Proving Lower Bounds on Circuits Using Projection Switching Lemma

M.Arunothia, 13378

Sai Kishan Pampana, 13458

Submitted to Prof. Raghunath Tewari for partial fulfilment of the course requirements for CS498A, IITK

Abstract

Projection switching lemma is a varied version of the general switching lemma that uses random projections to give decision tree depth bound of an r-DNF. In our earlier work we had studied Hastad's presentation of switching lemma that used random restriction to show a bounded conversion from a CNF to a DNF and we also saw how it helped in proving lower bounds on circuit complexity problems (like parity function). In this work, we understand the proof of projection switching lemma while contrasting it with the simple switching lemma. We also understand how the introduction of random projections help in proving circuit size lower bound for the graph connectivity problem.

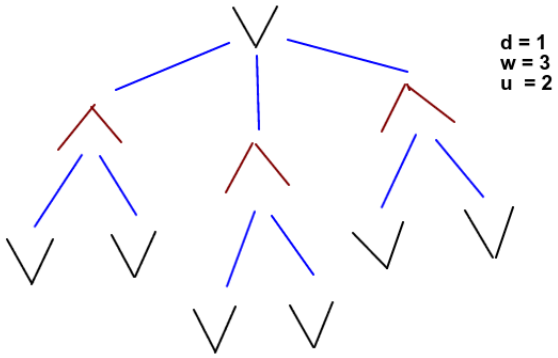
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Contents

1 Basics

1.1 $SkewedSipser_{u,d}$

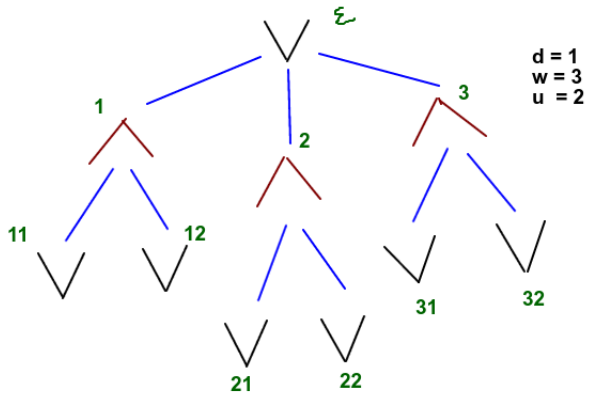
- The circuit has $2d+1$ levels of alternating AND and OR gates.
- The root level and leaf level are that of OR gates.
- AND gates have a fan-in of u.
- OR gates have a fan-in of w but for the leave-level where it has a fan-in of $w^{33/100}$.



1.2 $SkewedSipser_{u,d}$ Addressing

The addressing starts from the root which is addressed as ϵ. The ith child of a node is addressed by it's parent's address concatenated with i.

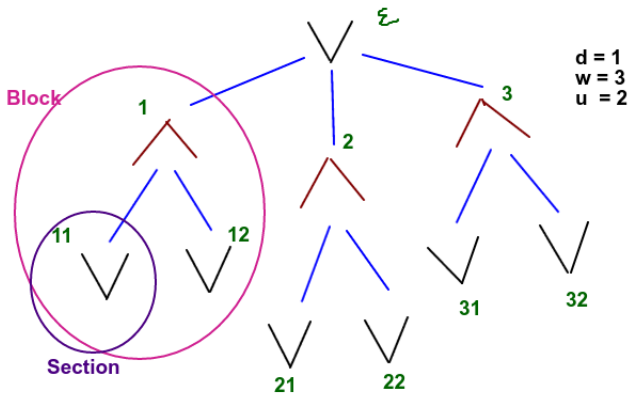
$$A(d) := (b_0, a_1, b_1, \cdots, a_d, b_d) : a_i \in [u], b_0, \cdots, b_{d-1} \in [w], b_d \in [w^{33/100}]$$



1.3 Addressing using blocks and sections

We will refer to the set of $uw^{33/100}$ addresses of variables below an $AND^{(1)}$ gate as a block, and the set of $w^{33/100}$ addresses of variables below an $OR^{(1)}$ gate as a section.

$$A(d) = B(d) \text{ X } A', \text{ where}$$
$$B(d) = (b_0, a_1, b_1, \cdots, a_{d-1}, b_{d-1}) : a_i \in [u], b_i \in [w], A = [u] \text{ X } [w^{33/100}]$$
$$A(d, \beta) = (\beta, \tau) : \tau \in A' \text{ and } A(d, \beta, a) = (\beta, a, b) : b \in [w^{33/100}]$$



1.4 Random Projections

1.4.1 Projection Operators

Given a restriction $\rho \in \{0,1,*\}^{A(d)}$, the projection operator $proj_\rho$ maps a function $f : \{0,1\}^{A(d)} \to \{0,1\}$ to a function $proj_rho(f) : \{0,1\}^{B(d)} \to \{0,1\}$, where $proj_\rho(f)(y) = f(x)$, where $x_{\beta,\tau} = y_\beta$ if $\rho_{\beta,\tau} = *$ otherwise it is equal to $\rho_{\beta,\tau}$.

1.5 Distribution $D_u(q)$

A draw of ρ from $D_u(q)$ gives a restriction to a block A' that $\in \{0,1,*\}^{A'}$

- With probability q, ρ is $\{*\}^{A'}$.
- With probability $1-q$, ρ is such that, exactly one section is filled with 0's (or 1's) while the rest are filled with 1's (or 0's).

1.6 Distribution $D_u^{(d)}(q)$

Consider an independent $D_u(q)$ for each $\beta(d)$ block.

2 Projection Switching Lemma [1]

Theorem 1. *For $2 \le u \le w$, let F be an r-DNF over the variables $\{x_{\beta,\tau}\}$, $(\beta, \tau\ ) \in A(d)$, where $r \le u-1$. Then for all $s \ge 1$ and $q \in (0,1)$, we have*

$$Pr_{p \leftarrow D_u^{(d)}(q)}[|Proj_p F| \ge s] \le \left(\frac{8urq}{1-q}\right)^s$$

where $|X|$ denotes the decision tree depth of X.

Proof. To prove switching lemma, the author's use the following algorithm to find the decision tree depth of $Proj_\rho(F)$

- If $Proj_\rho(F) \equiv 0$ or 1, output 0 or 1, respectively.
- Otherwise, let T be the first term in F such that T_ρ is non-constant and $T_{\rho\rho'} \equiv 1$ for some $\rho' \in supp(D_u^{(d)}(q))$.
  - There should always be a term such that T_ρ is non-constant because otherwise, first step would have executed already.
  - Also, $T_{\rho\rho'} \equiv 1$ for some $\rho' \in supp(D_u^{(d)}(q))$ should definitely happen for atleast one term T because, otherwise this would lead to the presence of both y_β and $\neg y_\beta$ under the same section, which leads to a contradiction to the definition of random projection.
  - Hence, such a choice is always feasible.
- Define

$$\eta = \{\beta \in B(d) : x_{\beta,\tau} \text{ or } \neg x_{\beta,\tau} \text{ occurs in } T_\rho \text{ for some } \tau\}.$$

- We enumerate every possible assignment to these y_β's and make a recursive call for enumeration.

It can be seen from this algorithm that the number of recursive calls (depth-wise) gives us the depth of the decision tree required.

We define the set B as follows -

Definition 1. $B = \{\rho \in supp(D_u^{(d)}) : \textit{decision tree depth of } CanonicalDT(F, \rho)\} \geq s$

Notice that bound on the size of set B is what is required. To obtain this bound, we define a θ function.

$$\theta : B \to \{0, 1, *\}^{A(d)} \text{ X } \{0, 1\}^s \text{ X } \{0, 1\}^{s(log(r)+1)}$$

which has the following two properties -

- Injection.
- Weight Increase.

  $$\frac{Pr[X=\theta_1(\rho)]}{Pr[X=\rho]} \geq \alpha$$

  for any desired α

We should prove the existence of at least one such θ function. Hence, consider the following construction using the *canonicalDT* function.

- $\theta_1(\rho) = \rho\sigma^{(1)} \cdots \sigma^{(s)} \in \{0, 1, *\}^{A(d)}$.
  - Here $\sigma^{(i)}$'s are the restrictions applied on the ith term in the algorithm.
  - It can hence be seen that ρ and $\rho\sigma$ differ in exactly s many blocks, with the former having all $*$s in them while the latter having 0/1s in them.
  - Hence, we get **weight increase condition** satisfied with $\alpha = \frac{1-q}{2qu}$
- $\theta_2(\rho)$: In the canonical decision tree, we consider the first path that crosses depth s. We take π along this path and truncate it at length s. Then $\theta_2(\rho)$ is defined to be $binary(\pi) \in \{0, 1\}^s$
- $\theta_3(\rho) = encode(\eta_1) \odot \cdots \odot encode(\eta_s) \in \{0, 1\}^{s(1+logr)}$ where, $encode(\eta_i) = location(\beta_1) \odot 0 \odot location(\beta_2) \odot 0 \cdots location(\beta_t) \odot 1 \in \{0, 1\}^{|\eta_i|(1+logr)}$
  - Basically, we encode the full details about η_i by this method, which will help establish the property of injection.

Notice, that **Injection** property is also satisfied as the construction is unique to a given ρ.

Once, we have shown that such a θ function exist, we use it to prove the bound on set **B**.

$$\text{Let } B_O = \{\rho \in B : \theta_2(\rho), \theta_3(\rho) = O\} \in B$$

We can bound the size of B_O using the weight increase and one-one property of the θ function

$$\begin{aligned} Pr[X \in B_O] &= \textstyle\sum_{\rho \in B_O} Pr[X = \rho] \\ &\leq (1/\alpha) * \textstyle\sum_{\rho \in B_O} Pr[X = \theta_1(\rho)] \leq 1/\alpha. \end{aligned}$$

Now, we use the definition of B_O to bound the size of B

$$Pr[X \in B] = \textstyle\sum_O Pr[X \in B_O] \leq 2^s * (2r)^s * (\alpha)^s$$

Hence, this completes the proof of Projection Switching Lemma. □

The following are some notable observations made from the proof

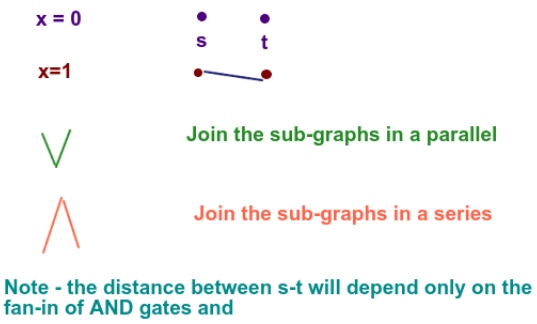
- The proof heavily depends on the introduction of the θ function.
- Once this is done, we see that the problem has been converted from bounding **Set B** to bounding **Set** B_O.
- Bounding **Set** B_O is aided by the property of the θ function being used as we now know the probability of being as $\theta_1(\rho)$ to that of being as ρ itself that too in terms of q.
- Hence, we are able to bound the size of **Set B**.

3 Graph-Connectivity Problem

Definition 2. *Any depth-d circuit for determining whether an n-node graph has an $s-t$ path of length at most k must have size $n^{\Omega(k^{1/d}/d)}$*

3.1 Connection with *SkewedSipser* Problem

Here we will see a reduction from the SkewedSipser to that of a graph connectivity. The idea here is that we take the SkewedSipser$_{u,d}$ and create a graph with a start node s and an end node t. We say that there is a path from s to t in the created graph iff the SkewedSipder evaluate to 1.
The multi-graph construction is as given below -



From these observations we obtain the following connection between SkewedSipser$_{u,d}$ and small-distance connectivity, which is key to our lower bound:

Lemma 1. *The multi-graph $G(SkewedSipser_{u,d}, z)$ contains an $s-to-t$ path of length at most u^d if and only if $SkewedSipser_{u,d}(z) = 1$*

Lemma 2. *Every shortest path from s to t in the multi-graph has length exactly u^d.*

4 Theorems Involved

We will mainly be working with two theorems
Theorem 1:For any $k(n) \leq n^{1/5}$ and any $d = d(n)$, any depth-d circuit computing $STCONN(k(n))$ must have size of $n^{\Omega(k^{1/d}/d)}$. Furthermore, for any $k(n) \leq n$ and any $d = d(n)$, any depth-d circuit computing $STCONN(k(n))$ must have size $n^{\Omega(k^{1/5d}/d)}$

The above theorem is the main theorem that we need to prove, but doing so is not so easy so will be using a different theorem with the help of which we will prove the theorem 1.

Theorem-2: Let $d(w) \geq 1$ and $2 \leq u(w) \leq w^{33/100}$, where $w \to \infty$. Then any depth-d circuit computing $skewedSipser_{u,d}$ has size at least $w^{\Omega(u)} = n^{\Omega(u/d)}$

4.1 Proof of Theorem 1

Our main aim is to prove the **Theorem 1**, but to do this we take the help of **Theorem 2**. We first show that we can prove theorem 1 if we assume that theorem 2 is true and then we will later give a short proof of theorem 2. Here we will give a short intution as to why we will be able to solve the theorem 1 using 2.
So the underlying procedure is:

- Consider the cases of $d \leq 2log(k)/log(log(k))$ and $(k/2)^{1/d} \geq 2$.
- Construct a **newSkewedSipser** function with suitable parameters such that this when put under certain restriction reduces to the skewedSipser function that we defined previously.
- By constructing the newSkewedSipser function we now say that any lower bounds that are applicable to the skewedSipser function will also be applicable to the newSkewedSipser function.

- This way we get the required bound.

The newSkewedSiper function will be having 2d+2 layers of alternating AND and OR gates. The last gate is an AND gate with a fan-in of 2. We construct the new Sipser function by defining the parameter of the Sipser function u, w in terms of the size n and the valus of k.

Now proof of Theorem 1 will be done once we are able to prove the Theorem 2

4.2 Proof of Theorem 2

The proof majorly depends on the following lemma -

Lemma 3. *For every* $1 \leq l \leq d$*, we have that* $Proj_\rho(SkewedSipser_{u,l})$ *contains* $SkewedSipser_{u,l-1}$ *as a subfunction with probability at least* 0.9 *over a random restriction* $\rho \leftarrow D_u^{(l)} q$*.*

- The underlying factor here is the way we define random projections, by this we mean that we draw a random projection for the skewedSipser by independently drawing restriction for each block of the function.
- So now if we want SkewedSipser$_{u,l-1}$ to be a subfunction of SkewedSipser$_{u,l}$, then for the OR$^{(2)}$ gate pout of the w blocks at least w$^{33/100}$ retain their structure, this can be done by choosing the appropriate q

The proof is based on contradiction. First we assume there is a depth-d circuit C of size at most S that computes $SkewedSipser_{u,d}$. C without loss of generality is alternating and levelled.

- Hard function retains structure - For $1 \leq l \leq d-2$, proj(SkewedSipser$_{u,l}$) contains SkewedSipser$_{u,l-1}$ as a subfunction, and we can hence have a deterministic trimming that can give us exactly SkewedSipser$_{u,l-1}$ from the projection
- Circuit collapses - For $1 \leq l \leq d-2$ proj(C$_{l+1}$ has depth l, bottom fan-in u-1, and has at most S gates at distance at least 2 from the inputs. Also we can say that Cl is a simplified version of the projection after the deterministic trimming associated with proj(SkewedSipser$_{u,l}$)

The properties imply that C computes $SkewedSipser_{u,l-1}$ for all $1 \leq l \leq d-2$. This yields the desired contradiction since C_1 , a decision tree of depth at most $u-1$, cannot compute $SkewedSipser_{u,0}$.

5 Conclusion

In the previous few sections we have shown how we can use the projection switching lemma to bound the size of a circuit that is computing the STCONN(k(n)), apart form this the projection switching lemma can also be used to bound some other properties of the circuit. In the above discussion we come to see the importance of random projections, the problem of STCONN could not have been solved by simply using random restriction. Here random projections help to maintain the complex structure of the SkewedSipser which helps us to develope the contradiction that we needed.

6 Further Extensions of this Project

We can try to apply projection switching lemma to the following graph problems and see how it works for them.

- Maximal Matching Problem.
- Red-Blue Path Problem.
  - Given a graph G whose edges are colored either red or blue and two fixed vertices s and t in G, is there a path from s to t in G that alternates between red and blue edges.

We can work on experimenting with further variations of switching lemma, like

- what happens if we include MOD gates?
- What other kind of restrictions can be thought of?
- Why only random projection worked in this case? Is there anything else that might work?

References

[1] Xi Chen, Igor C. Oliveira, Rocco A. Servedio, and Li-Yang Tan. Near-optimal small-depth lower bounds for small distance connectivity. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 612–625, New York, NY, USA, 2016. ACM.

[2] J Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, pages 6–20, New York, NY, USA, 1986. ACM.

[3] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[4] Benjamin Rossman. The average sensitivity of bounded-depth formulas. *CoRR*, abs/1508.07677, 2015.