



Computational Complexity Theory


Lecture 10: Parity not in AC^0

Department of Computer Science,
Indian Institute of Science

Recap: Class NC

- **NC** stands for Nick's Class – named after Nick Pippenger.
- **Definition.** For $i \in \mathbb{N}$, a language L is in NC^i if there is a polynomial function $q(\cdot)$ and a constant c s.t. L is decided by a $q(n)$ -size circuit family $\{C_n\}_{n \in \mathbb{N}}$, where depth of C_n is at most $c \cdot (\log n)^i$ for every $n \in \mathbb{N}$.
- **Definition.** $NC = \bigcup_{i \in \mathbb{N}} NC^i$.
- **PARITY** is in $NC^1 = \text{poly}(n)$ -size Boolean formulas.

Recap: Class AC

- **Definition.** For $i \in \mathbb{N} \cup \{0\}$, a language L is in AC^i if there is a polynomial function $q(\cdot)$ and a constant c s.t. L is decided by a $q(n)$ -size unbounded fan-in circuit family $\{C_n\}_{n \in \mathbb{N}}$, where depth of C_n is at most $c \cdot (\log n)^i$ for every $n \in \mathbb{N}$.
- **Definition.** $AC = \bigcup_{i \geq 0} AC^i$. (stands for *Alternating Class*)
- **Observation.** $AC^i \subseteq NC^{i+1} \subseteq AC^{i+1}$ for all $i \geq 0$.


Replace an unbounded fan-in gate by a binary tree of bounded fan-in gates.


Recap: Class AC

- **Definition.** For $i \in \mathbb{N} \cup \{0\}$, a language L is in AC^i if there is a polynomial function $q(\cdot)$ and a constant c s.t. L is decided by a $q(n)$ -size unbounded fan-in circuit family $\{C_n\}_{n \in \mathbb{N}}$, where depth of C_n is at most $c \cdot (\log n)^i$ for every $n \in \mathbb{N}$.
- **Definition.** $AC = \bigcup_{i \geq 0} AC^i$.
- In this lecture, we'll show that **PARITY** is not in AC^0 , i.e., $AC^0 \subsetneq NC^1$.

Recap: The Parity function

- $\text{PARITY}(x_1, x_2, \dots, x_n) = x_1 \oplus x_2 \oplus \dots \oplus x_n$.
- **Fact.** $\text{PARITY}(x_1, x_2, \dots, x_n)$ can be computed by a circuit of size $O(n)$ and a formula of size $O(n^2)$.
- **Theorem.** (*Khrapchenko 1971*) Any formula computing $\text{PARITY}(x_1, x_2, \dots, x_n)$ has size $\Omega(n^2)$.

Recap: The Parity function

- $\text{PARITY}(x_1, x_2, \dots, x_n) = x_1 \oplus x_2 \oplus \dots \oplus x_n$.
- **Fact.** $\text{PARITY}(x_1, x_2, \dots, x_n)$ can be computed by a circuit of size $O(n)$ and a formula of size $O(n^2)$.


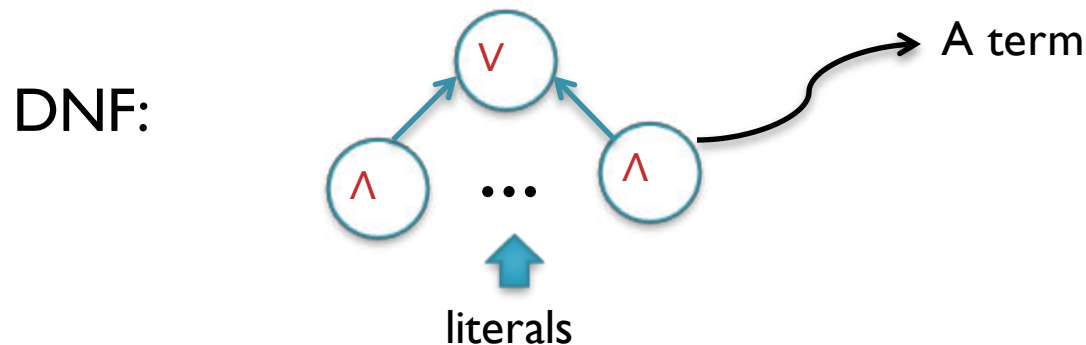
has depth $O(\log n)$ has depth $O(\log n)$
- **Theorem.** (*Khrapchenko 1971*) Any formula computing $\text{PARITY}(x_1, x_2, \dots, x_n)$ has size $\Omega(n^2)$.

Recap: The Parity function

- $\text{PARITY}(x_1, x_2, \dots, x_n) = x_1 \oplus x_2 \oplus \dots \oplus x_n$.
- **Fact.** $\text{PARITY}(x_1, x_2, \dots, x_n)$ can be computed by a circuit of size $O(n)$ and a formula of size $O(n^2)$.
- **Theorem.** (*Khrapchenko 1971*) Any formula computing $\text{PARITY}(x_1, x_2, \dots, x_n)$ has size $\Omega(n^2)$.
- Can poly-size constant depth circuits compute **PARITY?** **No!**

Depth 2 circuit for Parity

- Without loss of generality, a depth 2 circuit is either a DNF or a CNF.



- Any Boolean function can be computed by a DNF (similarly, CNF) with 2^n terms (respectively, clauses).
- Can we do better for depth 2 circuits computing **PARITY**?

Depth 2 circuit for Parity

- Without loss of generality, a depth 2 circuit is either a DNF or a CNF.
- **Obs.** Any DNF computing **PARITY** has $\geq 2^{n-1}$ terms.
- **Proof.** Let ϕ be a DNF computing **PARITY**. Then, every term in ϕ has n literals (otherwise, the value of **PARITY** can be fixed by fixing less than n variables which is false).

Depth 2 circuit for Parity

- Without loss of generality, a depth 2 circuit is either a DNF or a CNF.
- **Obs.** Any DNF computing **PARITY** has $\geq 2^{n-1}$ terms.
- **Proof.** Let ϕ be a DNF computing **PARITY**. Then, every term in ϕ has n literals (otherwise, the value of **PARITY** can be fixed by fixing less than n variables which is false). Such a term corresponds to a *unique* assignment that makes the term evaluate to 1. Terms corresponding to assignments that set odd number of variables to 1 must be present in ϕ .

Depth 3 circuit for Parity

- **Obs.** There's a $2^{O(\sqrt{n})}$ size depth 3 circuit for **PARITY**.

- **Proof.**

$$\text{PARITY} = \underbrace{x_1 \oplus x_2 \oplus \dots \oplus x_{\sqrt{n}}}_{y_1} \oplus \dots \oplus \underbrace{x_{n-\sqrt{n}+1} \oplus x_{n-\sqrt{n}+2} \oplus \dots \oplus x_n}_{y_{\sqrt{n}}}$$

- Divide & conquer: Compute y_i and $\neg y_i$ by $2^{O(\sqrt{n})}$ size DNFs on the x literals. Compute $y_1 \oplus \dots \oplus y_{\sqrt{n}}$ by a $2^{O(\sqrt{n})}$ size CNF on the y literals. “Attach” the CNF with the DNFs and “merge” the two middle layers of \vee gates.

Depth 3 circuit for Parity

- **Obs.** There's a $2^{O(\sqrt{n})}$ size depth 3 circuit for **PARITY**.

- **Proof.**

$$\begin{array}{ccccccc}
 x_1 \oplus x_2 \oplus \dots \oplus x_{\sqrt{n}} \oplus \dots \oplus x_{n-\sqrt{n}} \oplus x_2 \oplus \dots \oplus x_n \\
 \underbrace{\hspace{10em}} & & \underbrace{\hspace{10em}} \\
 \text{PARITY} = & y_1 & \oplus & \dots & \oplus & y_{\sqrt{n}}
 \end{array}$$

- Divide & conquer: Compute y_i and $\neg y_i$ by $2^{O(\sqrt{n})}$ size DNFs on the x literals. Compute $y_1 \oplus \dots \oplus y_{\sqrt{n}}$ by a $2^{O(\sqrt{n})}$ size CNF on the y literals. “Attach” the CNF with the DNFs and “merge” the two middle layers of \vee gates.

Is the $2^{O(\sqrt{n})}$ upper bound on the size of depth 3 circuits computing **PARITY** tight? “Yes”

Depth d circuit for Parity

- **Obs.** There's a $\exp(n^{1/(d-1)})$ size depth d circuit for **PARITY**, where $\exp(x) = 2^x$.
- **Proof sketch.** “Divide & conquer” for $d-1$ levels. Alternate between CNFs and DNFs. “Attach” the CNFs and the DNFs appropriately, and then “merge” the intermediate layers to bring the depth down to d .
- Is the $\exp(n^{1/(d-1)})$ upper bound on the size of depth d circuits computing **PARITY** tight? “Yes”

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- Furst, Saxe and Sipser showed a quasi-polynomial lower bound.
- Ajtai showed an exponential lower bound, but the bound wasn't optimal.
- Finally, Hastad showed an optimal lower bound.


Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- Gives a super-polynomial lower bound for depth d circuits for d up to $O(\log n / \log \log n)$.
- A lower bound for circuits of depth $d = O(\log n)$ implies a Boolean formula lower bound!

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof idea.** A **random assignment** to a “large” fraction of the variables makes a constant depth circuit of polynomial size evaluate to a constant (i.e., the circuit stops depending on the unset variables). On the other hand, we cannot make **PARITY** evaluate to a constant by setting less than n variables.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof idea.** A **random assignment** to a “large” fraction of the variables makes a constant depth circuit of polynomial size evaluate to a constant (i.e., the circuit stops depending on the unset variables).

- We'll prove this fact using Hastad's **Switching lemma**. But first let us discuss some structural simplifications of depth d circuits.

Simplifying depth d circuits

- **Fact 1.** If $f(x_1, \dots, x_n)$ is computable by a circuit of depth d and size s , then f is also computable by a circuit C of depth d and size $O(s)$ such that C has no \neg gates and the inputs to C are x_1, \dots, x_n and $\neg x_1, \dots, \neg x_n$.

Simplifying depth d circuits

- **Fact 1.** If $f(x_1, \dots, x_n)$ is computable by a circuit of depth d and size s , then f is also computable by a circuit C of depth d and size $O(s)$ such that C has no \neg gates and the inputs to C are x_1, \dots, x_n and $\neg x_1, \dots, \neg x_n$.
- **Fact 2.** If f is computable by a circuit of depth d and size s , then f is also computable by a formula of depth d and size $O(s)^d$.

Simplifying depth d circuits

- **Fact 1.** If $f(x_1, \dots, x_n)$ is computable by a circuit of depth d and size s , then f is also computable by a circuit C of depth d and size $O(s)$ such that C has no \neg gates and the inputs to C are x_1, \dots, x_n and $\neg x_1, \dots, \neg x_n$.
- **Fact 2.** If f is computable by a circuit of depth d and size s , then f is also computable by a formula of depth d and size $O(s)^d$.
- **Fact 3.** If f is computable by a formula of depth d and size s , then f is computable by a formula C of depth d and size $O(sd)$ that has alternating layers of \vee and \wedge gates with inputs feeding into *only* the bottom layer.

Simplifying depth d circuits

- **Fact 1.** If $f(x_1, \dots, x_n)$ is computable by a circuit of depth d and size s , then f is also computable by a circuit C of depth d and size $O(s)$ such that C has no \neg gates and the inputs to C are x_1, \dots, x_n and $\neg x_1, \dots, \neg x_n$.
- **Fact 2.** If f is computable by a circuit of depth d and size s , then f is also computable by a formula of depth d and size $O(s)^d$.
- **Fact 3.** If f is computable by a formula of depth d and size s , then f is computable by a formula C of depth d and size $O(sd)$ that has alternating layers of \vee and \wedge gates with inputs feeding into *only* the bottom layer.

Homework: Prove the above facts.

Random restrictions

- A restriction σ is a partial assignment to a subset of the n variables.
- A random restriction σ that leaves m variables alive/unset is obtained by picking a random subset $S \subseteq [n]$ of size $n-m$ and setting every variable in S to 0/1 uniformly and independently.
- Let f_σ denote the function obtained by applying the restriction σ on f .

The Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,
$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$

The Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,
$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$
- We can interchange “CNF” and “DNF” in the above statement by applying the lemma on $\neg f$.

The Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,
$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$
- We can interchange “CNF” and “DNF” in the above statement by applying the lemma on $\neg f$.
- Before proving the lemma, let us see how it is used to prove lower bound for depth d circuits.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** Bottom-up application of the switching lemma.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** W.l.o.g C is in the simplified form and the bottom/last layer consists of v gates. $\text{Size}(C) = s$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** W.l.o.g C is in the simplified form and the bottom/last layer consists of v gates. $\text{Size}(C) = s$.
- **Step 0:** Pick every variable independently with prob. $1/2$ and set it to **0/1** uniformly. C_1 be the resulting ckt.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** W.l.o.g C is in the simplified form and the bottom/last layer consists of v gates. $\text{Size}(C) = s$.
- **Step 0:** Pick every variable independently with prob. $1/2$ and set it to **0/1** uniformly. C_1 be the resulting ckt.
- Let t be a parameter that we'll fix later in the analysis. If a v gate in the last layer has fan-in $> t$, then the probability it doesn't evaluate to **1** is $\leq (3/4)^t$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** W.l.o.g C is in the simplified form and the bottom/last layer consists of v gates. $\text{Size}(C) = s$.
- **Step 0:** Pick every variable independently with prob. $1/2$ and set it to **0/1** uniformly. C_1 be the resulting ckt.
- Let t be a parameter that we'll fix later in the analysis. If a v gate in the last layer has fan-in $> t$, then the probability it doesn't evaluate to **1** is $\leq (3/4)^t$. So,
$$\Pr[\text{a fan-in } > t \text{ last layer } v \text{ gate survives}] \leq s(3/4)^t.$$

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** W.l.o.g C is in the simplified form and the bottom/last layer consists of v gates. $\text{Size}(C) = s$.
- **Step 0:** Pick every variable independently with prob. $1/2$ and set it to **0/1** uniformly. C_1 be the resulting ckt.
- Let t be a parameter that we'll fix later in the analysis. If a v gate in the last layer has fan-in $> t$, then the probability it doesn't evaluate to **1** is $\leq (3/4)^t$. So,
$$\Pr[\text{a fan-in } > t \text{ last layer } v \text{ gate survives}] \leq s(3/4)^t.$$

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** W.l.o.g C is in the simplified form and the bottom/last layer consists of v gates. $\text{Size}(C) = s$.
- **Step 0:** Pick every variable independently with prob. $1/2$ and set it to **0/1** uniformly. C_1 be the resulting ckt.
- With probability $\geq 1 - s(3/4)^t$, every \wedge gate of the second-last layer of C_1 computes a **t**-CNF.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** W.l.o.g C is in the simplified form and the bottom/last layer consists of v gates. $\text{Size}(C) = s$.
- **Step 0:** Pick every variable independently with prob. $1/2$ and set it to **0/1** uniformly. C_1 be the resulting ckt.
- With probability $\geq 1 - s(3/4)^t$, every \wedge gate of the second-last layer of C_1 computes a **t**-CNF.
- Let n_1 be the no. of unset variables after Step 0. By Chernoff bound, $n_1 \geq n/4$ with probability $1 - 2^{-\Omega(n)}$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** W.l.o.g C is in the simplified form and the bottom/last layer consists of v gates. $\text{Size}(C) = s$.
- **Step 0:** Pick every variable independently with prob. $1/2$ and set it to **0/1** uniformly. C_1 be the resulting ckt.
- With probability $\geq 1 - s(3/4)^t$, every \wedge gate of the second-last layer of C_1 computes a **t**-CNF.
- Let n_1 be the no. of unset variables after Step 0. By Chernoff bound, $n_1 \geq n/4$ with probability $1 - 2^{-\Omega(n)}$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\wedge \text{ gates of the second-last layer of } C_1) \leq s$.
- **Step 1:** Apply a random restriction σ_1 on the n_1 variables that leaves $n_2 = pn_1$ variables alive, where $p < 1/2$ will be fixed later.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\wedge \text{ gates of the second-last layer of } C_1) \leq s$.
- **Step 1:** Apply a random restriction σ_1 on the n_1 variables that leaves $n_2 = pn_1$ variables alive, where $p < 1/2$ will be fixed later.
- By the Switching lemma, probability that any of the t -CNFs computed at the second-last layer of C_1 cannot be expressed as a t -DNF is $\leq s \cdot (16pt)^t$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\wedge \text{ gates of the second-last layer of } C_1) \leq s$.
- **Step 1:** Apply a random restriction σ_1 on the n_1 variables that leaves $n_2 = pn_1$ variables alive, where $p < 1/2$ will be fixed later.
- By the Switching lemma, probability that any of the t -CNFs computed at the second-last layer of C_1 cannot be expressed as a t -DNF is $\leq s \cdot (16pt)^t$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\wedge \text{ gates of the second-last layer of } C_1) \leq s$.
- **Step 1:** Apply a random restriction σ_1 on the n_1 variables that leaves $n_2 = pn_1$ variables alive, where $p < 1/2$ will be fixed later.
- Replace the t -CNFs by the corresponding t -DNFs.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\wedge \text{ gates of the second-last layer of } C_1) \leq s$.
- **Step 1:** Apply a random restriction σ_1 on the n_1 variables that leaves $n_2 = pn_1$ variables alive, where $p < 1/2$ will be fixed later.
- Replace the t -CNFs by the corresponding t -DNFs.
- Merge the \vee gates of the second-last layer with the \vee gates of the layer above. C_2 be the resulting ckt.

Lower bound for depth d circuits

- **Theorem.** (Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\wedge \text{ gates of the second-last layer of } C_1) \leq s$.
- **Step 1:** Apply a random restriction σ_1 on the n_1 variables that leaves $n_2 = pn_1$ variables alive, where $p < 1/2$ will be fixed later.
- The no. of \vee gates of the second-last layer of the resulting circuit C_2 equals the no. of \vee gates of the third-last layer of C_1 . So, this no. is $\leq s$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\wedge \text{ gates of the second-last layer of } C_1) \leq s$.
- **Step 1:** Apply a random restriction σ_1 on the n_1 variables that leaves $n_2 = pn_1$ variables alive, where $p < 1/2$ will be fixed later.
- Merging reduces the depth to $d-1$.
- All the gates of the second-last layer of C_2 compute t -DNFs with probability $\geq 1 - s \cdot (16pt)^t$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** # (\vee gates of the second-last layer of C_2) $\leq s$.
- **Step 2:** Apply a random restriction σ_2 on the n_2 variables that leaves $n_3 = pn_2$ variables alive, where p is same as before.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\vee \text{ gates of the second-last layer of } C_2) \leq s$.
- **Step 2:** Apply a random restriction σ_2 on the n_2 variables that leaves $n_3 = pn_2$ variables alive, where p is same as before.
- By the Switching lemma, probability that any of the t -DNFs computed at the second-last layer of C_2 cannot be expressed as a t -CNF is $\leq s \cdot (16pt)^t$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\vee \text{ gates of the second-last layer of } C_2) \leq s$.
- **Step 2:** Apply a random restriction σ_2 on the n_2 variables that leaves $n_3 = pn_2$ variables alive, where p is same as before.
- By the Switching lemma, probability that any of the t -DNFs computed at the second-last layer of C_2 cannot be expressed as a t -CNF is $\leq s \cdot (16pt)^t$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\vee \text{ gates of the second-last layer of } C_2) \leq s$.
- **Step 2:** Apply a random restriction σ_2 on the n_2 variables that leaves $n_3 = pn_2$ variables alive, where p is same as before.
- Replace the \vee -DNFs by the corresponding \vee -CNFs.
- Merge the \wedge gates of the second-last layer with the \wedge gates of the layer above. C_3 be the resulting ckt.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\vee \text{ gates of the second-last layer of } C_2) \leq s$.
- **Step 2:** Apply a random restriction σ_2 on the n_2 variables that leaves $n_3 = pn_2$ variables alive, where p is same as before.
- The no. of \wedge gates of the second-last layer of the resulting circuit C_3 equals the no. of \wedge gates of the third-last layer of C_2 . So, this no. is $\leq s$ (why?).

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\vee \text{ gates of the second-last layer of } C_2) \leq s$.
- **Step 2:** Apply a random restriction σ_2 on the n_2 variables that leaves $n_3 = pn_2$ variables alive, where p is same as before.
- Merging reduces the depth to $d-2$.
- All the gates of the second-last layer of C_3 compute t -CNFs with probability $\geq 1 - s \cdot (16pt)^t$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** $\# (\wedge \text{ gates of the second-last layer of } C_3) \leq s$.
- **Step 3:** Apply a random restriction σ_3 on the n_3 variables that leaves $n_4 = pn_3$ variables alive, where p is same as before. Continue as before..

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** After **Step $d-2$** , we are left with a depth **2** circuit, i.e., a t -CNF or a t -DNF with probability $\geq 1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)}$.
- The number of variables alive is $p^{d-2}n_1 \geq (p^{d-2}n)/4$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** After **Step $d-2$** , we are left with a depth **2** circuit, i.e., a t -CNF or a t -DNF with probability $\geq 1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)}$.
- The number of variables alive is $p^{d-2}n_1 \geq (p^{d-2}n)/4$.
- Observe that by setting t more variables, we can now fix the value of the circuit. But, recall that the value of **PARITY** cannot be fixed by setting $< n$ variables.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** After **Step $d-2$** , we are left with a depth **2** circuit, i.e., a t -CNF or a t -DNF with probability $\geq 1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)}$.
- The number of variables alive is $p^{d-2}n_1 \geq (p^{d-2}n)/4$.
- Hence,
 either $1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)} \leq 0$,
 or $p^{d-2}n_1 \leq t$.

Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** After **Step $d-2$** , we are left with a depth **2** circuit, i.e., a t -CNF or a t -DNF with probability \geq

$$1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)}.$$


- The number of variables alive is $p^{d-2}n_1 \geq (p^{d-2}n)/4$.
- By choosing $t = O(n^{1/(d-1)})$ and $p = 1/(160t)$, we can make sure that

$$p^{d-2}n_1 > t.$$

$< 1/2$



Lower bound for depth d circuits

- **Theorem.** (*Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86*)
Any depth d circuit C computing **PARITY** has size $\exp(\Omega_d(n^{1/(d-1)}))$, where $\Omega_d()$ is hiding a $\text{poly}(d)^{-1}$ factor.
- **Proof.** After **Step $d-2$** , we are left with a depth **2** circuit, i.e., a t -CNF or a t -DNF with probability \geq
 $1 - s \cdot (d-2)(16pt)^t - 2^{-\Omega(n)}.$
- The number of variables alive is $p^{d-2}n_1 \geq (p^{d-2}n)/4.$
- Therefore, for $t = O(n^{1/(d-1)})$ and $p = 1/(160t),$
 $1 - s \cdot (d-2)(16pt)^t - 2^{-\Omega(n)} \leq 0,$
 $s = \exp(\Omega(n^{1/(d-1)})).$



Proof of the Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,
$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$
- **Proof.** We'll present a proof due to Razborov.

Proof of the Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,
$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$
- **Proof.** Let A_{ℓ} be the set of restrictions that keeps ℓ variables alive. Then, $|A_{\ell}| = \binom{n}{\ell} 2^{n-\ell}$.

Proof of the Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,
$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$
- **Proof.** Let A_{ℓ} be the set of restrictions that keeps ℓ variables alive. Then, $|A_{\ell}| = \binom{n}{\ell} \cdot 2^{n-\ell}$. Let $B_{m,k} \subseteq A_m$ be the set of “bad” restrictions, i.e., a $\sigma \in A_m$ is in $B_{m,k}$ iff f_{σ} can't be represented as a k -DNF.
- We need to upper bound $|B_{m,k}|$.

Proof of the Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,

$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$

- **Proof.** Let A_{ℓ} be the set of restrictions that keeps ℓ variables alive. Then, $|A_{\ell}| = \binom{n}{\ell} \cdot 2^{n-\ell}$. Let $B_{m,k} \subseteq A_m$ be the set of “bad” restrictions, i.e., a $\sigma \in A_m$ is in $B_{m,k}$ iff f_{σ} can't be represented as a k -DNF.
- We need to upper bound $|B_{m,k}|$.
- This is done by giving an **injective map** from $B_{m,k}$ to $A_{m-k} \times U$, where $U = \{0,1\}^{k(\log t + 2)}$. $|U| = (4t)^k$.

Proof of the Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,

$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$

- **Proof.** Then, $|B_{m,k}| \leq \binom{n}{m-k} \cdot 2^{n-m+k} \cdot (4t)^k$. and so

$$|B_{m,k}| / |A_m| \leq [(m! \cdot (n-m)!) / ((m-k)! \cdot (n-m+k)!)] \cdot 2^k \cdot (4t)^k$$

Proof of the Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,

$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$

- **Proof.** Then, $|B_{m,k}| \leq \binom{n}{m-k} \cdot 2^{n-m+k} \cdot (4t)^k$. and so

$$\begin{aligned} |B_{m,k}|/|A_m| &\leq [(m! \cdot (n-m)!)/(m-k)! \cdot (n-m+k)!] \cdot 2^k \cdot (4t)^k \\ &\leq (m/(n-m))^k \cdot 2^k \cdot (4t)^k \\ &= (p/(1-p))^k \cdot 2^k \cdot (4t)^k \quad (\text{as } m = pn) \\ &\leq p^k \cdot 2^k \cdot 2^k \cdot (4t)^k \quad (\text{as } p < 1/2) \\ &= (16pt)^k. \end{aligned}$$

Proof of the Switching Lemma

- **Switching lemma.** Let f be a t -CNF on n variables and σ a random restriction that leaves $m = pn$ variables alive, where $p < 1/2$. Then,
$$\Pr_{\sigma} [f_{\sigma} \text{ can't be represented as a } k\text{-DNF}] \leq (16pt)^k.$$
- **Proof.** Next, we show an injection from $B_{m,k}$ to $A_{m-k} \times U$, where $U = \{0,1\}^{k(\log t + 2)}$.

A definition and a notation

- **Definition.** A min-term of a function g is a restriction π such that $g_\pi = 1$, but **no** proper sub-restriction of π makes g evaluate to 1.
- **Obs.** If g can't be expressed as a k -DNF, then g has a min-term π of width $> k$ (i.e., π assigns 0/1 values to more than k variables). (*Homework*)

A definition and a notation

- **Definition.** A min-term of a function g is a restriction π such that $g_\pi = 1$, but no proper sub-restriction of π makes g evaluate to 1.
- **Obs.** If g can't be expressed as a k -DNF, then g has a min-term π of width $> k$ (i.e., π assigns 0/1 values to more than k variables). (*Homework*)
- **Notation.** If σ is a restriction that assigns 0/1 values to variables in $S_1 \subseteq [n]$ and π is a restriction that assigns 0/1 values to variables in $S_2 \subseteq [n] \setminus S_1$, then $\sigma \circ \pi$ is the “composed” restriction that assigns 0/1 values to $S_1 \cup S_2$ consistent with σ and π . $|\pi| := \text{width of } \pi$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$: (*Overview*)
 - **Step 1:** For $\sigma \in B_{m,k}$, let π be the lexicographically smallest min-term of f_σ of width $> k$. We'll carefully define a sub-restriction π' of π of width k .

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$: (*Overview*)
 - **Step 1:** For $\sigma \in B_{m,k}$, let π be the lexicographically smallest min-term of f_σ of width $> k$. We'll carefully define a sub-restriction π' of π of width k .
 - **Step 2:** Using π' , we'll carefully define a restriction ρ that assigns $0/1$ values to the same set of variables as π' .

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$: (*Overview*)
 - **Step 1:** For $\sigma \in B_{m,k}$, let π be the lexicographically smallest min-term of f_σ of width $> k$. We'll carefully define a sub-restriction π' of π of width k .
 - **Step 2:** Using π' , we'll carefully define a restriction ρ that assigns $0/1$ values to the same set of variables as π' .
 - **Step 3:** Using π' , define a $u \in U$. Finally, $\chi(\sigma) := (\sigma \circ \rho, u)$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 1:** For $\sigma \in B_{m,k}$, let π be the lexicographically smallest min-term of f_σ of width $> k$. Order the clauses of f , and order the $\leq t$ variables appearing within such a clause.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 1:** For $\sigma \in B_{m,k}$, let π be the lexicographically smallest min-term of f_σ of width $> k$. Order the clauses of f , and order the $\leq t$ variables appearing within such a clause. C_1 be the first surviving clause in f_σ and $\pi(1)$ the assignment to its surviving variables made by π .

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 1:** For $\sigma \in B_{m,k}$, let π be the lexicographically smallest min-term of f_σ of width $> k$. Order the clauses of f , and order the $\leq t$ variables appearing within such a clause. C_1 be the first surviving clause in f_σ and $\pi(1)$ the assignment to its surviving variables made by π . C_2 be the first surviving clause in $f_{\sigma \circ \pi(1)}$ and $\pi(2)$ the assignment to its surviving variables made by π .

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 1:** For $\sigma \in B_{m,k}$, let π be the lexicographically smallest min-term of f_σ of width $> k$. Order the clauses of f , and order the $\leq t$ variables appearing within such a clause. C_1 be the first surviving clause in f_σ and $\pi(1)$ the assignment to its surviving variables made by π . C_2 be the first surviving clause in $f_{\sigma \circ \pi(1)}$ and $\pi(2)$ the assignment to its surviving variables made by π . Continue like this.. Stop if $|\pi(1) \circ \dots \circ \pi(r)| \geq k$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step I:** If $|\pi(l) \circ \dots \circ \pi(r)| > k$, then “prune” $\pi(r)$ by restricting it to the set of “smallest” variables in C_r so that $|\pi(l) \circ \dots \circ \pi(r)| = k$. Define $\pi' := \pi(l) \circ \dots \circ \pi(r)$; $|\pi'| = k$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

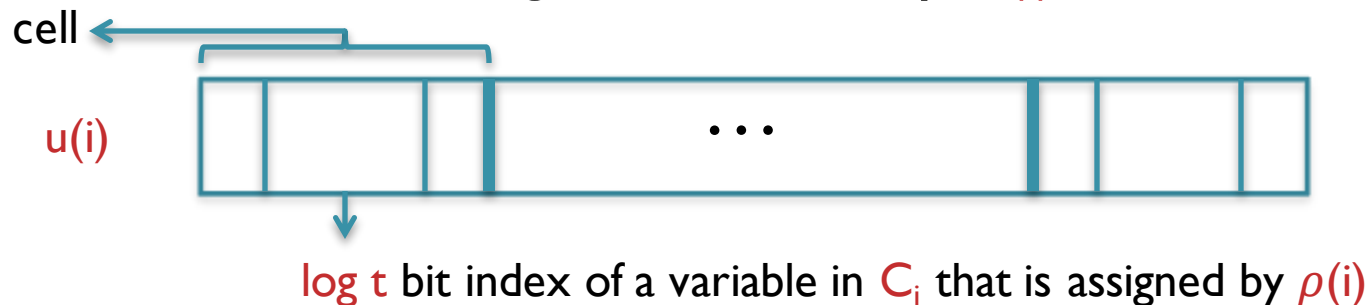
- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 2:** For $i \in [r]$, let S_i be the set of variables in the clause C_i that are assigned 0/1 values by $\pi(i)$. $|S_i| = |\pi(i)|$. Let $\rho(i)$ be the unique assignment to the variables in S_i that makes the corresponding literals in C_i zero. Define $\rho := \rho(1) \circ \dots \circ \rho(r)$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 2:** For $i \in [r]$, let S_i be the set of variables in the clause C_i that are assigned 0/1 values by $\pi(i)$. $|S_i| = |\pi(i)|$. Let $\rho(i)$ be the unique assignment to the variables in S_i that makes the corresponding literals in C_i zero. Define $\rho := \rho(1) \circ \dots \circ \rho(r)$.
 - **Remark*.** $\pi(i)$ and $\rho(i)$ are assignments to the same set of variables S_i . C_i remains unsatisfied under $\rho(i)$.

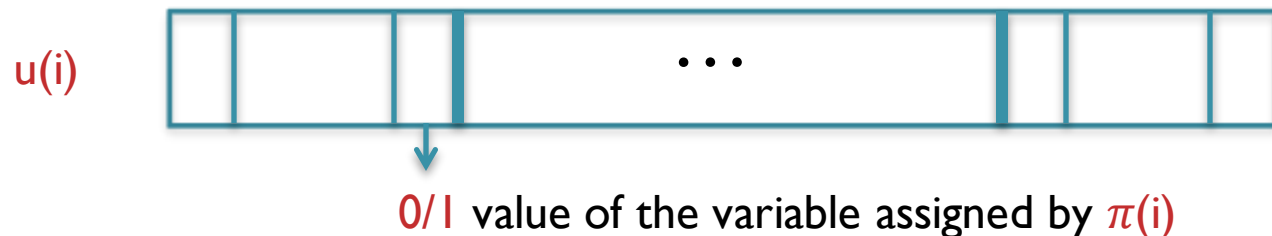
Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 3:** For $i \in [r]$, let $u(i)$ be the string obtained by listing the indices (within the clause C_i) of the variables assigned by $\rho(i)$ along with the values assigned to them by $\pi(i)$.



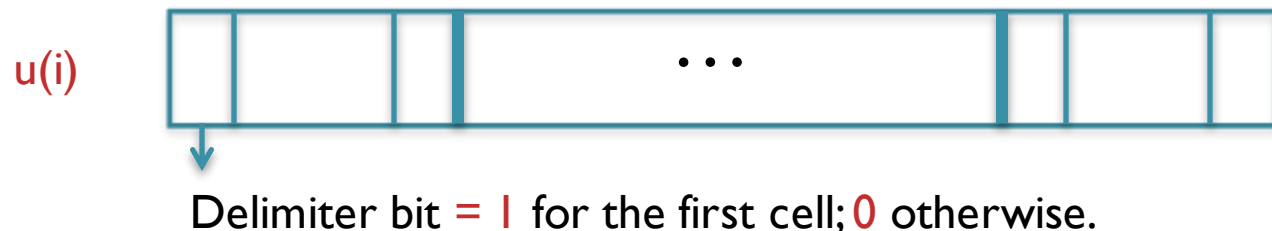
Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 3:** For $i \in [r]$, let $u(i)$ be the string obtained by listing the indices (within the clause C_i) of the variables assigned by $\rho(i)$ along with the values assigned to them by $\pi(i)$.



Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}$.
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 3:** For $i \in [r]$, let $u(i)$ be the string obtained by listing the indices (within the clause C_i) of the variables assigned by $\rho(i)$ along with the values assigned to them by $\pi(i)$.



Injection from $B_{m,k}$ to $A_{m-k} \times U$

- f is a t -CNF on n variables.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma \text{ can't be represented as a } k\text{-DNF}\}.$
- **Obs.** If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to $A_{m-k} \times U$:
 - **Step 3:** For $i \in [r]$, let $u(i)$ be the string obtained by listing the indices (within the clause C_i) of the variables assigned by $\rho(i)$ along with the values assigned to them by $\pi(i)$. Define u by concatenating $u(1), \dots, u(r)$ in order. Observe that $|u| = k(\log t + 2)$. Finally, $\chi(\sigma) := (\sigma \circ \rho, u)$. (**Remark.** The delimiter bits make it possible to extract $u(i)$ from u .)

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- We'll now show that it is possible to recover σ from $(\sigma \circ \rho, u)$ which implies χ is an injection.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- We'll now show that it is possible to recover σ from $(\sigma \circ \rho, u)$ which implies χ is an injection.
- **Obs***. For every $i \in [r]$, the first “unsatisfied” clause in $f_{\sigma \circ \pi(1) \circ \dots \circ \pi(i-1) \circ \rho(i) \circ \dots \circ \rho(r)}$ is C_i .
- **Proof**. Fix an $i \in [r]$. By construction, C_i is the first surviving clause in $f_{\sigma \circ \pi(1) \circ \dots \circ \pi(i-1)}$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- We'll now show that it is possible to recover σ from $(\sigma \circ \rho, u)$ which implies χ is an injection.
- **Obs***. For every $i \in [r]$, the first “unsatisfied” clause in $f_{\sigma \circ \pi(1) \circ \dots \circ \pi(i-1) \circ \rho(i) \circ \dots \circ \rho(r)}$ is C_i .
- **Proof**. Fix an $i \in [r]$. By construction, C_i is the first surviving clause in $f_{\sigma \circ \pi(1) \circ \dots \circ \pi(i-1)}$. C_i remains unsatisfied under $\rho(i)$ (**Remark***). Further, $\rho(i+1), \dots, \rho(r)$ do not touch any variable of C_i . Hence, C_i is the first unsatisfied clause in $f_{\sigma \circ \pi(1) \circ \dots \circ \pi(i-1) \circ \rho(i) \circ \dots \circ \rho(r)}$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- We'll now show that it is possible to recover σ from $(\sigma \circ \rho, u)$ which implies χ is an injection.
- **Obs***. For every $i \in [r]$, the first “unsatisfied” clause in $f_{\sigma \circ \pi(l) \circ \dots \circ \pi(i-1) \circ \rho(i) \circ \dots \circ \rho(r)}$ is C_i .
- Recovering σ from $(\sigma \circ \rho, u)$:
 - Pick the first unsatisfied clause in $f_{\sigma \circ \rho(l) \circ \dots \circ \rho(r)}$. This clause is C_l (**Obs***). Now by looking at $u(l)$, we can derive $\pi(l)$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- We'll now show that it is possible to recover σ from $(\sigma \circ \rho, u)$ which implies χ is an injection.
- **Obs***. For every $i \in [r]$, the first “unsatisfied” clause in $f_{\sigma \circ \pi(1) \circ \dots \circ \pi(i-1) \circ \rho(i) \circ \dots \circ \rho(r)}$ is C_i .
- Recovering σ from $(\sigma \circ \rho, u)$:
 - Pick the first unsatisfied clause in $f_{\sigma \circ \rho(1) \circ \dots \circ \rho(r)}$. This clause is C_1 (**Obs***). Now by looking at $u(1)$, we can derive $\pi(1)$. Construct $\sigma \circ \pi(1) \circ \rho(2) \circ \dots \circ \rho(r)$ from $\sigma \circ \rho(1) \circ \dots \circ \rho(r)$ and $\pi(1)$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- We'll now show that it is possible to recover σ from $(\sigma \circ \rho, u)$ which implies χ is an injection.
- **Obs***. For every $i \in [r]$, the first “unsatisfied” clause in $f_{\sigma \circ \pi(1) \circ \dots \circ \pi(i-1) \circ \rho(i) \circ \dots \circ \rho(r)}$ is C_i .
- Recovering σ from $(\sigma \circ \rho, u)$:
 - Pick the first unsatisfied clause in $f_{\sigma \circ \pi(1) \circ \rho(2) \circ \dots \circ \rho(r)}$. This clause is C_2 (**Obs***). Now by looking at $u(2)$, we can derive $\pi(2)$. Construct $\sigma \circ \pi(1) \circ \pi(2) \circ \rho(3) \circ \dots \circ \rho(r)$ from $\sigma \circ \pi(1) \circ \rho(2) \circ \dots \circ \rho(r)$ and $\pi(2)$.

Injection from $B_{m,k}$ to $A_{m-k} \times U$

- We'll now show that it is possible to recover σ from $(\sigma \circ \rho, u)$ which implies χ is an injection.
- **Obs***. For every $i \in [r]$, the first “unsatisfied” clause in $f_{\sigma \circ \pi(1) \circ \dots \circ \pi(i-1) \circ \rho(i) \circ \dots \circ \rho(r)}$ is C_i .
- Recovering σ from $(\sigma \circ \rho, u)$:
 - Continuing like this we can construct $\sigma \circ \pi(1) \circ \dots \circ \pi(r)$ and also find $\pi(1), \dots, \pi(r)$ in the process. From here, recovering σ is straightforward.

- Ref.

<https://sites.math.rutgers.edu/~skl233/courses/topics-SI3/lec3.pdf>