Computational Complexity Theory

Lecture 18: Parity not in AC^0 (contd.)

Department of Computer Science,
Indian Institute of Science

Recap: 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"

Recap: 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 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!

Recap: 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 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.

Will be proved in today's lecture

Recap: 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.

Recap: 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 $p < ½$. Then,

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

- We can interchange "CNF" and "DNF" in the above statement by applying the lemma on $\neg f$.

- We used the lemma in the last lecture to prove lower bound for depth d circuits.

Recap: 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 d^{-1} factor.
- Proof. W.l.o.g C is in the simplified form and the bottom/last layer consists of $\vee$ gates. $\text{Size}(C) = s$.
- **Step 0:** Pick every variable independently with prob. ½, and then 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 $\vee$ 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[a fan-in $> t$ last layer $\vee$ gate survives] $\leq \boxed{s(3/4)^t.}$

Recap: Lower bound for depth d circuits

- **Theorem.** *(Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86)* Any depth d circuit C computing PARITY has size $\exp(\boldsymbol{\Omega}_d(n^{1/(d-1)}))$, where $\boldsymbol{\Omega}_d()$ is hiding a d^{-1} factor.
- **Proof.** W.l.o.g C is in the simplified form and the bottom/last layer consists of $\vee$ gates. $\mathrm{Size}(C) = s$.
- **Step 0:** Pick every variable independently with prob. ½, and then 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, $\boxed{n_1 \geq n/4}$ with probability $1 - \boxed{2^{-\Omega(n)}}$.

Recap: Lower bound for depth d circuits

- **Theorem.** *(Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86)* Any depth d circuit C computing PARITY has size $\exp(\boldsymbol{\Omega}_d(n^{1/(d-1)}))$, where $\boldsymbol{\Omega}_d()$ is hiding a d^{-1} factor.
- **Proof.** # ($\wedge$ gates of the second-last layer of C_1) $\leq s$.
- **Step 1:** Apply a random restriction $\boldsymbol{\sigma}_1$ on the n_1 variables that leaves $n_2 = pn_1$ variables alive, where $p < ½$ 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 $\leq \boxed{s.(16pt)^t.}$

Recap: 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 d^{-1} factor.
- Proof. # ($\wedge$ 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 $p < ½$ 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.

Recap: 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 d^{-1} factor.
- Proof. # ($\wedge$ 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 $p < ½$ 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 $\geq 1 - s.(16pt)^t$.

Recap: Lower bound for depth d circuits

- Theorem. *(Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86)* Any depth d circuit C computing PARITY has size $\exp(\boldsymbol{\Omega}_d(n^{1/(d-1)}))$, where $\boldsymbol{\Omega}_d()$ is hiding a d^{-1} factor.
- Proof. # ($\vee$ gates of the second-last layer of C_2) $\leq s$.
- **Step 2:** Apply a random restriction $\boldsymbol{\sigma}_2$ 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 $\leq \boxed{s.(16pt)^t.}$

Recap: 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 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.
- Replace the t-DNFs by the corresponding t-CNFs.
- Merge the $\wedge$ gates of the second-last layer with the $\wedge$ gates of the layer above. C_3 be the resulting ckt.

Recap: Lower bound for depth d circuits

- Theorem. *(Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86)* Any depth d circuit C computing PARITY has size $\exp(\mathbf{\Omega}_d(n^{1/(d-1)}))$, where $\mathbf{\Omega}_d()$ is hiding a d^{-1} factor.
- Proof. # ($\vee$ gates of the second-last layer of C_2) $\leq s$.
- **Step 2:** Apply a random restriction $\boldsymbol{\sigma}_2$ 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 $\geq 1 - s.(16pt)^t$.

Recap: 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 d^{-1} factor.
- Proof. # ($\wedge$ 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..

Recap: Lower bound for depth d circuits

- Theorem. *(Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86)* Any depth d circuit C computing PARITY has size $\exp(\mathbf{\Omega}_d(n^{1/(d-1)}))$, where $\mathbf{\Omega}_d()$ is hiding a 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 $\geq$

$$1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)} - s(3/4)^t.$$

- The number of variables alive is $p^{d-2}n_1 \geq (p^{d-2}n)/4$.
- Hence,

either $1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)} - s(3/4)^t \leq 0$,

or $p^{d-2}n_1 \leq t$.

Recap: Lower bound for depth d circuits

- **Theorem.** *(Furst, Saxe, Sipser '81; Ajtai '83; Hastad '86)* Any depth d circuit C computing PARITY has size $\exp(\boldsymbol{\Omega}_d(n^{1/(d-1)}))$, where $\boldsymbol{\Omega}_d()$ is hiding a 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 $\geq$

$$1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)} - s(3/4)^t.$$

- 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 $p = 1/(160\ t)$, we can make sure that

$< ½$

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

Recap: 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 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 $\geq$

$$1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)} - s(3/4)^t.$$

- 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 $p = 1/(160\ t)$,

$$1 - s.(d-2)(16pt)^t - 2^{-\Omega(n)} - s(3/4)^t \leq 0,$$

$$\Rightarrow \quad s = \exp(\Omega(n^{1/(d-1)})).$$

$\blacksquare$

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 $p < ½$. 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 $p < ½$. Then,

  $\Pr_\sigma$ [f_σ can't be represented as a k-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 $p < ½$. Then,

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

- Proof. Let A_ℓ be the set of restrictions that keeps ℓ variables alive. Then, $|A_\ell| = \binom{n}{\ell}.2^{n-\ell}$. 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 $p < ½$. 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, $|A_\ell| = \binom{n}{\ell}.2^{n-\ell}$. 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 $p < ½$. Then,

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

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

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

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 $p < ½$. Then,

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

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

$$
\begin{aligned}
|B_{m,k}|/|A_m| &\leq [(m! \,.\, (n-m)!) \,/\, ((m-k)! \,.\, (n-m+k)!)].2^k.(4t)^k \\
&\leq (m/(n-m))^k \,.\, 2^k \,.\, (4t)^k \\
&= (p/(1-p))^k \,.\, 2^k \,.\, (4t)^k \qquad \text{(as } m = pn\text{)} \\
&\leq p^k \,.\, 2^k \,.\, 2^k \,.\, (4t)^k \qquad \text{(as } p < ½ \text{)} \\
&= (16pt)^k \,.
\end{aligned}
$$

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 $p < ½$. Then,

  $\Pr_\sigma$ [f_σ can't be represented as a k-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 $\boxed{\sigma \circ \pi}$ is the "composed" restriction that assigns $0/1$ values to $S_1 \uplus S_2$ consistent with σ and π. $\boxed{|\pi| := \text{width of } \pi}$.

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

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x U

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x U

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x 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} x U

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\boldsymbol{\sigma} \in A_m : f_{\boldsymbol{\sigma}}$ can't be represented as a k-DNF$\}$.
- Obs. If $\boldsymbol{\sigma} \in B_{m,k}$ then $f_{\boldsymbol{\sigma}}$ has a min-term of width > k.
- A map χ from $B_{m,k}$ to A_{m-k} x U : (*Overview*)

➢ **Step 1:** For $\boldsymbol{\sigma} \in B_{m,k}$, let π be the lexicographically smallest min-term of $f_{\boldsymbol{\sigma}}$ of width > k. We'll carefully define a <u>sub-restriction π'</u> of π of width k.

➢ **Step 2:** Using π', we'll carefully define a <u>restriction ρ</u> that assigns 0/1 values to the same set of variables as π'.

➢ **Step 3:** Using π', <u>define a u</u> ∈ U. Finally, $\chi(\boldsymbol{\sigma}) := (\boldsymbol{\sigma} \circ \rho, u)$.

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

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x 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} x U

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x 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} x U

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x U

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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 $|\pi(1)\circ\ldots\circ\pi(r)| \geq k$.

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

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x U :
  - **Step 1:** If $|\pi(1)\circ\ldots\circ\pi(r)| > k$, then "prune" $\pi(r)$ by restricting it to the set of "smallest" variables in C_r so that $|\pi(1)\circ\ldots\circ\pi(r)| = k$. Define $\pi' := \pi(1)\circ\ldots\circ\pi(r)$; $|\pi'| = k$.

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

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x 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 $\rho := \rho(1)\circ\ldots\circ\rho(r)$.

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

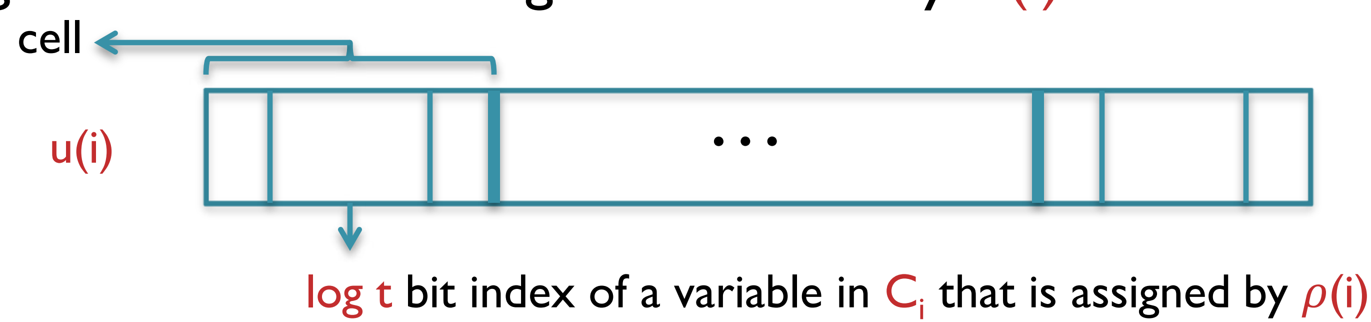
- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-DNF$\}$.
- Obs. If $\sigma \in B_{m,k}$ then f_σ has a min-term of width $> k$.
- A map χ from $B_{m,k}$ to A_{m-k} x 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 $\rho := \rho(1) \circ \ldots \circ \rho(r)$.

➢ 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} x U

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x 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 $\pi(i)$.

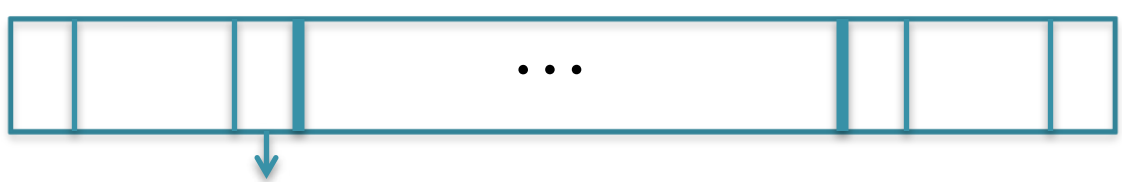


Injection from $B_{m,k}$ to A_{m-k} x U

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x U :

➢ **Step 3:** For $i \in [r]$, let $u(i)$ be the string obtained by listing the indices (<u>*within*</u> the clause C_i) of the variables assigned by $\rho(i)$ along with the values assigned to them by $\pi(i)$.

u(i)

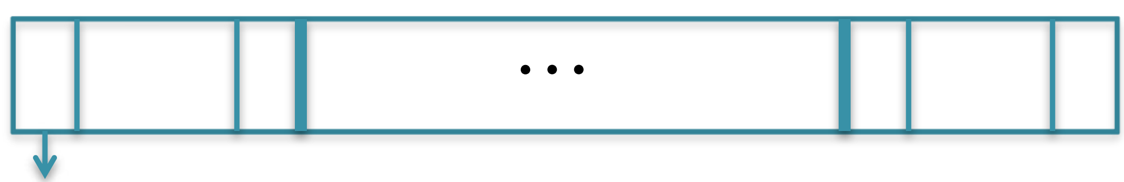


0/1 value of the variable assigned by $\pi(i)$

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

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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} x U :
  - **Step 3:** For $i \in [r]$, let $u(i)$ be the string obtained by listing the indices (<u>*within*</u> the clause C_i) of the variables assigned by $\rho(i)$ along with the values assigned to them by $\pi(i)$.

u(i)



Delimiter bit = 1 for the first cell; 0 otherwise.

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

- f is a t-CNF on n variables. $U = \{0,1\}^{k(\log t + 2)}$.
- A_ℓ = set of restrictions that keeps ℓ variables alive.
- $B_{m,k} = \{\sigma \in A_m : f_\sigma$ can't be represented as a k-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 $u(1), \ldots, u(r)$ 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} x U

- We'll now show that it is possible to recover $\boldsymbol{\sigma}$ from $(\boldsymbol{\sigma} \circ \rho\ , \mathrm{u})$ which will then imply χ is an injection.

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

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

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

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

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

- We'll now show that it is possible to recover $\boldsymbol{\sigma}$ from $(\boldsymbol{\sigma}\circ\rho\ , u)$ which will then imply χ is an injection.
- Obs*. For every $i \in [r]$, the first "unsatisfied" clause in $f_{\boldsymbol{\sigma}\circ\pi(1)\circ\ldots\circ\pi(i-1)\circ\rho(i)\circ\ldots\circ\rho(r)}$ is C_i .
- Recovering $\boldsymbol{\sigma}$ from $(\boldsymbol{\sigma}\circ\rho\ , u)$:

➢ Pick the first unsatisfied clause in $f_{\boldsymbol{\sigma}\circ\rho(1)\circ\ldots\circ\rho(r)}$. This clause is C_1 (Obs*). Now by looking at $u(1)$, we can derive $\pi(1)$.

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

- We'll now show that it is possible to recover $\boldsymbol{\sigma}$ from $(\boldsymbol{\sigma} \circ \rho\ , \mathrm{u})$ which will then imply χ is an injection.
- Obs*. For every $i \in [r]$, the first "unsatisfied" clause in $f_{\boldsymbol{\sigma} \circ \pi(1) \circ \ldots \circ \pi(i-1) \circ \rho(i) \circ \ldots \circ \rho(r)}$ is C_i .
- Recovering $\boldsymbol{\sigma}$ from $(\boldsymbol{\sigma} \circ \rho\ , \mathrm{u})$:

➢ Pick the first unsatisfied clause in $f_{\boldsymbol{\sigma} \circ \rho(1) \circ \ldots \circ \rho(r)}$. This clause is C_1 (Obs*). Now by looking at $u(1)$, we can derive $\pi(1)$. Construct $\boldsymbol{\sigma} \circ \pi(1) \circ \rho(2) \circ \ldots \circ \rho(r)$ from $\boldsymbol{\sigma} \circ \rho(1) \circ \ldots \circ \rho(r)$ and $\pi(1)$.

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

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

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

- We'll now show that it is possible to recover $\boldsymbol{\sigma}$ from $(\boldsymbol{\sigma} \circ \rho\ , u)$ which will then imply χ is an injection.
- Obs*. For every $i \in [r]$, the first "unsatisfied" clause in $f_{\boldsymbol{\sigma} \circ \pi(1) \circ \ldots \circ \pi(i-1) \circ \rho(i) \circ \ldots \circ \rho(r)}$ is C_i.
- Recovering $\boldsymbol{\sigma}$ from $(\boldsymbol{\sigma} \circ \rho\ , u)$:

➢ Continuing like this we can construct $\boldsymbol{\sigma} \circ \pi(1) \circ \ldots \circ \pi(r)$ and also find $\pi(1), \ldots, \pi(r)$ in the process. From here, recovering $\boldsymbol{\sigma}$ is straightforward.

- Ref. https://sites.math.rutgers.edu/~sk1233/courses/topics-S13/lec3.pdf