A Switching Lemma for DNFs over $\mathbb{F}_p$: the Canonical Decision Tree Approach

Abstract

We prove a switching lemma for DNFs over $\mathbb{F}_p$ via the canonical decision tree (CDT) method of Razborov, obtaining the M-independent bound

$$\Pr[\mathrm{DT}(f|_\rho) \geq s] \leq \left(\frac{2pwq}{1-q}\right)^s$$

for any width-w DNF f with *any number of terms* M, where ρ is a random restriction keeping each variable alive with probability q. The key observation is that the p-ary branching of the CDT over $\mathbb{F}_p$ is effectively binary: all $p-1$ nonzero branches lead to the same continuation, so the encoding requires only $2w$ (not pw) bits per step. For $p = 2$ this recovers the classical bound $(4wq/(1-q))^s$. As a corollary, depth-d circuits over $\mathbb{F}_p$ computing generalized parity require size $\exp(\Omega_p(n^{1/(d-1)}))$.

1 Introduction

Håstad's switching lemma [1] is the cornerstone of AC^0 lower bounds. For a width-w CNF or DNF f on $\{0,1\}^n$ and a random restriction ρ_q keeping each variable alive with probability q:

$$\Pr[\mathrm{DT}(f|_\rho) \geq s] \leq (Cwq)^s,$$

where C is an absolute constant and DT denotes decision tree depth. Two features are essential: the bound controls DT depth (not merely certificate complexity), and it is *independent* of the number of clauses M.

Extending this to $\mathbb{F}_p$ is nontrivial. Over $\mathbb{Z}_p^n$ (for a prime p), a literal is $\ell_i(x) = \mathbf{1}[x_i \neq 0]$, and random restrictions assign each variable to: alive with probability q, or dead with a uniform value in $\{0, \ldots, p-1\}$ with probability $(1-q)/p$ each. The recent work [6] proved the single-gate switching lemma $(C_p qK/s)^s$ for fan-in K gates, which already yields tight AC^0 lower bounds.

In this note we prove the full M-independent DNF/CNF switching lemma over $\mathbb{F}_p$, adapting Razborov's canonical decision tree proof [2, 3].

Theorem 1 (Main)**.** *Let $f = T_1 \vee \cdots \vee T_M$ be a width-w DNF over $\mathbb{F}_p^n$ (each term $T_j = \bigwedge_{i \in S_j} \ell_i$ with $|S_j| \leq w$). For $\rho \sim \rho_q$:*

$$\Pr[\mathrm{DT}(f|_\rho) \geq s] \leq \left(\frac{2pwq}{1-q}\right)^s.$$

An identical bound holds for width-w CNFs by duality.

For $p = 2$ and small q, this gives $(4wq)^s$, recovering the classical bound.

Remark 2 (Comparison with Boolean). In the Boolean case ($p = 2$), the constant in the switching lemma is $C \approx 4$–7 depending on the proof method. Our proof gives $C_p = 2p$; for $p = 2$ this is $C_2 = 4$, matching Razborov's original argument. Whether C_p can be improved to $O(1)$ independent of p is an interesting open question.

2 Setup

2.1 Literals and formulas over $\mathbb{F}_p$

We work over $\mathbb{Z}_p^n$ for a prime p. A *literal* is $\ell_i(x) = \mathbf{1}[x_i \neq 0]$. A *width-$w$ DNF* is $f = T_1 \vee \cdots \vee T_M$ where each *term* is $T_j = \bigwedge_{i \in S_j} \ell_i$ with $|S_j| \leq w$. Term T_j evaluates to 1 iff all variables in S_j are nonzero.

2.2 Random restrictions

A *random restriction* $\rho \sim \rho_q$ independently sets each variable x_i to:

- alive (denoted $*$) with probability q;
- dead with value $v \in \{0, \ldots, p-1\}$ with probability $(1-q)/p$ each.

We write $A(\rho)$ for the set of alive variables and $\sigma(\rho)$ for the dead assignment.

2.3 Effect on terms

Under restriction ρ, a term T_j is:

- *falsified*: some variable in S_j is dead with value 0;
- *fully satisfied*: every variable in S_j is dead with a nonzero value;
- *active*: no variable in S_j is dead-0, and at least one variable is alive.

The restricted function $f|_\rho$ depends only on the alive variables.

3 The canonical decision tree over $\mathbb{F}_p$

Definition 3 (CDT for DNFs)**.** *Given a width-w DNF $f = T_1 \vee \cdots \vee T_M$ and a restriction ρ, the* canonical decision tree $\mathcal{T}(f|_\rho)$ *is defined recursively:*

1. *If some term T_j is fully satisfied under ρ: output 1 (leaf, depth 0).*
2. *Otherwise, let T_j be the first active term (smallest j). Query the smallest-index alive variable $x_i \in S_j$.*
   - ***Branch** $x_i = 0$: term T_j is falsified (as are all other terms containing x_i). Recurse on $f|_{\rho'}$ where ρ' extends ρ by setting $x_i = 0$.*
   - ***Branch** $x_i \neq 0$: literal ℓ_i is satisfied. Recurse on $f|_{\rho'}$ where ρ' extends ρ by setting $x_i = 1$ (any nonzero value).*
3. *If no active term exists (all terms falsified): output 0 (leaf, depth 0).*

The depth $\mathrm{DT}(f|_\rho)$ *is the depth of $\mathcal{T}(f|_\rho)$.*

Proposition 4 (Binary effective branching)**.** *In the CDT, the subtrees rooted at branches $x_i = v$ for $v = 1, 2, \ldots, p-1$ are all identical. Hence the CDT is effectively a binary tree.*

Proof. The CDT state depends only on which terms are active and which variables are alive. For a variable x_i in active term T_j:

- Value 0: T_j becomes falsified.
- Any value $v \neq 0$: literal $\ell_i = \mathbf{1}[x_i \neq 0]$ becomes 1. Term T_j either becomes satisfied (if x_i was the last alive variable) or continues with one fewer alive variable. The effect is the same for all $v \neq 0$. $\square$

4 Proof of Theorem 1

We adapt Razborov's encoding argument [2, 3].

Bad restrictions. Let $\mathrm{BAD}_s = \{\rho : \mathrm{DT}(f|_\rho) \geq s\}$. For $\rho \in \mathrm{BAD}_s$, there exists a root-to-node path in $\mathcal{T}(f|_\rho)$ of depth s.

Extracting a path. Fix $\rho \in \mathrm{BAD}_s$. By Proposition 4, the CDT is an effectively binary tree. Choose the *lexicographically first* (left-first) path of depth s in the CDT. This path queries variables $x_{v_1}, \ldots, x_{v_s}$ in terms $T_{\alpha_1}, \ldots, T_{\alpha_s}$, with branch indicators $b_1, \ldots, b_s \in \{0, 1\}$ (where 0 = "value is 0" and 1 = "value is nonzero").

Injection. Define $\Phi(\rho) = (\tilde{\rho}, \tau)$ where:

- $\tilde{\rho}$ agrees with ρ except each x_{v_i} is set to *dead with value* 1 (instead of alive).
- The *encoding* is $\tau = (k_1, b_1, \ldots, k_s, b_s)$, where $k_i = \mathrm{pos}(x_{v_i}, S_{\alpha_i}) \in \{0, \ldots, w-1\}$ is the position of x_{v_i} within term T_{α_i} (in sorted order).

Preservation of falsified terms. We kill to value 1 (nonzero). This does not falsify any term, since falsification requires a dead variable with value 0. Formally: for any term T_j, if T_j is falsified under ρ (has some dead variable with value 0), then T_j is still falsified under $\tilde{\rho}$ (the same dead-0 variable persists). Conversely, if T_j is not falsified under ρ, the only change is that some alive variables become dead-1, which does not create any dead-0 variable.

Hence: *the set of falsified terms is identical under ρ and $\tilde{\rho}$.*

Injectivity. Given $(\tilde{\rho}, \tau)$, we reconstruct ρ iteratively:

1. Initialize $\rho := \tilde{\rho}$. Let $\mathcal{F}_0$ be the set of *base-falsified* terms (those with a dead-0 variable under $\tilde{\rho}$). Initialize the *CDT-falsified* set $\mathcal{F} := \mathcal{F}_0$ and a pointer to the first non-falsified term.
2. For $i = 1, \ldots, s$:
   (a) *Determine the current term.* Let T_{α_i} be the first term (smallest index) not in $\mathcal{F}$.
   (b) *Identify the variable.* The k_i-th variable (sorted order) in T_{α_i} is x_{v_i}. This variable is currently dead with value 1 in ρ.
   (c) *Revive.* Set $\rho(x_{v_i}) := *$ (alive).
   (d) *Update CDT-falsified set.* If $b_i = 0$ (zero branch), add to $\mathcal{F}$ *every* term T_j containing x_{v_i}. If $b_i = 1$ (nonzero branch), $\mathcal{F}$ is unchanged and we remain at term T_{α_i} (the next step will continue querying this term unless it becomes fully satisfied).
3. Output ρ.

This procedure is well-defined because:

- *Step 2(a)*: the base-falsified set $\mathcal{F}_0$ is identical under $\tilde{\rho}$ and ρ, since we only change dead-1 to alive (never creating or removing dead-0 variables). The CDT-falsified terms added in step 2(d) exactly mirror the CDT's own state: on the original ρ, querying x_{v_i} and branching $x_{v_i} = 0$ falsifies every term containing x_{v_i}, not just the current one.
- *Step 2(b)*: the variable at position k_i in the current term is dead-1 (it was set to dead-1 by the forward map and has not been revived in a previous step, since the CDT queries each variable at most once).

Since the procedure uniquely determines ρ from $(\tilde{\rho}, \tau)$, the map Φ is injective on each fiber $\mathrm{BAD}_\tau := \{\rho \in \mathrm{BAD}_s : \Phi(\rho) \text{ has encoding } \tau\}$.

Encoding count.

$$|\{\tau\}| \le (2w)^s,$$

since each step contributes w choices for the position k_i and 2 choices for the branch b_i. Crucially, b_i records only "zero vs. nonzero" (not the specific nonzero value), which suffices by Proposition 4.

Probability ratio. Each x_{v_i} changes from alive (probability q) to dead with value 1 (probability $(1-q)/p$):

$$\frac{\Pr[\rho]}{\Pr[\tilde{\rho}]} = \prod_{i=1}^{s} \frac{q}{(1-q)/p} = \left(\frac{pq}{1-q}\right)^s.$$

Combining.

$$\begin{aligned}
\Pr[\mathrm{BAD}_s] &= \sum_\tau \Pr[\mathrm{BAD}_\tau] = \sum_\tau \sum_{\rho \in \mathrm{BAD}_\tau} \Pr[\rho] \\
&= \sum_\tau \sum_{\rho \in \mathrm{BAD}_\tau} \Pr[\tilde{\rho}] \cdot \left(\frac{pq}{1-q}\right)^s \\
&\le \sum_\tau 1 \cdot \left(\frac{pq}{1-q}\right)^s = (2w)^s \cdot \left(\frac{pq}{1-q}\right)^s = \left(\frac{2pwq}{1-q}\right)^s.
\end{aligned}$$

5 AC^0 lower bound

Corollary 5. *For any prime p and constant depth $d \ge 2$, any depth-d circuit over $\mathbb{F}_p$ computing $\mathrm{PAR}_p(x) = \mathbf{1}[\sum_i x_i \not\equiv 0 \pmod p]$ has size $\exp(\Omega_p(n^{1/(d-1)}))$.*

Proof. The argument is standard. Let C be a depth-d circuit of size S with bottom fan-in w. Apply $d-1$ rounds of random restrictions with $q = c/(pw)$ for a small constant c.

At each round, every bottom gate (a width-w DNF or CNF) satisfies

$$\Pr[\mathrm{DT} \ge s] \le (2pwq/(1-q))^s = (2c/(1-q))^s \le (3c)^s$$

for small q. Setting $s = \lceil C' \ln S \rceil$ for large C', this is $< 1/S^2$. By a union bound over S gates, all gates simplify to DT depth $\le s$ with positive probability.

Each gate is replaced by its depth-s decision tree expansion, reducing the circuit depth by 1 and setting the new bottom fan-in to $s = O(\ln S)$. After $d-1$ rounds:

$$n' = n \cdot \prod_{i=1}^{d-1} q_i \ge \frac{n}{(Cp)^{d-1} \cdot w \cdot (\ln S)^{d-2}}$$

alive variables survive. For PAR_p to remain non-constant, we need $n' \ge 1$, giving $w \cdot (\ln S)^{d-2} \le n/(Cp)^{d-1}$. Since $w \le S$, this forces $S \ge \exp(\Omega_p(n^{1/(d-1)}))$. $\square$

6 Discussion

6.1 The key observation: binary effective branching

The CDT for a DNF over $\mathbb{F}_p$ branches p ways at each query, but $p-1$ of these branches are identical (all nonzero values have the same effect on clause satisfaction). This is the crucial observation: the encoding needs only 1 bit per step to specify the branch (zero vs. nonzero), rather than $\log_2 p$ bits.

In contrast, for more general "$\mathbb{F}_p$-valued" gate types (e.g., functions that distinguish between different nonzero values), the p-ary branching would be genuinely p-ary, and the encoding would require $\log_2 p$ bits per step. This would give a bound of $(p^2 wq/(1-q))^s$.

6.2 Killing to value 1

The choice to kill staircase variables to dead-1 (any fixed nonzero value) is essential. Killing to value 0 (as in the injection of [7]) preserves the set of *surviving* clauses, but the CDT structure changes because the active terms differ. Killing to a nonzero value preserves the set of *falsified* terms (since falsification requires a dead-0 variable), which is what the CDT-based inverse needs.

6.3 Open problems

1. **Optimal constant.** Our bound gives $C_p = 2p$. For $p = 2$, this is $C_2 = 4$, matching Razborov's proof. Can C_p be made $O(1)$? In the Boolean case, Razborov's bound of 4 is not tight; improved bounds give $C_2 \leq 7/2$ [4]. The *correct* constant over $\mathbb{F}_p$ is likely $O(p)$ since the probability ratio $pq/(1-q)$ inherently involves p.

2. **Projections and depth hierarchy.** RST [5] extended the switching lemma to random *projections* to prove average-case depth hierarchy for the Boolean PH relative to a random oracle. Extending this to $\mathbb{F}_p$ would give analogous results for $\mathbb{F}_p$-valued computation models.

3. **$\mathbf{ACC^0}$.** Our switching lemma handles AND/OR gates over $\mathbb{F}_p$. The frontier of AC^0-style lower bounds is ACC^0 (circuits with MOD_m gates). Connecting the $\mathbb{F}_p$ framework to ACC^0 lower bounds remains a major open problem.

References

[1] J. Håstad, *Almost optimal lower bounds for small depth circuits*, in Proc. 18th STOC, pp. 6–20, 1986.

[2] A. Razborov, Personal communication; see Beame [3] for exposition.

[3] P. Beame, *A switching lemma primer*, Technical Report UW-CSE-95-07-01, University of Washington, 1994.

[4] P. Beame, *Improved switching lemma bounds*, unpublished.

[5] B. Rossman, R. A. Servedio, L.-Y. Tan, *An average-case depth hierarchy theorem for Boolean circuits*, in Proc. 56th FOCS, pp. 1030–1048, 2015.

[6] Y. Wang, *A Fourier-analytic switching lemma over $\mathbb{F}_p$ and the AC^0 lower bound for generalized parity*, ECCC Report TR26-014, 2026.

[7] Y. Wang, *Switching lemmas over $\mathbb{F}_p$ and tight AC^0 lower bounds*, Preprint, 2026.