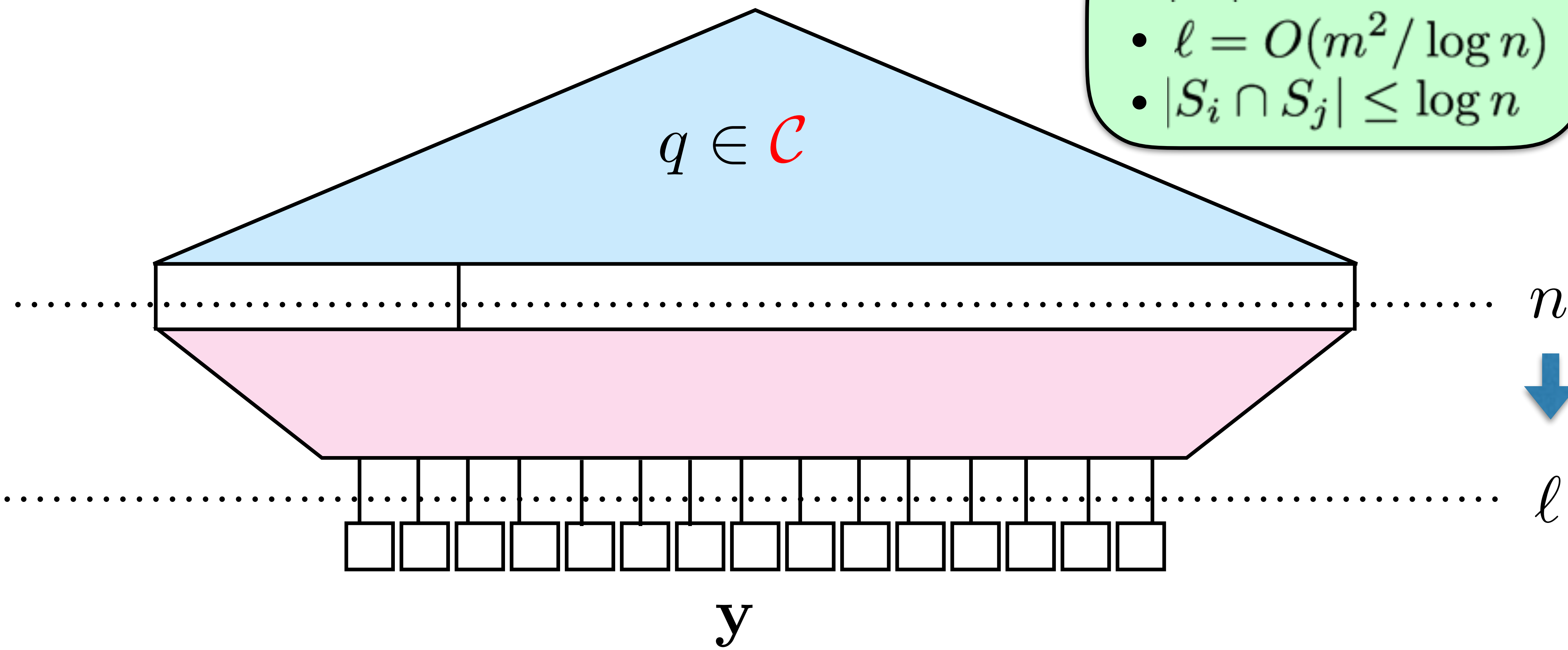
- $|S_i| = m$
- $\ell = O(m^2 / \log n)$
- $|S_i \cap S_j| \leq \log n$



Reducing #Variables

Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$

- $|S_i| = m$
- $\ell = O(m^2 / \log n)$
- $|S_i \cap S_j| \leq \log n$

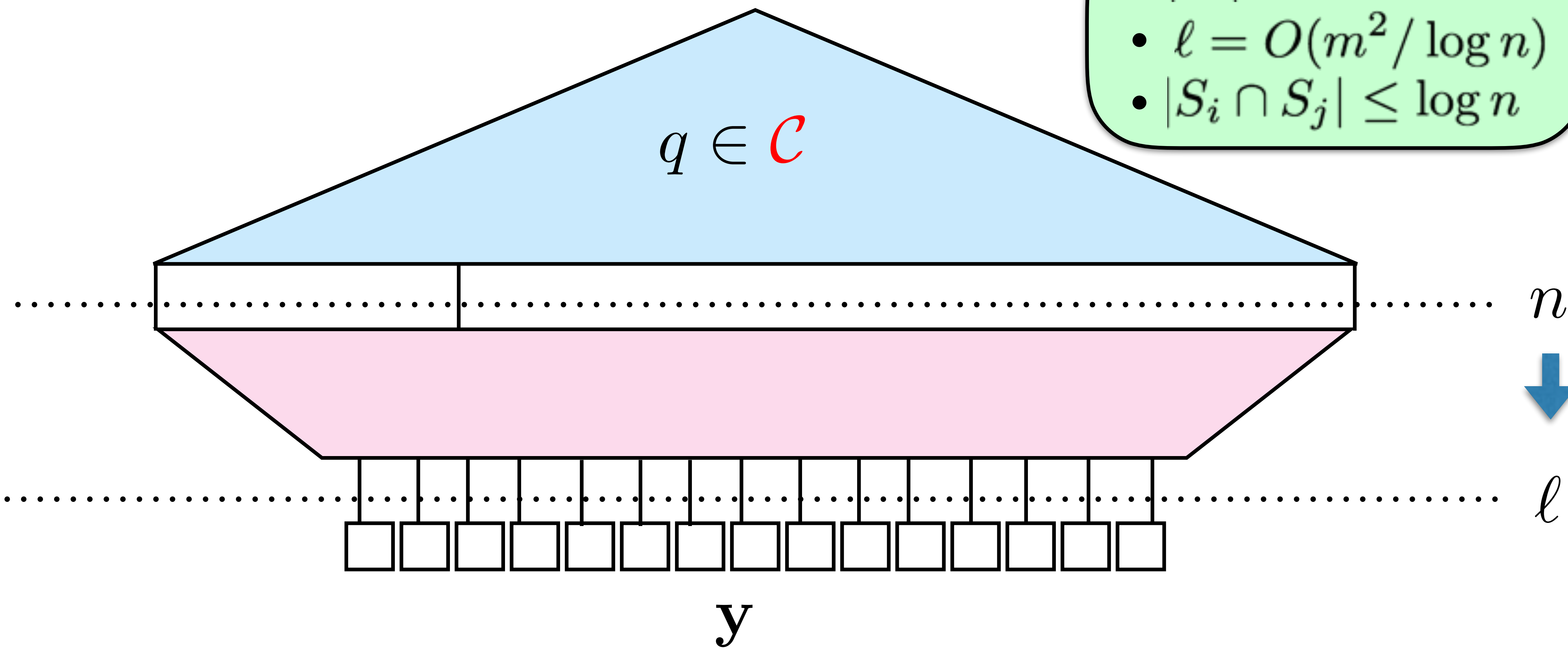


$$Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big)$$

Reducing #Variables

Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$

- $|S_i| = m$
- $\ell = O(m^2/\log n)$
- $|S_i \cap S_j| \leq \log n$



$$Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big)$$

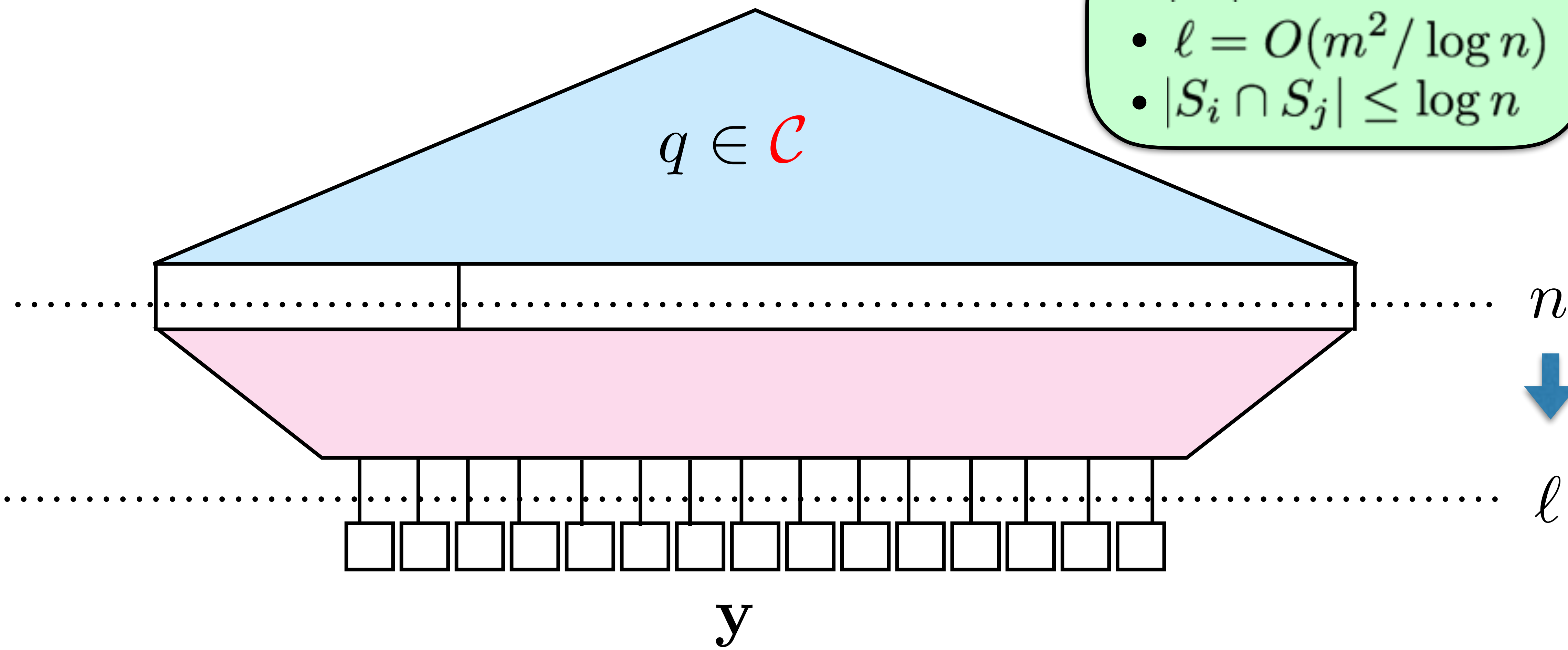
Lemma [Schwartz, Zippel]

There exists hitting set of size $d^{O(\ell)}$ for degree d ℓ-variate polynomials.

Reducing #Variables

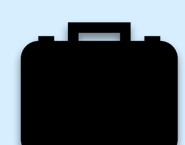
Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$

- $|S_i| = m$
- $\ell = O(m^2 / \log n)$
- $|S_i \cap S_j| \leq \log n$



$$Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big)$$

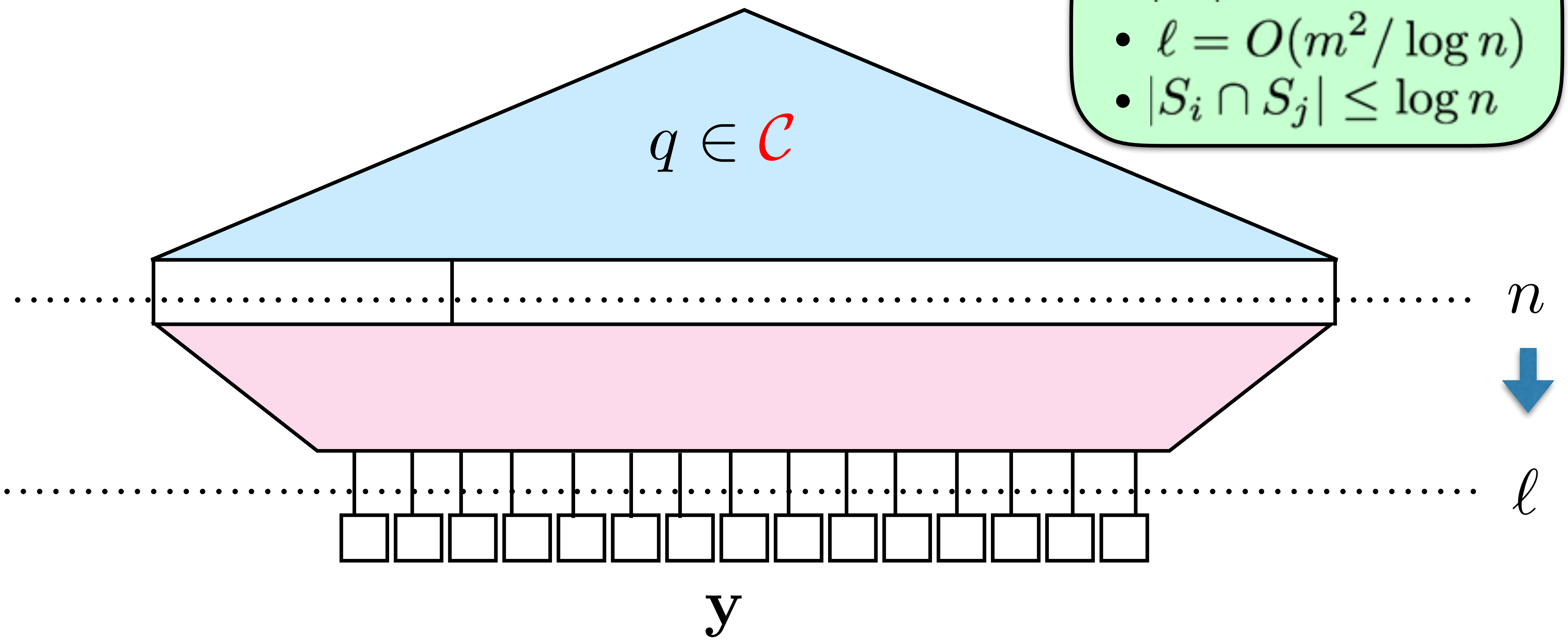
Lemma [Schwartz, Zippel]

$\ell = o(n) \quad \Rightarrow \quad 2^{o(n)}$ time  PIT

Reducing #Variables

Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$ for $\mathcal{C}$

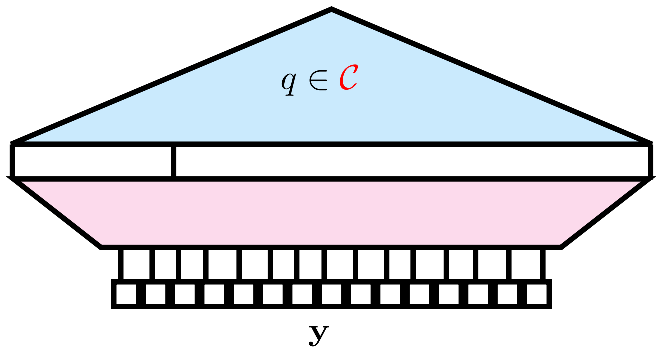
- $|S_i| = m$
- $\ell = O(m^2/\log n)$
- $|S_i \cap S_j| \le \log n$



$$Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big)$$

Want: If $q \neq 0$, then $Q \neq 0$.

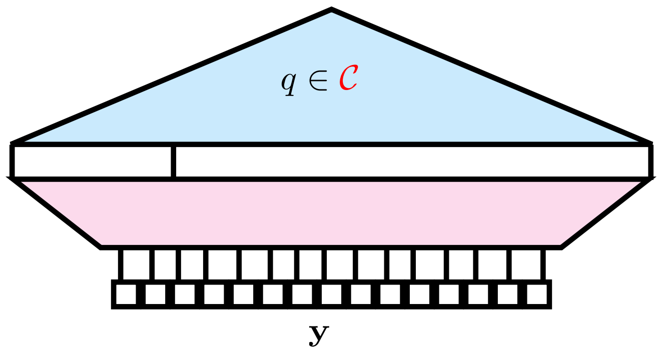
Key Lemma



$Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big)$

Goal: If $q \not\equiv 0$, then $Q \not\equiv 0$.

Key Lemma



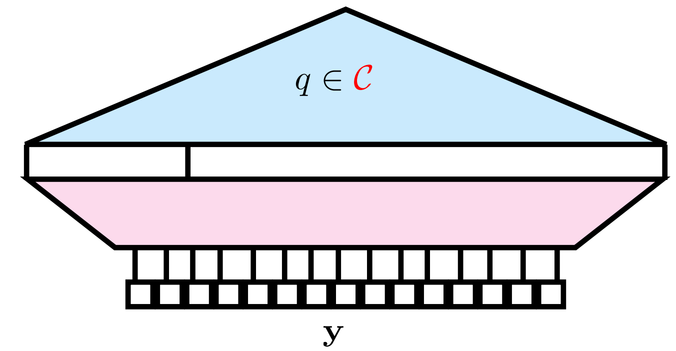
$Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \dots, f(\mathbf{y}|_{S_n})\big)$

Goal: If $q \not\equiv 0$, then $Q \not\equiv 0$.

Proof structure:

Suppose $Q \equiv 0$, then f can be computed by a small circuit.

Key Lemma



$Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big)$

Goal: If $q \not\equiv 0$, then $Q \not\equiv 0$.

Proof structure:

Suppose $Q \equiv 0$, then f can be computed by a small circuit.

To show f having a small circuit, we need **polynomial factorization**!

Proof Sketch of the Key Lemma

$\exists q \in \mathcal{C},\ Q \equiv 0$

$\Downarrow$

f has a small circuit

Proof Sketch of the Key Lemma

$\exists q \in \mathcal{C},\ Q \equiv 0$

$\Downarrow$

f has a small circuit

If $Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big) \equiv 0$

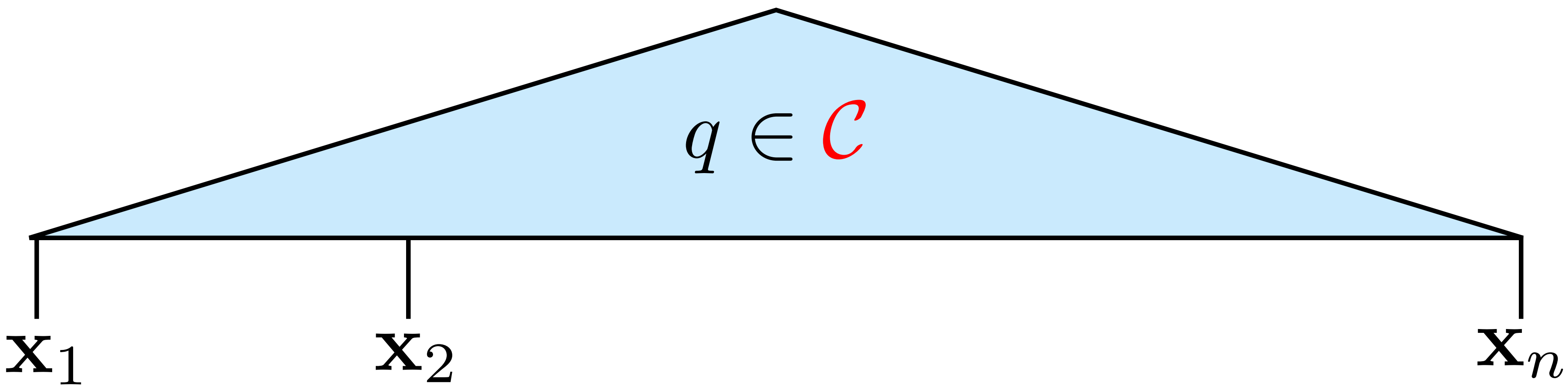
Proof Sketch of the Key Lemma

$\exists q \in \mathcal{C},\ Q \equiv 0$

⬇

f has a small circuit

If $Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big) \equiv 0$



$\not\equiv 0$

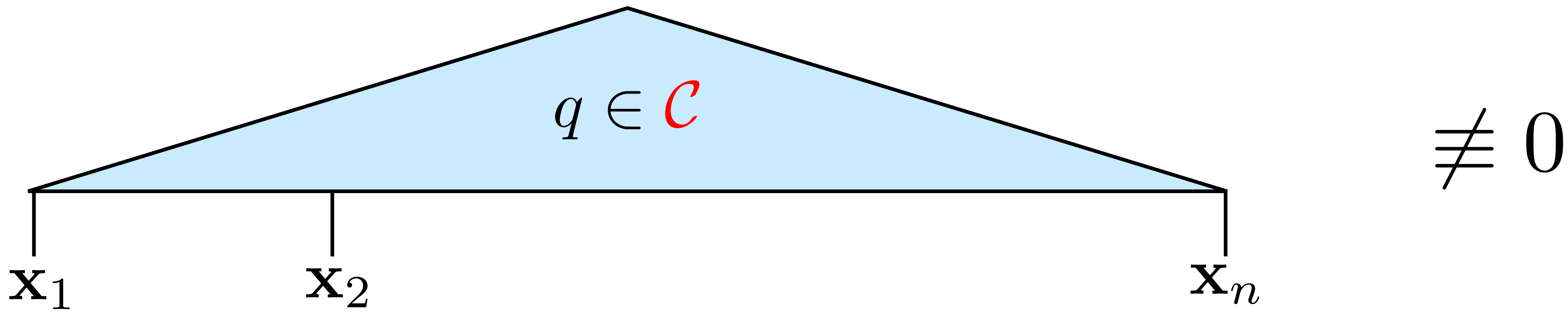
Proof Sketch of the Key Lemma

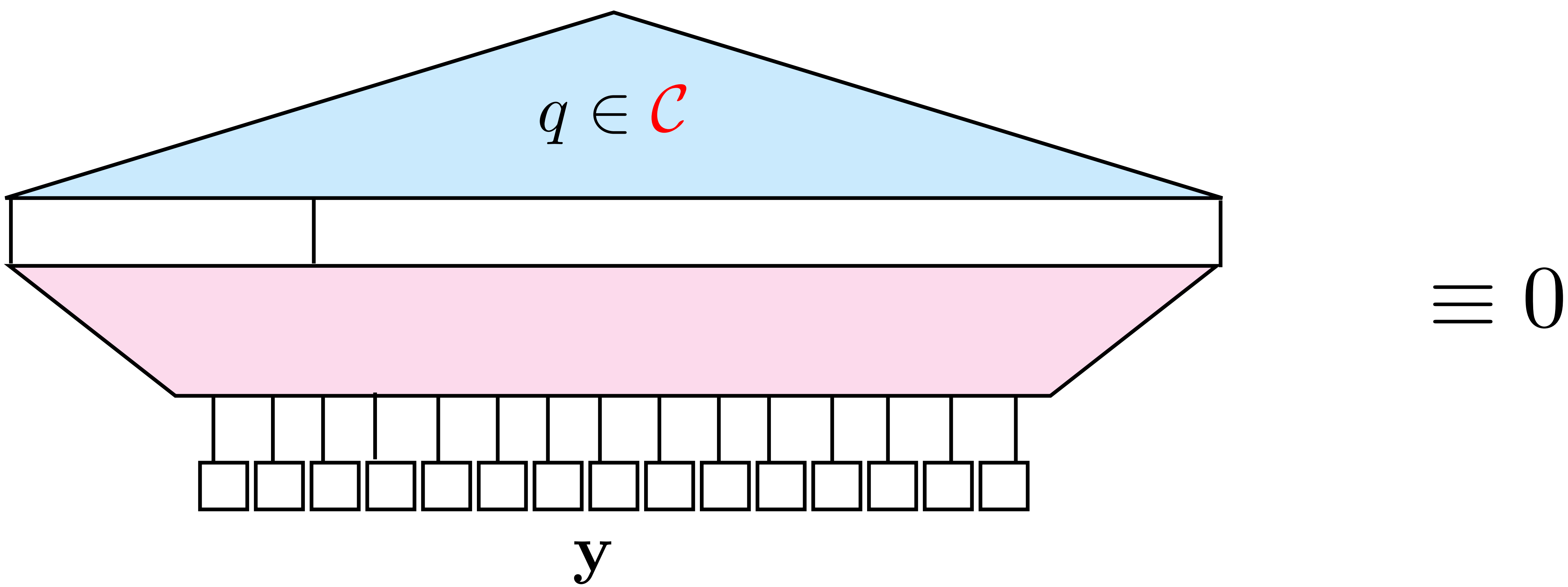
$\exists q \in \mathcal{C},\ Q \equiv 0$

$\Downarrow$

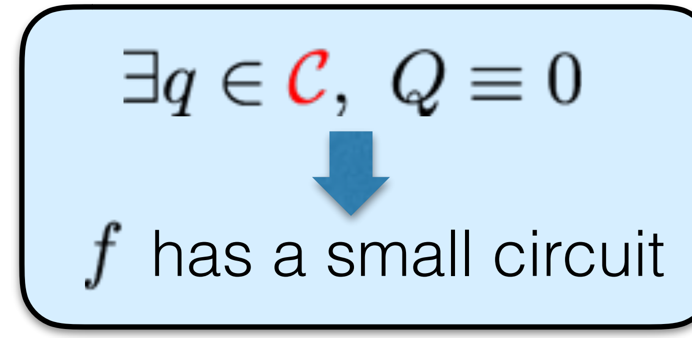
f has a small circuit

If $Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big) \equiv 0$

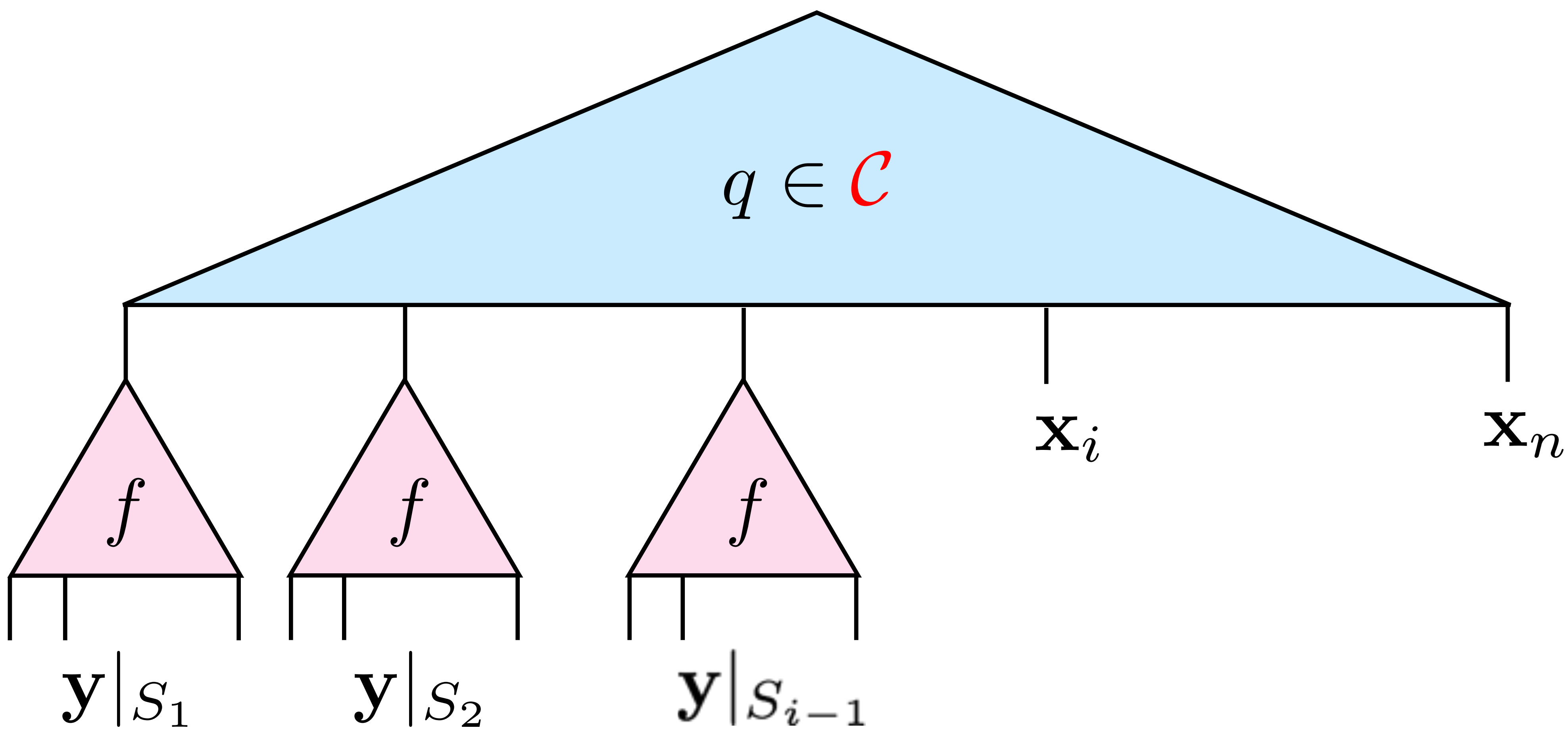




Proof Sketch of the Key Lemma



By hybrid argument, there exists i s.t.



Proof Sketch of the Key Lemma

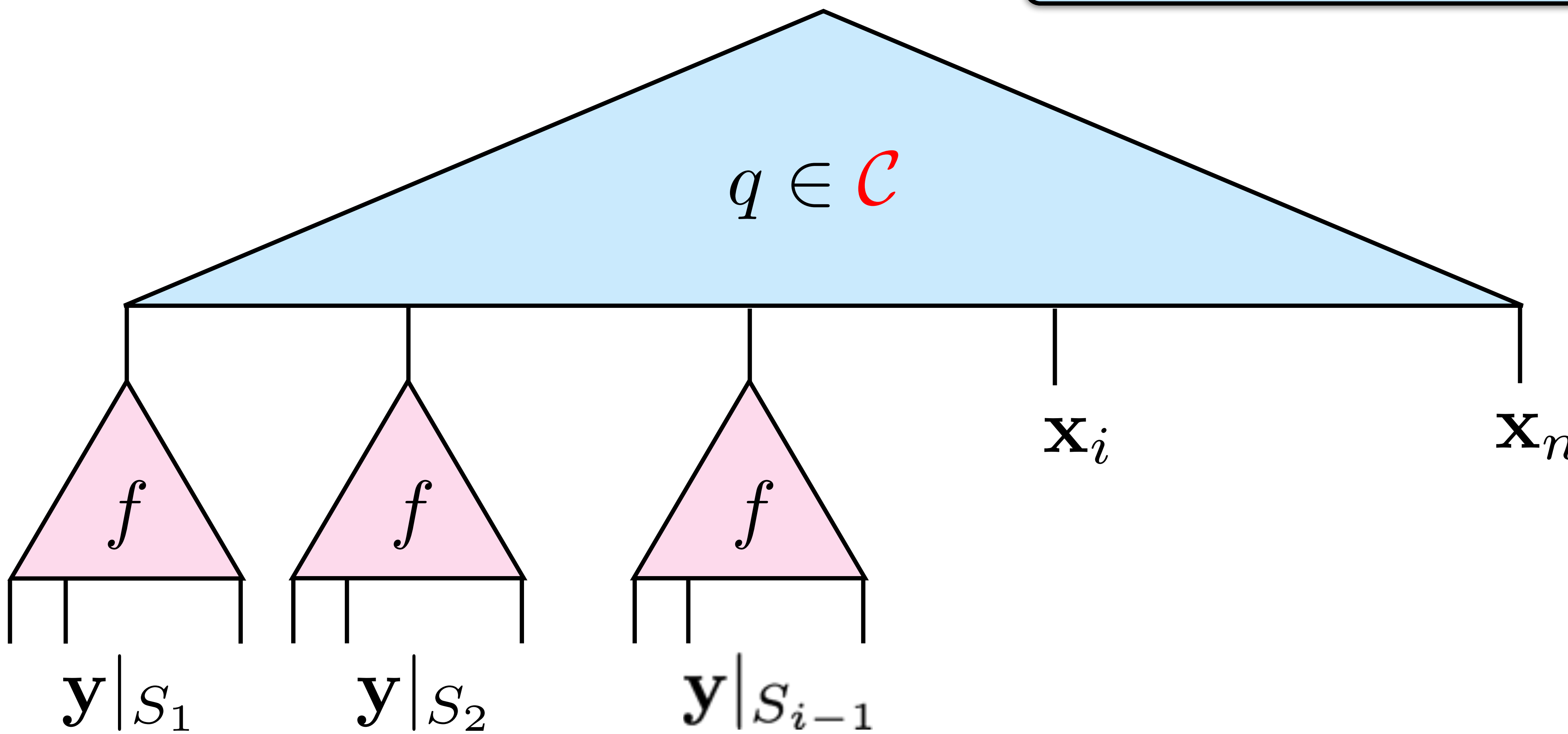
$\exists q \in \mathcal{C},\ Q \equiv 0$

f has a small circuit

By hybrid argument, there exists i s.t.

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i)$

$\mathbf{z} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n, \mathbf{y}\}$



Proof Sketch of the Key Lemma

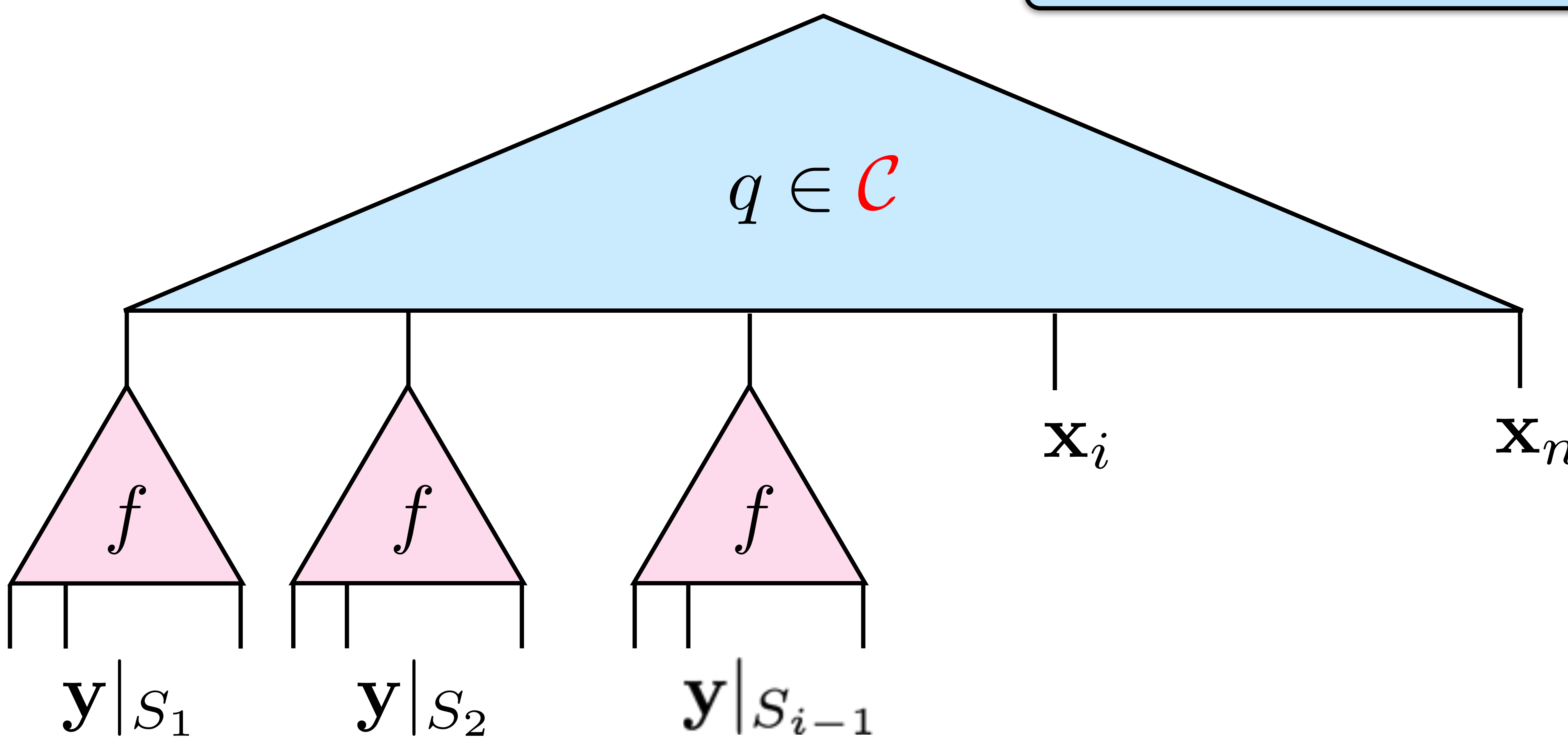
$\exists q \in \mathcal{C},\ Q \equiv 0$

f has a small circuit

By hybrid argument, there exists i s.t.

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i)$

$\mathbf{z} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n, \mathbf{y}\}$



- $\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \not\equiv 0$

Proof Sketch of the Key Lemma

$\exists q \in \mathcal{C},\ Q \equiv 0$

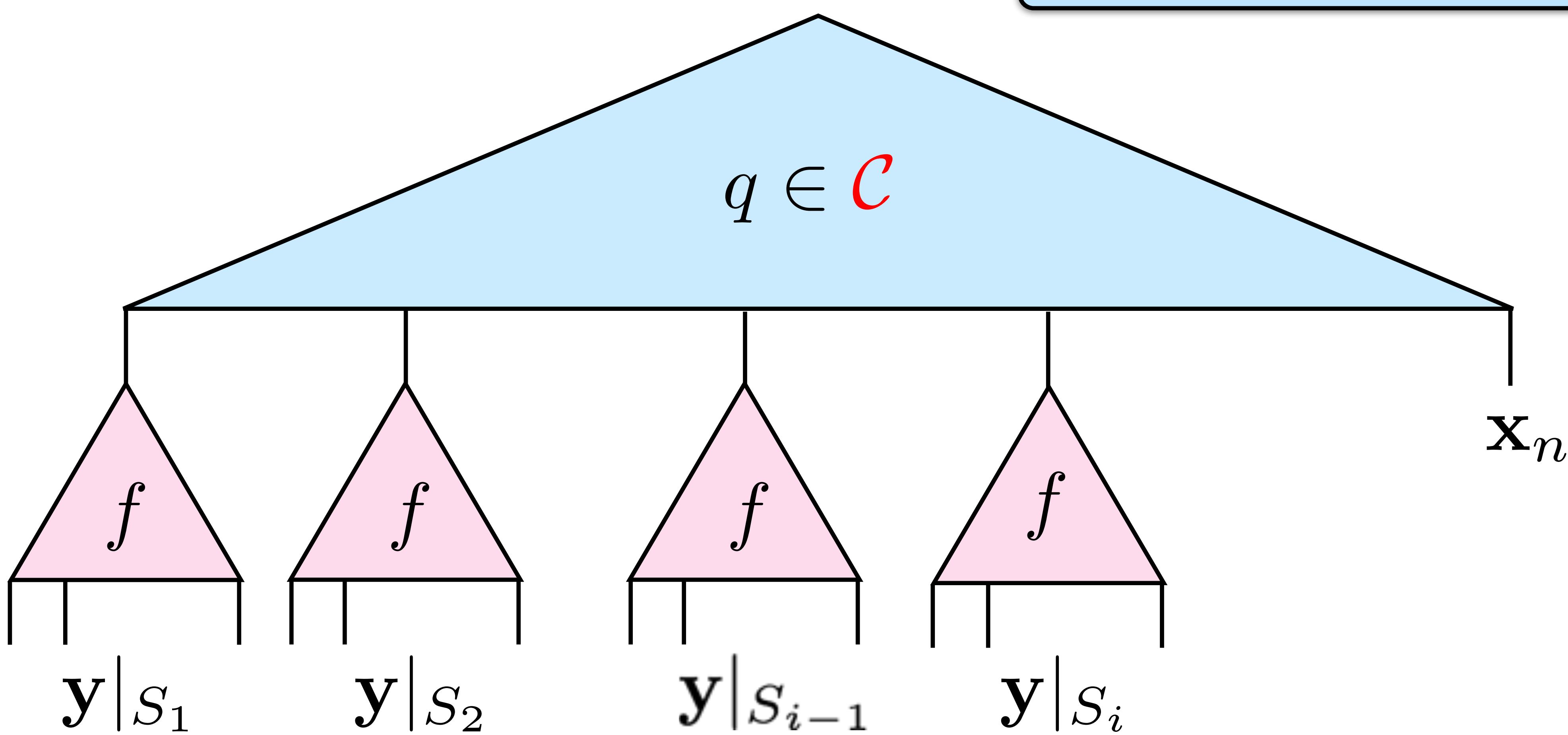
$\Downarrow$

f has a small circuit

By hybrid argument, there exists i s.t.

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i)$

$\mathbf{z} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n, \mathbf{y}\}$



- $\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \not\equiv 0$
- $\tilde{Q}(\mathbf{z}, f(\mathbf{z})) \equiv 0$

Proof Sketch of the Key Lemma

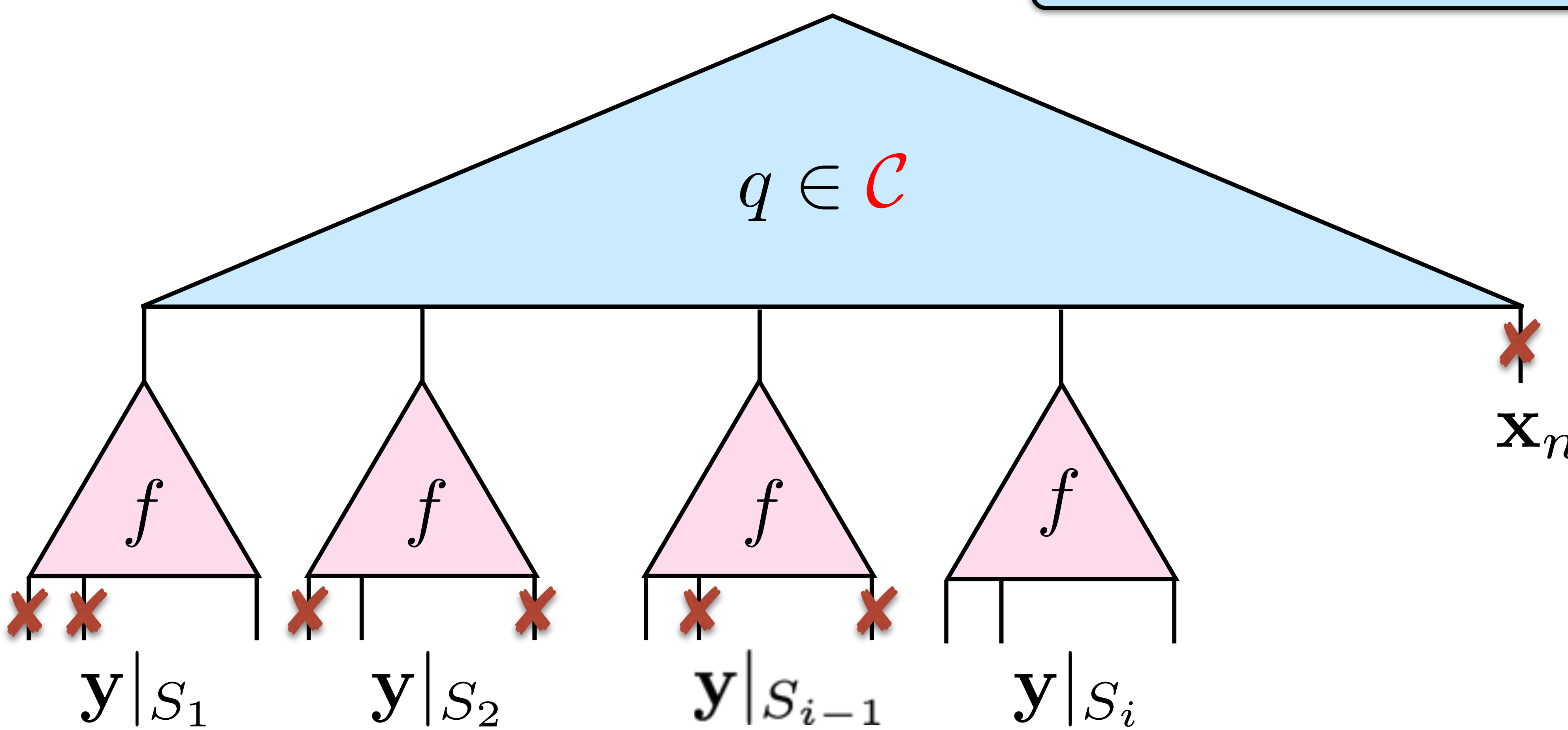
$\exists q \in \mathcal{C},\ Q \equiv 0$

f has a small circuit

By hybrid argument, there exists i s.t.

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i)$

$\mathbf{z} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n, \mathbf{y}\}$



- $\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \not\equiv 0$
- $\tilde{Q}(\mathbf{z}, f(\mathbf{z})) \equiv 0$

Proof Sketch of the Key Lemma

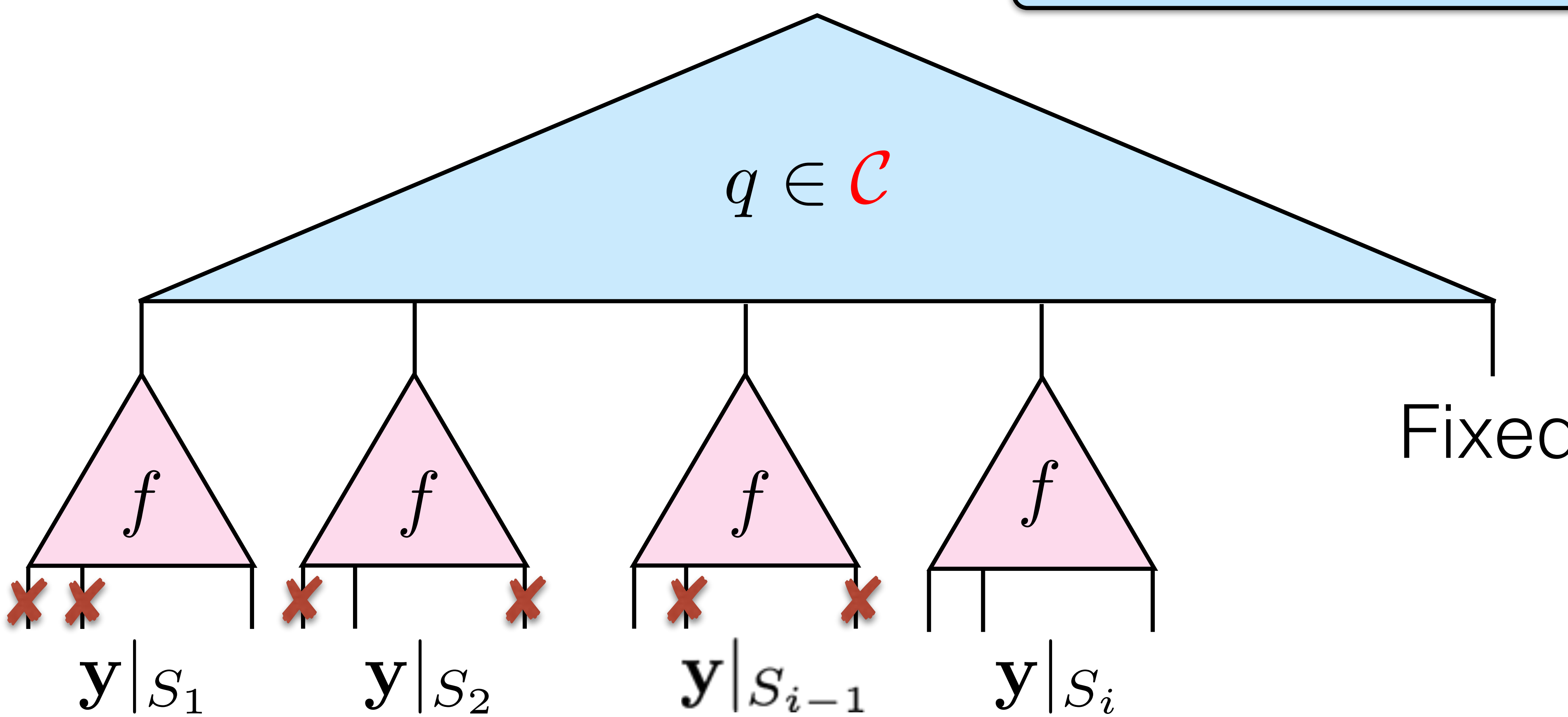
$\exists q \in \mathcal{C},\ Q \equiv 0$

f has a small circuit

By hybrid argument, there exists i s.t.

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i)$

$\mathbf{z} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n, \mathbf{y}\}$



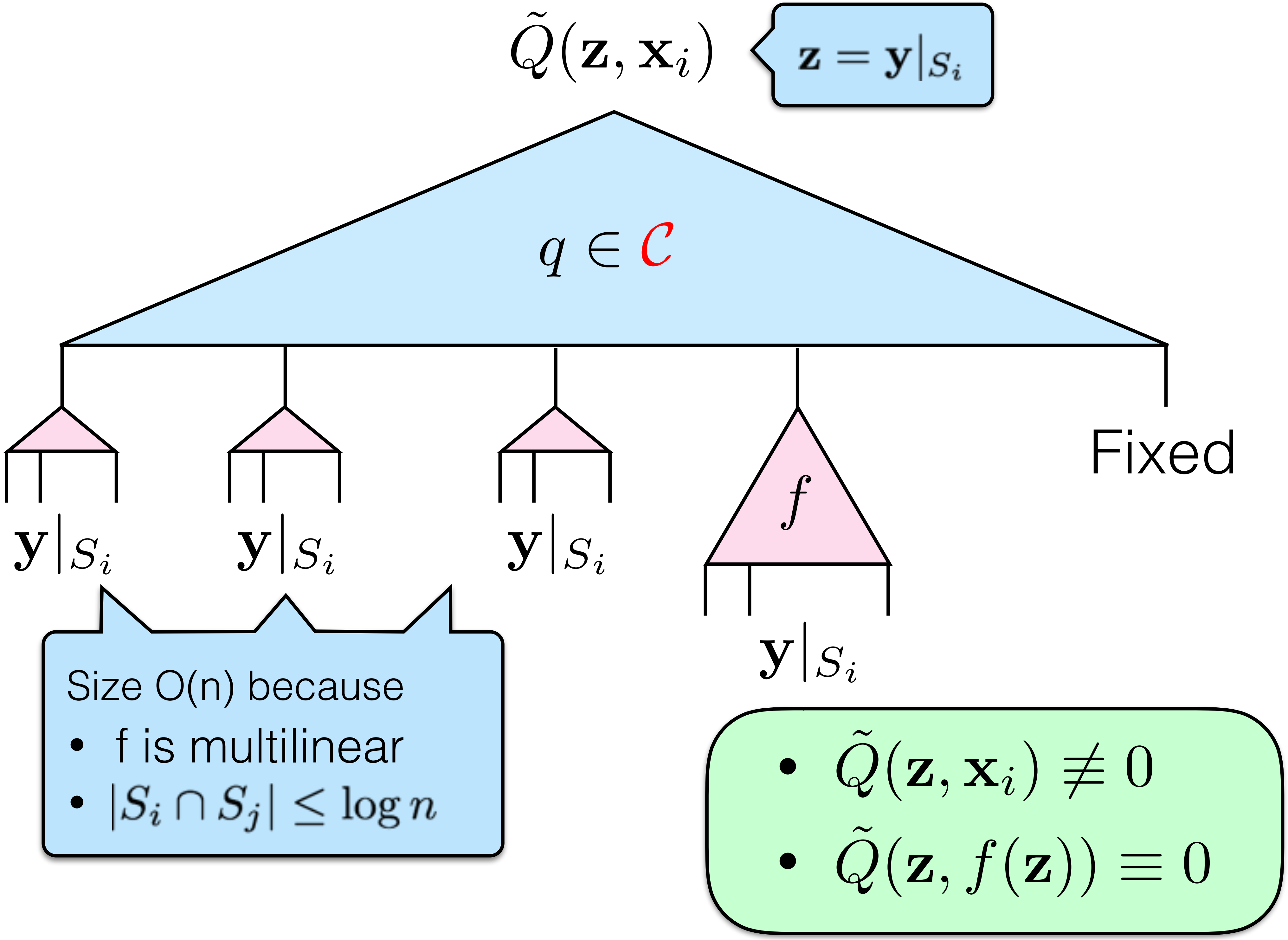
- $\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \not\equiv 0$
- $\tilde{Q}(\mathbf{z}, f(\mathbf{z})) \equiv 0$

Proof Sketch of the Key Lemma

$\exists q \in \mathcal{C},\ Q \equiv 0$

f has a small circuit

By hybrid argument, there exists i s.t.



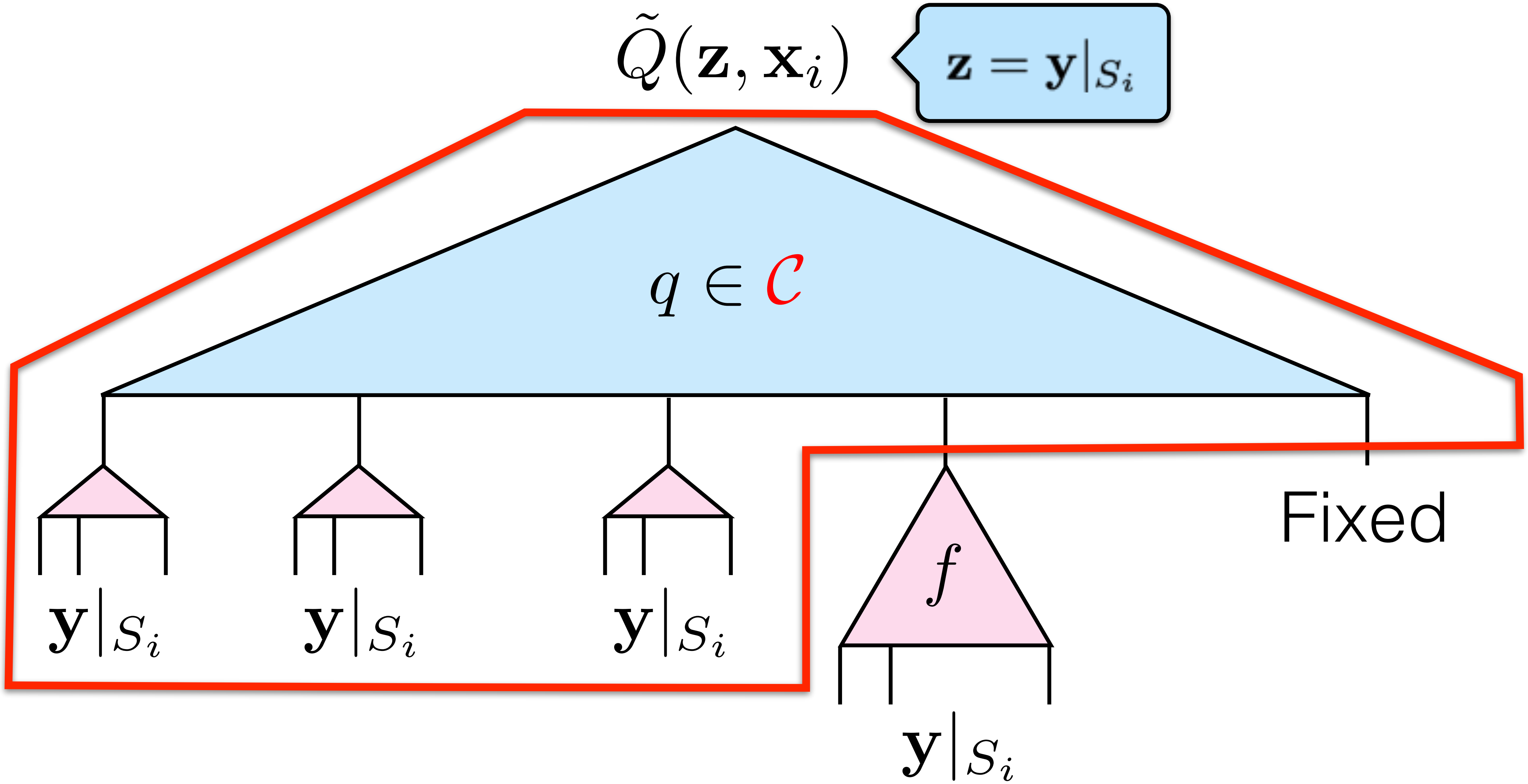
Proof Sketch of the Key Lemma

$\exists q \in \mathcal{C}, \; Q \equiv 0$

$\Downarrow$

f has a small circuit

By hybrid argument, there exists i s.t.



$\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \in \mathcal{C}$

- $\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \not\equiv 0$
- $\tilde{Q}(\mathbf{z}, f(\mathbf{z})) \equiv 0$

Reducing to Polynomial Factorization

$$\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \in \mathcal{C}$$

- $\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \not\equiv 0$
- $\tilde{Q}(\mathbf{z}, f(\mathbf{z})) \equiv 0$

Reducing to Polynomial Factorization

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \in \mathcal{C}$

- $\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \not\equiv 0$
- $\tilde{Q}(\mathbf{z}, f(\mathbf{z})) \equiv 0$

$\mathbf{x}_i - f(\mathbf{z})$ divides $\tilde{Q}(\mathbf{z}, \mathbf{x}_i)$

Reducing to Polynomial Factorization

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \in \mathcal{C}$

- $\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \not\equiv 0$
- $\tilde{Q}(\mathbf{z}, f(\mathbf{z})) \equiv 0$

$\mathbf{x}_i - f(\mathbf{z})$ divides $\tilde{Q}(\mathbf{z}, \mathbf{x}_i)$

Reducing to **polynomial factorization**!

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \in \mathcal{C}$

$\Downarrow$

f has a small circuit

Polynomial Factorization

Polynomial Factorization (Simplified setting)

Goal: For any $P(\mathbf{z}, y) \in \mathcal{C}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \mathcal{C}'$.

Polynomial Factorization (Simplified setting)

Goal: For any $P(\mathbf{z}, y) \in \mathcal{C}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \mathcal{C}'$.

existential

Polynomial Factorization (Simplified setting)

Goal: For any $P(\mathbf{z}, y) \in \mathcal{C}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP

Polynomial Factorization (Simplified setting)

Goal: For any $P(\mathbf{z}, y) \in \mathcal{C}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09]	$\text{Depth-}\Delta$ with bounded individual degree	$\text{Depth-}\Delta + 3$

Polynomial Factorization (Simplified setting)

Goal: For any $P(\mathbf{z}, y) \in \mathcal{C}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09]	$\text{Depth-}\Delta$ with bounded individual degree	$\text{Depth-}\Delta + 3$
[DSS18, **C**KS18]	$\mathrm{VF}(n^{\log n}))$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n})$)	$\mathrm{VF}(n^{\log n}))$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n})$)

Polynomial Factorization (Simplified setting)

Goal: For any $P(\mathbf{z}, y) \in \mathcal{C}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09]	$\text{Depth-}\Delta$ with bounded individual degree	$\text{Depth-}\Delta + 3$
[DSS18, **C**KS18]	$\mathrm{VF}(n^{\log n})$ $(\text{resp. } \mathrm{VBP}(n^{\log n}), \mathrm{VNP}(n^{\log n}))$	$\mathrm{VF}(n^{\log n})$ $(\text{resp. } \mathrm{VBP}(n^{\log n}), \mathrm{VNP}(n^{\log n}))$
[**C**KS18]	$\text{Depth-}\Delta$ with degree $O(\log^2 n / \log^2 \log n)$	$\text{Depth-}\Delta + 3$

Polynomial Factorization (Simplified setting)

Goal: For any $P(\mathbf{z}, y) \in \mathcal{C}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09]	$\text{Depth-}\Delta$ with bounded individual degree	$\text{Depth-}\Delta + 3$
[DSS18, **C**KS18]	$\text{VF}(n^{\log n})$ (resp. $\text{VBP}(n^{\log n})$, $\text{VNP}(n^{\log n})$)	$\text{VF}(n^{\log n})$ (resp. $\text{VBP}(n^{\log n})$, $\text{VNP}(n^{\log n})$)
[**C**KS18]	$\text{Depth-}\Delta$ with degree $O(\log^2 n / \log^2 \log n)$	$\text{Depth-}\Delta + 3$
[**C**KS18]	VNP	VNP

Polynomial Factorization (Simplified setting)

Goal: For any $P(\mathbf{z}, y) \in \mathcal{C}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \mathcal{C}'$.

existential

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09]	$\text{Depth-}\Delta$ with bounded individual degree	$\text{Depth-}\Delta + 3$
[DSS18, **C**KS18]	$\mathrm{VF}(n^{\log n})$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n})$)	$\mathrm{VF}(n^{\log n})$ (resp. $\mathrm{VBP}(n^{\log n})$, $\mathrm{VNP}(n^{\log n})$)
[**C**KS18]	$\text{Depth-}\Delta$ with degree $O(\log^2 n / \log^2 \log n)$	$\text{Depth-}\Delta + 3$
[**C**KS18]	VNP	VNP

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Example: For any $P(\mathbf{z}, y) \in \mathrm{VP}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$. Show that $f \in \mathrm{VP}$.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Example: For any $P(\mathbf{z}, y) \in \mathrm{VP}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$. Show that $f \in \mathrm{VP}$.

- P as an univariate polynomial: $P(\mathbf{z}, y) = \sum_{i=0}^{k} C_i(\mathbf{z}) y^i$.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Example: For any $P(\mathbf{z}, y) \in \mathrm{VP}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$. Show that $f \in \mathrm{VP}$.

- P as an univariate polynomial: $P(\mathbf{z}, y) = \sum_{i=0}^{k} C_i(\mathbf{z}) y^i$.
- Derivative does not vanish: $\dfrac{\partial P(\mathbf{0}, f(\mathbf{0}))}{\partial y} = \delta \neq 0$.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Example: For any $P(\mathbf{z}, y) \in \mathrm{VP}$ such that $P(\mathbf{z}, f(\mathbf{z})) = 0$. Show that $f \in \mathrm{VP}$.

- P as an univariate polynomial: $P(\mathbf{z}, y) = \sum_{i=0}^{k} C_i(\mathbf{z}) y^i$.
- Derivative does not vanish: $\dfrac{\partial P(\mathbf{0}, f(\mathbf{0}))}{\partial y} = \delta \neq 0$.

Induction hypothesis:
Maintain $h_i \in \mathrm{VP}$ for $i = 0, 1, \ldots, d = \deg(f)$ such that

$$\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f].$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Goal: $\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f]$ and $h_i \in \mathrm{VP}$.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Goal: $\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f]$ and $h_i \in \mathrm{VP}$.

Base case: $h_0 \in \mathbb{F},\ h_0 \in \mathrm{VP}$.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Goal: $\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f]$ and $h_i \in \mathrm{VP}$.

Base case: $h_0 \in \mathbb{F},\ h_0 \in \mathrm{VP}$.

Induction step: $h_i = h_{i-1} - \dfrac{P(\mathbf{z}, h_{i-1}(\mathbf{z}))}{\delta}$.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Goal: $\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f]$ and $h_i \in \mathrm{VP}$.

Base case: $h_0 \in \mathbb{F}, \;\; h_0 \in \mathrm{VP}$.

Induction step: $h_i = h_{i-1} - \dfrac{P(\mathbf{z}, h_{i-1}(\mathbf{z}))}{\delta}$.

- ($h_i \in \mathrm{VP}$): There's only additive poly(n,s,d) blow-up.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Goal: $\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f]$ and $h_i \in \mathrm{VP}$.

Base case: $h_0 \in \mathbb{F}, \; h_0 \in \mathrm{VP}$.

Induction step: $h_i = h_{i-1} - \dfrac{P(\mathbf{z}, h_{i-1}(\mathbf{z}))}{\delta}$.

- ($h_i \in \mathrm{VP}$): There's only additive poly(n,s,d) blow-up.
- ($\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f]$): Taylor's expansion.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

Goal: $\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f]$ and $h_i \in \mathrm{VP}$.

Base case: $h_0 \in \mathbb{F}, \; h_0 \in \mathrm{VP}$.

Induction step: $h_i = h_{i-1} - \dfrac{P(\mathbf{z}, h_{i-1}(\mathbf{z}))}{\delta}$.

- ($h_i \in \mathrm{VP}$): There's only additive poly(n,s,d) blow-up.
- ($\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f]$): Taylor's expansion.

Want $h_i - h_{i-1} = \mathcal{H}_i[f - h_{i-1}]$.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following,
we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\dfrac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}$.

Pf: By Taylor's expansion.

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\frac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}$.

Pf: By Taylor's expansion.

$$0 = P(h_{i-1} + f - h_{i-1})$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\dfrac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}$.

Pf: By Taylor's expansion.

$$
\begin{aligned}
0 &= P(h_{i-1} + f - h_{i-1}) \\
&= P(h_{i-1}) + \frac{\partial}{\partial y} P(h_{i-1}) \cdot (f - h_{i-1}) \\
&+ \frac{\partial^2}{\partial y^2} P(h_{i-1}) \cdot (f - h_{i-1})^2 + \cdots
\end{aligned}
$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\dfrac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}$.

Pf: By Taylor's expansion.

$$0 = P(h_{i-1}) + \frac{\partial}{\partial y} P(h_{i-1}) \cdot (f - h_{i-1})$$
$$+ \frac{\partial^2}{\partial y^2} P(h_{i-1}) \cdot (f - h_{i-1})^2 + \cdots$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\frac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}.$

By induction $\deg(f - h_{i-1}) > i - 1.$

Pf: By Taylor's expansion.

$$0 = P(h_{i-1}) + \frac{\partial}{\partial y} P(h_{i-1}) \cdot (f - h_{i-1})$$
$$+ \frac{\partial^2}{\partial y^2} P(h_{i-1}) \cdot (f - h_{i-1})^2 + \cdots$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\dfrac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}$.

Pf: By Taylor's expansion.

$$0 = \mathcal{H}_{\leq i}\left[P(h_{i-1}) + \frac{\partial}{\partial y}P(h_{i-1}) \cdot (f - h_{i-1})\right]$$

$$+ \mathcal{H}_{\leq i}\left[\frac{\partial^2}{\partial y^2}P(h_{i-1}) \cdot (f - h_{i-1})^2 + \cdots\right]$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\dfrac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}$.

Pf: By Taylor's expansion.

$$
\begin{aligned}
0 = \mathcal{H}_{\leq i}\left[P(h_{i-1}) + \frac{\partial}{\partial y}P(h_{i-1}) \cdot (f - h_{i-1})\right] \\
+ \mathcal{H}_{\leq i}\left[\frac{\partial^2}{\partial y^2}P(h_{i-1}) \cdot (f - h_{i-1})^2 + \cdots\right]
\end{aligned}
$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\dfrac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}$.

By induction $\deg(f - h_{i-1}) > i - 1$.

Pf: By Taylor's expansion.

$$0 = \mathcal{H}_{\leq i}\left[P(h_{i-1})\right] + \mathcal{H}_{\leq i}\left[\frac{\partial}{\partial y} P(h_{i-1}) \cdot (f - h_{i-1})\right]$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\dfrac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}.$

Pf: By Taylor's expansion.

By induction $\deg(f - h_{i-1}) > i - 1.$

$$0 = \mathcal{H}_{\leq i}\left[P(h_{i-1})\right] + \mathcal{H}_{\leq i}\left[\frac{\partial}{\partial y} P(h_{i-1}) \cdot (f - h_{i-1})\right]$$

$$= \mathcal{H}_{\leq i}\left[P(h_{i-1})\right] + \mathcal{H}_0\left[\frac{\partial}{\partial y} P(h_{i-1})\right] \cdot \mathcal{H}_i\left[(f - h_{i-1})\right]$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\dfrac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}.$

By induction $\deg(f - h_{i-1}) > i - 1.$

Pf: By Taylor's expansion.

$$0 = \mathcal{H}_{\leq i}\left[P(h_{i-1})\right] + \mathcal{H}_{\leq i}\left[\frac{\partial}{\partial y}P(h_{i-1}) \cdot (f - h_{i-1})\right]$$

$$= \mathcal{H}_{\leq i}\left[P(h_{i-1})\right] + \mathcal{H}_0\left[\frac{\partial}{\partial y}P(h_{i-1})\right] \cdot \mathcal{H}_i\left[(f - h_{i-1})\right]$$

$$\frac{\partial P(\mathbf{0}, f(\mathbf{0}))}{\partial y} = \delta \neq 0.$$

Key Idea: Newton Iteration (Sloppy Hensel Lifting)

In the following, we ignore $\mathbf{z}$ for simplicity.

Lemma: $\mathcal{H}_i[f - h_{i-1}] = -\frac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}$.

By induction $\deg(f - h_{i-1}) > i - 1$.

Pf: By Taylor's expansion.

$$0 = \mathcal{H}_{\leq i}\left[P(h_{i-1})\right] + \mathcal{H}_{\leq i}\left[\frac{\partial}{\partial y} P(h_{i-1}) \cdot (f - h_{i-1})\right]$$

$$= \mathcal{H}_{\leq i}\left[P(h_{i-1})\right] + \mathcal{H}_0\left[\frac{\partial}{\partial y} P(h_{i-1})\right] \cdot \mathcal{H}_i\left[(f - h_{i-1})\right]$$

$$= \mathcal{H}_{\leq i}\left[P(h_{i-1})\right] + \delta \cdot \mathcal{H}_i\left[(f - h_{i-1})\right]$$

Extend to Other Circuit Classes?

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09]	$\text{Depth-}\Delta$ with bounded individual degree	$\text{Depth-}\Delta + 3$
[DSS18, **C**KS18]	$\mathrm{VF}(n^{\log n})$ $(\text{resp. } \mathrm{VBP}(n^{\log n}), \mathrm{VNP}(n^{\log n}))$	$\mathrm{VF}(n^{\log n})$ $(\text{resp. } \mathrm{VBP}(n^{\log n}), \mathrm{VNP}(n^{\log n}))$
[**C**KS18]	$\text{Depth-}\Delta$ with degree $O(\log^2 n / \log^2 \log n)$	$\text{Depth-}\Delta + 3$
[**C**KS18]	VNP	VNP

Extend to Other Circuit Classes?

Issue: How to efficiently implement the **update step**?

Recall that we have

$$\mathcal{H}_i[f - h_{i-1}] = -\frac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}.$$

Extend to Other Circuit Classes?

Issue: How to efficiently implement the **update step**?

Recall that we have

$$\mathcal{H}_i[f - h_{i-1}] = -\frac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}.$$

Example: Does $\mathcal{H}_i[P(\mathbf{z}, h_{i-1})]$ have a small *constant-depth* circuit?

Extend to Other Circuit Classes?

Issue: How to efficiently implement the **update step**?

Recall that we have

$$\mathcal{H}_i[f - h_{i-1}] = -\frac{\mathcal{H}_i[P(\mathbf{z}, h_{i-1}(\mathbf{z}))]}{\delta}.$$

Example: Does $\mathcal{H}_i[P(\mathbf{z}, h_{i-1})]$ have a small *constant-depth* circuit?

Key: Avoid **recursion**!

Generator Lemma [DSY09, **C**KS18]

Goal: Efficiently implement $\mathcal{H}_i[P(\mathbf{z}, h_{i-1})]$.

Generator Lemma [DSY09, **C**KS18]

Goal: Efficiently implement $\mathcal{H}_i[P(\mathbf{z}, h_{i-1})]$.

Lemma [DSY09]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $\mathsf{poly}(s, d)$ such that

$$\mathcal{H}_i[f] = \mathcal{H}_{\leq i}[A_i(C_0, C_1, \ldots, C_k)],$$

where $P(\mathbf{z}, y) = \sum_{i=0}^{k} C_i(\mathbf{z}) y^i$.

Individual degree

Generator Lemma [DSY09, **C**KS18]

Goal: Efficiently implement $\mathcal{H}_i[P(\mathbf{z}, h_{i-1})]$.

Lemma [DSY09]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $\mathsf{poly}(s, d)$ such that

$$\mathcal{H}_i[f] = \mathcal{H}_{\leq i}[A_i(C_0, C_1, \ldots, C_k)],$$

where $P(\mathbf{z}, y) = \sum_{i=0}^{k} C_i(\mathbf{z}) y^i$.

Individual degree

$\{C_i\}$ can be **reused** and **efficiently extracted** from P!

Generator Lemma [DSY09, **C**KS18]

Lemma [DSY09]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $\mathsf{poly}(s, d)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(C_0, C_1, \ldots, C_k)].$$

Generator Lemma [DSY09, **C**KS18]

Lemma [DSY09]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $\mathsf{poly}(s, d)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(C_0, C_1, \ldots, C_k)].$$

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $O(d^2 i)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(g_0, g_1, \ldots, g_d)]$$

where

Generator Lemma [DSY09, **C**KS18]

Lemma [DSY09]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $\mathrm{poly}(s, d)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(C_0, C_1, \ldots, C_k)].$$

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $O(d^2 i)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(g_0, g_1, \ldots, g_d)]$$

where

$$g_i = \mathcal{H}_{\leq d}\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right] - \mathcal{H}_0\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right].$$

Efficiently extracted from P.

Generator Lemma [DSY09, **C**KS18]

Lemma [DSY09]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $\mathsf{poly}(s, d)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(C_0, C_1, \ldots, C_k)].$$

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $O(d^2 i)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(g_0, g_1, \ldots, g_d)]$$

where

$$g_i = \mathcal{H}_{\leq d}\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right] - \mathcal{H}_0\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right].$$

Efficiently extracted from P.

Generator Lemma [DSY09, **C**KS18]

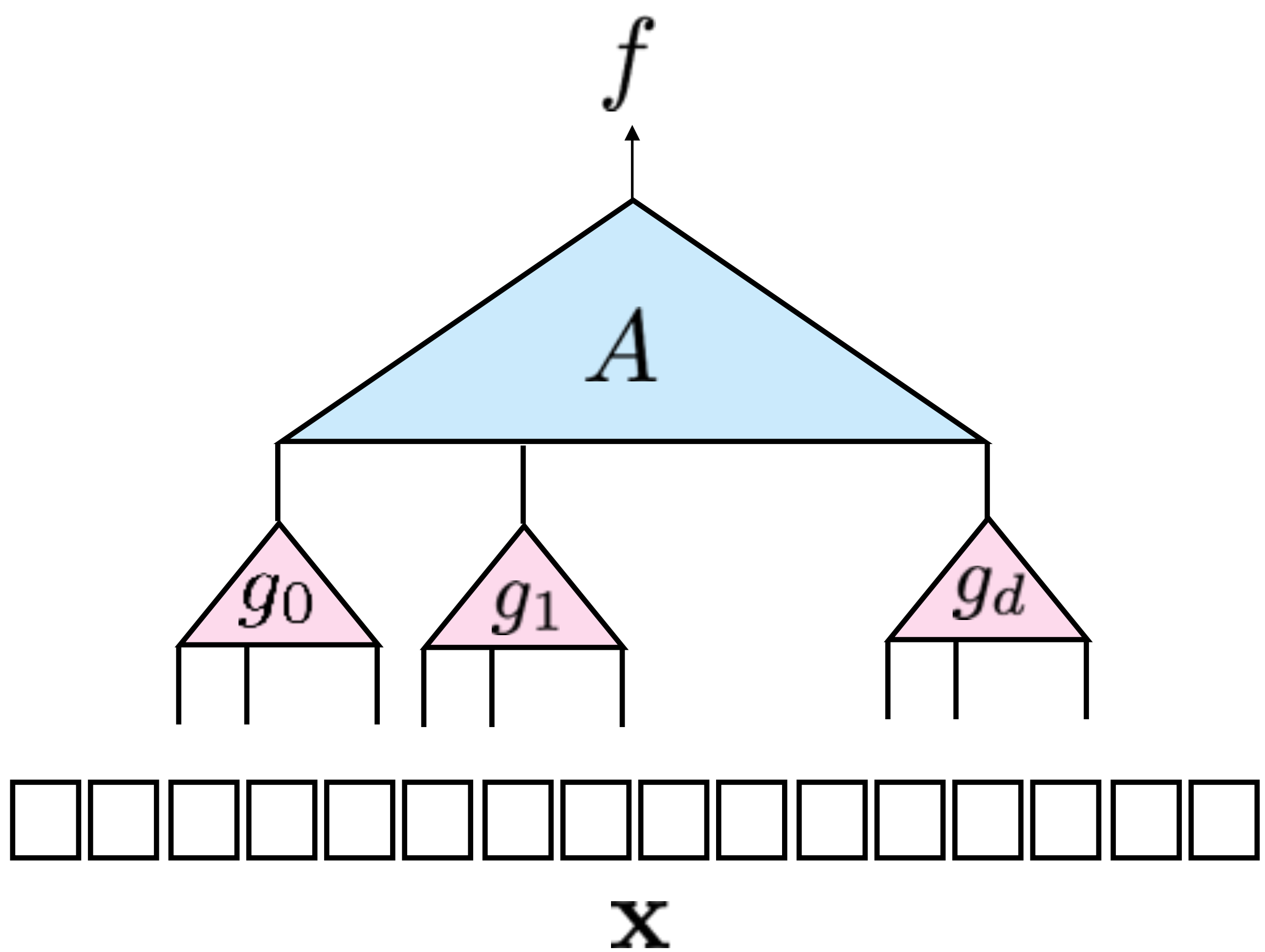
Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A of size $O(d^2 i)$ such that

$$f = A(g_0, g_1, \ldots, g_d) \, .$$

Generator Lemma [DSY09, **C**KS18]

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A of size $O(d^2 i)$ such that

$$f = A(g_0, g_1, \ldots, g_d).$$

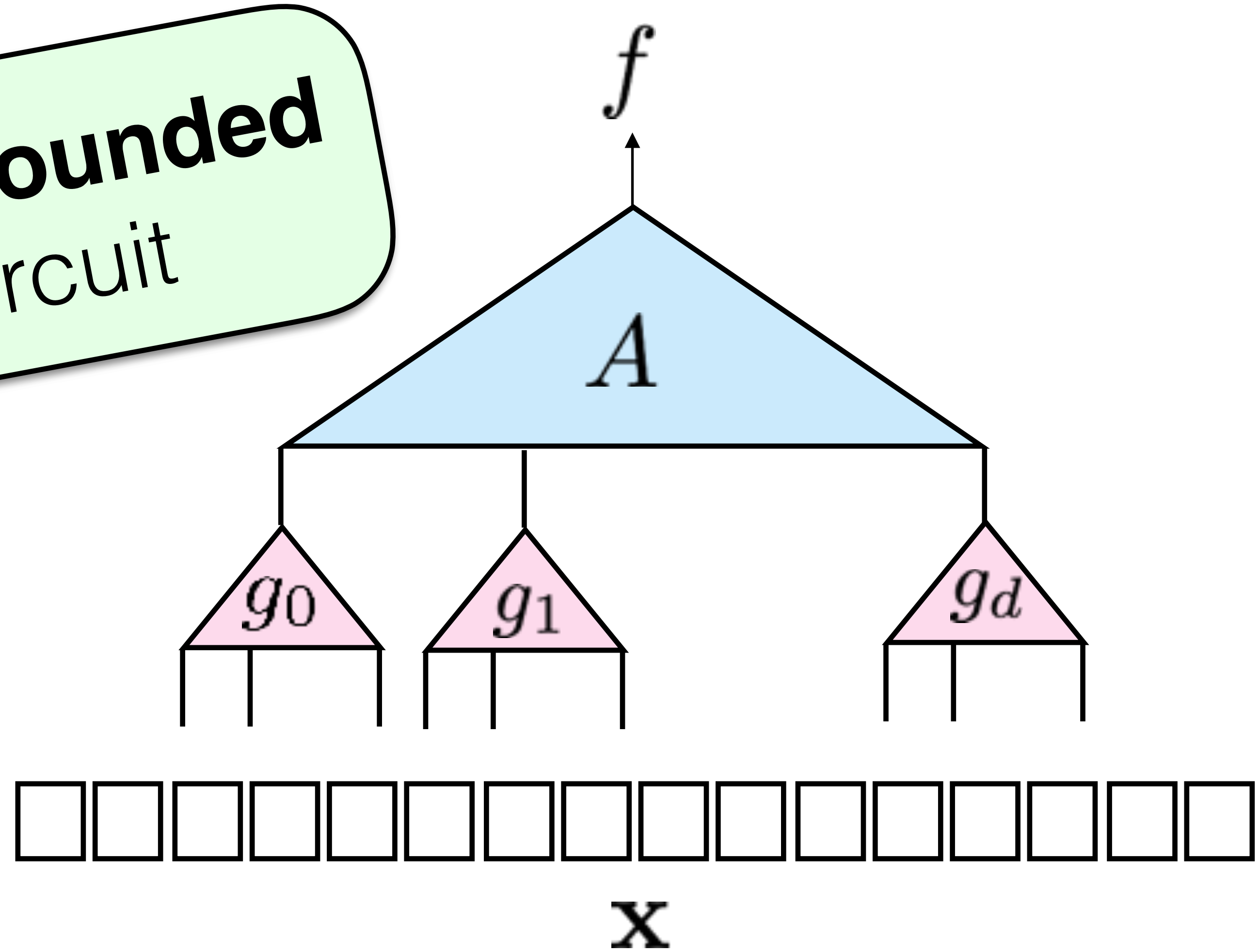


Generator Lemma [DSY09, **C**KS18]

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A of size $O(d^2 i)$ such that

$$f = A(g_0, g_1, \ldots, g_d).$$

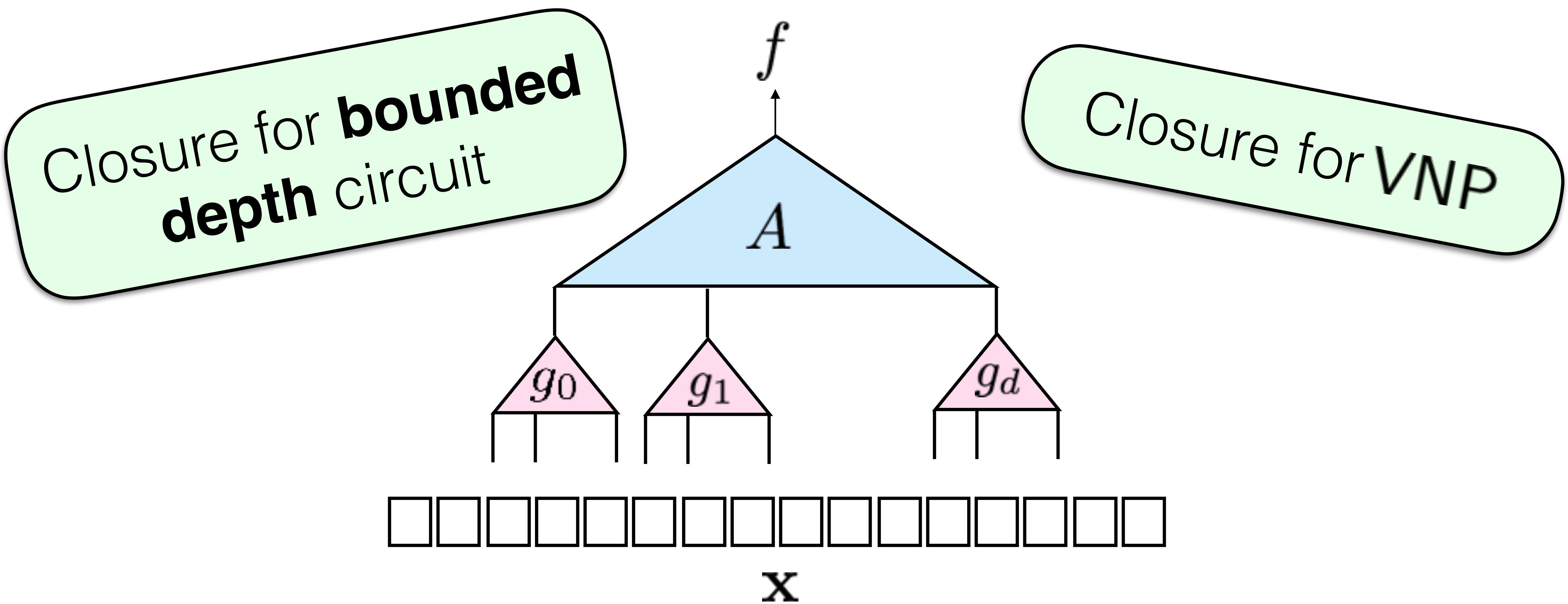
Closure for **bounded depth** circuit



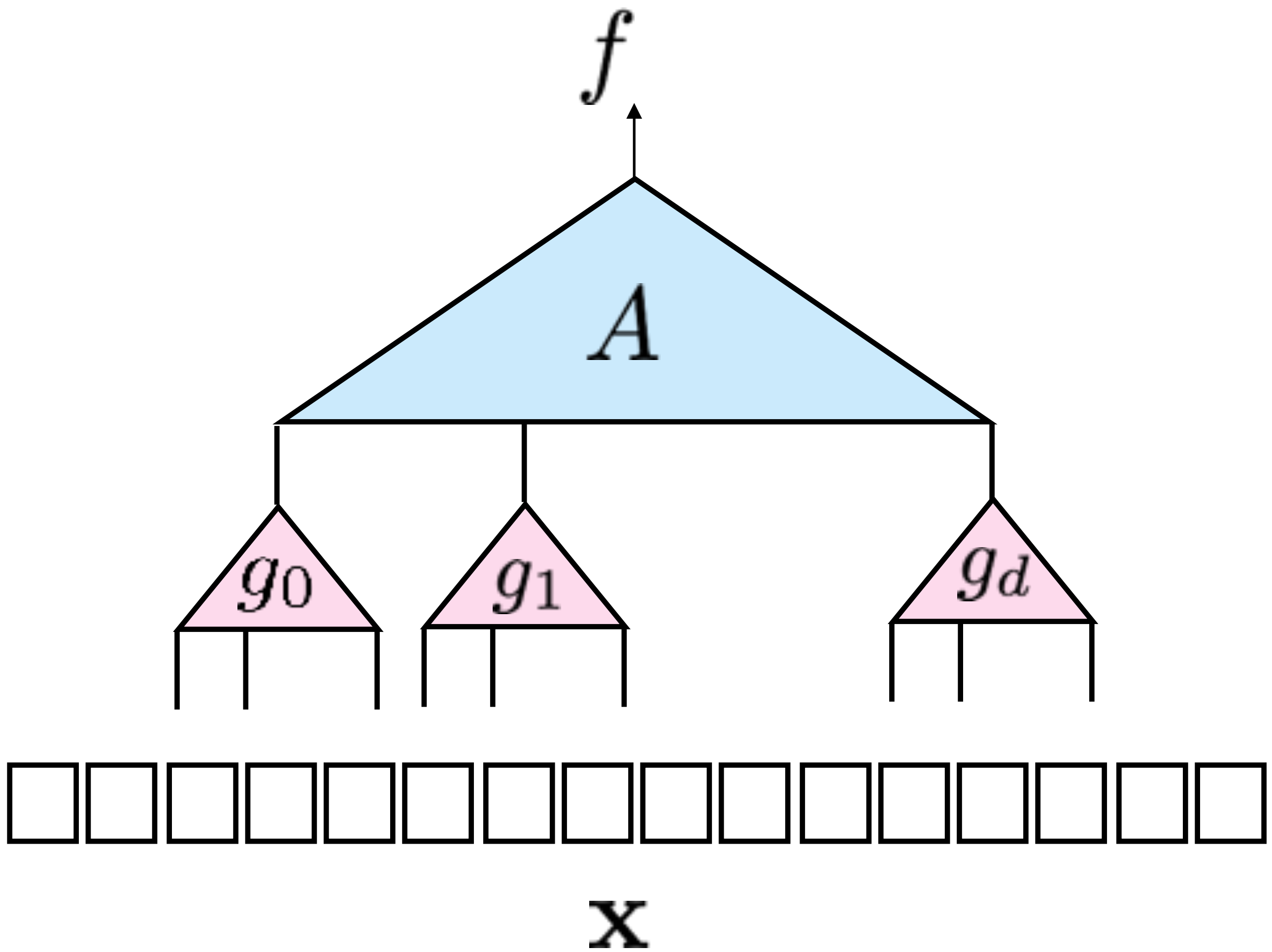
Generator Lemma [DSY09, **C**KS18]

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A of size $O(d^2 i)$ such that

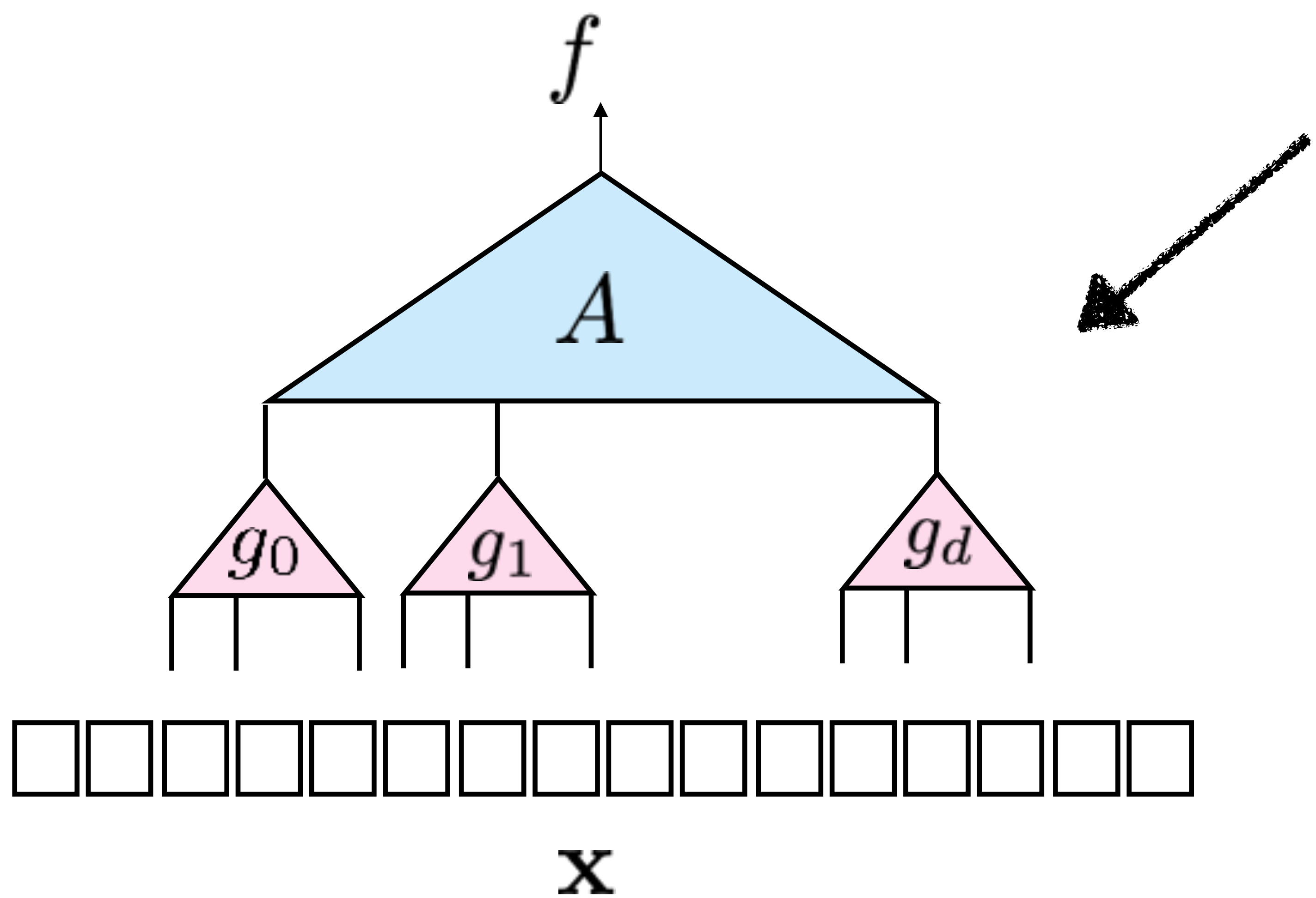
$$f = A(g_0, g_1, \ldots, g_d).$$



Closure for Bounded Depth Circuits [**C**KS18]

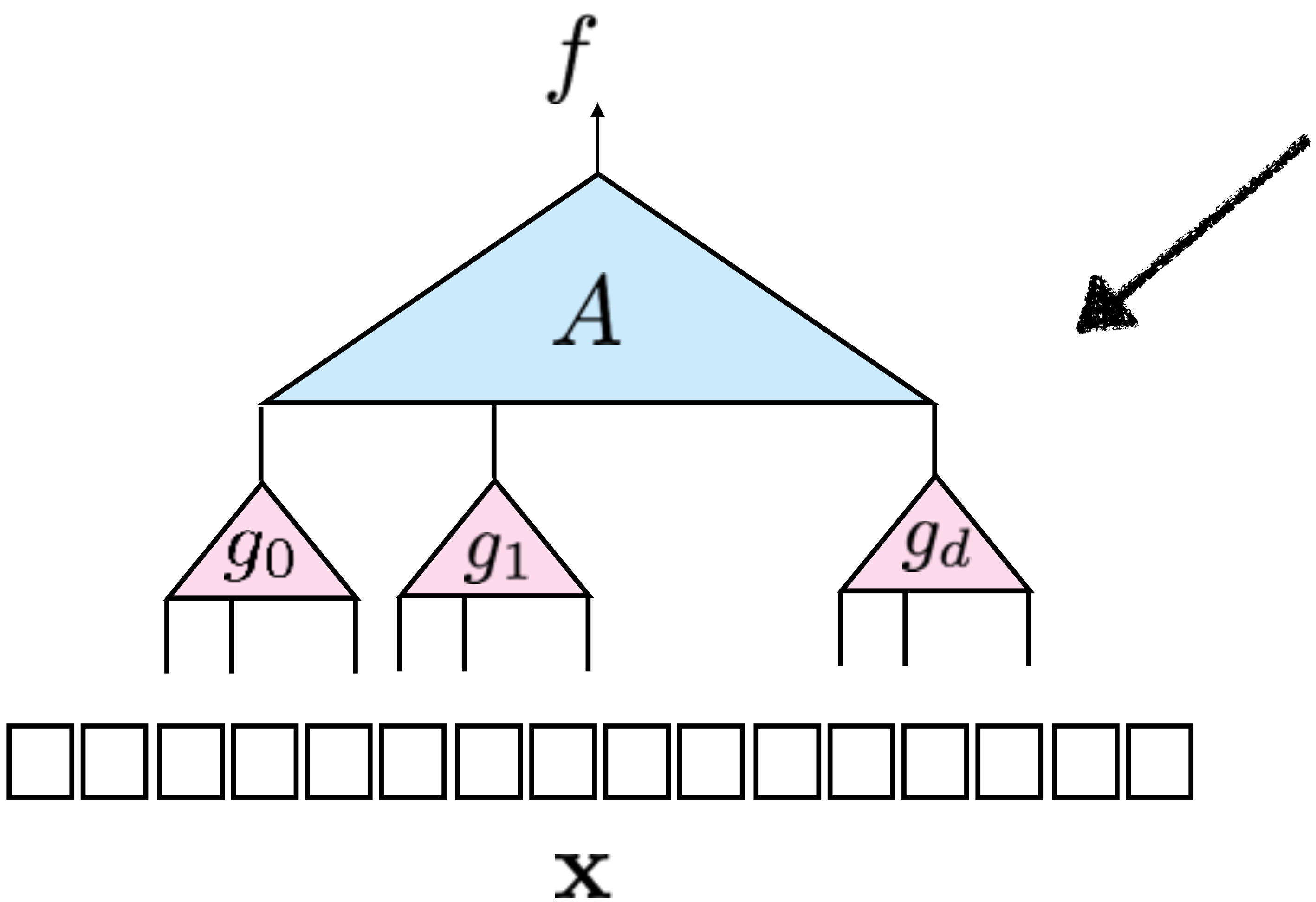


Closure for Bounded Depth Circuits [**C**KS18]



Depth reduction [GKKS13]

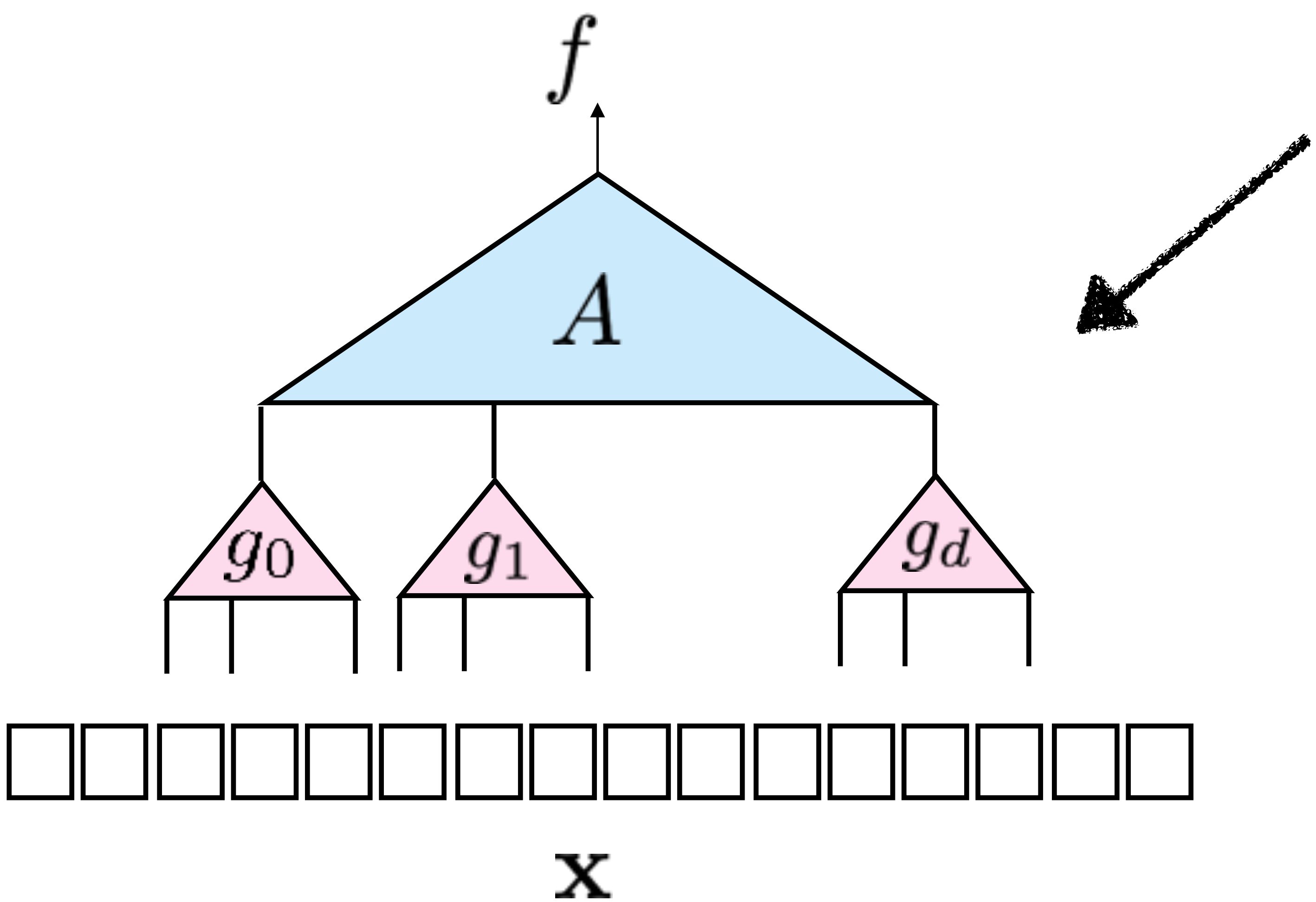
Closure for Bounded Depth Circuits [**C**KS18]



Depth reduction [GKKS13]

- size s
- degree d

Closure for Bounded Depth Circuits [**C**KS18]



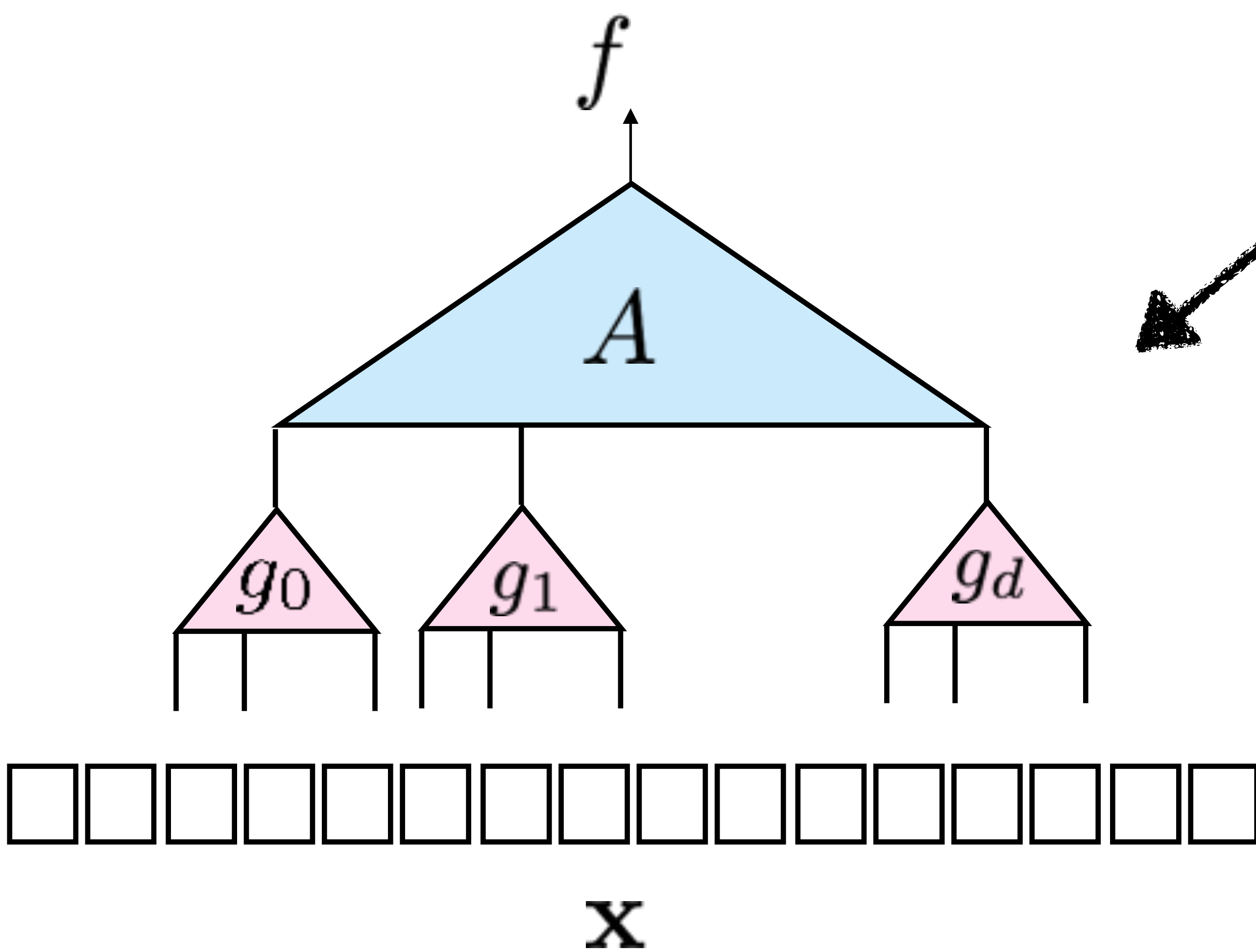
Depth reduction [GKKS13]

- size s
- degree d

⬇

- size $(snd)^{O(\sqrt{d})}$
- depth 3, *i.e.*, $\Sigma\Pi\Sigma$

Closure for Bounded Depth Circuits [**C**KS18]



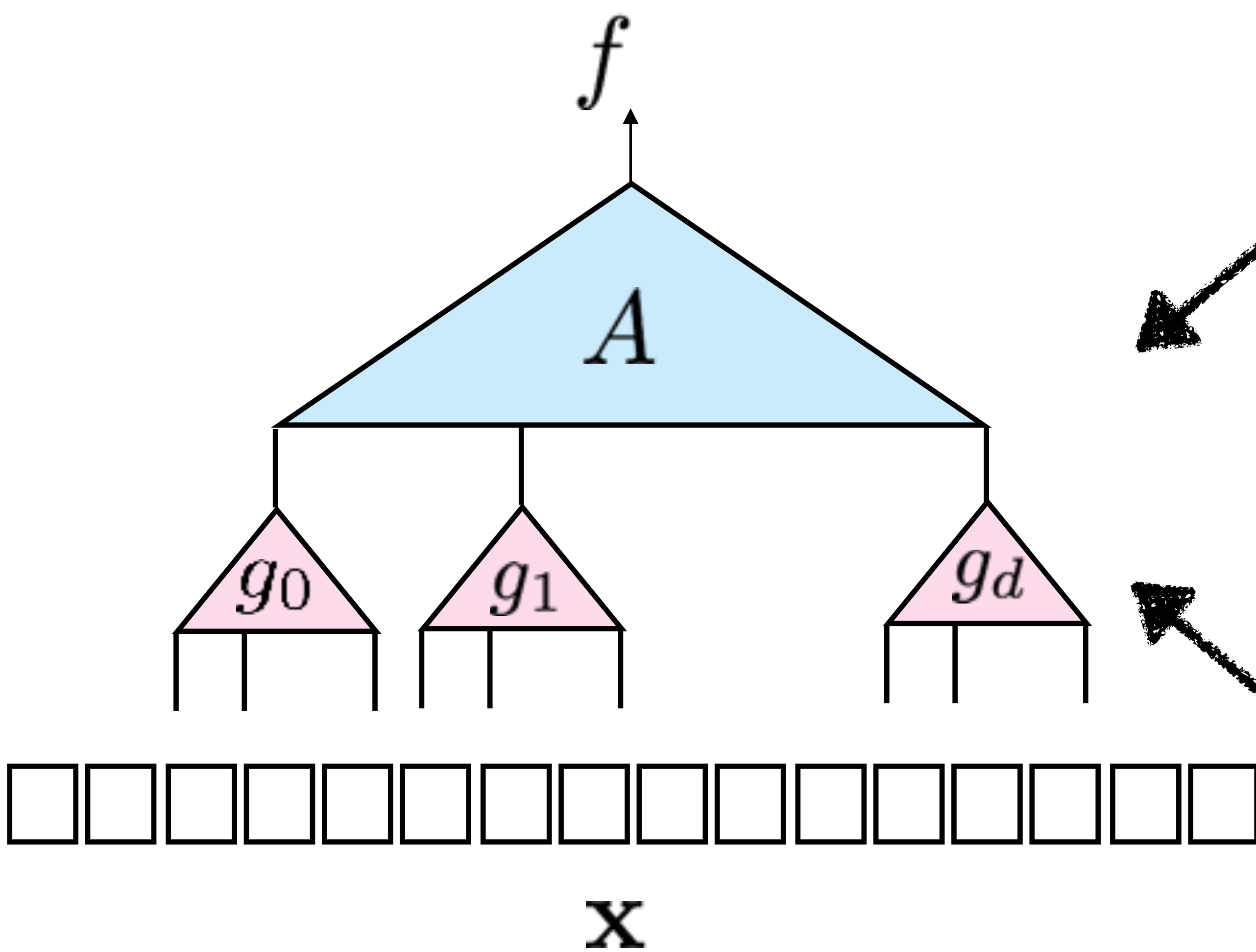
Depth reduction [AV08, Koi12, Tav15]

- size s
- degree d

↓

- size $(snd)^{O(d^{1/k})}$
- depth 2k

Closure for Bounded Depth Circuits [**C**KS18]



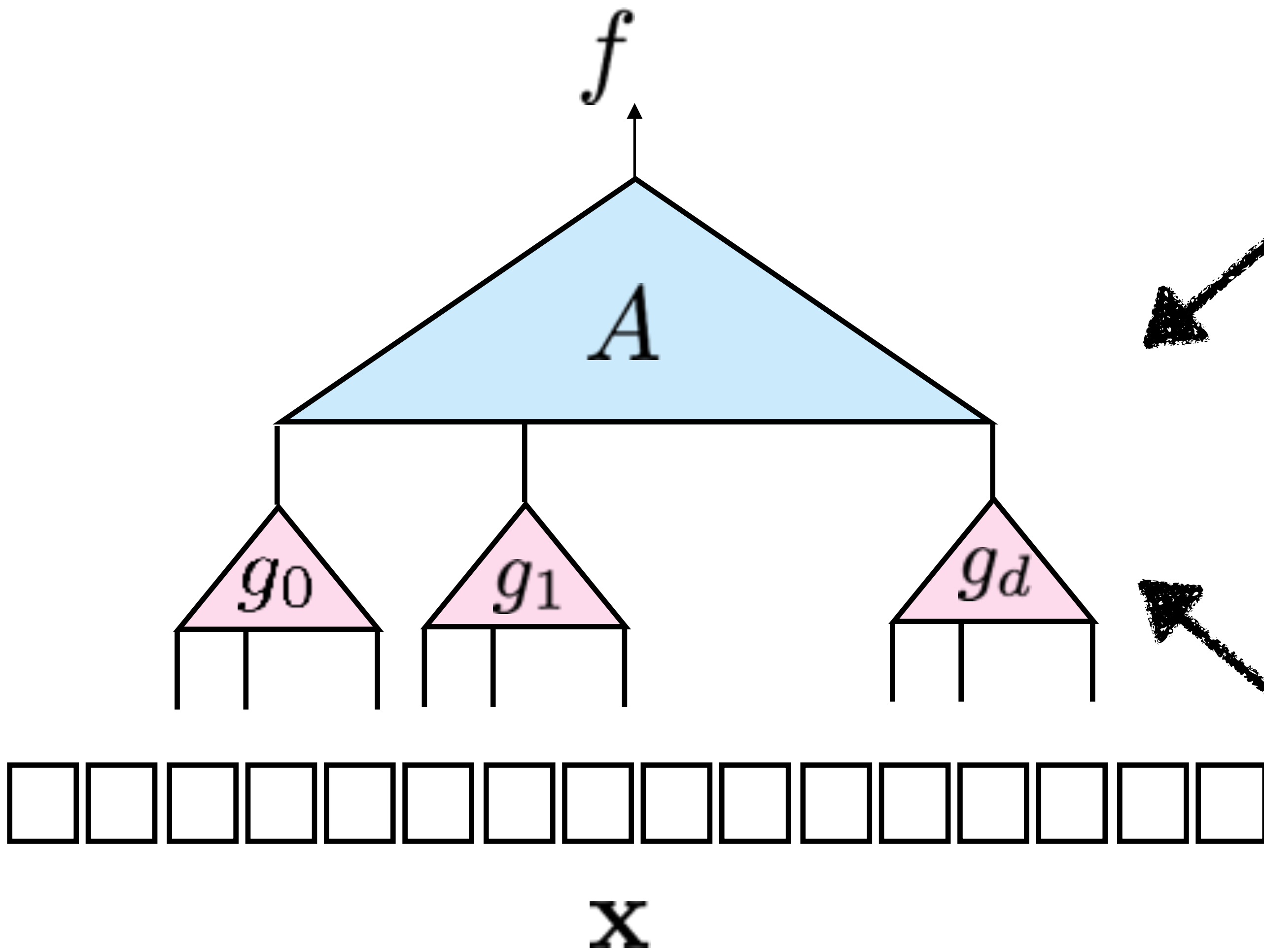
Depth reduction [AV08, Koi12, Tav15]

- size s
- degree d

- size $(snd)^{O(d^{1/k})}$
- depth 2k

Interpolation

Closure for Bounded Depth Circuits [**C**KS18]



Depth reduction [AV08, Koi12, Tav15]

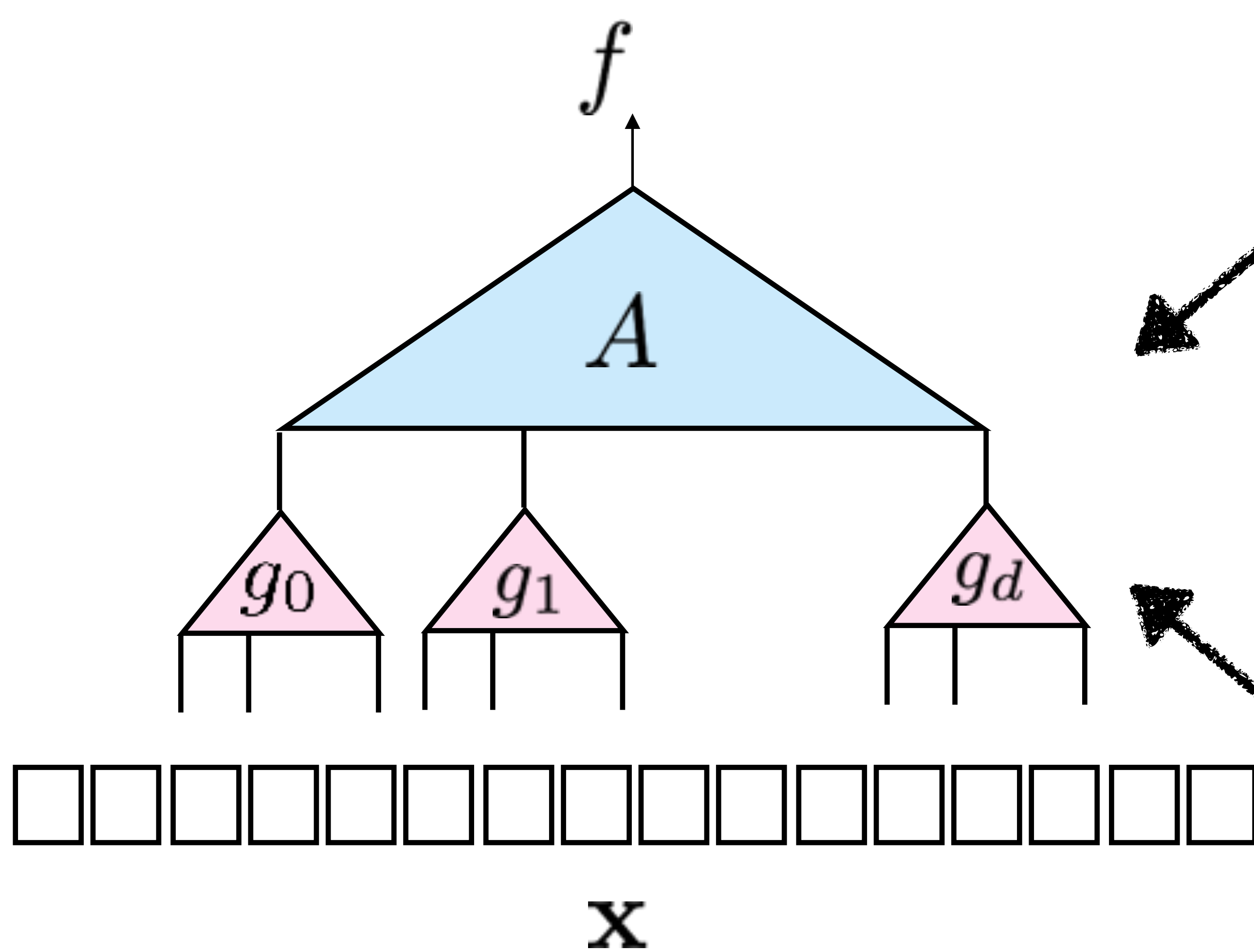
- size s
- degree d

- size $(snd)^{O(d^{1/k})}$
- depth 2k

Interpolation

- size has polynomial blowup
- depth has O(1) blowup

Closure for Bounded Depth Circuits [**C**KS18]



Depth reduction [AV08, Koi12, Tav15]

- size s
- degree d

- size $(snd)^{O(d^{1/k})}$
- depth 2k

Interpolation

- size has polynomial blowup
- depth has O(1) blowup

The factor f has a small bounded depth circuit!

Closure for VNP [**C**KS18]

Closure for VNP [**C**KS18]

Definition (VNP): We say $f \in \mathsf{VNP}$ if $\exists Q \in \mathsf{VP}$ s.t.

.

Closure for VNP [**C**KS18]

Definition (VNP): We say $f \in$ VNP if $\exists Q \in$ VP s.t.

$$f(\mathbf{x}) = \sum_{\mathbf{c} \in \{0,1\}^{|\mathbf{y}|}} Q(\mathbf{x}, \mathbf{c}).$$

Exponential sum

Closure for VNP [**C**KS18]

Definition (VNP): We say $f \in$ VNP if $\exists Q \in$ VP s.t.

$$f(\mathbf{x}) = \sum_{\mathbf{c} \in \{0,1\}^{|\mathbf{y}|}} Q(\mathbf{x}, \mathbf{c}).$$

Exponential sum

Theorem (Valiant): For every f having circuit of size s and degree d, there's **formula** Q of size poly(s, d) s.t.

$$f(\mathbf{x}) = \sum_{\mathbf{c} \in \{0,1\}^{|\mathbf{y}|}} Q(\mathbf{x}, \mathbf{c}).$$

Closure for VNP [**C**KS18]

Definition (VNP): We say $f \in$ VNP if $\exists Q \in$ VP s.t.

$$f(\mathbf{x}) = \sum_{\mathbf{c} \in \{0,1\}^{|\mathbf{y}|}} Q(\mathbf{x}, \mathbf{c}).$$

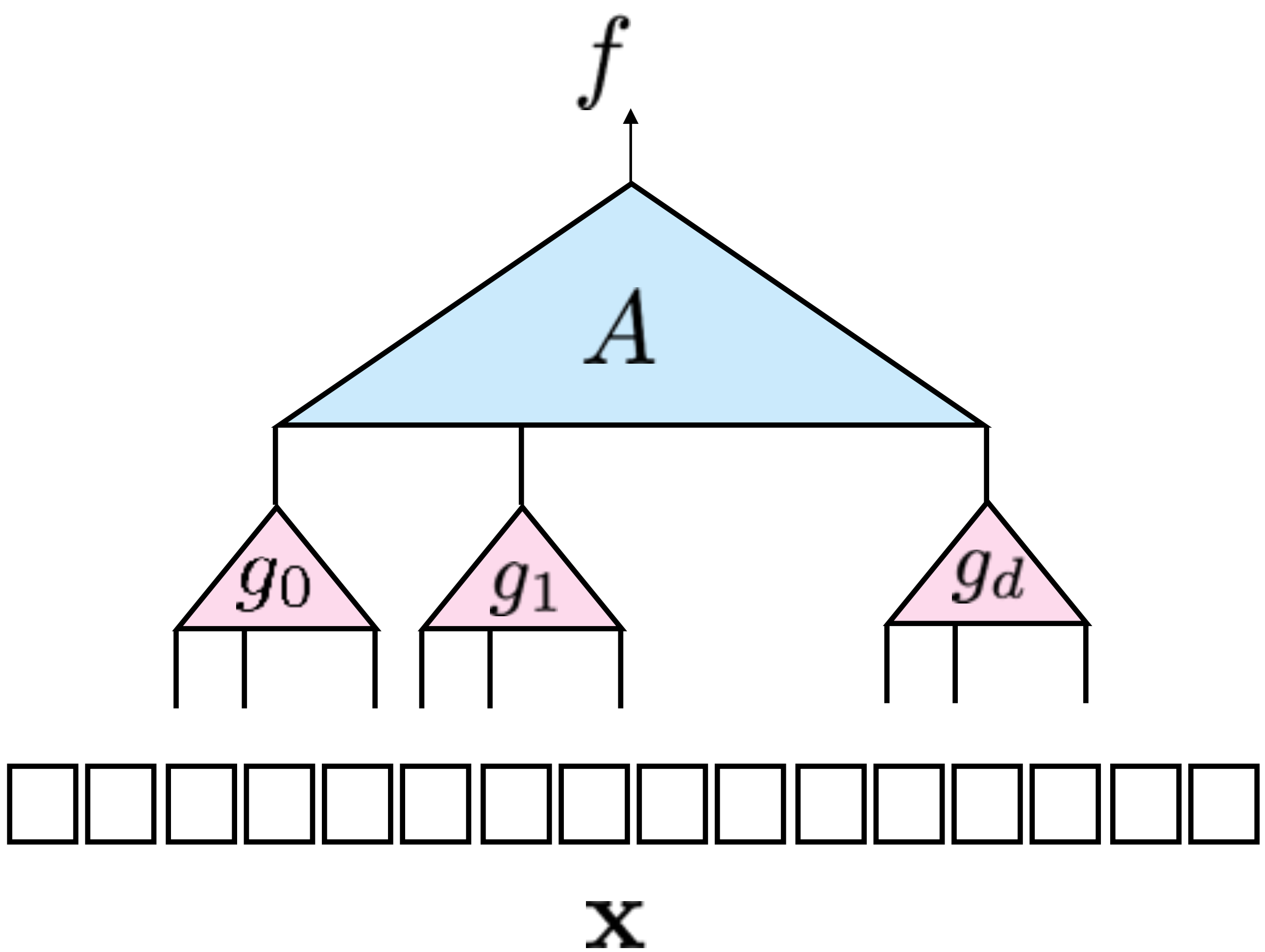
Exponential sum

Theorem (Valiant): For every f having circuit of size s and degree d, there's **formula** Q of size poly(s, d) s.t.

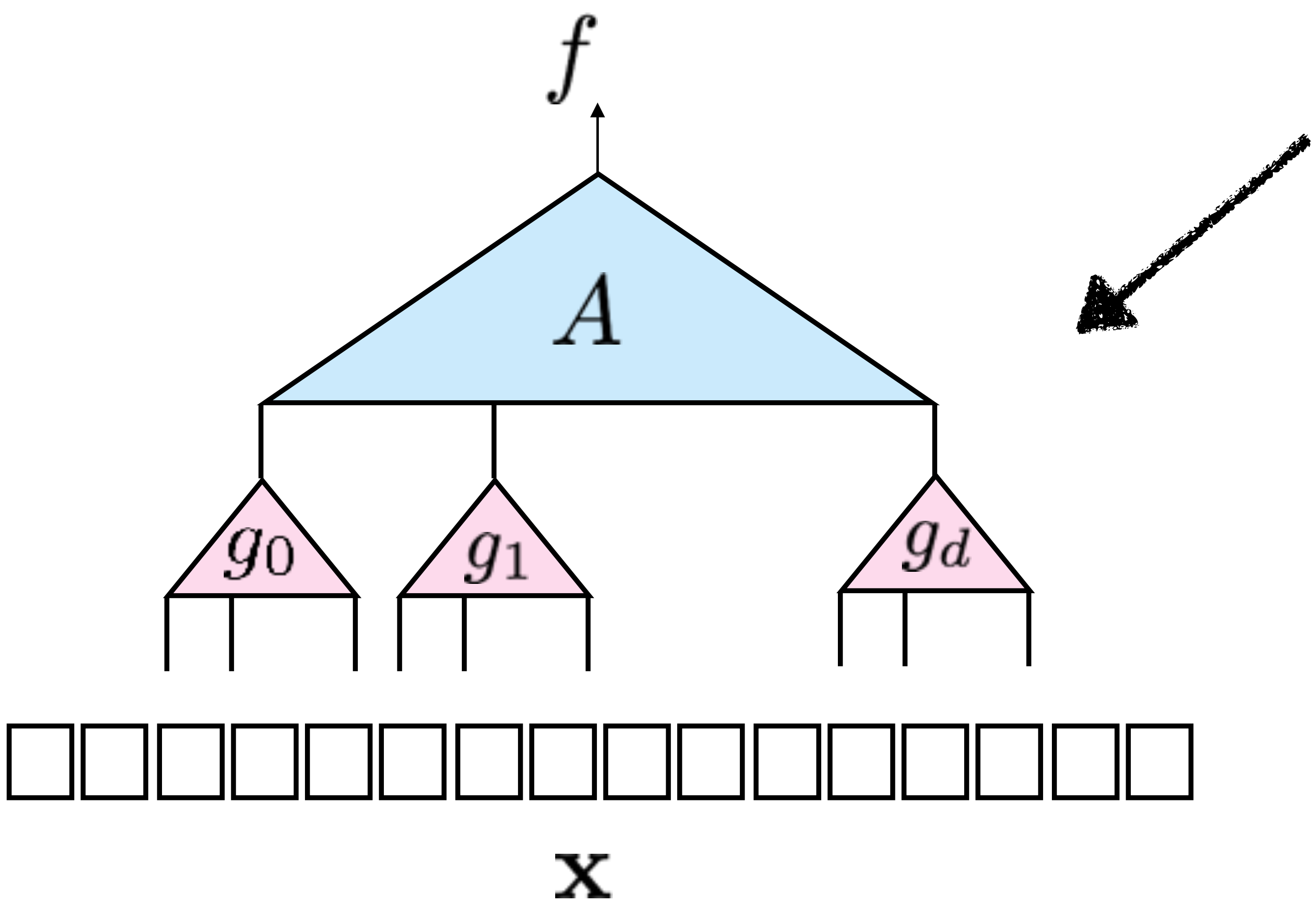
$$f(\mathbf{x}) = \sum_{\mathbf{c} \in \{0,1\}^{|\mathbf{y}|}} Q(\mathbf{x}, \mathbf{c}).$$

Formula is useful for **composing** exponential sum!

Closure for VNP [**C**KS18]

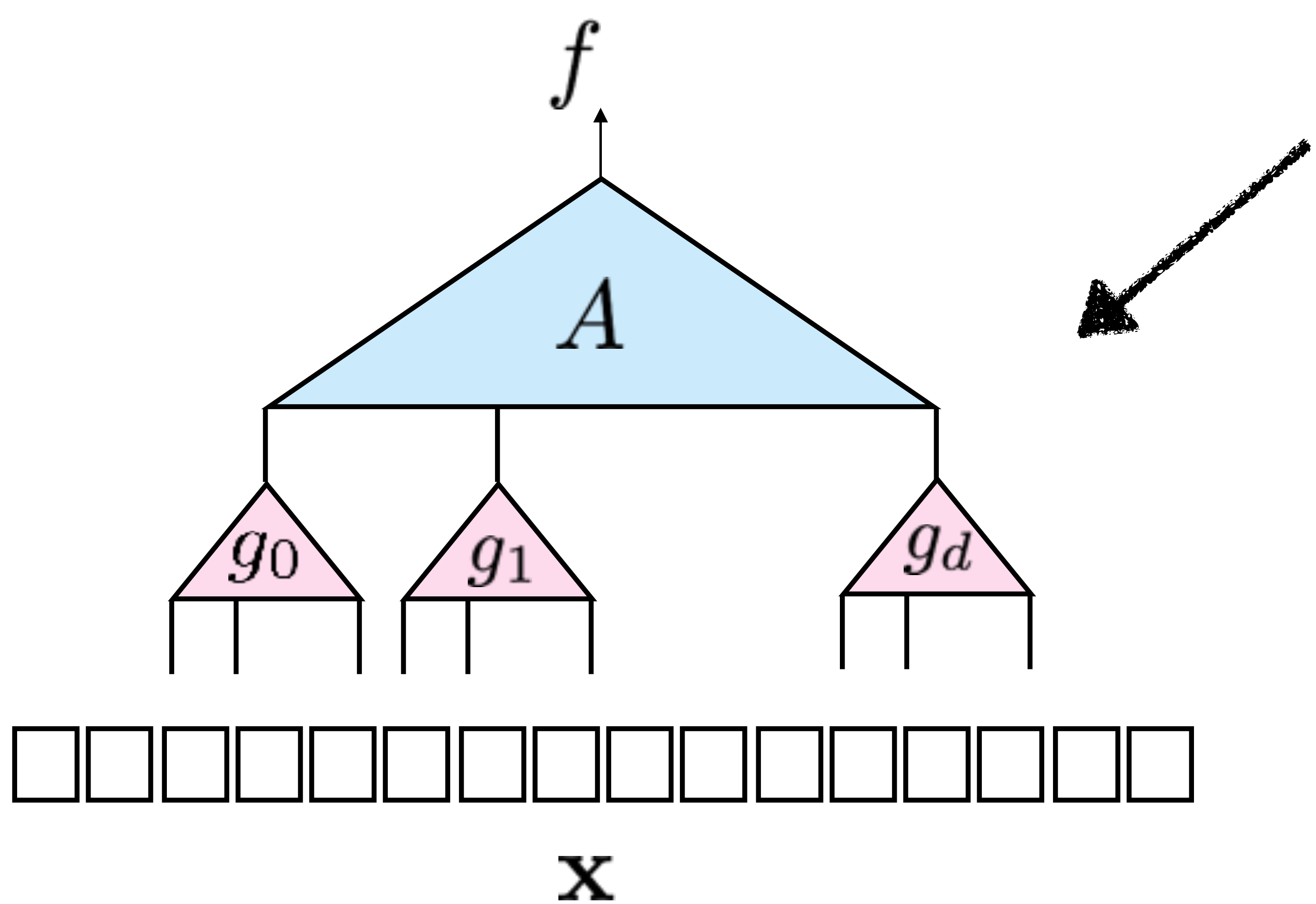


Closure for VNP [**C**KS18]



Valiant Theorem

Closure for VNP [**C**KS18]

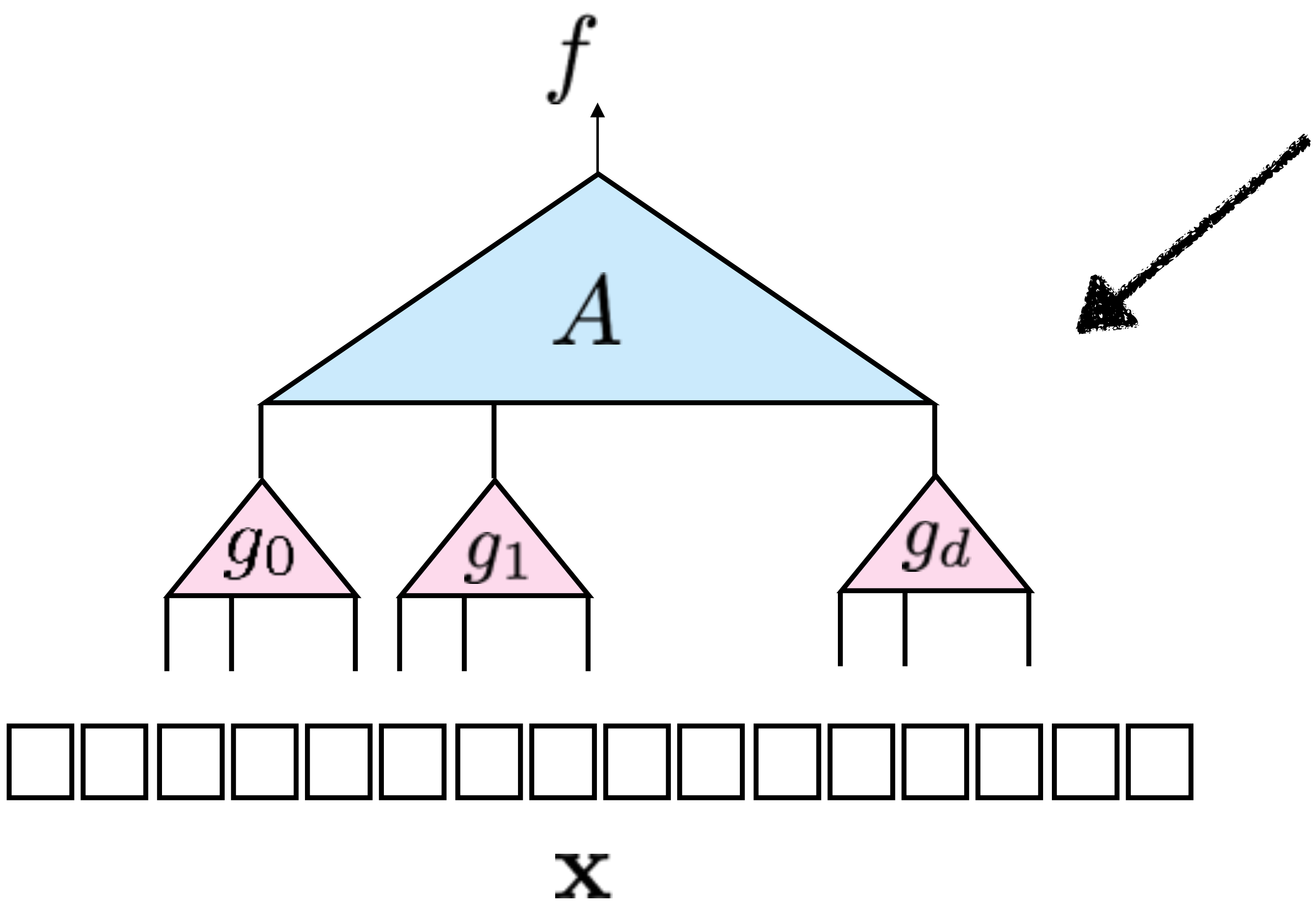


Valiant Theorem

$A(\mathbf{z})$

- size s
- degree d

Closure for VNP [**C**KS18]



Valiant Theorem

$A(\mathbf{z})$

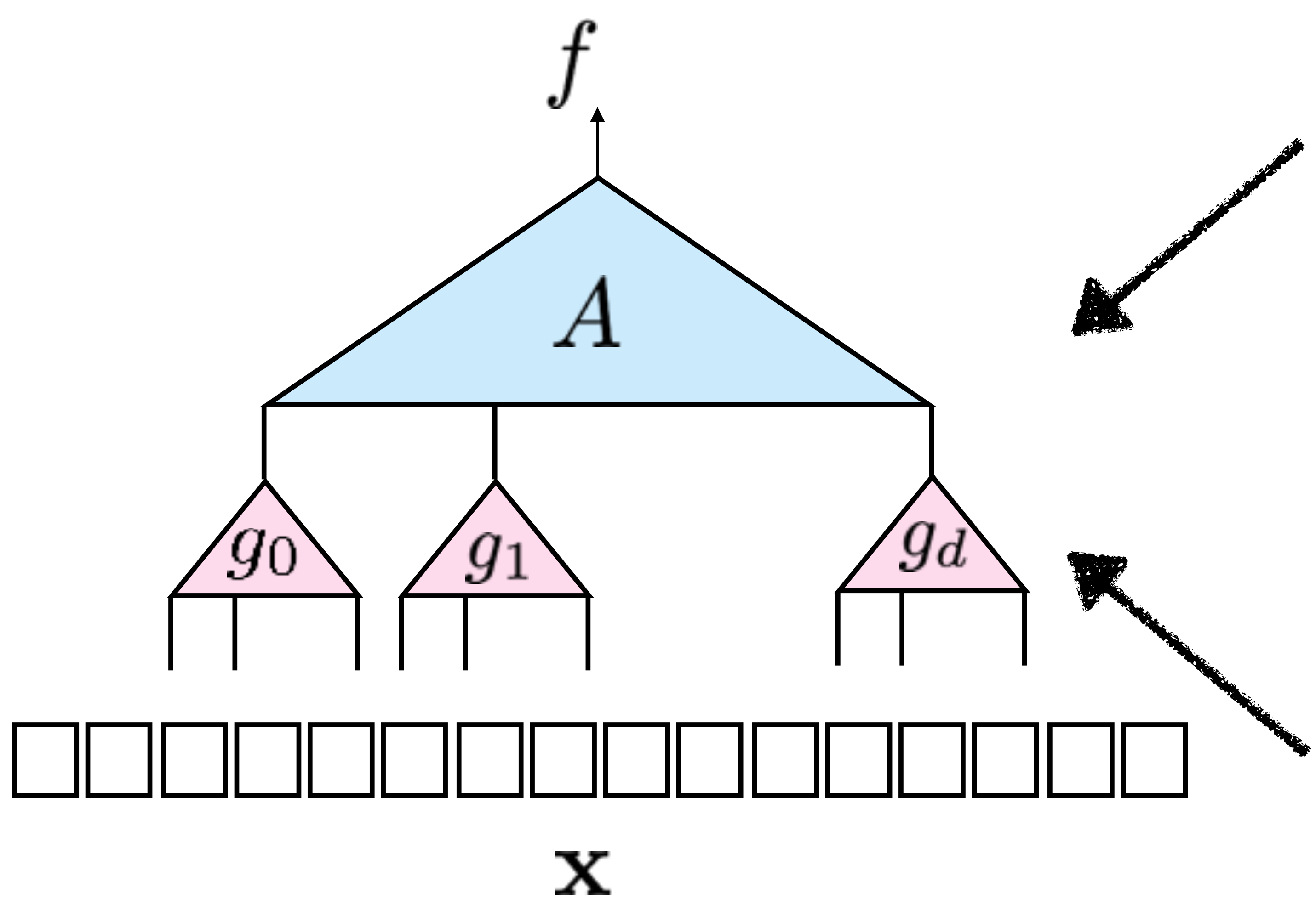
- size s
- degree d

$\Downarrow$

$$\sum_{\mathbf{c}\in\{0,1\}^{|\mathbf{y}|}} Q(\mathbf{z},\mathbf{c})$$

- size **poly**(s, d)
- Q is **formula**

Closure for VNP [**C**KS18]



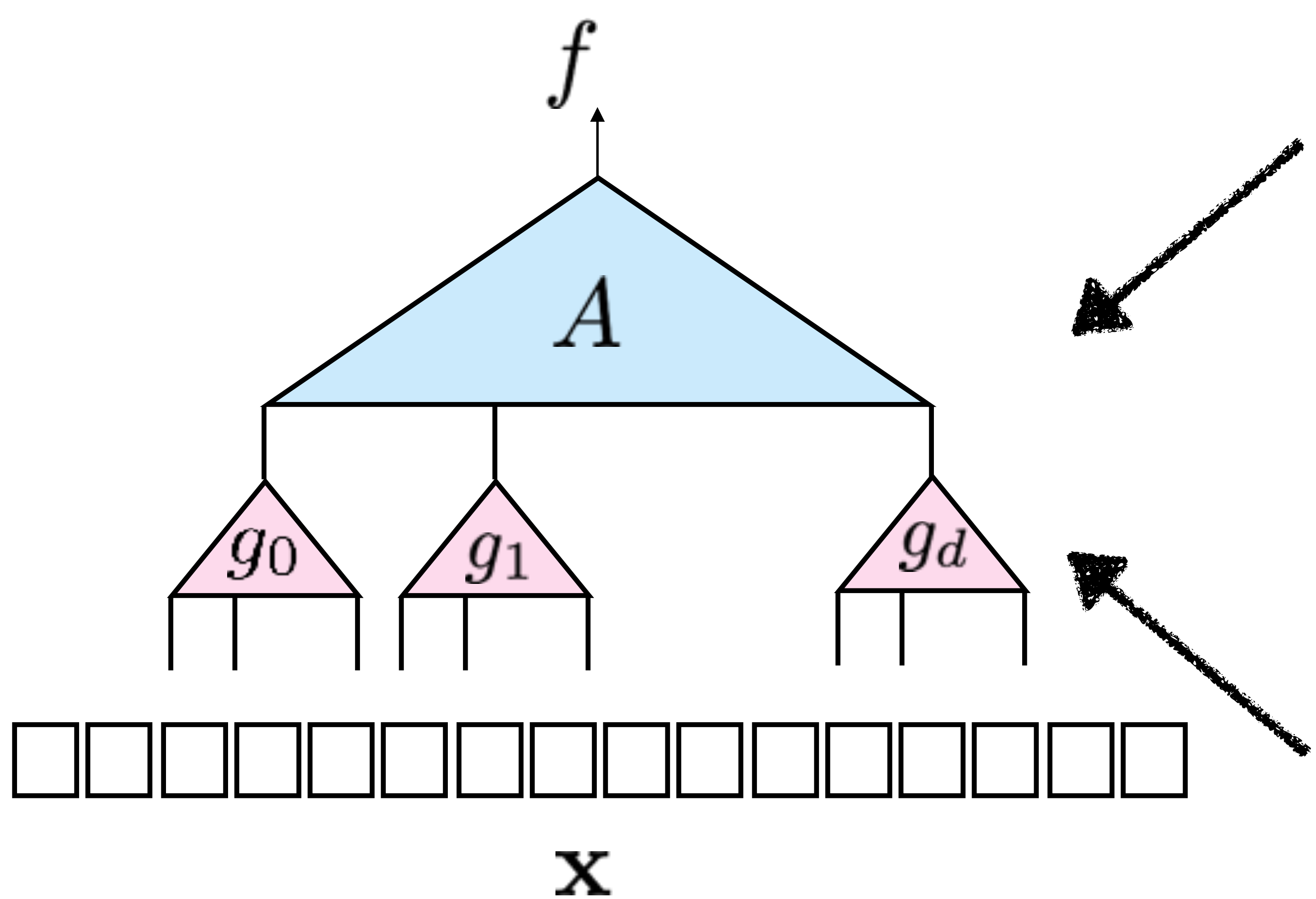
Valiant Theorem

$A(\mathbf{z})$
- size s
- degree d

$\sum_{\mathbf{c}\in\{0,1\}^{|\mathbf{y}|}} Q(\mathbf{z},\mathbf{c})$
- size poly(s, d)
- Q is **formula**

Interpolation for exponential sum

Closure for VNP [**C**KS18]



Valiant Theorem

$A(\mathbf{z})$

- size s
- degree d

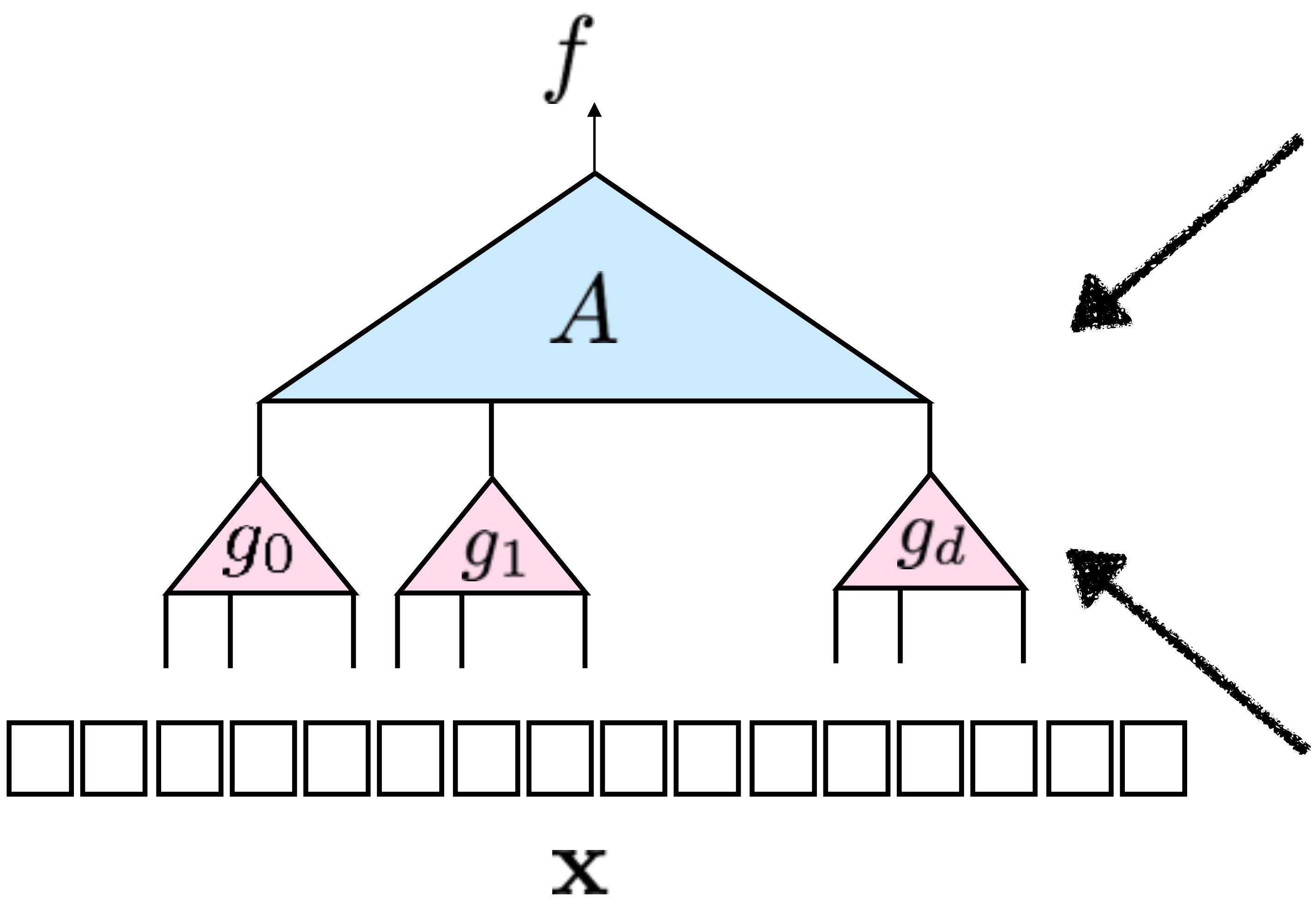
$\sum_{\mathbf{c}\in\{0,1\}^{|\mathbf{y}|}} Q(\mathbf{z},\mathbf{c})$

- size **poly**(s, d)
- Q is **formula**

Interpolation for exponential sum

- size has polynomial blowup

Closure for VNP [**C**KS18]



Valiant Theorem

$A(\mathbf{z})$
- size s
- degree d

$\Downarrow$

$\sum_{\mathbf{c}\in\{0,1\}^{|\mathbf{y}|}} Q(\mathbf{z},\mathbf{c})$
- size $\mathrm{poly}(s, d)$
- Q is **formula**

Interpolation for exponential sum

- size has polynomial blowup

The factor f has a small exponential sum!

Proof of Generator Lemma

Proof of Generator Lemma

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $O(d^2 i)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(g_0, g_1, \ldots, g_d)]$$

where

$$g_i = \mathcal{H}_{\leq d}\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right] - \mathcal{H}_0\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right].$$

Proof of Generator Lemma

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $O(d^2 i)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(g_0, g_1, \ldots, g_d)]$$

where

$$g_i = \mathcal{H}_{\leq d}\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right] - \mathcal{H}_0\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right].$$

Notation: Let $h_i = A_i(g_0, g_1, \ldots, g_d)$.

Proof of Generator Lemma

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $O(d^2 i)$ such that

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(g_0, g_1, \ldots, g_d)]$$

where

$$g_i = \mathcal{H}_{\leq d}\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right] - \mathcal{H}_0\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right].$$

Notation: Let $h_i = A_i(g_0, g_1, \ldots, g_d)$.

Goal: Show that A_i is small and $\mathcal{H}_{\leq i}[h_i] = \mathcal{H}_{\leq i}[f]$.

Proof of Generator Lemma

Lemma [CKS18]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i of size $O(d^2 i)$ such that

$$\mathcal{H}_{\le i}[f] = \mathcal{H}_{\le i}[A_i(g_0, g_1, \ldots, g_d)]$$

where

$$g_i = \mathcal{H}_{\le d}\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right] - \mathcal{H}_0\left[\frac{\partial^i}{\partial y^i} P(\mathbf{z}, \mathcal{H}[f])\right].$$

Notation: Let $h_i = A_i(g_0, g_1, \ldots, g_d)$.

Goal: Show that A_i is small and $\mathcal{H}_{\le i}[h_i] = \mathcal{H}_{\le i}[f]$.

Induction step: $\mathcal{H}_{\le i}[f] = \mathcal{H}_{\le i}\left[h_{i-1} - \frac{P(h_{i-1})}{\delta}\right]$.

Proof of Generator Lemma

Notation: Let $h_i = A_i(g_0, g_1, \ldots, g_d)$.

Induction step: $\mathcal{H}_{\le i}[f] = \mathcal{H}_{\le i}\left[h_{i-1} - \frac{P(h_{i-1})}{\delta}\right]$.

Proof of Generator Lemma

Notation: Let $h_i = A_i(g_0, g_1, \ldots, g_d)$.

Induction step: $\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}\left[h_{i-1} - \frac{P(h_{i-1})}{\delta}\right]$.

1. Write $h_{i-1} = f_0 + \tilde{h}$ and apply Taylor's expansion on $P(h_{i-1})$, we have

$$P(h_{i-1}) = f_0 + \frac{\partial}{\partial y} P(f_0)\tilde{h} + \cdots + \frac{\partial^i}{\partial y^i} P(f_0)\tilde{h}^i + \cdots .$$

Proof of Generator Lemma

Notation: Let $h_i = A_i(g_0, g_1, \ldots, g_d)$.

Induction step: $\mathcal{H}_{\le i}[f] = \mathcal{H}_{\le i}\left[h_{i-1} - \frac{P(h_{i-1})}{\delta}\right]$.

1. Write $h_{i-1} = f_0 + \tilde{h}$ and apply Taylor's expansion on $P(h_{i-1})$, we have

$$P(h_{i-1}) = f_0 + \frac{\partial}{\partial y}P(f_0)\tilde{h} + \cdots + \frac{\partial^i}{\partial y^i}P(f_0)\tilde{h}^i + \cdots .$$

2. As $\deg(\tilde{h}) \ge 1$, we have

$$\mathcal{H}_{\le i}[P(h_{i-1})] = \mathcal{H}_{\le i}\left[f_0 + \mathcal{H}_{\le d}\left[\frac{\partial}{\partial y}P(f_0)\right]\tilde{h} + \cdots + \mathcal{H}_{\le d}\left[\frac{\partial^i}{\partial y^i}P(f_0)\right]\tilde{h}^i\right].$$

Proof of Generator Lemma

Notation: Let $h_i = A_i(g_0, g_1, \ldots, g_d)$.

Induction step: $\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}\left[h_{i-1} - \frac{P(h_{i-1})}{\delta}\right]$.

1. Write $h_{i-1} = f_0 + \tilde{h}$ and apply Taylor's expansion on $P(h_{i-1})$, we have

$$P(h_{i-1}) = f_0 + \frac{\partial}{\partial y}P(f_0)\tilde{h} + \cdots + \frac{\partial^i}{\partial y^i}P(f_0)\tilde{h}^i + \cdots .$$

2. As $\deg(\tilde{h}) \geq 1$, we have

$$\mathcal{H}_{\leq i}[P(h_{i-1})] = \mathcal{H}_{\leq i}\left[f_0 + \mathcal{H}_{\leq d}\left[\frac{\partial}{\partial y}P(f_0)\right]\tilde{h} + \cdots + \mathcal{H}_{\leq d}\left[\frac{\partial^i}{\partial y^i}P(f_0)\right]\tilde{h}^i\right] .$$

3. Thus, A_i can be written as circuit over $\{A_{i-1}, g_0, \ldots, g_i\}$ of size $O(d^2)$.

Proof of Generator Lemma

Notation: Let $h_i = A_i(g_0, g_1, \dots, g_d)$.

Induction step: $\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}\left[h_{i-1} - \frac{P(h_{i-1})}{\delta}\right]$.

1. Write $h_{i-1} = f_0 + \tilde{h}$ and apply Taylor's expansion on $P(h_{i-1})$, we have

$$P(h_{i-1}) = f_0 + \frac{\partial}{\partial y}P(f_0)\tilde{h} + \cdots + \frac{\partial^i}{\partial y^i}P(f_0)\tilde{h}^i + \cdots .$$

2. As $\deg(\tilde{h}) \geq 1$, we have

$$\mathcal{H}_{\leq i}[P(h_{i-1})] = \mathcal{H}_{\leq i}\left[f_0 + \mathcal{H}_{\leq d}\left[\frac{\partial}{\partial y}P(f_0)\right]\tilde{h} + \cdots + \mathcal{H}_{\leq d}\left[\frac{\partial^i}{\partial y^i}P(f_0)\right]\tilde{h}^i\right] .$$

3. Thus, A_i can be written as circuit over $\{A_{i-1}, g_0, \dots, g_i\}$ of size $O(d^2)$.

4. As the size blow-up from A_{i-1} is additive, A_i has circuit of size $O(d^2 i)$.

Conclusion

Conclusion

- **Closure for bounded depth poly-log degree circuit**
    - ✦ New hardness versus randomness connection.

Conclusion

- **Closure for bounded depth poly-log degree circuit**
  - ✦ New hardness versus randomness connection.
- **Closure for VNP**
  - ✦ Note that if VNP is not closed under factoring, then $\mathrm{VP} \neq \mathrm{VNP}$.

Conclusion

- **Closure for bounded depth poly-log degree circuit**
  - New hardness versus randomness connection.
- **Closure for VNP**
  - Note that if VNP is not closed under factoring, then $\mathsf{VP} \neq \mathsf{VNP}$.
- **Key technique**
  - A more efficient generator lemma.

Open Problems & Future Directions

Open Problems & Future Directions

- **Bounded depth circuits**
  - Remove the degree conditions in [DSY09,**C**KS18]?

Open Problems & Future Directions

- **Bounded depth circuits**
  - ✦ Remove the degree conditions in [DSY09,**C**KS18]?
- **Formulas and Branching programs**
  - ✦ Remove the quasi-poly blow up in [DSS18]?

Open Problems & Future Directions

- **Bounded depth circuits**
  - ✦ Remove the degree conditions in [DSY09,**C**KS18]?
- **Formulas and Branching programs**
  - ✦ Remove the quasi-poly blow up in [DSS18]?
- **High-degree regime**
  - ✦ Factoring conjecture [Bürgisser 00]: Let $f = gh$ have $\mathsf{poly}(n)$ size circuit, does its **low-degree factor** also have small circuit too?

Open Problems & Future Directions

Thank you!

- **Bounded depth circuits**
  - ✦ Remove the degree conditions in [DSY09,**C**KS18]?
- **Formulas and Branching programs**
  - ✦ Remove the quasi-poly blow up in [DSS18]?
- **High-degree regime**
  - ✦ Factoring conjecture [Bürgisser 00]: Let $f = gh$ have $\mathsf{poly}(n)$ size circuit, does its **low-degree factor** also have small circuit too?