Hardness vs Randomness for Bounded Depth Arithmetic Circuits

Chi-Ning Chou Mrinal Kumar Noam Solomon

Harvard University

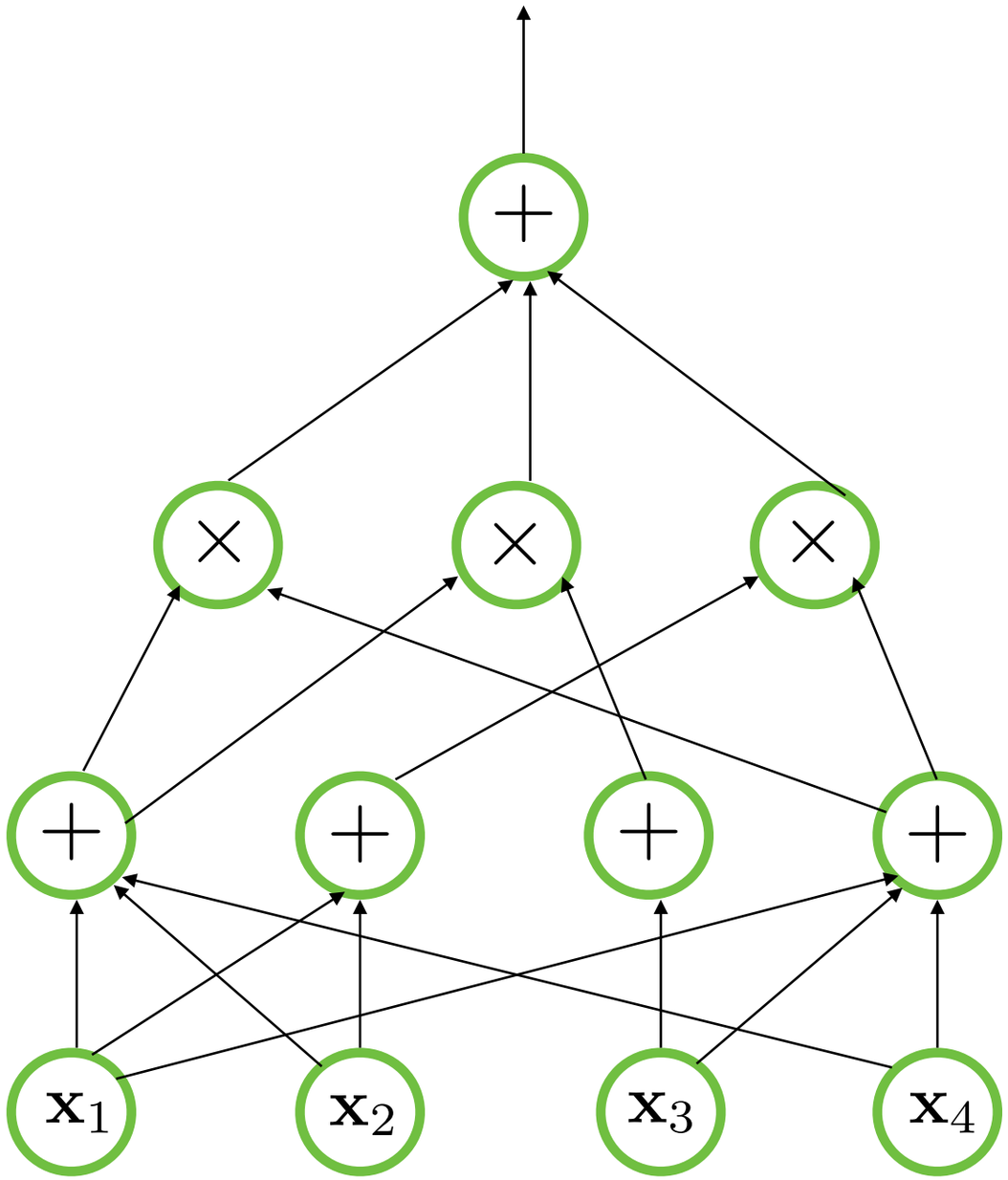
Outline

- Arithmetic circuits and algebraic complexity classes
- Polynomial identity testing (PIT)
- Hardness vs Randomness for arithmetic circuits
- Polynomial factorization
- Open problems

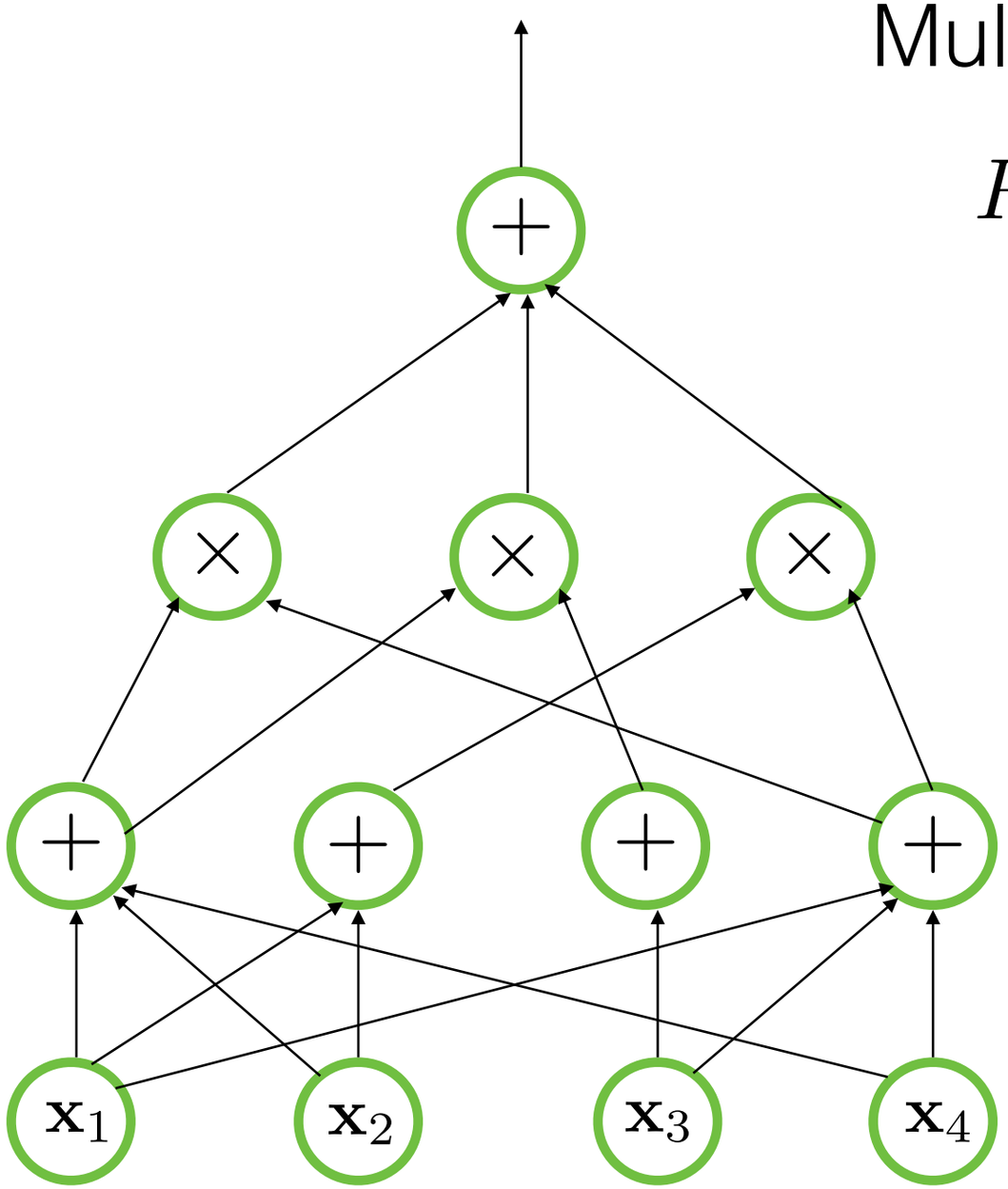
Outline

- Arithmetic circuits and algebraic complexity classes
- Polynomial identity testing (PIT)
- Hardness vs Randomness for arithmetic circuits
- Polynomial factorization
- Open problems

Arithmetic circuits



Arithmetic circuits



Multivariate polynomial

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

Arithmetic circuits

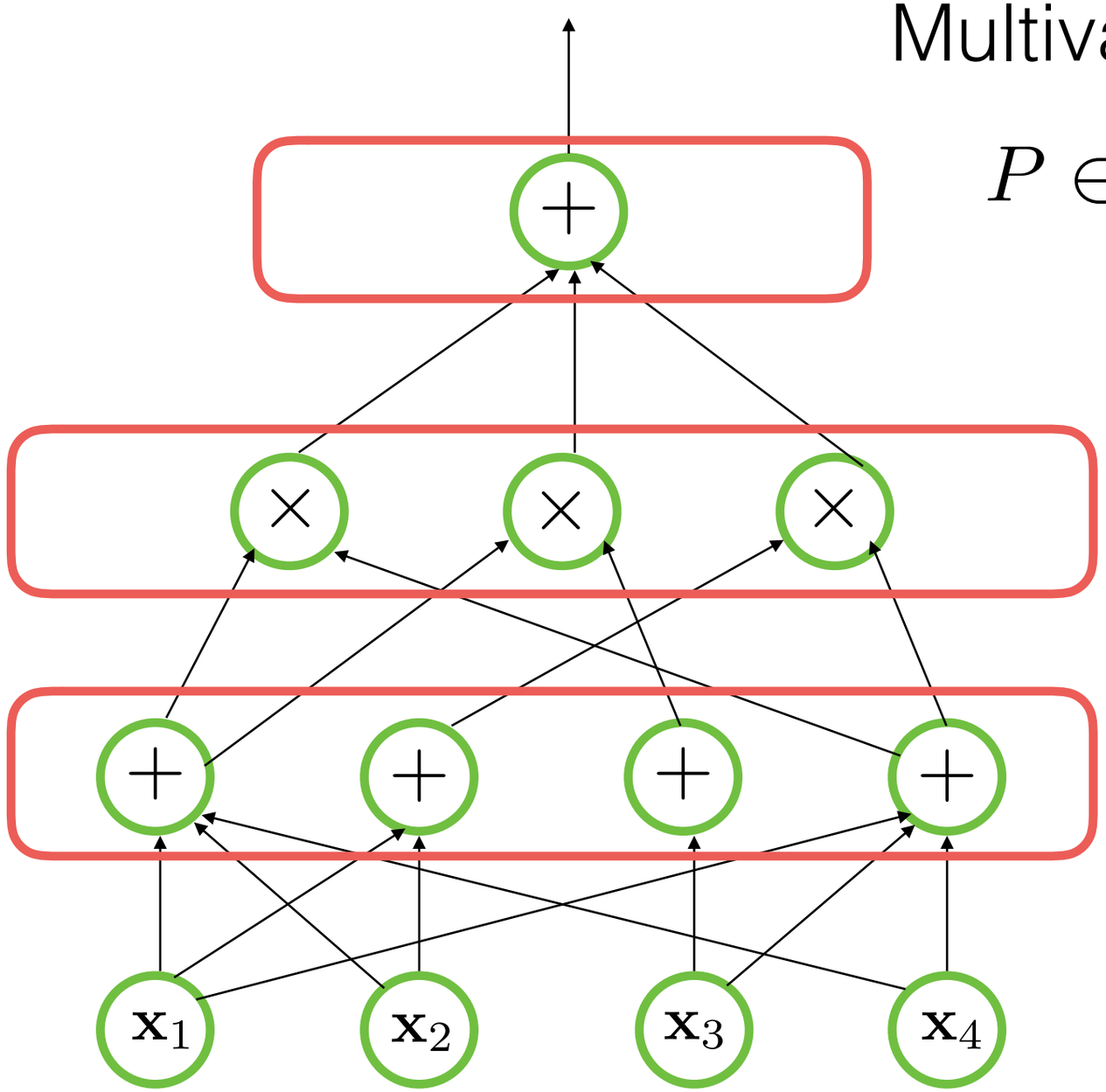
Multivariate polynomial

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

Σ

Π

Σ



Arithmetic circuits

Multivariate polynomial

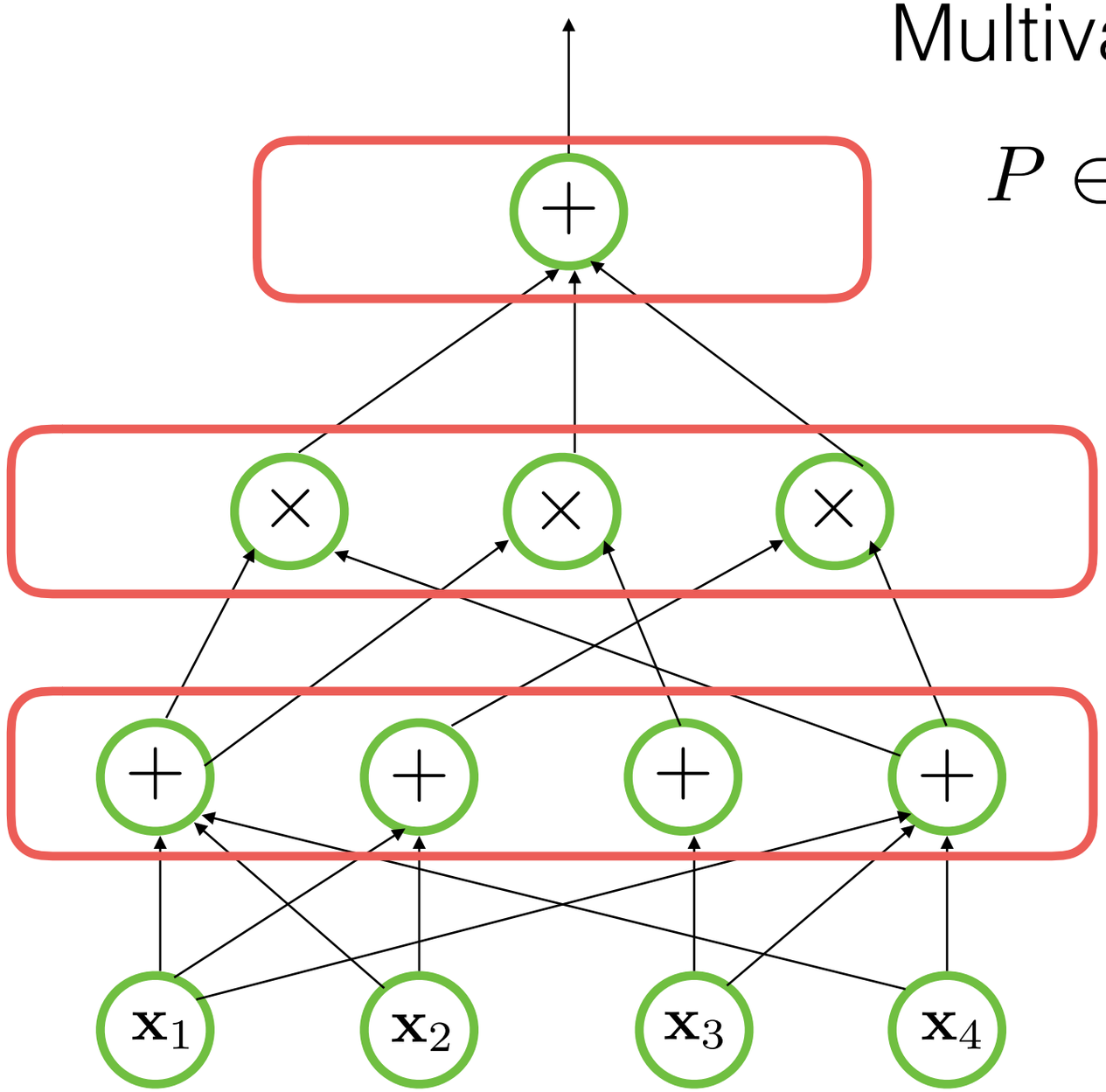
$$P \in \mathbb{F}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$$

Depth - 3

Σ

Π

Σ

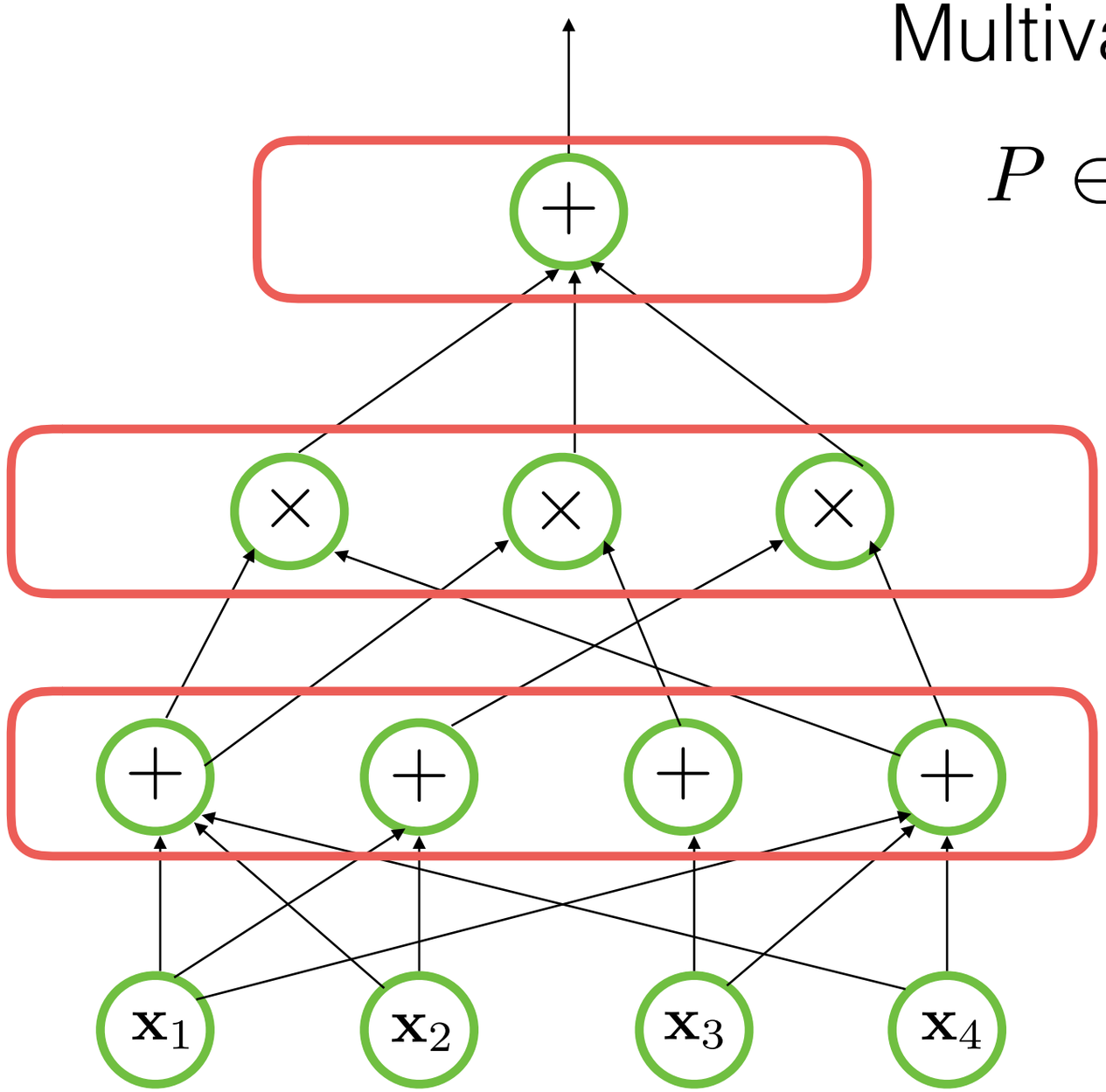


Arithmetic circuits

Σ

Π

Σ



Multivariate polynomial

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

Depth - 3

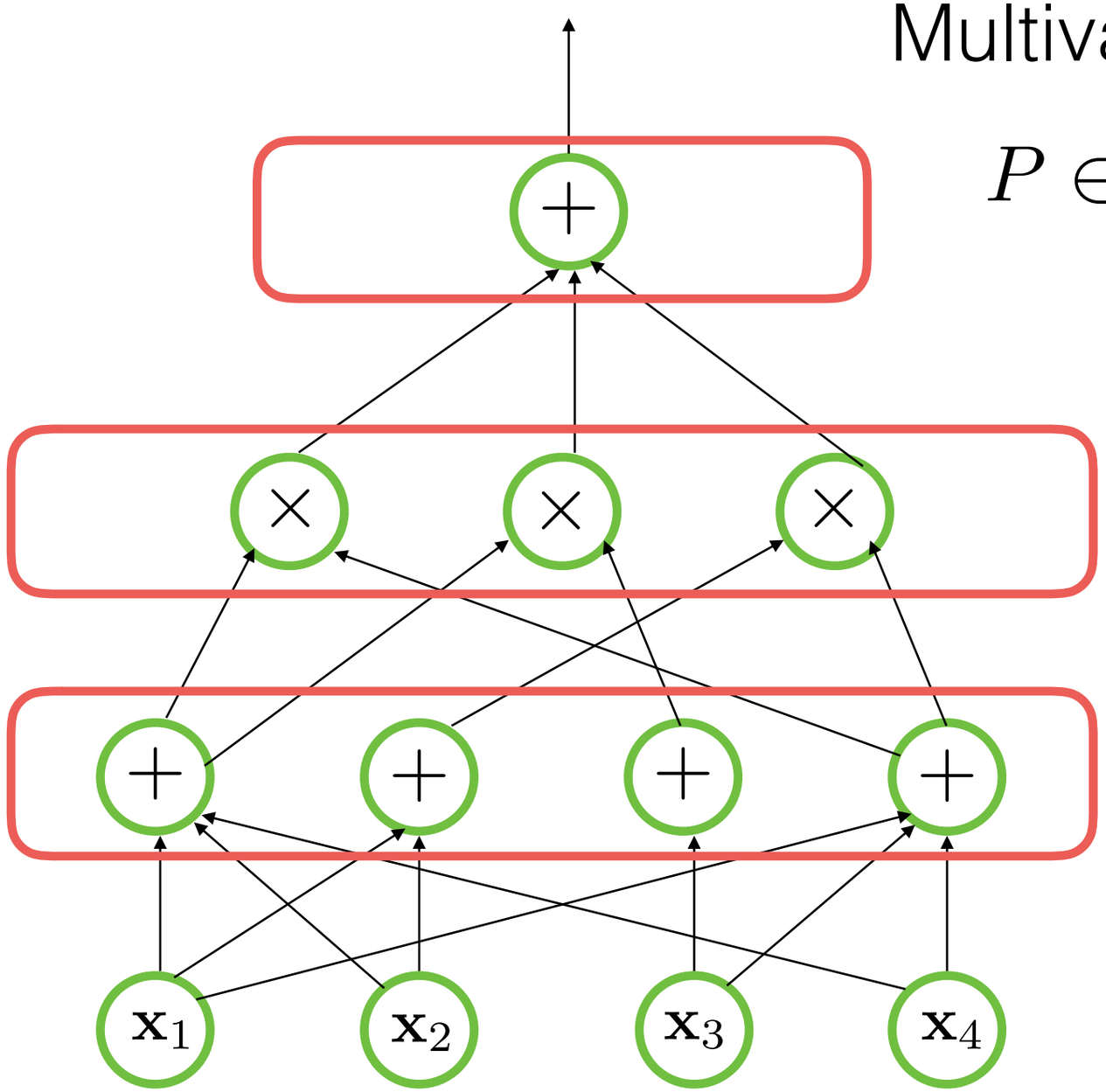
Size - 8

Arithmetic circuits

Σ

Π

Σ



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

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

Algebraic complexity classes

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

For simplicity, denote $f = f_n$.

Algebraic complexity classes

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

For simplicity, denote $f = f_n$.

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

Algebraic complexity classes

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

For simplicity, denote $f = f_n$.

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

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

Algebraic complexity classes

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

For simplicity, denote $f = f_n$.

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

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

- Many more such as VF, VBP, VNP…

Hardness - Lower bounds

Hardness - Lower bounds

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

Hardness - Lower bounds

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

- [Strassen 73, Baur & Strassen 83] An **n log n** lower bound for general arithmetic circuits.

Hardness - Lower bounds

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

- [Strassen 73, Baur & Strassen 83] An **n log n** lower bound for general arithmetic circuits.
- [Kalorkoti 87] A **quadratic** lower bound for arithmetic formula.

Hardness - Lower bounds

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

- [Strassen 73, Baur & Strassen 83] An **n log n** lower bound for general arithmetic circuits.
- [Kalorkoti 87] A **quadratic** lower bound for arithmetic formula.
- [Kumar 17] A **quadratic** lower bound for homogeneous algebraic branching programs.

Hardness - Lower bounds

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

- [Strassen 73, Baur & Strassen 83] An **n log n** lower bound for general arithmetic circuits.
- [Kalorkoti 87] A **quadratic** lower bound for arithmetic formula.
- [Kumar 17] A **quadratic** lower bound for homogeneous algebraic branching programs.
- [NW'95, GKKS'14, FLMS'14, KS'14] **Exponential** lower bounds for depth-3 and depth-4 circuits.

Outline

- Arithmetic circuits and algebraic complexity classes
- Polynomial identity testing (PIT)
- Hardness vs Randomness for arithmetic circuits
- Polynomial factorization
- Open problems

Randomness - Polynomial identity testing (PIT)

Randomness - Polynomial identity testing (PIT)

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

Randomness - Polynomial identity testing (PIT)

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

Randomness - Polynomial identity testing (PIT)

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

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

Randomness - Polynomial identity testing (PIT)

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 $\mathrm{Depth}\text{-}\Delta$.

Randomness - Polynomial identity testing (PIT)

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

Randomness - Polynomial identity testing (PIT)

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 = Hitting Set

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

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

Randomness - Polynomial identity testing (PIT)

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** $f \in \mathcal{C}$

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

Randomness - Polynomial identity testing (PIT)

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** $f \in \mathcal{C}$

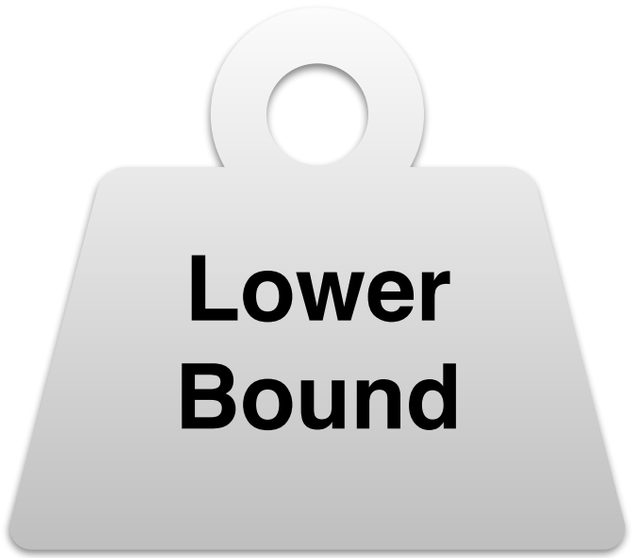
$$\exists \mathbf{a} \in \mathcal{P}, \ f(\mathbf{a}) \neq 0.$$

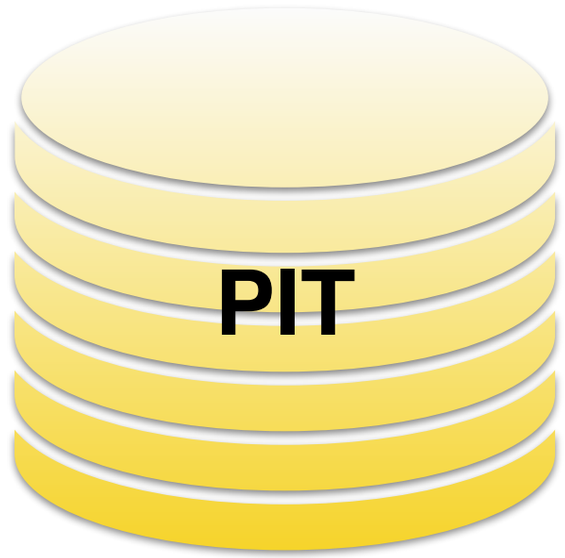
Outline

- Arithmetic circuits and algebraic complexity classes
- Polynomial identity testing (PIT)
- Hardness vs Randomness for arithmetic circuits
- Polynomial factorization
- Open problems

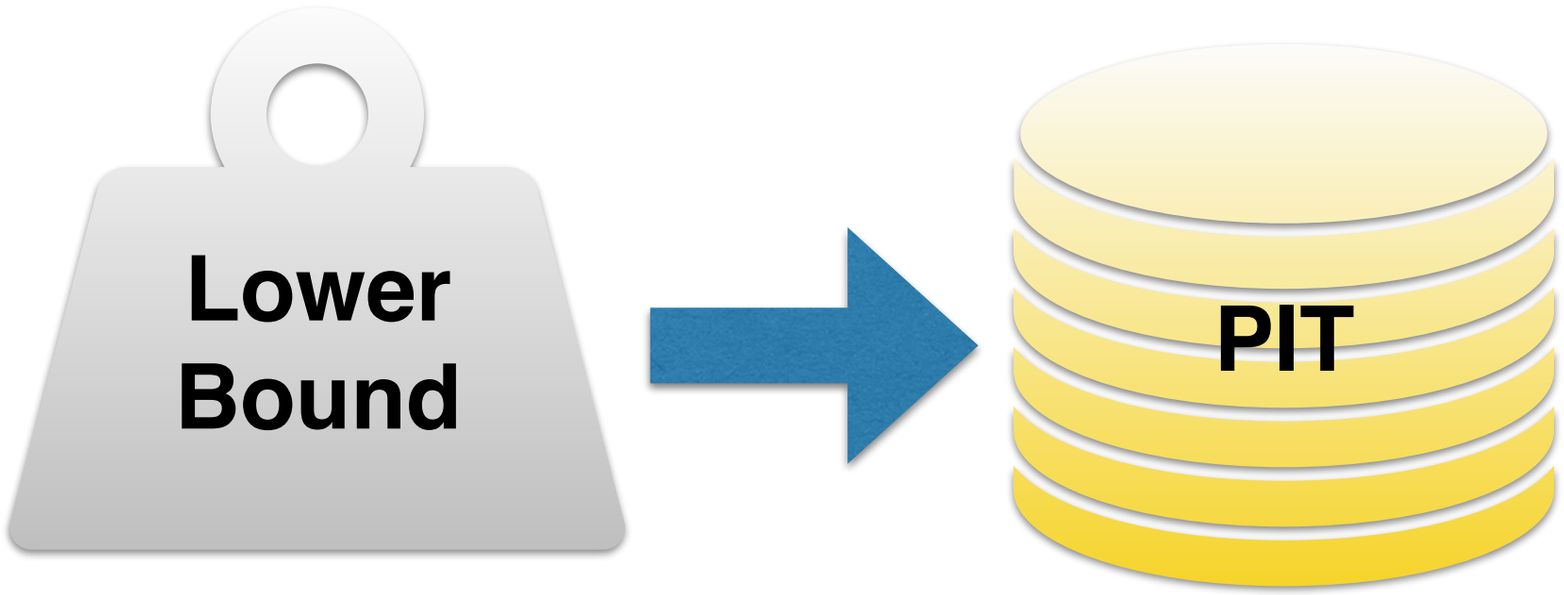
Hardness vs Randomness

Hardness vs Randomness

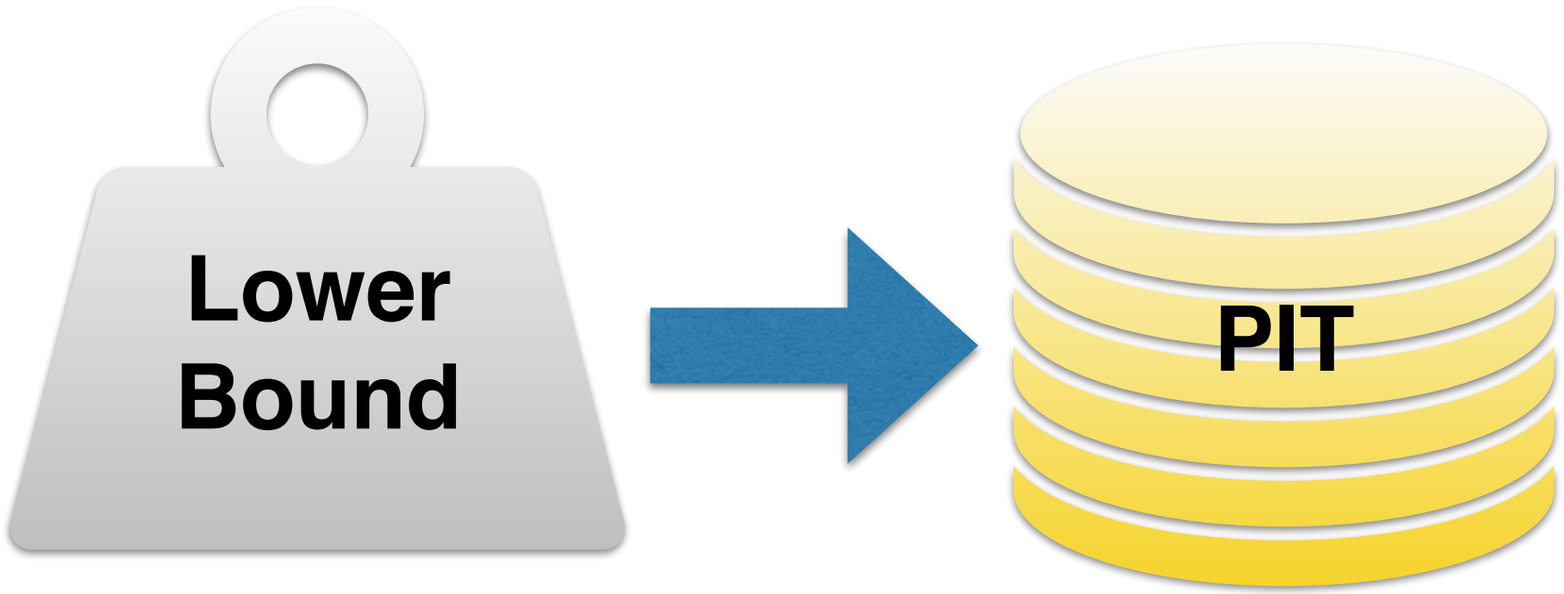




Hardness vs Randomness

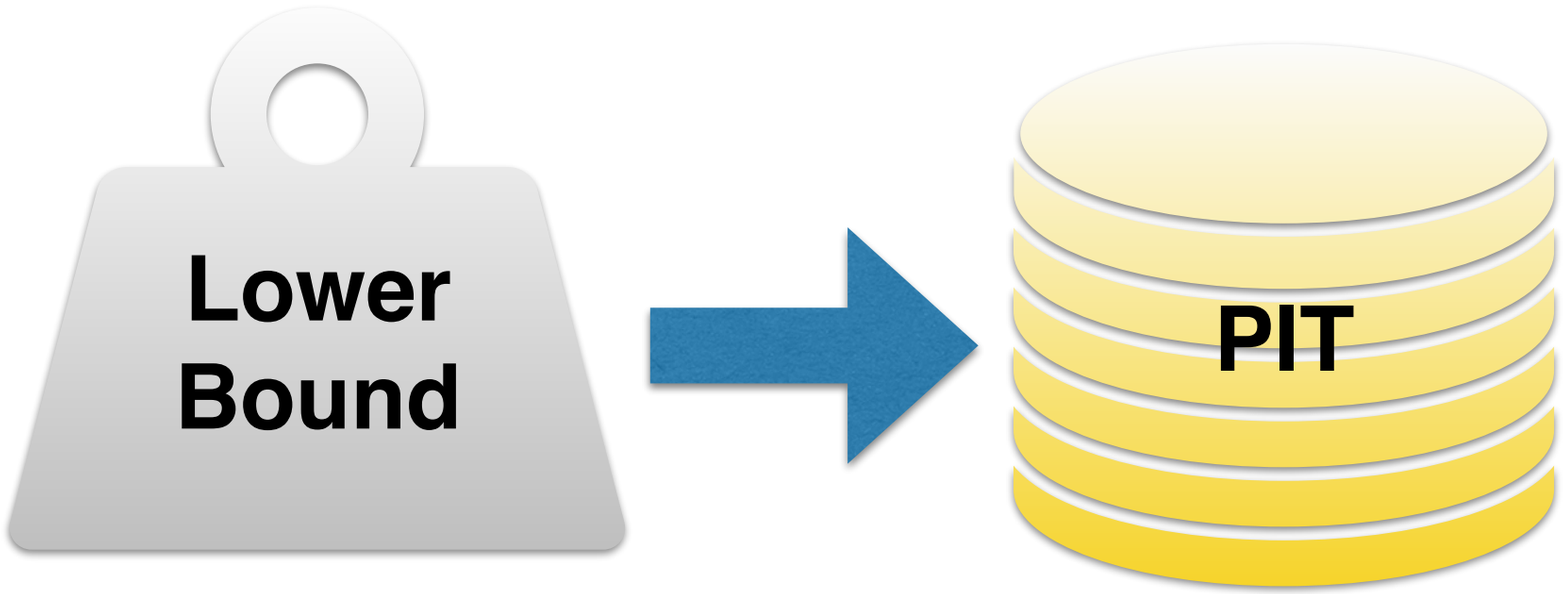


Hardness vs Randomness



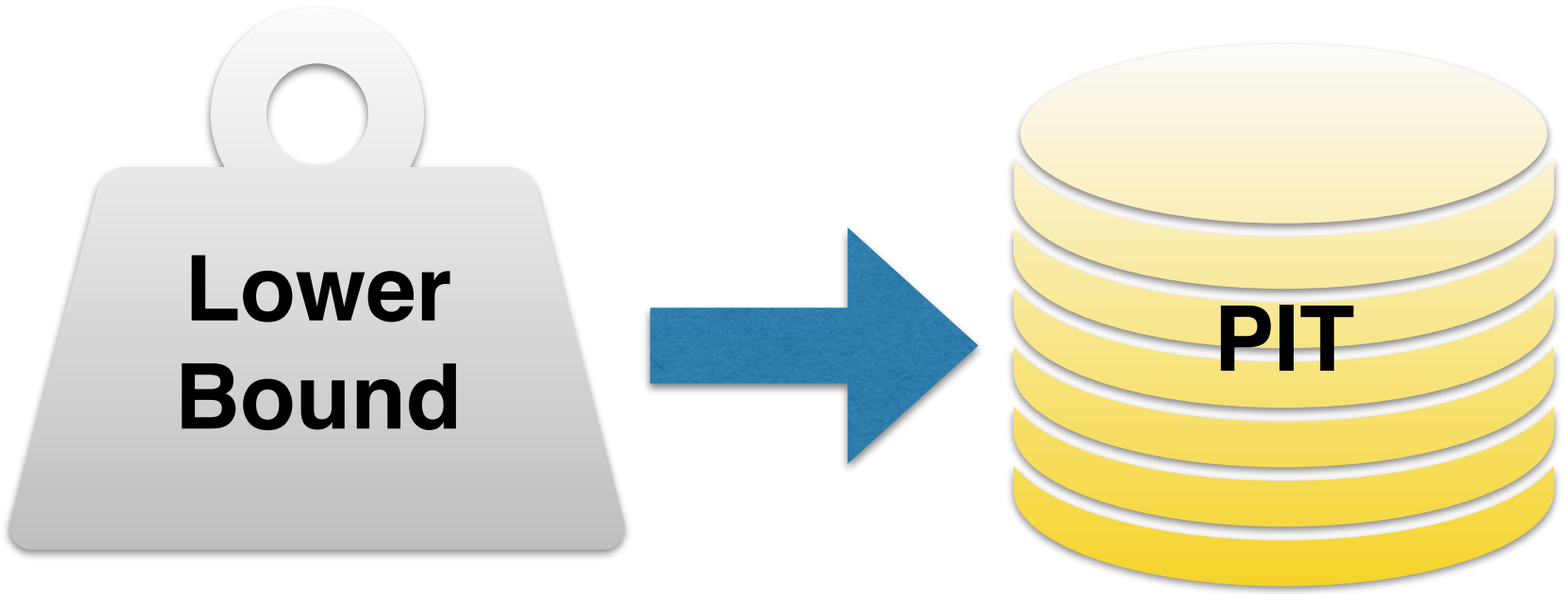
- [KI'04]: Permanent not in VP => PIT for VP

Hardness vs Randomness



- [KI'04]: Permanent not in VP => PIT for VP
- [DSY'09]: $\omega(\mathrm{poly}(n))$ for $\mathrm{Depth}\text{-}\Delta$ => PIT for $\mathrm{Depth}\text{-}\Delta - 5$

Hardness vs Randomness



- [KI'04]: Permanent not in VP => PIT for VP
- [DSY'09]: $\omega(\mathrm{poly}(n))$ for $\mathrm{Depth}\text{-}\Delta$ => PIT for $\mathrm{Depth}\text{-}\Delta - 5$

multilinear

with bounded individual degree

Our result

Theorem: For any $\Delta \geq 6$,

Our result

Theorem: For any $\Delta \geq 6$,

$\omega(\mathrm{poly}(n))$ lower bound for $\mathrm{Depth}\text{-}\Delta$

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

Our result

Theorem: For any $\Delta \geq 6$,

$\omega(\mathrm{poly}(n))$ lower bound for $\mathrm{Depth}\text{-}\Delta$

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

Sub-exponential time PIT for $\mathrm{Depth}\text{-}\Delta - 5$

Our result

Theorem: For any $\Delta \geq 6$,

$\omega(\mathrm{poly}(n))$ lower bound for $\mathrm{Depth}\text{-}\Delta$

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

$O(\log^k n / \log^k \log n)$

Sub-exponential time PIT for $\mathrm{Depth}\text{-}\Delta - 5$

$\Delta - 2k - 2$

Our result

Don't be bothered by the constant in depth!

Theorem: For any $\Delta \geq 6$,

$\omega(\mathrm{poly}(n))$ lower bound for $\mathrm{Depth}\text{-}\Delta$

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

$O(\log^k n / \log^k \log n)$

Sub-exponential time PIT for $\mathrm{Depth}\text{-}\Delta - 5$

$\Delta - 2k - 2$

Compare with [Dvir-Shpilka-Yehudayoff'09]

	[DSY'09]	This work
Lower bound for $\text{Depth-}\Delta$		With degree $O(\log^2 n / \log^2 \log n)$
PIT for $\text{Depth-}\Delta - 5$	With bounded individual degree	

Hardness vs Randomness framework [KI'04, DSY'09]

Hardness vs Randomness framework [KI'04, DSY'09]

Nisan-Wigderson generator

Reduce #variables from $n \to \ell$

Hardness vs Randomness framework [KI'04, DSY'09]

Nisan-Wigderson generator

Reduce #variables from $n \to \ell$

Schwartz-Zippel lemma

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

Hardness vs Randomness framework [KI'04, DSY'09]

Reduce to **factoring problem**!

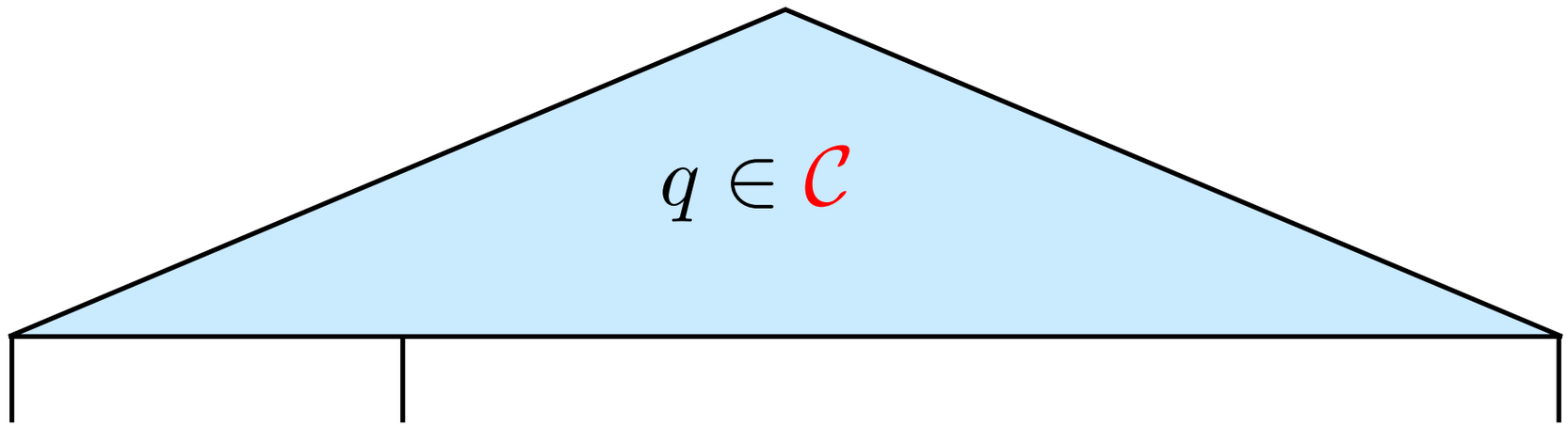
Nisan-Wigderson generator

Reduce #variables from $n \to \ell$

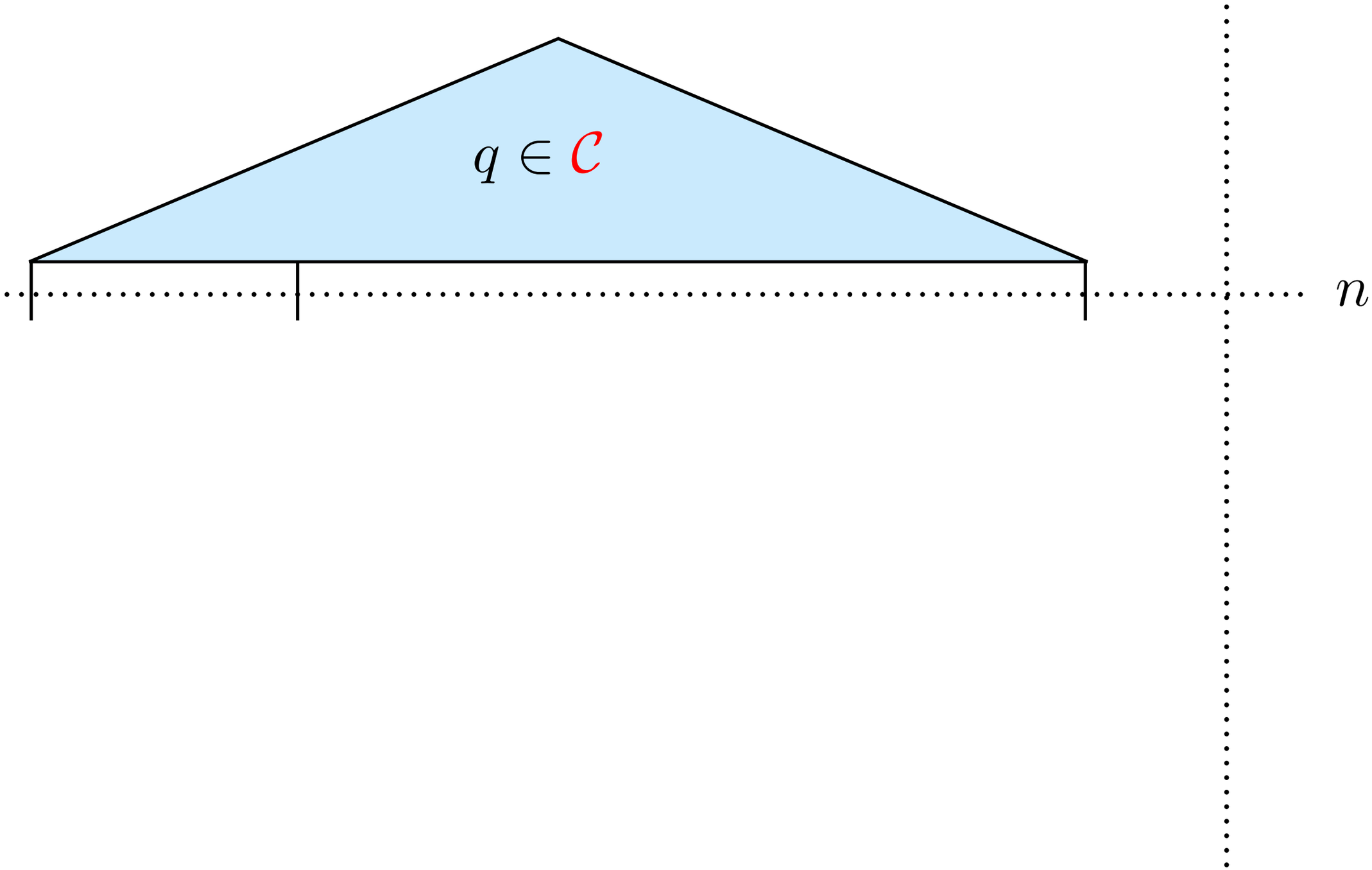
Schwartz-Zippel lemma

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

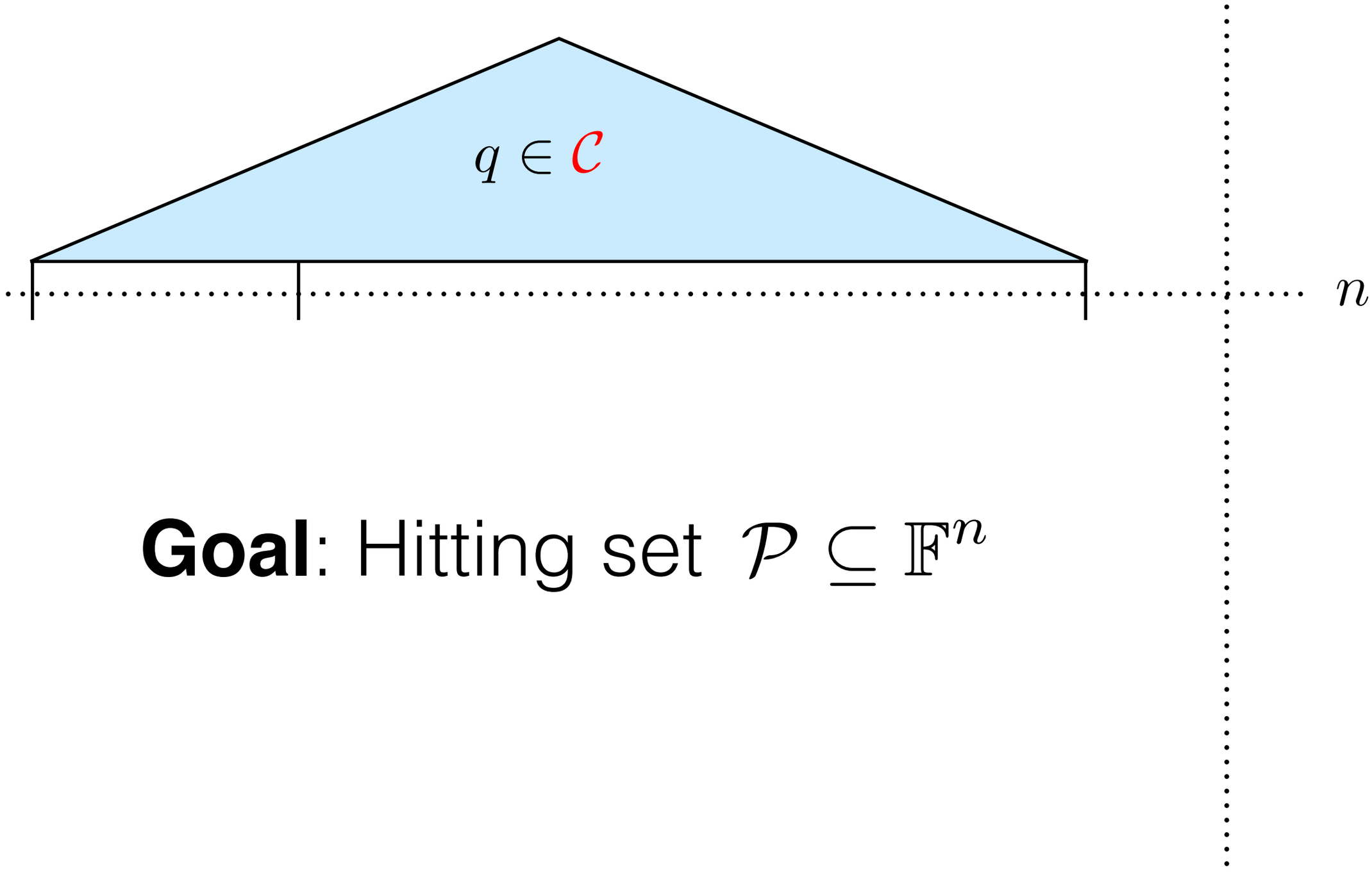
NW generator - reducing #variables



NW generator - reducing #variables

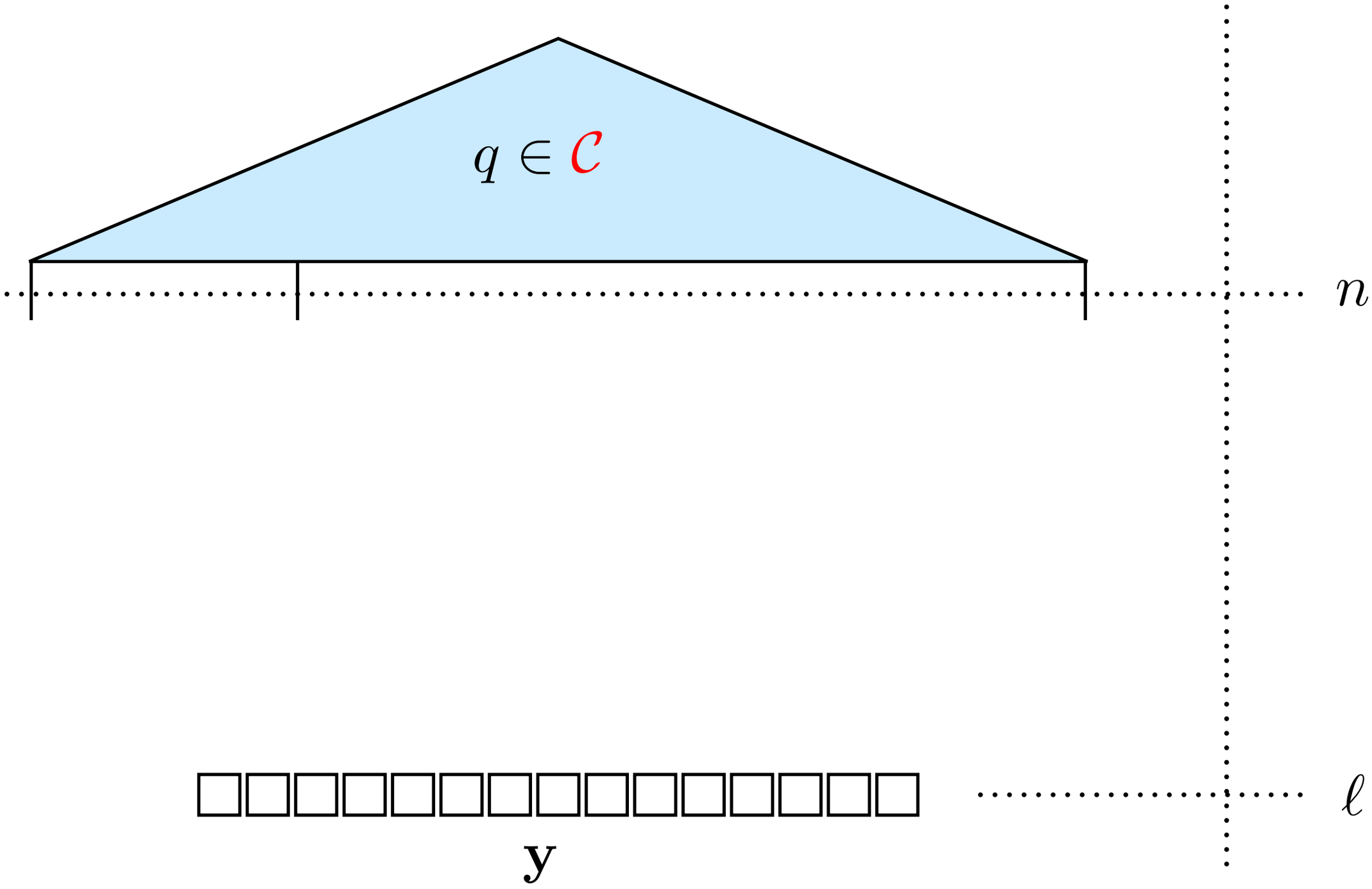


NW generator - reducing #variables

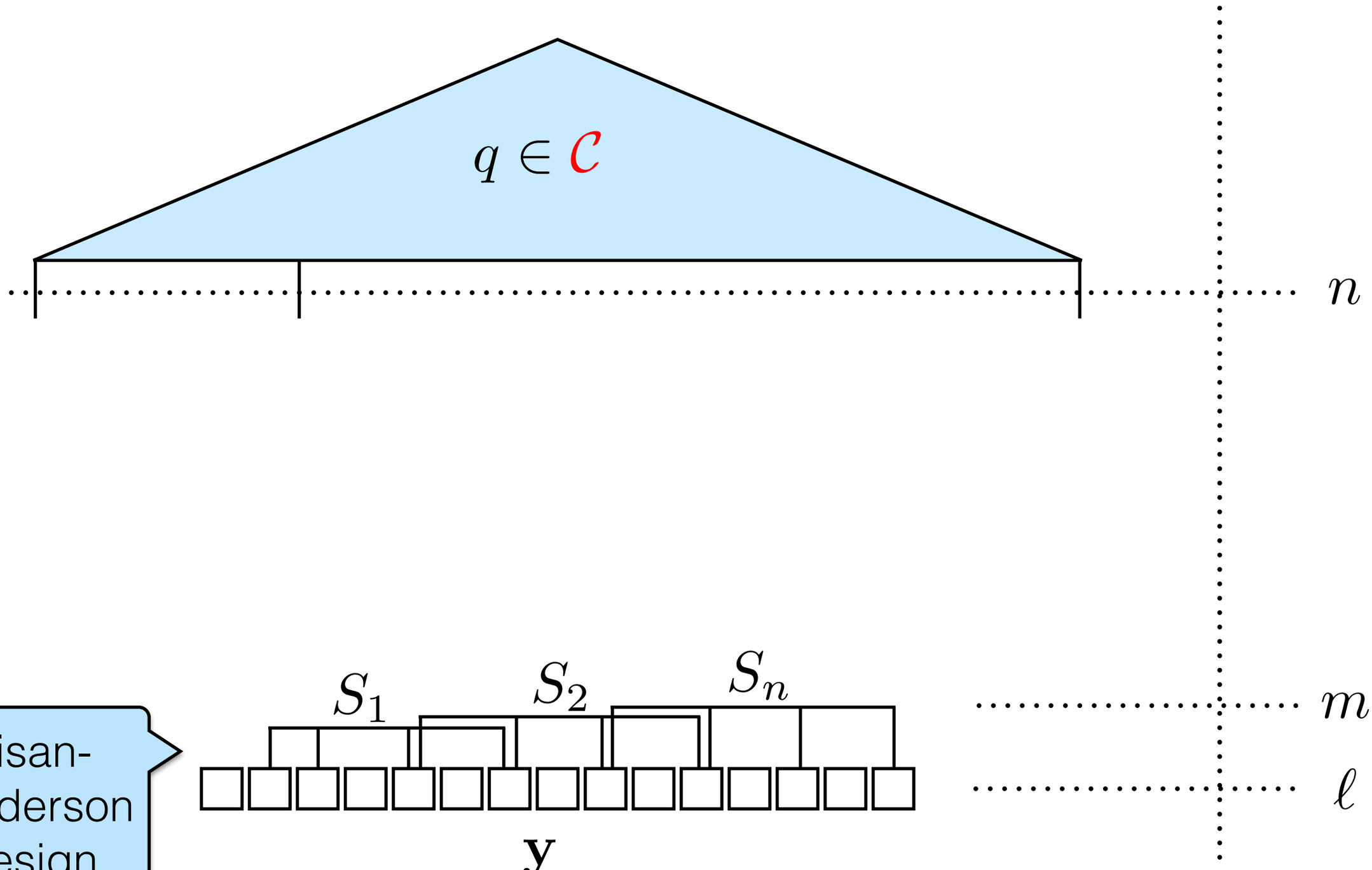


Goal: Hitting set $\mathcal{P} \subseteq \mathbb{F}^n$

NW generator - reducing #variables

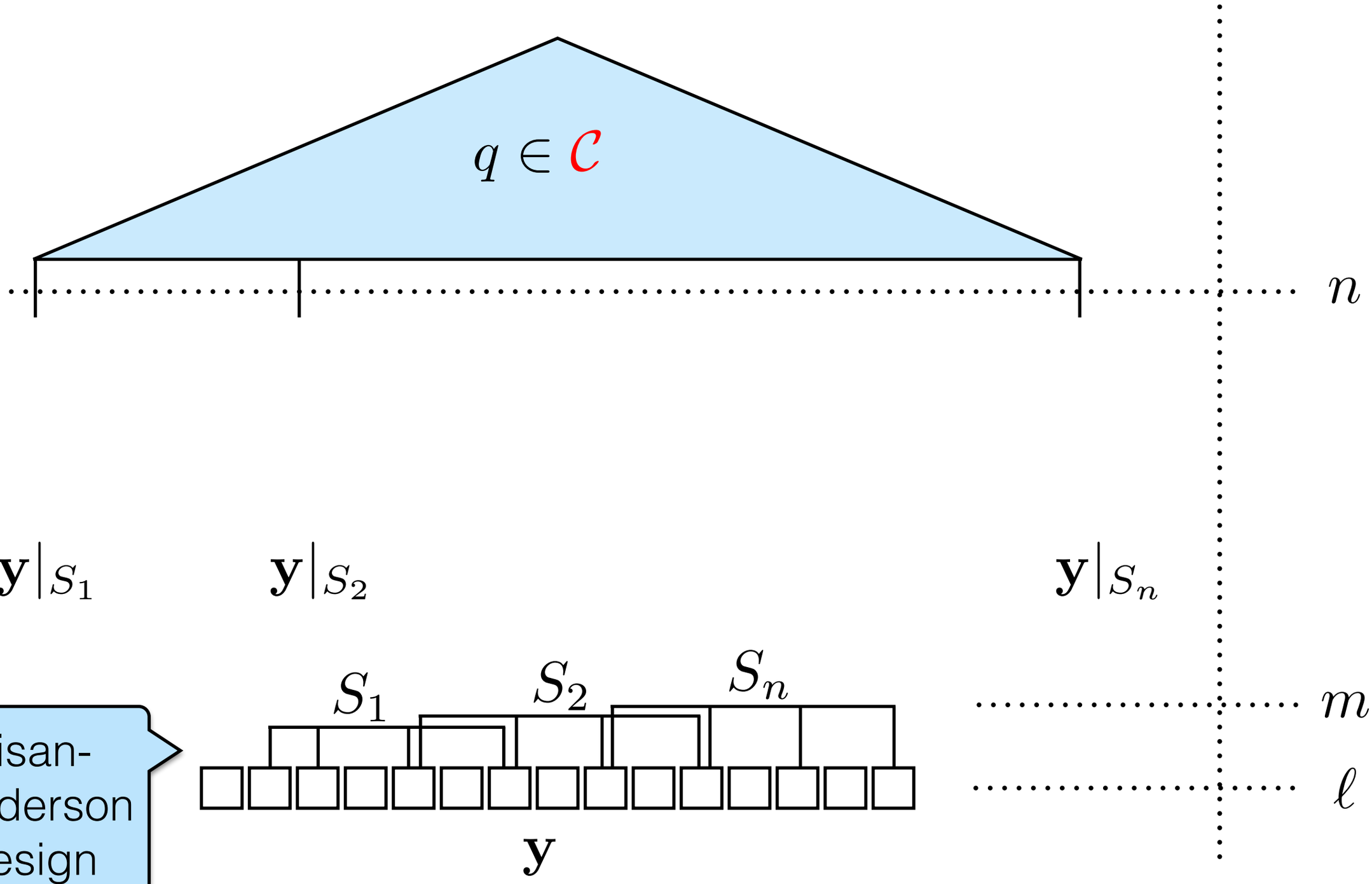


NW generator - reducing #variables



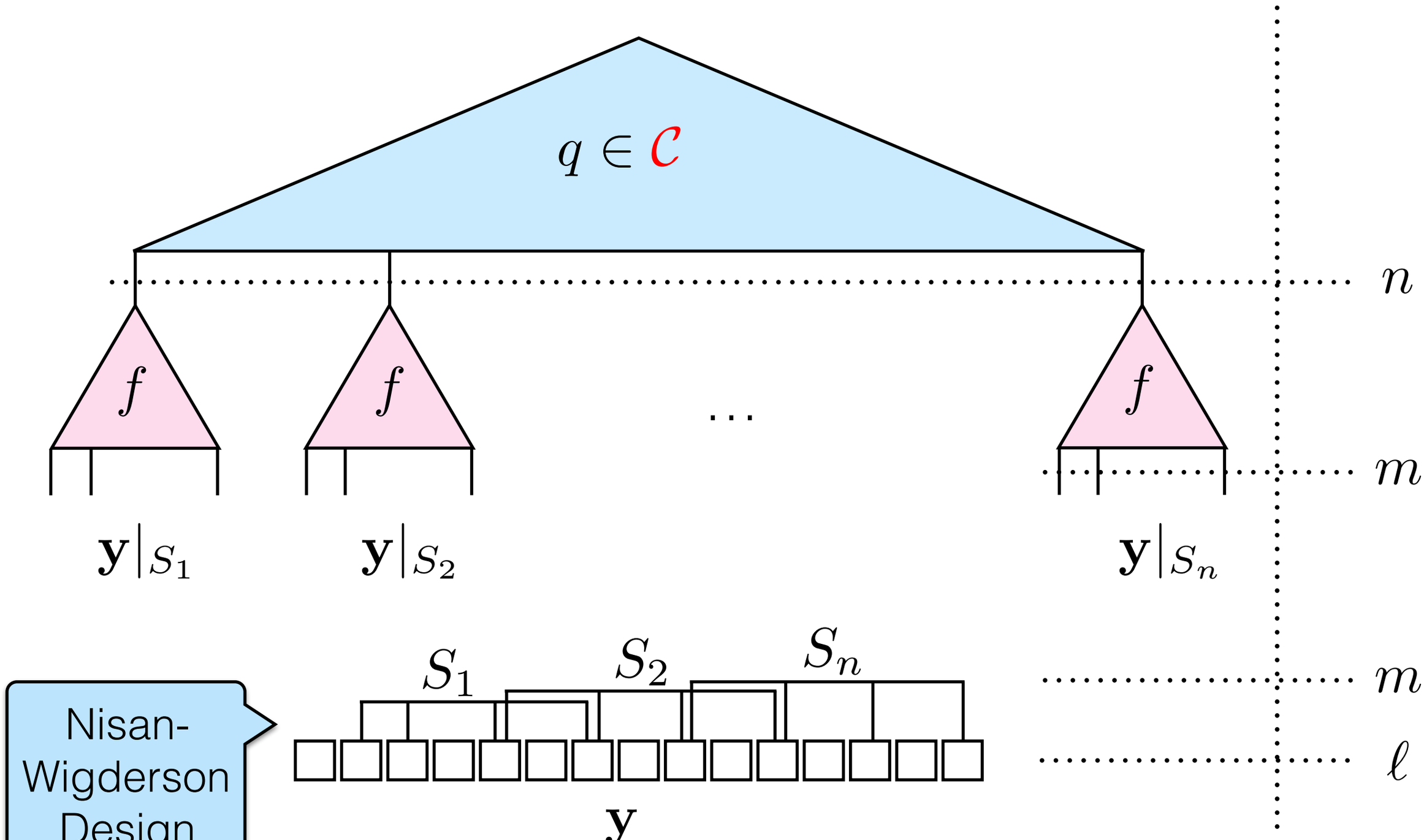
Nisan-Wigderson Design

NW generator - reducing #variables



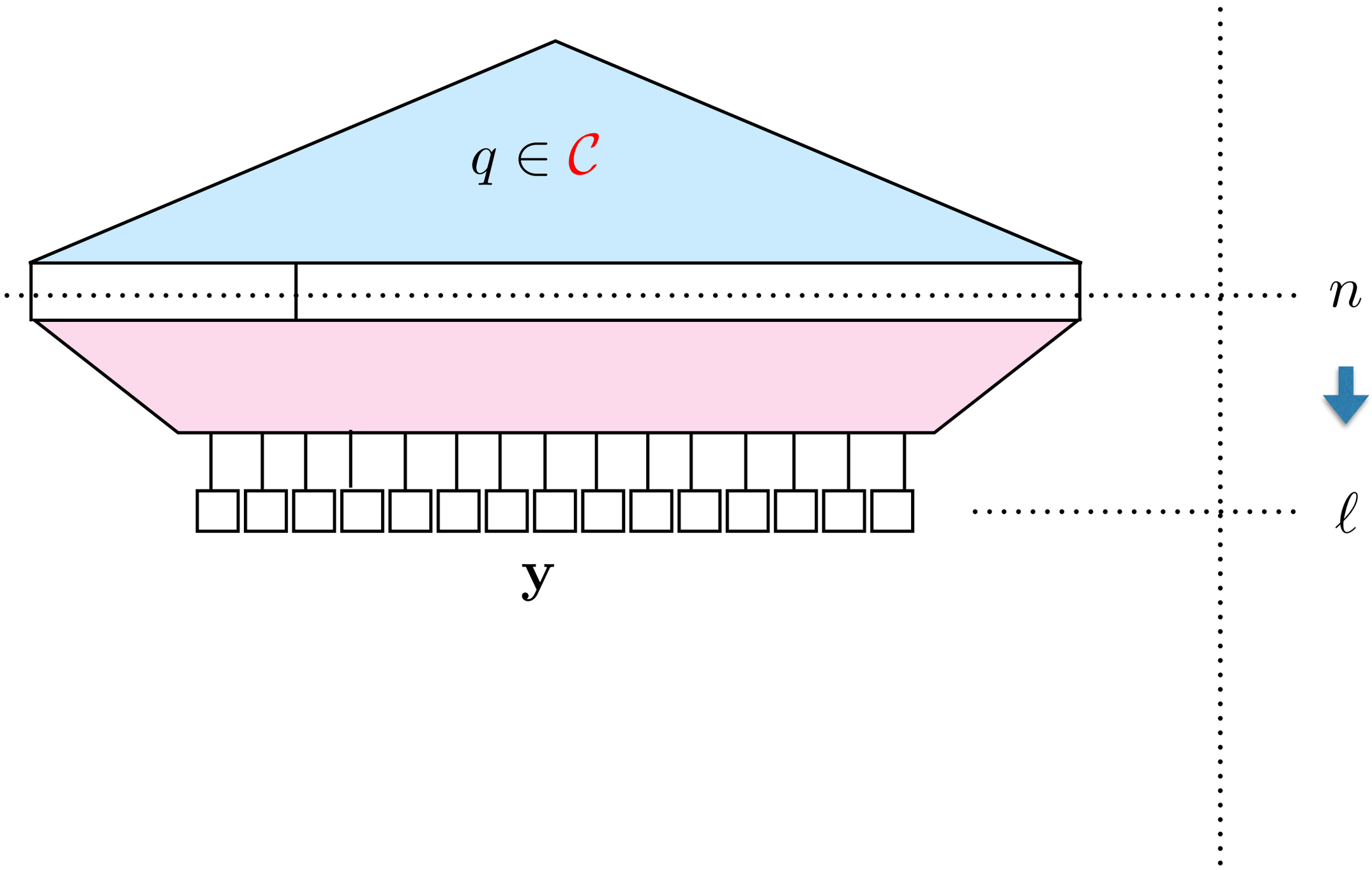
Nisan-Wigderson Design

NW generator - reducing #variables

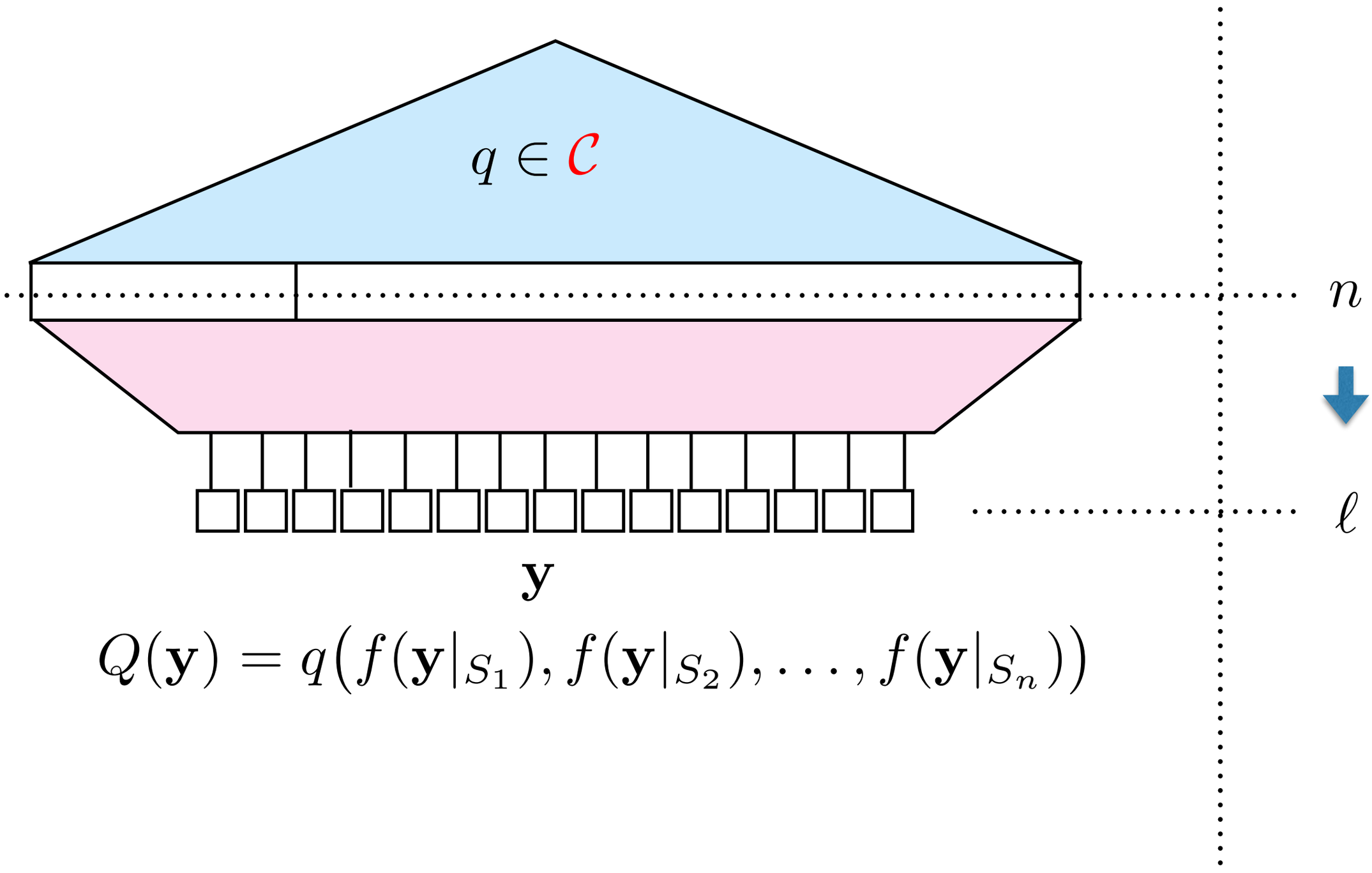


Nisan-Wigderson Design

NW generator - reducing #variables

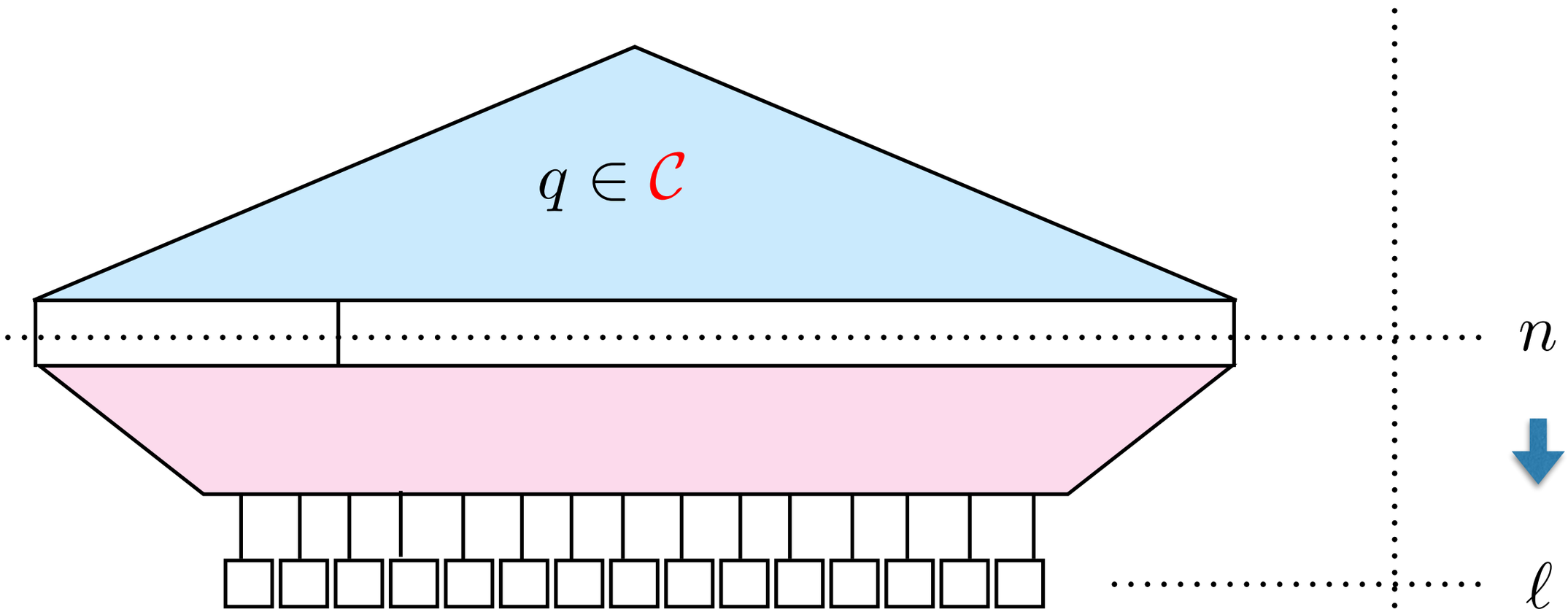


NW generator - reducing #variables



$$Q(\mathbf{y}) = q\big(f(\mathbf{y}|s_1), f(\mathbf{y}|s_2), \ldots, f(\mathbf{y}|s_n)\big)$$

NW generator - reducing #variables



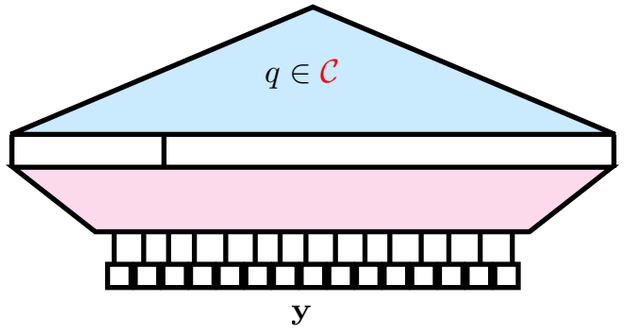
$\mathbf{y}$

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

Want: If $q \not\equiv 0$ then $Q \not\equiv 0$.

Key lemma

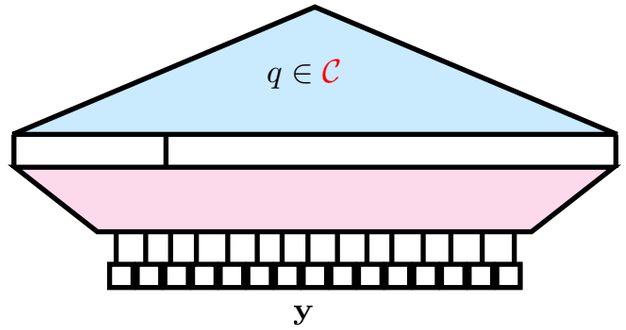
Key lemma



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

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

Key lemma

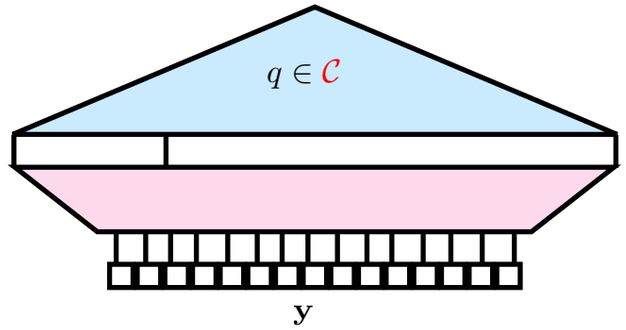


$$Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big)$$

Lemma: Let non-zero $q \in \text{Depth-}\Delta$ and a m-variate *multilinear* polynomial f of degree d. If

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

Key lemma



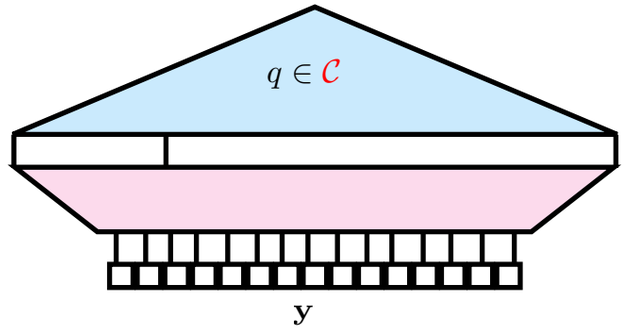
$$Q(\mathbf{y}) = q\big(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\big)$$

Lemma: Let non-zero $q \in \text{Depth-}\Delta$ and a m-variate *multilinear* polynomial f of degree d. If

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

Then, f can be computed by a size $\text{poly}(n, d^{\sqrt{d}})$ and depth $\Delta + 5$ circuit.

Key lemma



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

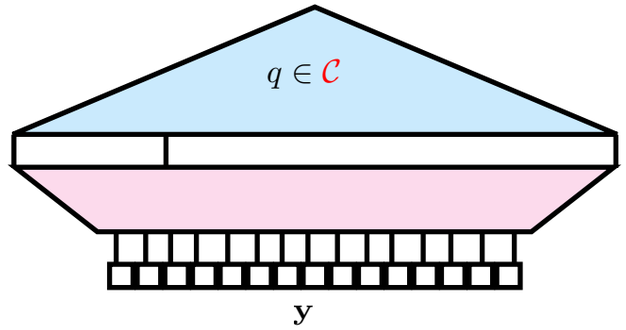
Lemma: Let non-zero $q \in \text{Depth-}\Delta$ and a m-variate *multilinear* polynomial f of degree d. If

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

Then, f can be computed by a size $\text{poly}(n, d^{\sqrt{d}})$ and depth $\Delta + 5$ circuit.

$$f \notin \text{Depth-}\Delta + 5 \quad \Rightarrow \quad Q(\mathbf{y}) \not\equiv 0, \ \forall q \in \text{Depth-}\Delta$$

Key lemma



Lemma: Let non-zero $q \in \text{Depth-}\Delta$ and a m-variate *multilinear* polynomial f of degree d. If

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

Then, f can be computed by a size $\text{poly}(n, d^{\sqrt{d}})$ and depth $\Delta + 5$ circuit.

$$f \notin \text{Depth-}\Delta + 5 \quad \Rightarrow \quad Q(\mathbf{y}) \not\equiv 0, \ \forall q \in \text{Depth-}\Delta$$

Schwartz-Zippel

Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

$$f \in \text{Depth-}\Delta + 5$$

Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

$$\Downarrow$$

$$f \in \text{Depth-}\Delta + 5$$

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

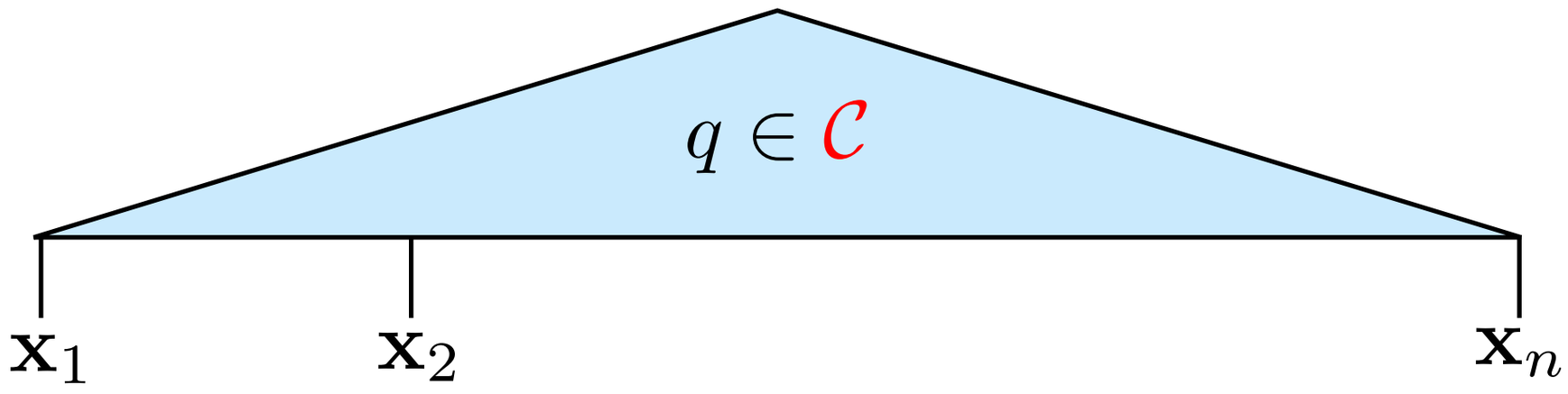
Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

$$\Downarrow$$

$$f \in \text{Depth-}\Delta + 5$$

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



$\not\equiv 0$

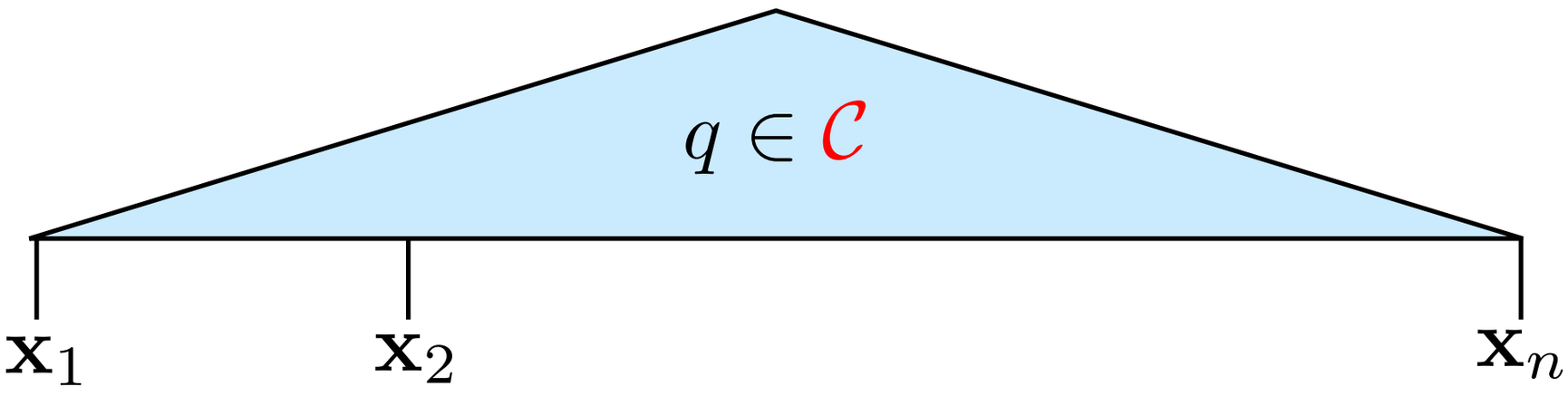
Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

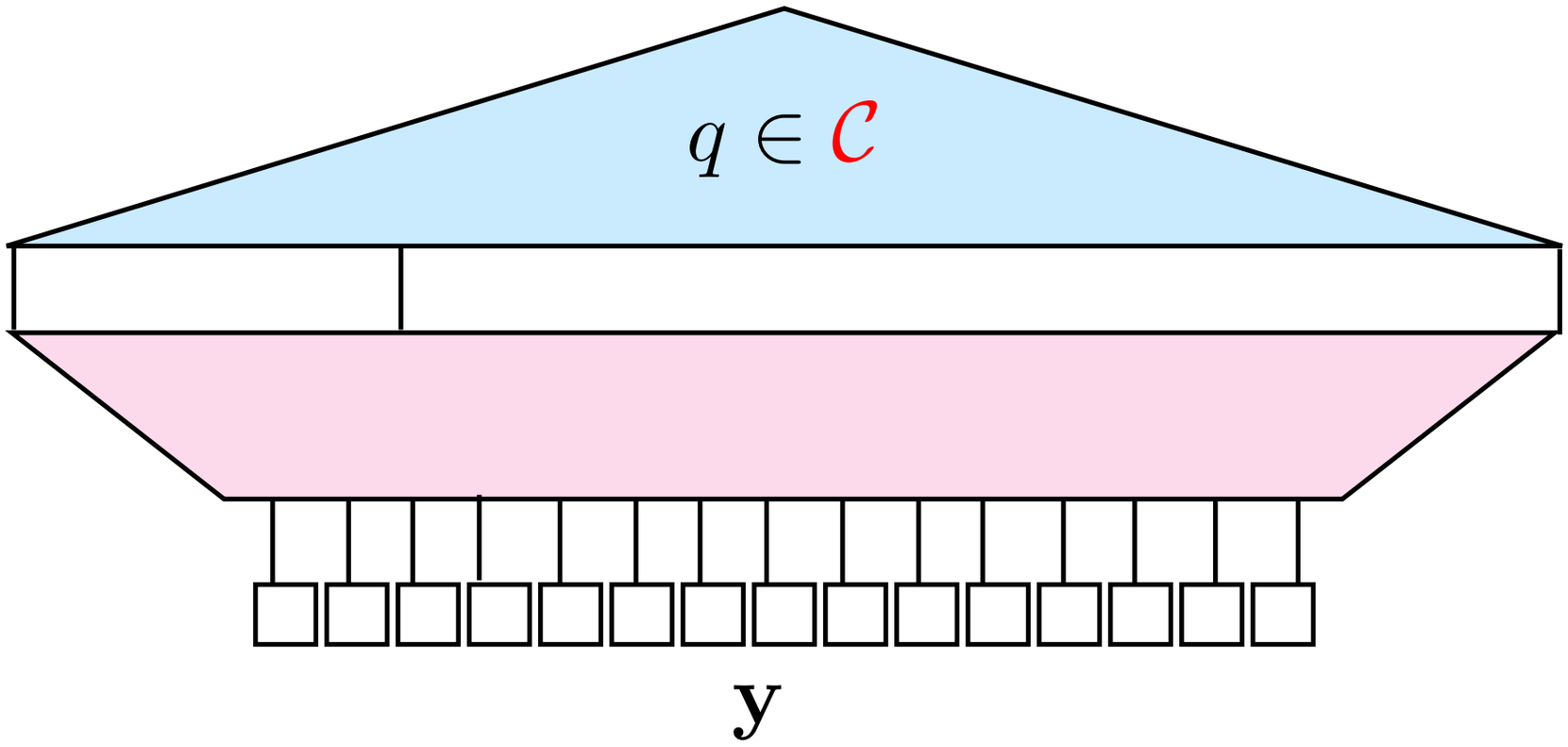
$$\Downarrow$$

$$f \in \text{Depth-}\Delta + 5$$

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



$\not\equiv 0$



$\equiv 0$

Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

$$f \in \text{Depth-}\Delta + 5$$

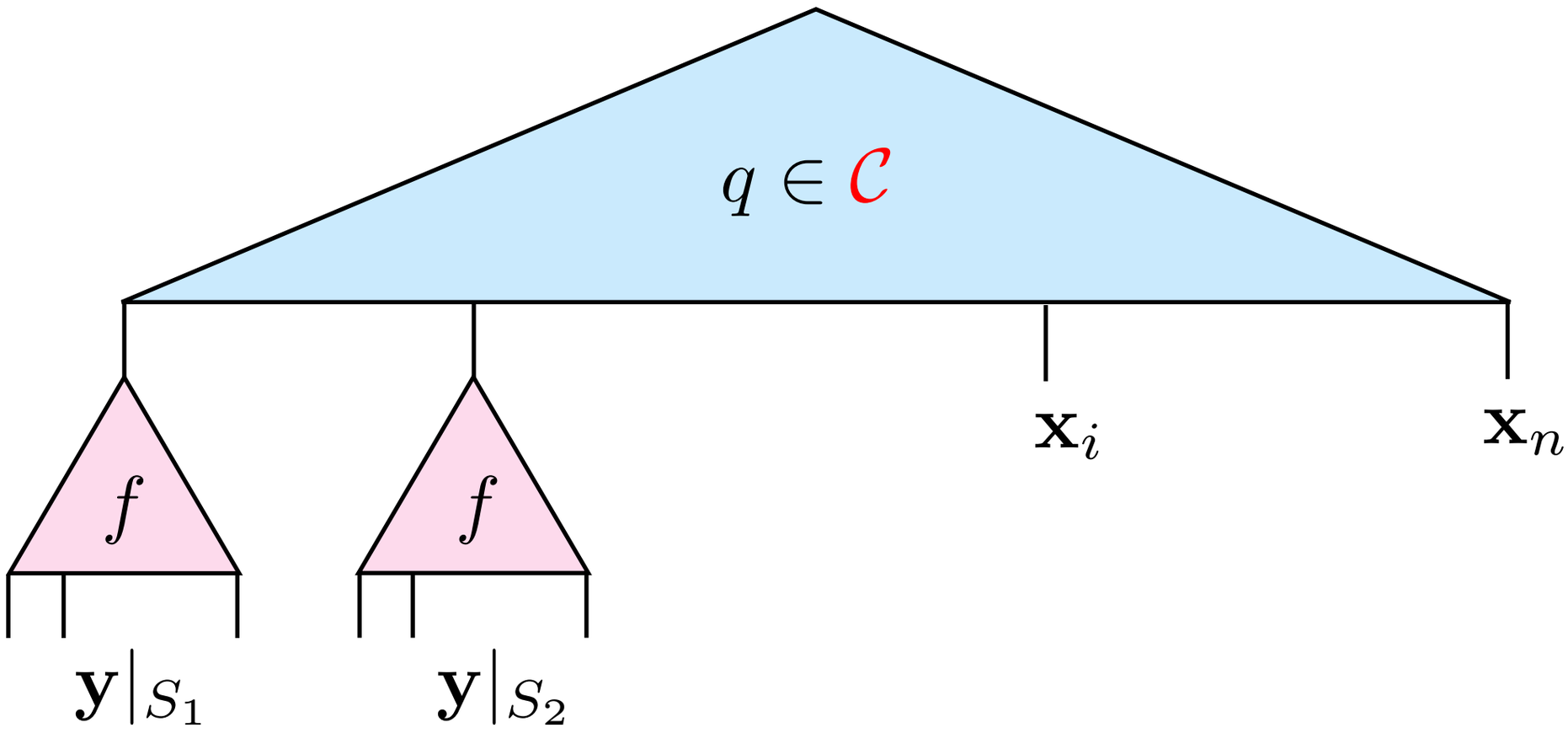
Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

$$\Downarrow$$

$$f \in \text{Depth-}\Delta + 5$$

By hybrid argument, there exists



Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

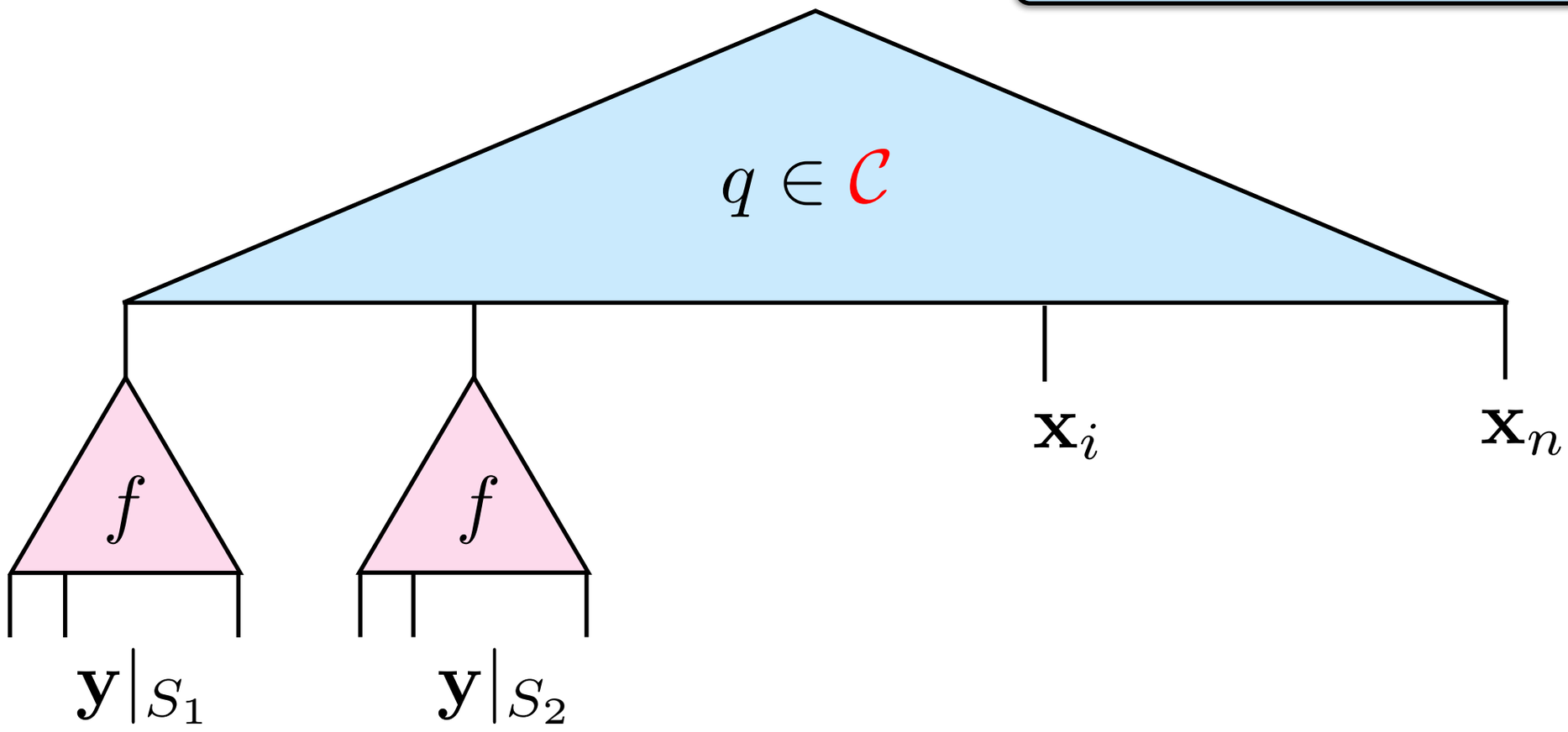
$$\Downarrow$$

$$f \in \text{Depth-}\Delta + 5$$

By hybrid argument, there exists

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

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



Proof sketch of the key lemma

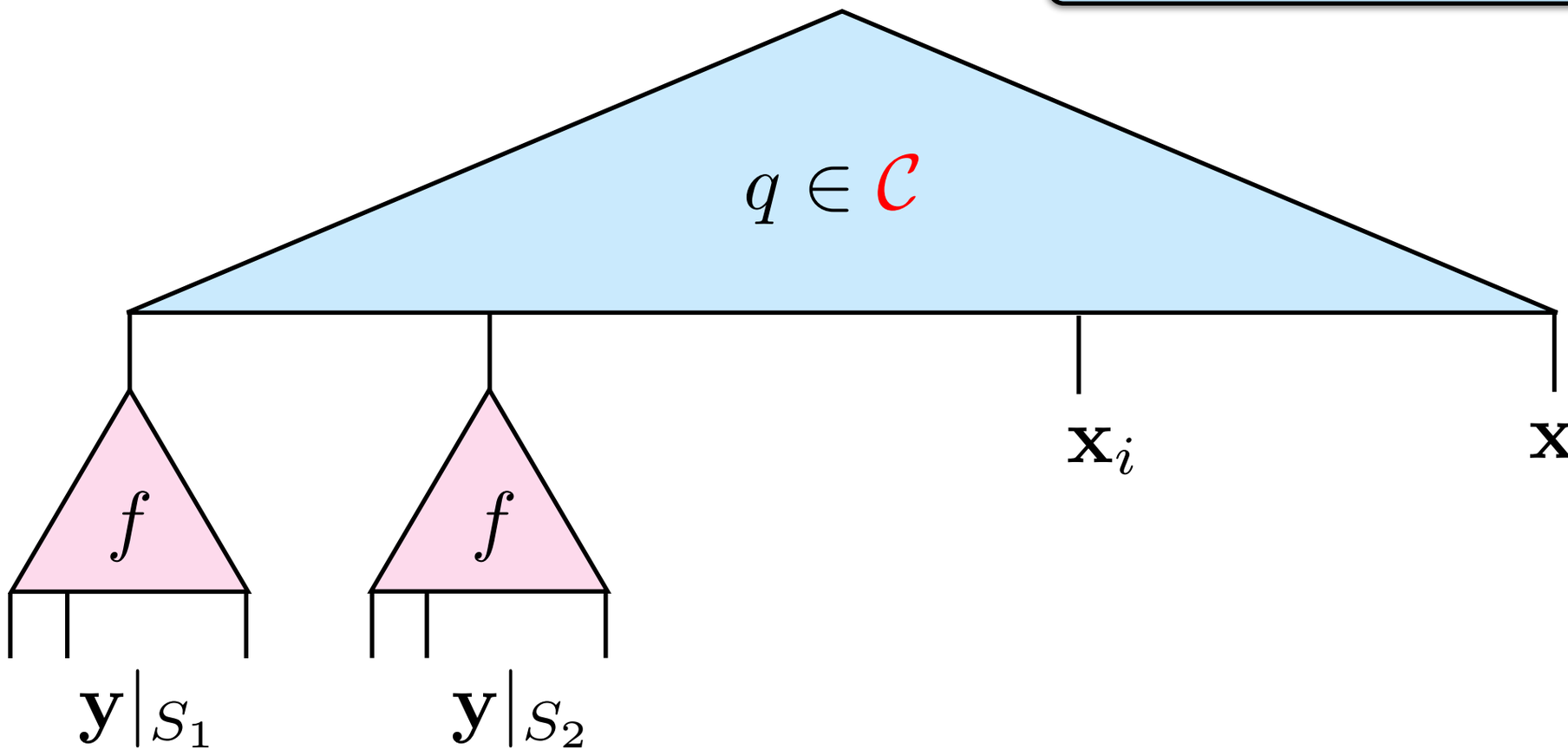
$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

$$\Downarrow$$

$$f \in \text{Depth-}\Delta + 5$$

By hybrid argument, there exists

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i)$ $\quad \mathbf{z} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n, \mathbf{y}\}$



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

Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

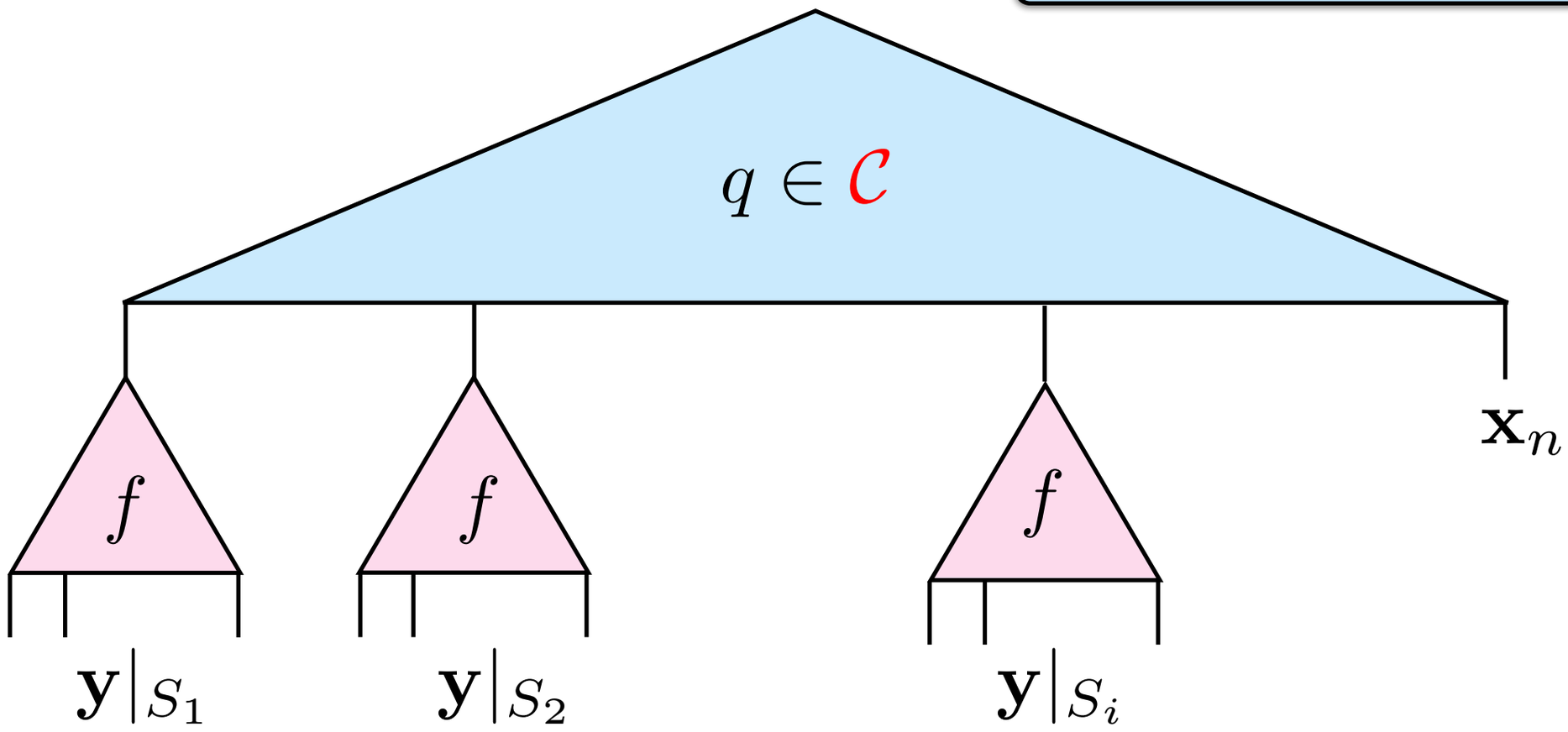
$$\Downarrow$$

$$f \in \text{Depth-}\Delta + 5$$

By hybrid argument, there exists

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

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



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

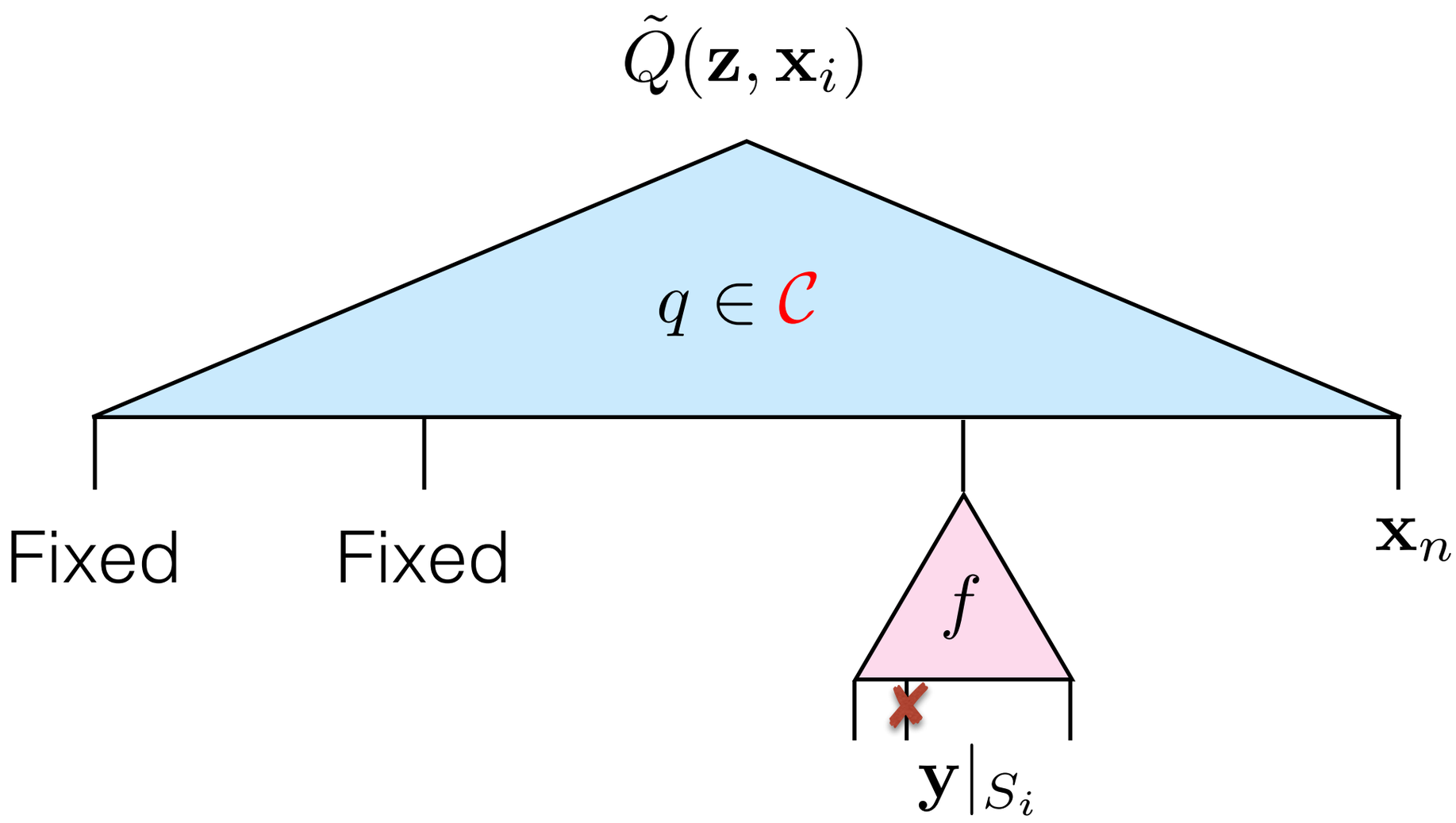
Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

$$\Downarrow$$

$$f \in \text{Depth-}\Delta + 5$$

By hybrid argument, there exists



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

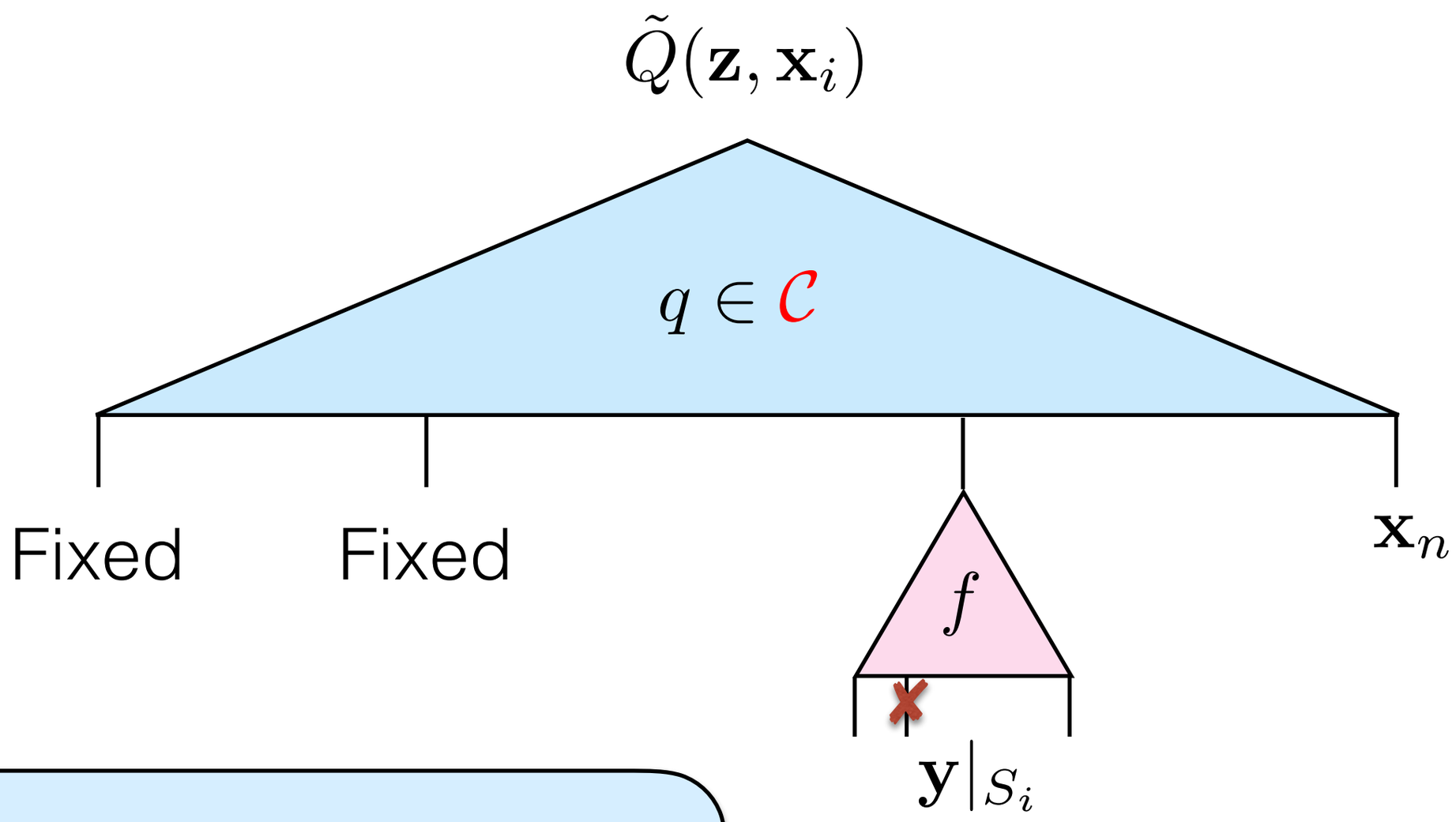
Proof sketch of the key lemma

$$\exists q \in \text{Depth-}\Delta,\ Q(\mathbf{y}) \equiv 0$$

$$\Downarrow$$

$$f \in \text{Depth-}\Delta + 5$$

By hybrid argument, there exists



$^*\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \in \text{Depth-}\Delta + 1$

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

Proof sketch of the key lemma

$$\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \in \text{Depth-}\Delta + 1$$

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

Proof sketch of the key lemma

$$\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \in \text{Depth-}\Delta + 1$$

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

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

Proof sketch of the key lemma

$\tilde{Q}(\mathbf{z}, \mathbf{x}_i) \in$ Depth-$\Delta + 1$

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

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

Reducing to **polynomial factorization!**

Outline

- Arithmetic circuits and algebraic complexity classes
- Polynomial identity testing (PIT)
- Hardness vs Randomness for arithmetic circuits
- Polynomial factorization
- Open problems

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 $f \in \mathcal{C}'$.

	$\mathcal{C}$	$\mathcal{C}'$
[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 $f \in \mathcal{C}'$.

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

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 $f \in \mathcal{C}'$.

	$\mathcal{C}$	$\mathcal{C}'$
[Kal89]	VP	VP
[DSY09]	$\text{Depth-}\Delta$ with bounded individual degree	$\text{Depth-}\Delta + 3$
[DSS18]	$\mathrm{VF}(n^{\log n}))$ (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 $f \in \mathcal{C}'$.

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

non-deterministic (existential)

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

Factorization for bounded depth circuits

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

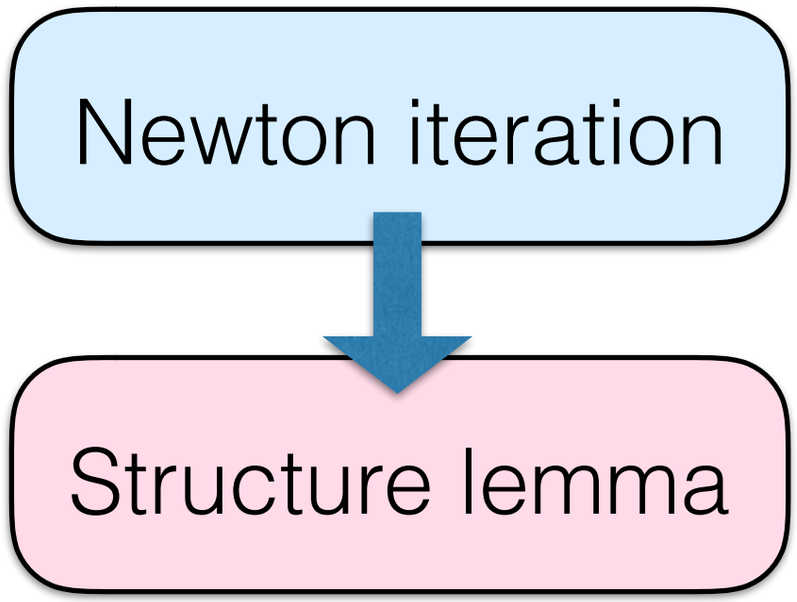
Factorization for bounded depth circuits

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

Newton iteration

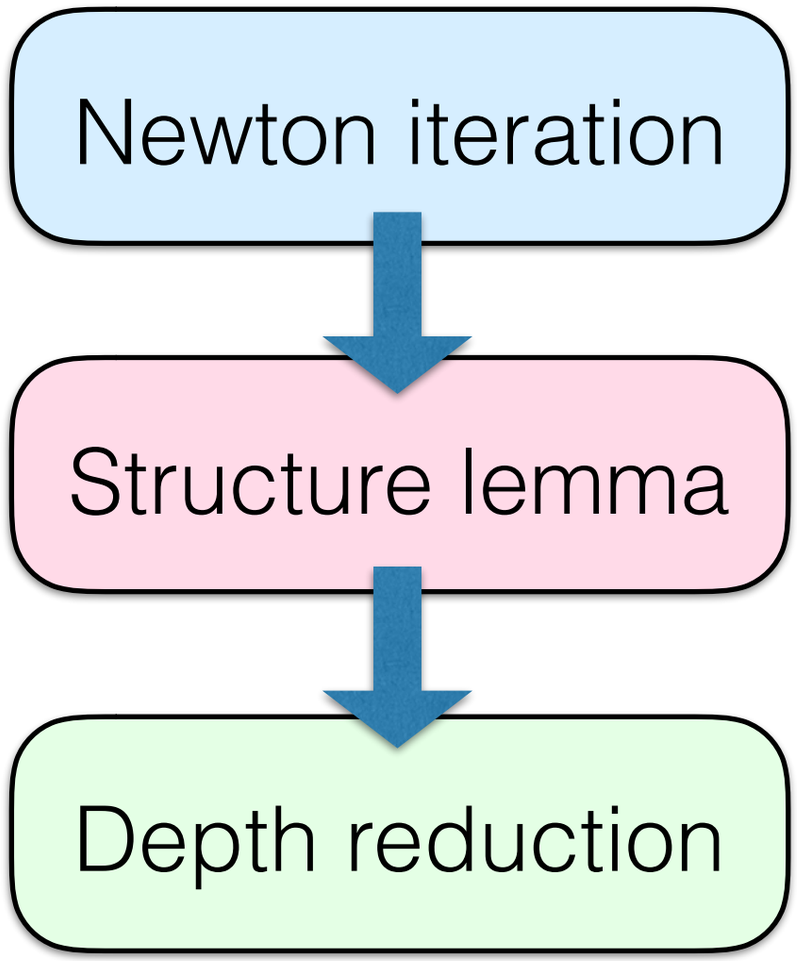
Factorization for bounded depth circuits

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.



Factorization for bounded depth circuits

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.



Newton iteration (Sloppy Hensel Lifting)

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

Newton iteration (Sloppy Hensel Lifting)

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

Def: (Homogeneous components)

Newton iteration (Sloppy Hensel Lifting)

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

Def: (Homogeneous components)
The degree **i** homogeneous component is the collection of monomials of degree **i**.

Newton iteration (Sloppy Hensel Lifting)

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

Def: (Homogeneous components)
The degree **i** homogeneous component is the collection of monomials of degree **i**.

Example: $f(x_1, x_2, x_3) = x_1^3 x_2 + x_1 x_2 x_3 + x_2^2 + x_1 x_3 + x_3^4$

Newton iteration (Sloppy Hensel Lifting)

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

Def: (Homogeneous components)
The degree **i** homogeneous component is the collection of monomials of degree **i**.

Example: $f(x_1, x_2, x_3) = x_1^3 x_2 + x_1 x_2 x_3 + x_2^2 + x_1 x_3 + x_3^4$

- $\mathcal{H}_0[f] = 0$

Newton iteration (Sloppy Hensel Lifting)

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

Def: (Homogeneous components)
The degree **i** homogeneous component is the collection of monomials of degree **i**.

Example: $f(x_1, x_2, x_3) = x_1^3 x_2 + x_1 x_2 x_3 + x_2^2 + x_1 x_3 + x_3^4$

- $\mathcal{H}_0[f] = 0$
- $\mathcal{H}_1[f] = 0$

Newton iteration (Sloppy Hensel Lifting)

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

Def: (Homogeneous components)
The degree **i** homogeneous component is the collection of monomials of degree **i**.

Example: $f(x_1, x_2, x_3) = x_1^3 x_2 + x_1 x_2 x_3 + x_2^2 + x_1 x_3 + x_3^4$

- $\mathcal{H}_0[f] = 0$
- $\mathcal{H}_1[f] = 0$
- $\mathcal{H}_2[f] = x_2^2 + x_1 x_3$

Newton iteration (Sloppy Hensel Lifting)

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

Def: (Homogeneous components)
The degree **i** homogeneous component is the collection of monomials of degree **i**.

Example: $f(x_1, x_2, x_3) = x_1^3 x_2 + x_1 x_2 x_3 + x_2^2 + x_1 x_3 + x_3^4$

- $\mathcal{H}_0[f] = 0$
- $\mathcal{H}_1[f] = 0$
- $\mathcal{H}_2[f] = x_2^2 + x_1 x_3$
- $\mathcal{H}_3[f] = x_1 x_2 x_3$

Newton iteration (Sloppy Hensel Lifting)

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

Def: (Homogeneous components)
The degree **i** homogeneous component is the collection of monomials of degree **i**.

Example: $f(x_1, x_2, x_3) = x_1^3 x_2 + x_1 x_2 x_3 + x_2^2 + x_1 x_3 + x_3^4$

- $\mathcal{H}_0[f] = 0$
- $\mathcal{H}_1[f] = 0$
- $\mathcal{H}_2[f] = x_2^2 + x_1 x_3$
- $\mathcal{H}_3[f] = x_1 x_2 x_3$
- $\mathcal{H}_4[f] = x_1^3 x_2 + x_3^4$

Newton iteration (Sloppy Hensel Lifting)

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

Newton iteration (Sloppy Hensel Lifting)

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

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

Newton iteration (Sloppy Hensel Lifting)

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

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

* Homogenization & partial derivative preserve depth

Newton iteration (Sloppy Hensel Lifting)

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

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

Intuition: Taylor's expansion.

* Homogenization & partial derivative preserve depth

Newton iteration (Sloppy Hensel Lifting)

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

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

Intuition: Taylor's expansion.

Q: How to efficiently update?

* Homogenization & partial derivative preserve depth

Structure lemma

Structure lemma

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

Structure lemma

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

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

Structure lemma

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

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

Lemma [DSY'09]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i such that

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

Structure lemma

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

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

Lemma [DSY'09]: For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i such that

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

Individual degree

Structure lemma

Structure lemma

Lemma (This work): For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i such that

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

Structure lemma

Lemma (This work): For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i such that

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

where

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

Structure lemma

Lemma (This work): For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i such that

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

where

size $O(d^6)$ degree at most d

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

Structure lemma

Lemma (This work): For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i such that

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

size $O(d^6)$ degree at most d

where

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

Depth $\Delta + 1$ with top Σ layer

Structure lemma

Lemma (This work): For each $i = 1, 2, \ldots, d = \deg(f)$, there exists polynomial A_i such that

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

size $O(d^6)$ degree at most d

where

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

Depth $\Delta + 1$ with top Σ layer

* Homogenization & partial derivative preserve depth

Structure lemma

Structure lemma

$$P(\mathbf{z}, y) \in \text{Depth-}\Delta$$

$$P(\mathbf{z}, f(\mathbf{z})) = 0$$

Structure lemma

$$P(\mathbf{z}, y) \in \text{Depth-}\Delta$$

$$P(\mathbf{z}, f(\mathbf{z})) = 0$$

$$f = h_d$$

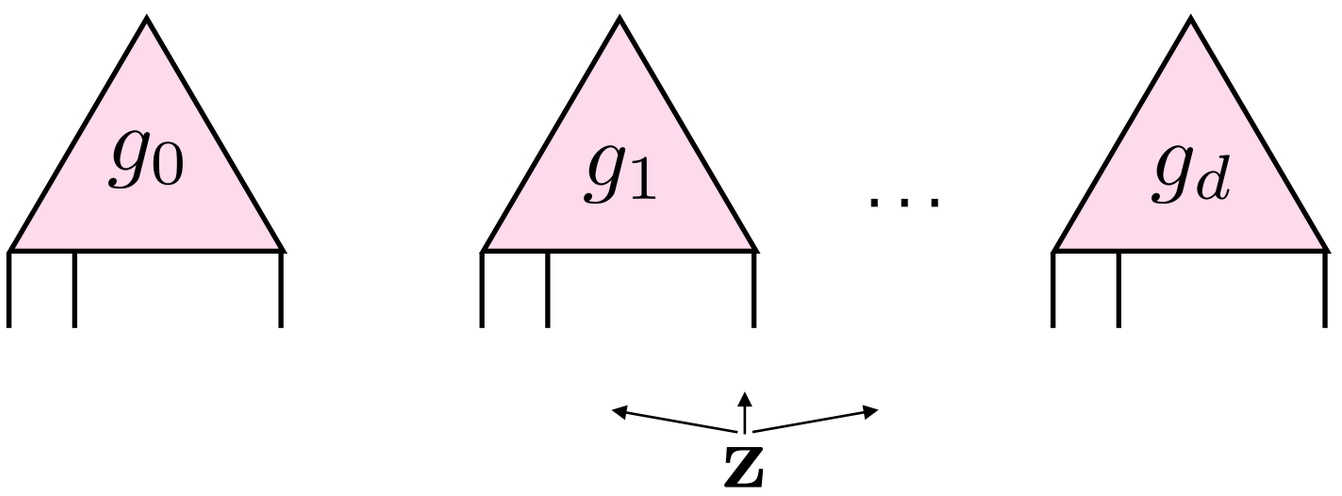
$$\mathbf{z}$$

Structure lemma

$$P(\mathbf{z}, y) \in \text{Depth-}\Delta$$

$$P(\mathbf{z}, f(\mathbf{z})) = 0$$

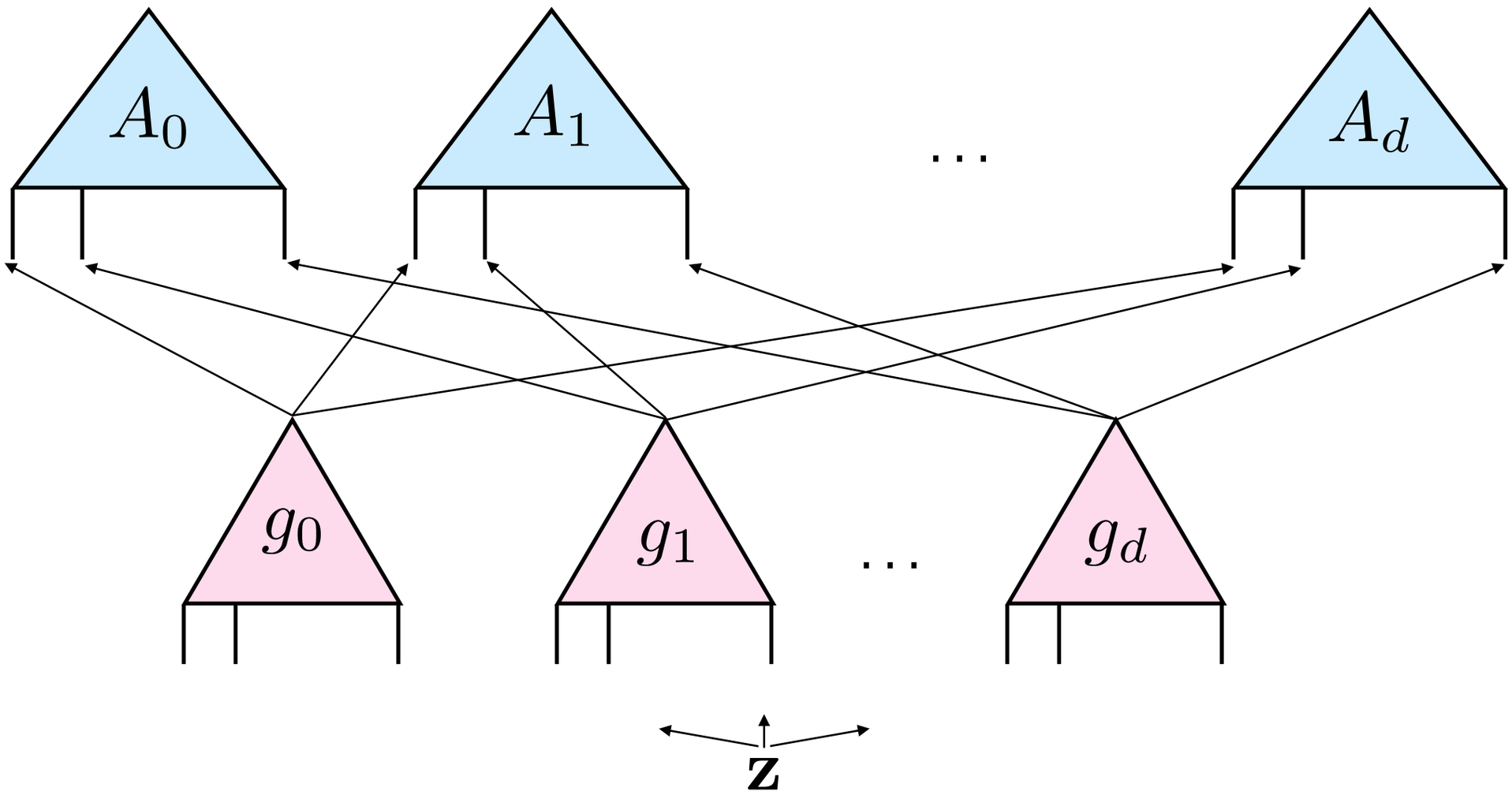
$$f = h_d$$



Structure lemma

$$P(\mathbf{z}, y) \in \text{Depth-}\Delta$$
$$P(\mathbf{z}, f(\mathbf{z})) = 0$$

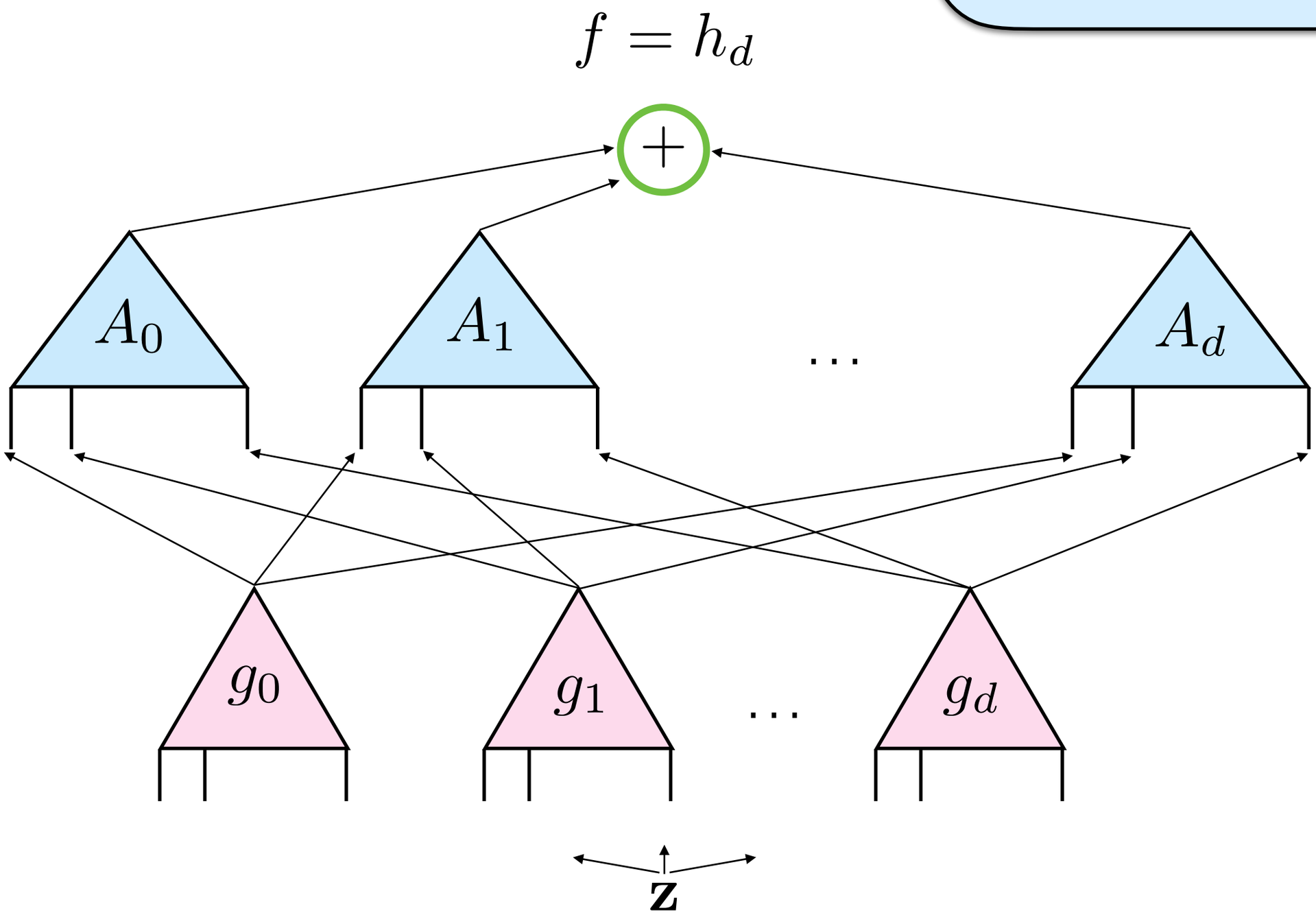
$$f = h_d$$



Structure lemma

$$P(\mathbf{z}, y) \in \text{Depth-}\Delta$$

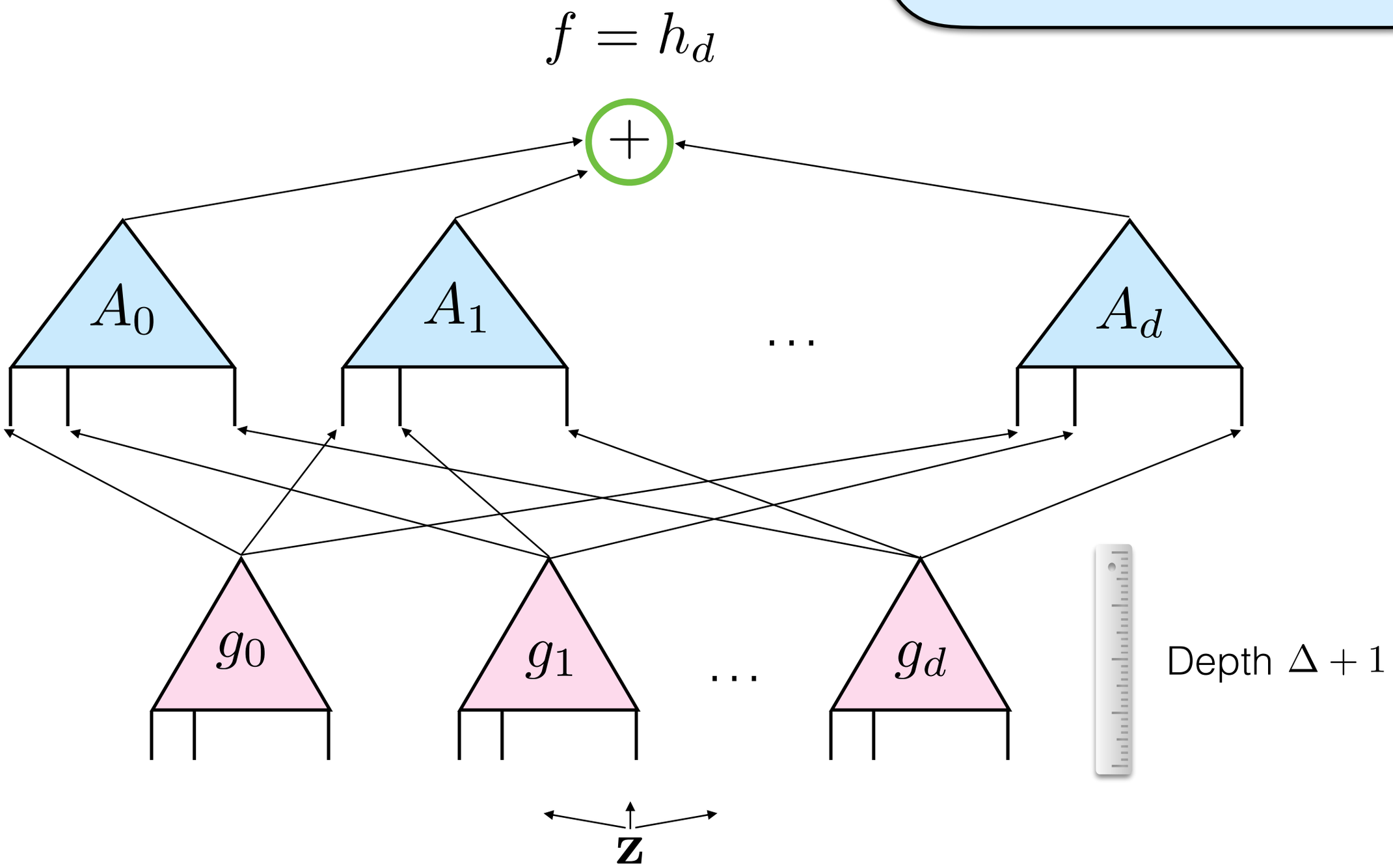
$$P(\mathbf{z}, f(\mathbf{z})) = 0$$



Structure lemma

$$P(\mathbf{z}, y) \in \text{Depth-}\Delta$$

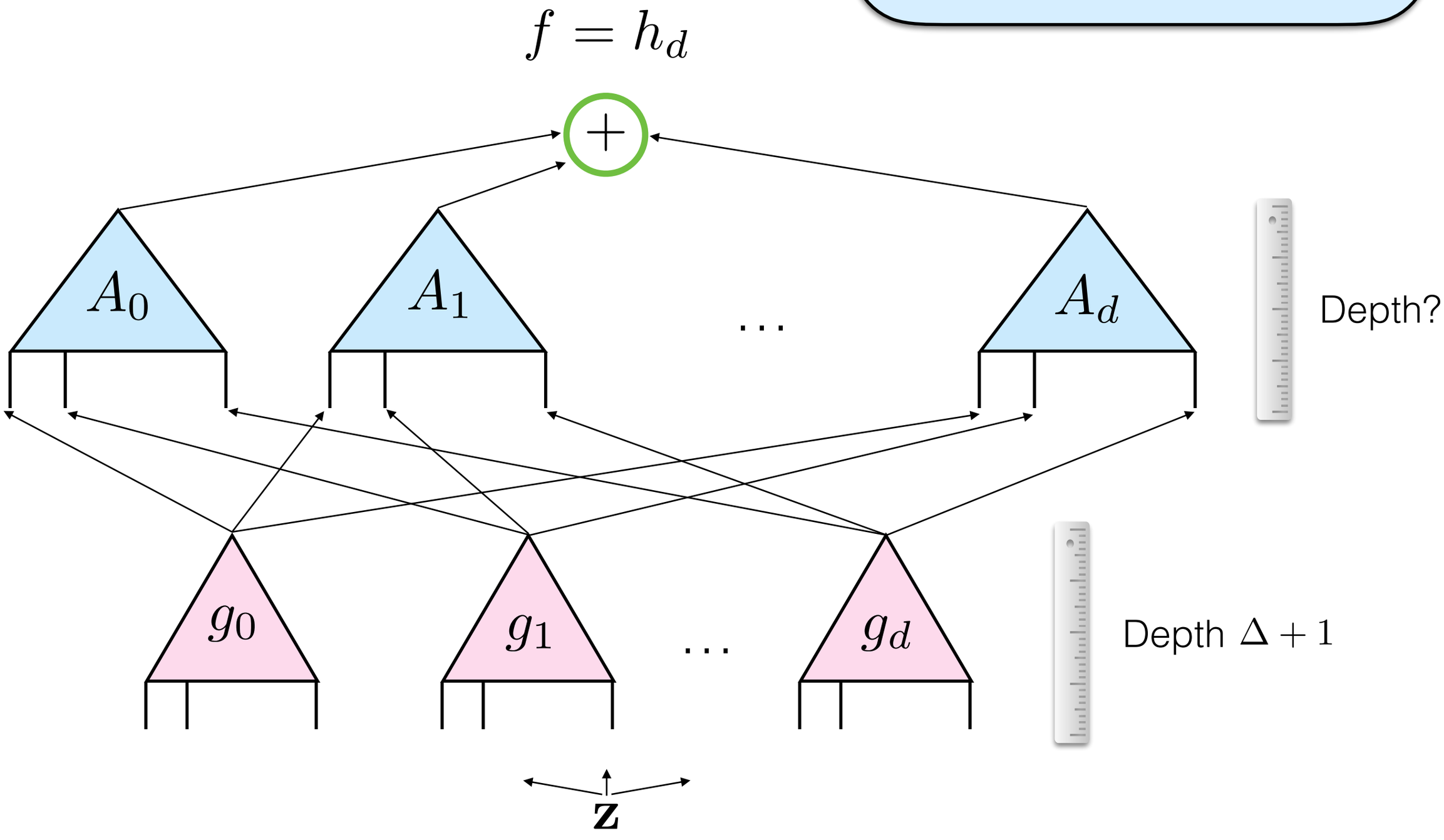
$$P(\mathbf{z}, f(\mathbf{z})) = 0$$



Structure lemma

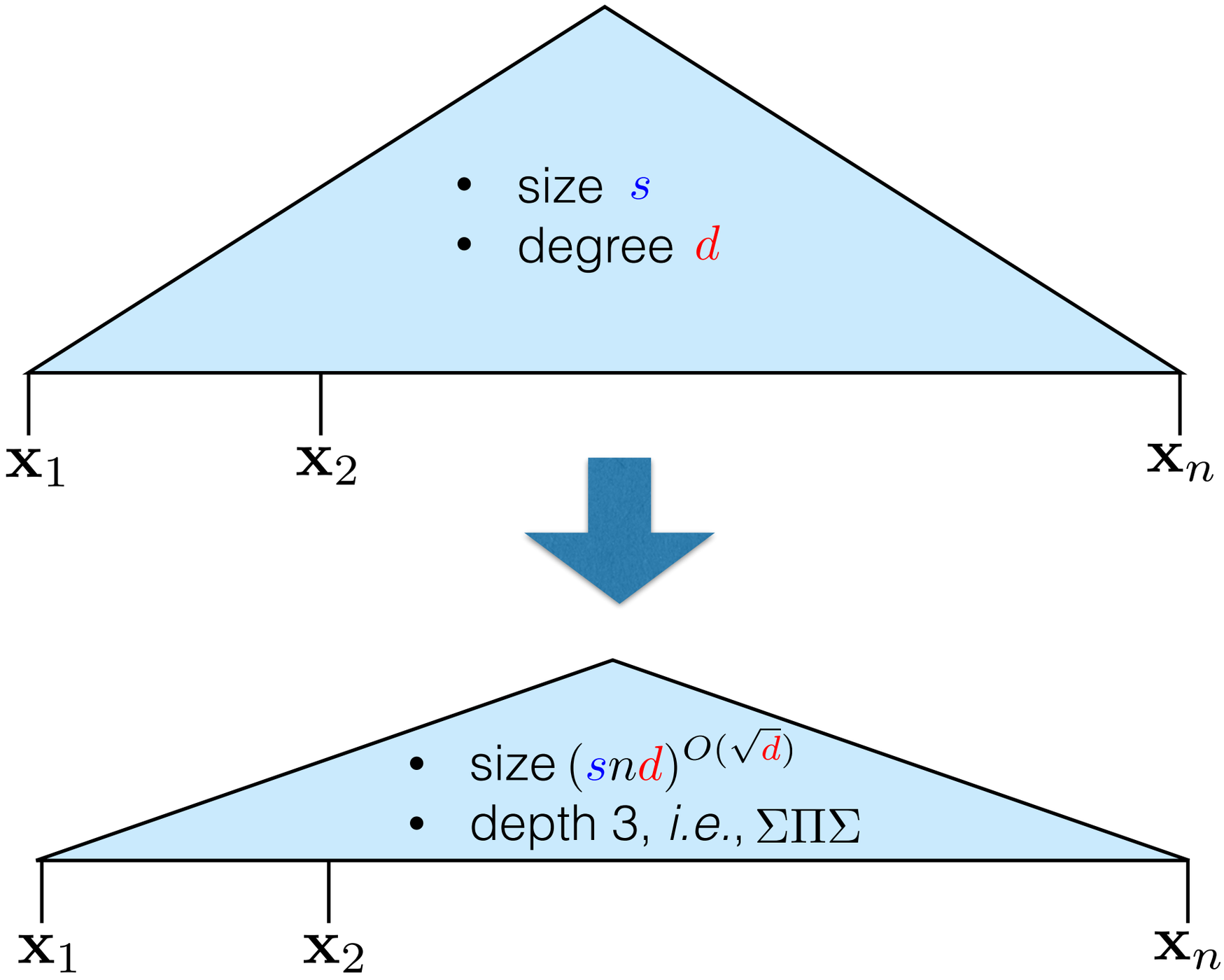
$$P(\mathbf{z}, y) \in \text{Depth-}\Delta$$

$$P(\mathbf{z}, f(\mathbf{z})) = 0$$

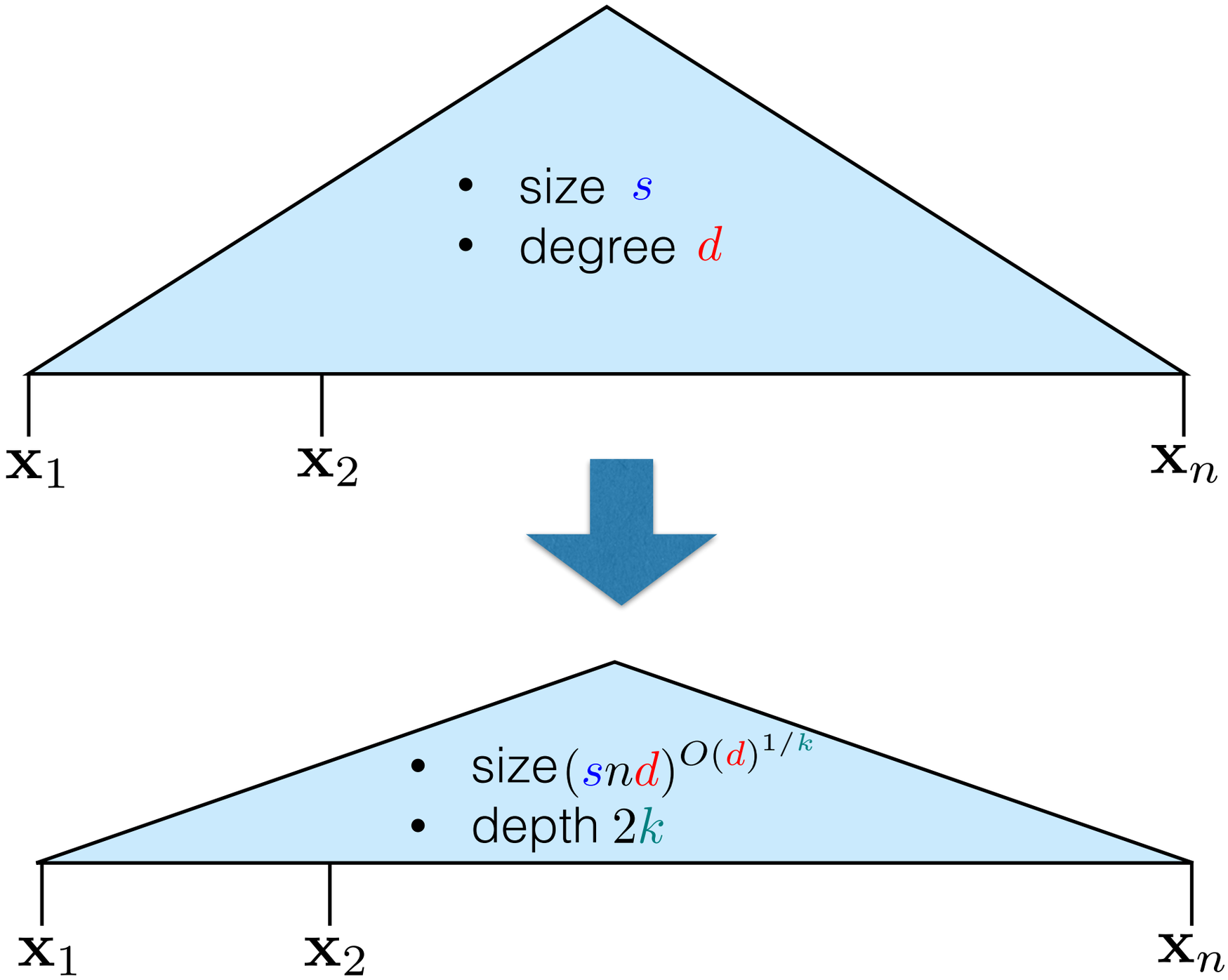


Depth reduction [Gupta-Kamath-Kayal-Saptharishi'13]

Depth reduction [Gupta-Kamath-Kayal-Saptharishi'13]

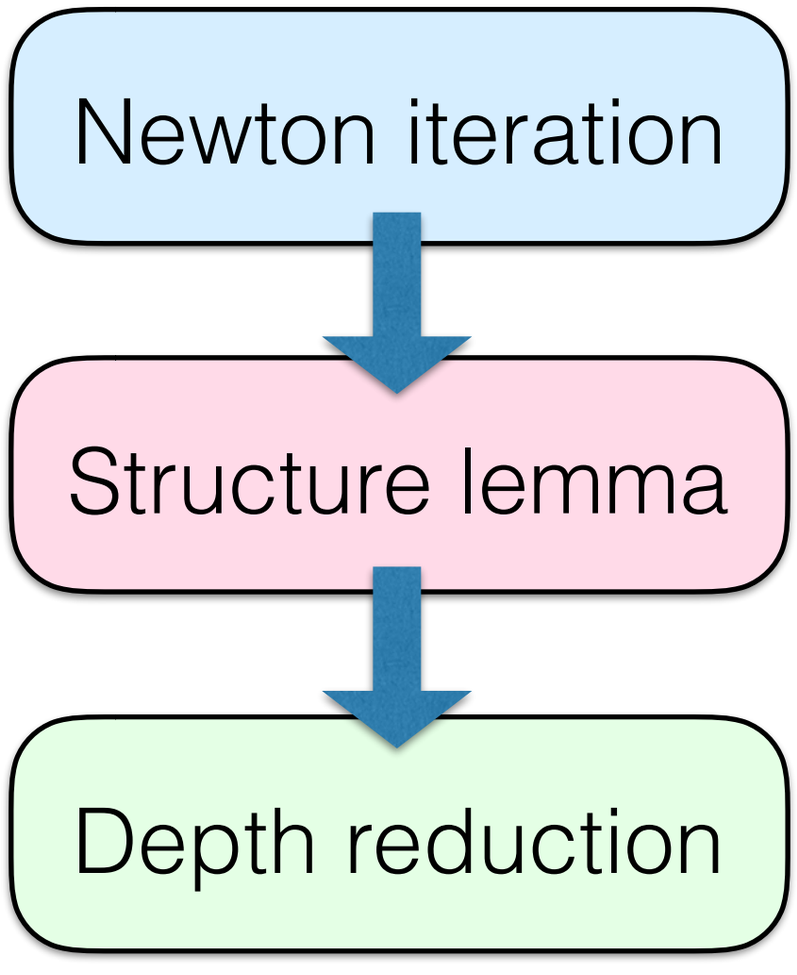


Depth reduction
[Agrawal-Vinay'08, Koiran'12, Tavenas'13]



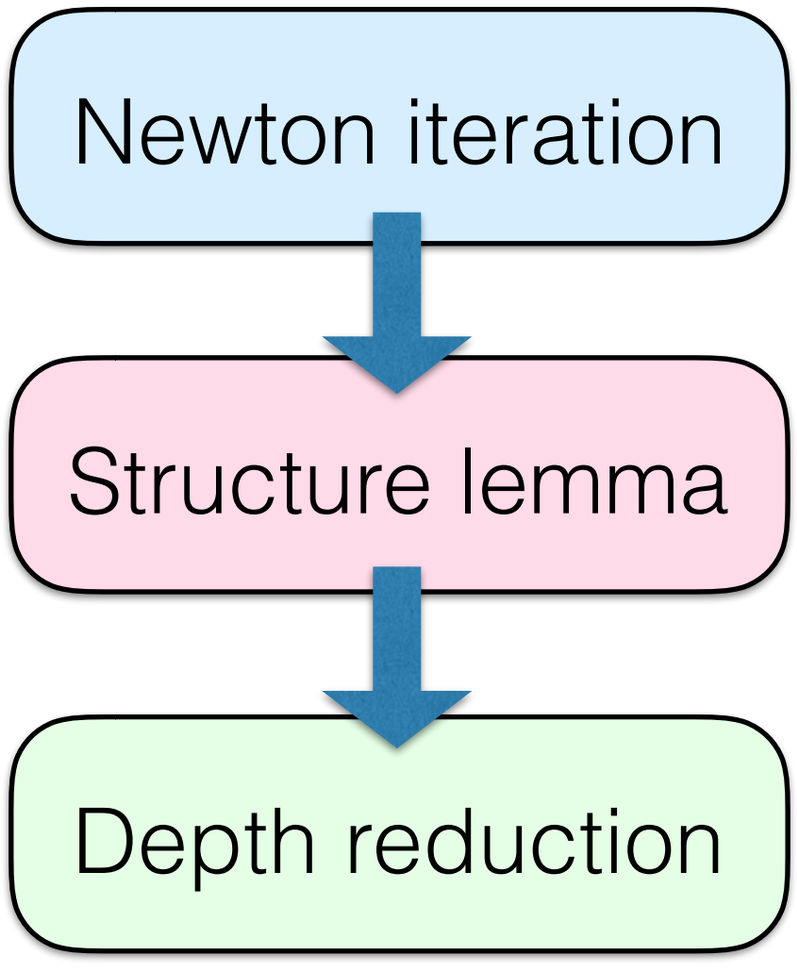
Factorization for bounded depth circuits (Wrap up)

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.



Factorization for bounded depth circuits (Wrap up)

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.



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

Factorization for bounded depth circuits (Wrap up)

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

Newton iteration

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

Structure lemma

$$h_i - h_{i-1} = A_i(g_0, g_1, \ldots, g_d)$$

Depth reduction

Factorization for bounded depth circuits (Wrap up)

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

Newton iteration

Structure lemma

Depth reduction

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

size $O(d^6)$ degree at most d

$$h_i - h_{i-1} = A_i(g_0, g_1, \ldots, g_d)$$

Depth $\Delta + 1$ with top Σ layer

Factorization for bounded depth circuits (Wrap up)

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

Newton iteration

Structure lemma

Depth reduction

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

size $O(d^6)$ degree at most d

$$h_i - h_{i-1} = A_i(g_0, g_1, \ldots, g_d)$$

Depth $\Delta + 1$ with top Σ layer

$$h_i - h_{i-1} = \tilde{A}_i(g_0, g_1, \ldots, g_d)$$

Factorization for bounded depth circuits (Wrap up)

Goal: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
Show that $f \in \text{Depth-}\Delta + O(1)$.

Newton iteration

Structure lemma

Depth reduction

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

size $O(d^6)$ degree at most d

$$h_i - h_{i-1} = A_i(g_0, g_1, \ldots, g_d)$$

Depth $\Delta + 1$ with top Σ layer

$$h_i - h_{i-1} = \tilde{A}_i(g_0, g_1, \ldots, g_d)$$

Depth $\Delta + 3$ size $\text{poly}(n, d^{\sqrt{d}})$ with top Σ layer

Conclusion

Conclusion

Theorem: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$.
If $d^{\sqrt{d}} = \text{poly}(n)$, then $f \in \text{Depth-}\Delta + 3$.

Conclusion

Theorem: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$. If $d^{\sqrt{d}} = \text{poly}(n)$, then $f \in \text{Depth-}\Delta + 3$.

Theorem: For any $\Delta \geq 6$. If there's a $\omega(\text{poly}(n))$ lower bound for $\text{Depth-}\Delta$ with degree $O(\log^2 n / \log^2 \log n)$, then there's a sub-exponential time PIT for $\text{Depth-}\Delta - 5$.

Conclusion

Theorem: For any $P(\mathbf{z}, y) \in \text{Depth-}\Delta$ s.t. $P(\mathbf{z}, f(\mathbf{z})) = 0$. If $d^{\sqrt{d}} = \text{poly}(n)$, then $f \in \text{Depth-}\Delta + 3$.

Theorem: For any $\Delta \geq 6$. If there's a $\omega(\text{poly}(n))$ lower bound for $\text{Depth-}\Delta$ with degree $O(\log^2 n / \log^2 \log n)$, then there's a sub-exponential time PIT for $\text{Depth-}\Delta - 5$.

	[DSY'09]	This work
Lower bound for $\text{Depth-}\Delta$		With degree $O(\log^2 n / \log^2 \log n)$
PIT for $\text{Depth-}\Delta - 5$	With bounded individual degree	

Outline

- Arithmetic circuits and algebraic complexity classes
- Polynomial identity testing (PIT)
- Hardness vs Randomness for arithmetic circuits
- Polynomial factorization
- Open problems

Open problems

Open problems

- **Hardness vs Randomness**

Open problems

- **Hardness vs Randomness**
  - ✦ Remove the degree condition(s)?

Open problems

- **Hardness vs Randomness**
  - ✦ Remove the degree condition(s)?
  - ✦ More circuit classes?

Open problems

- **Hardness vs Randomness**
  - ✦ Remove the degree condition(s)?
  - ✦ More circuit classes?
- **Polynomial factorization**

Open problems

- **Hardness vs Randomness**
  - ✦ Remove the degree condition(s)?
  - ✦ More circuit classes?
- **Polynomial factorization**
  - ✦ Remove the degree condition(s)?

Open problems

- **Hardness vs Randomness**
  - ✦ Remove the degree condition(s)?
  - ✦ More circuit classes?
- **Polynomial factorization**
  - ✦ Remove the degree condition(s)?
  - ✦ Sparse polynomials?

Open problems

- **Hardness vs Randomness**
  - ✦ Remove the degree condition(s)?
  - ✦ More circuit classes?
- **Polynomial factorization**
  - ✦ Remove the degree condition(s)?
  - ✦ Sparse polynomials?
  - ✦ Closure results for VF, VBP?

Open problems

- **Hardness vs Randomness**
  - ✦ Remove the degree condition(s)?
  - ✦ More circuit classes?
- **Polynomial factorization**
  - ✦ Remove the degree condition(s)?
  - ✦ Sparse polynomials?
  - ✦ Closure results for VF, VBP?

Thank you!