CS 2429 - Propositional Proof Complexity

Lecture #9: 7 November 2002

Lecturer: Toniann Pitassi

Scribe Notes by: Sheikh M. N. Alam

1 Introduction

In this lecture we will begin to prove exponential lower bounds for bounded-depth Frege proofs (also called AC_0-Frege proofs) of the propositional pigeonhole principle. That is, we will prove the following theorem.

Theorem 1. *Any AC_0-Frege proof of PHP_n^{n+1} requires exponential size.*

There are many equivalent definitions of a depth-k Frege system. We will focus on one particular one for the purposes of this lower bound, although the lower bound applies to a general class of depth-k Frege systems over the standard DeMorgan basis (AND, OR and NOT). In our Frege system, formulas will be restricted to the connectives $\vee$ and $\neg$. The $\vee$ connective is fan-in two, but the depth is defined so that the depth does not increase unless we switch connective type. In particular, the *depth* of a formula (or circuit) f is defined as follows:

Definition If f has no connectives then its depth is 0, otherwise depth of f is the maximum number of alternation of connectives along any path from root to leaf plus 1.

Formulas or circuits in standard CNF or DNF form, for instance, have depth 2. Our Frege system will be Shoenfields system. There is one axiom, $\neg A \vee A$ for any formula A, and four rules: (1) Expansion rule: from p, the formula $q \vee p$ can be derived; (2) Contraction rule: from $p \vee p$, the formula p can be derived; (3) Associative rule: from $p \vee (q \vee r)$, the formula $(p \vee q) \vee r$ can be derived; and (4) The Cut rule: from $p \vee q$ and $\neg p \vee r$, the formula $q \vee r$ can be derived.

If Γ is a sequence of formulas, then the *size* of Γ is the number of distinct subformulas in Γ. The depth of a proof in our Frege system is the maximum depth of a line in the proof.

2 Overview

An overview of the proof is as follows. Assume for sake of contradiction that $\mathcal{P}$ is a small Frege proof of the pigeonhole principle of depth d. We will apply a sequence of restrictions to all of the subformulas in the proof, and an associated interpretation of each of the subformulas under the restriction, so as to obtain an evaluation of each formula in the proof. The evaluation of each subformula will be a matching decision tree of small height. A key property of these decision trees will be that they are all 1-trees. But on the other hand, the decision tree that will be associated with the pigeonhole principle will always be a 0-tree, and hence we obtain a contradiction.

The above very brief overview combines ideas from circuit complexity and model/proof theory. From circuit complexity, we use the idea of applying a restriction and an associated switching lemma in order to simplify the formula. From model/proof theory, we apply the idea of interpreting each formula in a local fashion that is consistent with the negation of the pigeonhole principle.

In order to explain the switching lemma part of the argument, we will begin by introducing the restriction method, and first see how to apply it in the much simpler context of proving lower bounds for bounded-depth circuits. In the subsequent lecture, we will tackle the proof theory part of the argument.

3 The Restriction Method

The restriction method is used in both circuit complexity and proof complexity for proving lower bounds.

Definition A *restriction* ρ is a partial assignment of values to a set of boolean variables $\{x_1, x_2, ..., x_n\}$, i.e. $\rho : \{x_1, x_2, ..., x_n\} \rightarrow \{0, 1, *\}$ where $\rho(x_i) = *$ indicates that the variable x_i is not assigned any value by this restriction.

When we apply a restriction ρ to a boolean function f we get a boolean function $f\lceil_\rho$ which is the result of substituting $\rho(x_i)$ for x_i for all places where $\rho(x_i) \neq *$. We say that all variables x_i such that $\rho(x_i) = *$ are *unset* and obviously the resulting function becomes a function of the unset variables.

Restrictions simplify formulas, circuits, or functions that we have. The simplification we obtain by restricting a small set of variables is typically substantially more than the number of variables we set. For example, given $f = (\bigvee_i x_i \vee \bigvee_j \neg x_j)$, a single assignment $\rho(x_i) = 1$ or $\rho(x_j) = 0$ makes $f\lceil_\rho$ a constant. To prove small circuits C cannot compute a complex function f, we show that there is a restriction ρ such that $f\lceil_\rho$ is still complicated but $C\lceil_\rho$ is so simple that it obviously cannot compute $f\lceil_\rho$. In this way we can prove a lower bound.

3.1 Decision Trees

When using restriction method, we will associate a decision tree with each gate of a circuit or each formula appearing in a proof.

Definition A *decision tree* T over $x_1, ...x_n$ is a binary tree such that

1. each internal node of T is labelled with some variable x_i,
2. edges out of a node x_i are labelled by $x_i = 0$ or $x_i = 1$,
3. no two nodes on a path have the same variable label, and
4. leaf nodes are labelled 0 or 1.

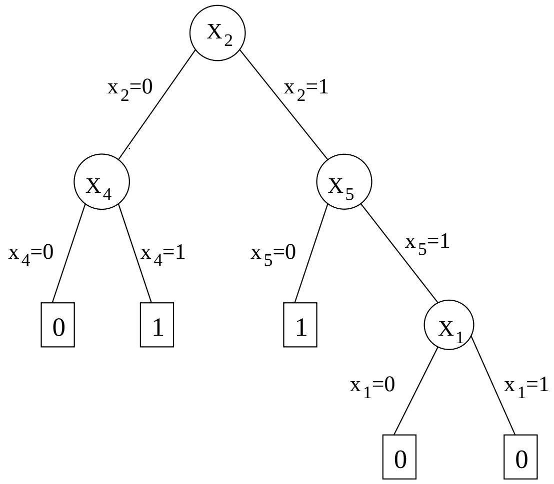

Figure 1: Decision tree for function f.

Each root to leaf path of a decision tree defines a partial restriction ρ to $\{x_1, ..., x_n\}$. More precisely, for $v \in \{0, 1\}$, $x_i \leftarrow v$ is in ρ iff on that root to leaf path, the out edge labelled $x_i = v$ is taken.

Definition Depth of a decision tree is the height of the tree.

Definition A decision tree T over $\{x_1, ..., x_n\}$ computes a boolean function f of $\{x_1, ..., x_n\}$ iff for every root to leaf path (or branch) B of T, the restriction ρ corresponding to branch B has the property that $f\lceil_\rho$ equals the leaf label of B.

Definition A t-DNF formula is the disjunction of terms having maximum term size t (number of literals in the term is at most t). A t-CNF formula is the conjunction of clauses having maximum size t.

Decision trees give a natural way of describing the function they compute as a CNF or DNF formula. If a boolean function f can be represented/computed by a height h decision tree, then f can be represented by an h-DNF by associating a term with each branch with leaf label 1 and also by an h-CNF formula by associating a clause with each branch with leaf label 0.

Example The function f in Figure 1 can be represented in DNF as $f = \bar{x_2}x_4 \vee x_2\bar{x_5}$ and in CNF as $f = (x_2 \vee x_4) \wedge (\bar{x_2} \vee \bar{x_5} \vee x_1) \wedge (\bar{x_2} \vee \bar{x_5} \vee \bar{x_1})$.

3.2 Lower Bound for Parity

The parity function can be defined as follows:

$$Parity(x_1, ..., x_n) = 1 \text{ if } (x_1, ..., x_n) \bmod 2 = 1$$

To prove the lower bound for *Parity*, we will proceed in the following way. We want to show that no circuit having size at most s and depth at most d computes *Parity*. Here, our circuits are assumed to be over the connectives $\vee$ and $\neg$ although now $\vee$ as having unbounded fanin, and depth will be defined in the usual way. We will prove that by contradiction. Fix a circuit S of size s and depth d.

1. At first note the following important property of $Parity$, for any restriction ρ, $Parity\lceil_\rho$ is either parity or its negation on the variables that are still not assigned a value, i.e.,

$$[Parity(x_1, ..., x_n)]\lceil_\rho = Parity(x_{i_1}, ..., x_{i_{n^\epsilon}}) \text{ or } \neg Parity(x_{i_1}, ..., x_{i_{n^\epsilon}})$$

   where $x_{i_1}, ..., x_{i_{n^\epsilon}}$ are variables left unset by ρ.

2. Then show there exists a restriction $\rho = \rho_1\rho_2 \cdots \rho_{d-1}$ such that the number of variables left unset by ρ is at least n^ϵ for some ϵ.

3. By Switching Lemma, which we will discuss shortly, $S\lceil_\rho$ can be represented by a simple circuit, i.e., by a t-DNF formula where $t \ll n^\epsilon$. This contradicts the fact that any DNF formula computing parity (or the negation of parity) of n bits has to have terms of $size > n - 1$.

Therefore, no such small circuit S exists.

To find restrictions for parity, we start at the inputs of the circuit and work upwards one layer at a time. As we go along, we maintain a current restriction ρ_i and a decision tree $T_i(g)$ for each gate g in the first i layers such that $T_i(g)$ computes $g\lceil_{\rho_i}$.

For layer 0, the gates are input variables, ρ_0 is empty and all decision trees have height 1. As we move up from layer $i-1$ to layer i, any new gate h is either a negation or an OR. If $h = \neg g$, we let $T_i(h)$ be $T_i(g)$ with the labels on its leaves flipped from 0 to 1 and vice versa. The case when $h = (g_1 \vee ... \vee g_l)$ is more complex. It might happen that $h\lceil_{\rho_i}$ requires tall decision trees even if $T_i(g_j)$ are short. We therefore look for a further small restriction π to the inputs in the hope of simplifying $h\lceil_{\rho_i}$ so that we might get a shorter tree. We would like to choose *one* π that simultaneosly does this for *all* unbounded fan-in OR's in the i-th layer (or which there are at most S).

We will set $\rho_{i+1} = \rho_i\pi$ and by our assumed properties of π, short $T_{i+1}(h)$ exist for all gates h in this layer. For all gates g below this layer, we will set $T_{i+1}(g) = T_i(g)\lceil_\pi$. We now continue upward in the normal fashion and end by setting $\rho = \rho_d$ for the depth d circuit. Since we have been choosing π's which gurantee short trees, if the circuit is small, the tree we end up with will be shorter than the number of inputs that ρ leaves unset. By our earlier observation about restrictions of parity, such a decision tree must be incorrect. This yields the lower bound.

Now we need to show how to get that restriction π. By Hastad Switching lemma such a restriction can be found if the depth is limited. Using standard probabilistic method we can show the existence of one such π. The idea is to choose a random small π and prove that the probability that it fails to shorten the decision tree for any single OR gate h is less than $1/S$. Now, There are at most S OR gates in any layer. So the probability that there exists an OR gate in this layer which is not shortened by π is strictly less than 1. So we conclude that there must exist a small π that works.

Thus it was shown by Hastad, that any depth-d circuit for parity has exponential size. Using the same argument with different parameter settings, the following theorem was also proven.

Theorem 2. *Polynomial-size circuits for Parity require* $\Omega(\log n/\log\log n)$ *depth.*

So if the depth is greater than log depth, then it is necessary to blow up the size of the circuit.

4 Switching Lemma

In this section we will state and prove the Hastad Switching Lemma, the key ingredient in the proof outlined above, for showing that the Parity function cannot be computed with polynomial-size bounded-depth circuits.

Let $\mathcal{R}_n^l$ to be the set of all restrictions ρ on a domain of n variables that have exactly l unset variables.

Hastad's switching lemma states that for any fixed r-DNF f, the probability that for a restriciton $\rho \in \mathcal{R}_n^l$, $f\lceil_\rho$ does not have a height s decision tree representing it is *small*.

$$\left[small \sim \left[\frac{lr}{n}\right]^s < \frac{1}{s}, \;\; s \sim n^{o(1)} \right]$$

Fix some r-DNF f and fix restriction $\rho \in \mathcal{R}_n^l$. A restriction ρ is applied to f in order, so that $f\lceil_\rho$ is the DNF formula whose terms consist of those terms of f that are not falsified by ρ, each shortened by removing any variables that are satisfied by ρ, and taken in the order of occurrence of the original terms on which they are based.

A *canonical decision tree* for $f\lceil_\rho$, $T\left(f\lceil_\rho\right)$ is as follows:

1. if $f\lceil_\rho$ is the constant function 0 or 1 (contains no term or has an empty first term, respectively) then the corresponding decision tree consists of a single leaf mode labelled by the appropriate constant value.

2. If the first term C_1 of $f\lceil_\rho$ is not empty then let $f'\lceil_\rho$ be the remainder of $f\lceil_\rho$ so that $f\lceil_\rho = C_1 \vee f'\lceil_\rho$. Let K be the set of variables appearing in C_1. The decision tree starts with a complete binary tree for K, which queries the variables in K in the order induced by the order of the indices. Each leaf i in the tree is associated with a restriction $\rho\sigma_i$ which sets the variables of K according to the path from the root to i. For each $\rho\sigma_i$ we replace the leaf node , i, by the subtree corresponding to $f\lceil_{\rho\sigma_i}$. (Note that for the unique $\rho\sigma_i$ which satisfies C_1 the leaf i will remain a leaf and be labelled 1. For all other choices of $\rho\sigma_i$, the tree that replaces i is the tree corresponding to $f\lceil_{\rho\sigma_i}$ which is same as the tree corresponding to $f'\lceil_{\rho\sigma_i}$.

Example Let $f = x_1x_2 \vee x_5x_7 \vee x_3x_4 \vee x_6x_5$ and the restriction ρ is $x_1 = 0$, $x_3 = 1$. Then $f\lceil_\rho = x_5x_7 \vee x_4 \vee x_6x_5$. The corresponding canonical decision tree for $f\lceil_\rho$ is shown in Figure 2.

We'll show that for any DNF formula f, for an appropriately chosen restriction ρ, the height of $T\left(f\lceil_\rho\right)$, $\left|T\left(f\lceil_\rho\right)\right|$, is small with high probability. This lemma is a switching lemma due to Hastad because it will allow us to obtain a DNF formula with short terms for $\neg f\lceil_\rho$ by taking the terms corresponding to the paths in $T\left(f\lceil_\rho\right)$ that have leaf labels 0.

Lemma 3 (Hastad's Switching Lemma). *Let f be a DNF formula in n variables with terms of length at most r (r-DNF). For $s \geq 0$, $l = pn$, and $p \leq 1/7$,*

$$\frac{\left|\left\{\rho \in \mathcal{R}_n^l \;:\; |T(f\lceil_\rho)| \geq s\right\}\right|}{|\mathcal{R}_n^l|} < (7pr)^s.$$

Before giving the proof of the switching lemma we give the following definition.

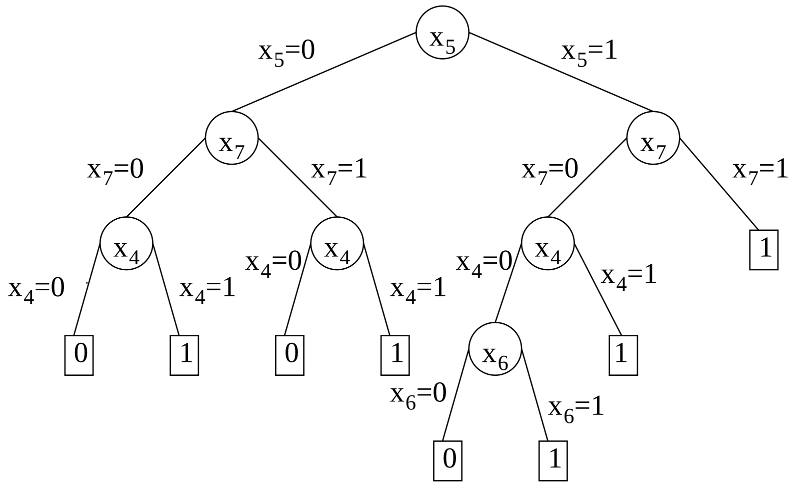

Figure 2: Canonical decision tree for $f\lceil_\rho = x_5x_7 \vee x_4 \vee x_6x_5$.

Definition *stars(r,s)* is the set of all sequences $\beta = (\beta_1, ..., \beta_k)$ such that for each j, $\beta_j \in \{*, -\}^r \backslash \{-\}^r$ and such that the total number of $*$'s in all the β_j is s.

Lemma 4. $|stars(r,s)| < (r/\ln 2)^s$.

Proof For convenience in the proof we shall include the empty string in $stars(r,0)$ which would otherwise be empty. It is sufficient to show that $|stars(r,s)| \le \gamma^s$ for $(1+1/\gamma)^r = 2$ because we have,

$$\begin{aligned}
\ln(1+1/\gamma) &= \frac{\ln 2}{r} \\
\text{i.e, } 1+1/\gamma = e^{\frac{\ln 2}{r}} &< e^\gamma \qquad \text{as } 1+x < e^x \text{ for } x \neq 0 \\
\text{i.e, } \frac{\ln 2}{r} &< \gamma
\end{aligned}$$

Induction on s. The base case $s = 0$ follows trivially. Now suppose that $s > 0$. It is easy to see from the definition that for any $\beta \in starts(r,s)$, if β_1 has $i \le s$ *'s then $\beta = (\beta_1, \beta')$ where $\beta' \in stars(r, s-i)$. (For $i = s$ we have used our augmentation of $stars(r,0)$.) There are $\binom{r}{i}$ choices of β_1 so

$$\begin{aligned}
|stars(r,s)| &= \sum_{i=1}^{min(r,s)} \binom{r}{i} |stars(r, s-i)| \\
&\le \sum_{i=1}^{r} \binom{r}{i} \gamma^{s-i} \\
&= \gamma^s \sum_{i=1}^{r} \binom{r}{i} (1/\gamma)^i \\
&= \gamma^s[(1+1/\gamma)^r - 1] \\
&= \gamma^s
\end{aligned}$$

by the inductive hypothesis and the definition of γ. $\square$

Proof: (Hastad Switching Lemma) We only need to consider $s > 0$. Let $S \in \mathcal{R}_n^l$ be the set of restrictions ρ such that $|T(f\lceil_\rho)| \geq s$ i.e., S is the set of bad restricitons for f under which the input DNF formula f is not sufficiently simplified. We'll show that a bad restriction can be mapped to an element of a small set in such a way that knowledge of the formula permits one to reconstruct the original bad restriction from the image of this map and thus the number of such bad restrictions is small. We'll show that $|S|$ is small (so $\frac{|S|}{|\mathcal{R}_n^l|}$) by constructing a 1-1 map from S to B $-$ set of all string of a fixed size where $|B| \ll |\mathcal{R}_n^l|$. More precisely we define a 1-1 map

$$S \;\rightarrow\; \mathcal{R}_n^{l-s} \;\times\; stars(r,s) \;\times\; 2^s.$$

Let $f = C_1 \vee C_2 \vee C_3 ....$ Suppose that $\rho \in S$ and let π be the restriction associated with the lexicographically first path in $T(f\lceil_\rho)$ that has length $\geq s$ (any way of canonically associated such a long path will do.) Trim the last few varibales set in π along the path from the root so that $|\pi| = s$. We use formula f and π to determine the image of ρ. The image of ρ is defined by following the path π in the canonical decision tree for $f\lceil_\rho$ and using the structure of that tree (see Figure 3).

Let C_{i_1} be the first term of f that is not set to 0 by ρ. Then $C_{i_1}\lceil_\rho$ will be the first term in $f\lceil_\rho$. Since $|\pi| > 0$, such a term must exist and will not be the empty term. Let K be the set of variables in $C_{i_1}\lceil_\rho$ and let σ_1 be the unique restriction of the variables in K that satisfies $C_{i_1}\lceil_\rho$. Let π_1 be the portion of π that sets the variables in K. We have two cases based on whether or not $\pi_1 = \pi$.

1. If $\pi_1 \neq \pi$ then by the construction of π, π_1 sets the variables in K. Note also that $C_{i_1}\lceil_{\rho\sigma_1} = 1$ but since $\pi_1 \neq \pi$, $\pi_1 \neq \sigma_1$, and thus $C_{i_1}\lceil_{\rho\pi_1} = 0$.
2. if $\pi_1 = \pi$ then it is possible that π does not set all of the variables in K. In this case we shorten σ_1 to the variables in K that appear in π_1. Now all we know is that $C_{i_1}\lceil_{\rho\sigma_1} \neq 0$.

Define $\beta_1 \in \{*, -\}^k$ based on the fixed ordering of the variables in term C_{i_1} by letting the j-th component of β_1 be $*$ if and only if the j-th variable in C_{i_1} is set by σ_1. Note that since $C_{i_1}\lceil_\rho$ is not the empty term there is at least one $*$ in β_1. From C_{i_1} and β_1 we can reconstruct σ_1.

Now, by the definition of $T(f\lceil_\rho)$, $\pi \setminus \pi_1$ labels a path in the canonical tree $T(f\lceil_{\rho\pi_1})$. If $\pi_1 \neq \pi$, we repeat the above argument, with $\pi \setminus \pi_1$ in place of π, $\rho\pi_1$ in place of ρ and find a term C_{i_2} which is the first term of f not set to 0 by $\rho\pi_1$. Based on this we generate π_2, σ_2, and β_2 as before. We repeat this process until the round k in which $\pi_1\pi_2...\pi_k = \pi$.

Let $\sigma = \sigma_1\sigma_2...\sigma_k$. We finally define $\delta \in \{0,1\}^s$ to be a vector that indicates for each variable set by π (which are the same as those set by σ) whether it is set to the same value as σ sets it.

The image of ρ under the 1-1 map we define is a triple , $\langle \rho\sigma_1...\sigma_k, (\beta_1, ..., \beta_k), \delta\rangle$. Clearly $\rho\sigma = \rho\sigma_1...\sigma_k \in \mathcal{R}_n^{l-s}$ and $(\beta_1, ..., \beta_k) \in stars(r,s)$ so the map is as required.

It remains to show that the map we have just defined is indeed 1-1. To do this, we show how to recover ρ from its image. The reconstruction is iterative. In the general stage of the reconstruction we will have recovered $\pi_1, ...\pi_{m-1}, \sigma_1, ..., \sigma_{m-1}$, and will have constructed $\rho\pi_1...\pi_{m-1}\sigma_m...\sigma_k$. Recall that for $m < k$, $C_{i_m}\lceil_{\rho\pi_1..\pi_{m-1}\sigma_m} = 1$ and $C_j\lceil_{\rho\pi_1..\pi_{m-1}\sigma_m} = 0$ for all $j < i_m$. This clearly also holds when we append $\sigma_{m+1}...\sigma_k$ to the restriction. when $m = k$, something similar occurs except the only guarantee is that $C_{i_m}\lceil_{\rho\pi_1..\pi_{k-1}\sigma_k} \neq 0$. Thus we can recover i_m as the index of the first term of f that is not set to 0 by $\rho\pi_1...\pi_{m-1}\sigma_m..\sigma_k$.

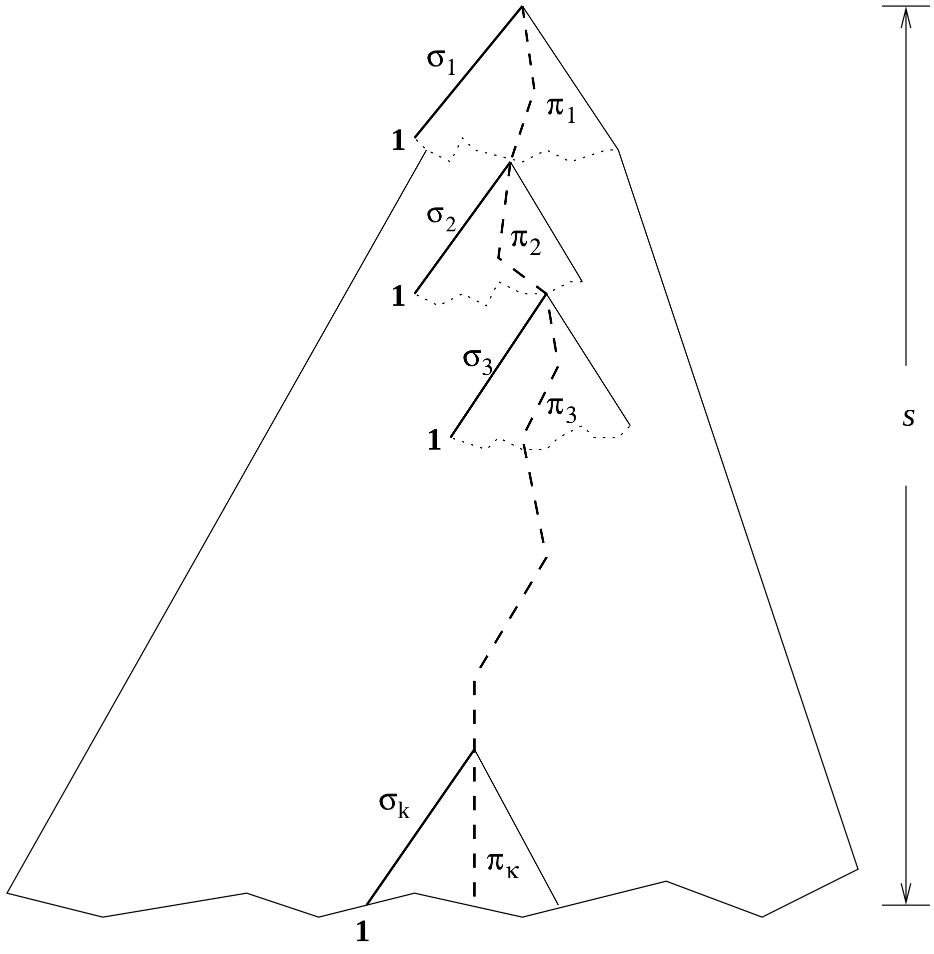

Figure 3: Canonical decision tree $T(f\lceil_\rho)$

Now, based on C_{i_m} and β_m we can determine σ_m. Since we know $\sigma_1, ..., \sigma_m$, using the vector δ we can determine π_m. We can now change $\rho\pi_1...\pi_{m-1}\sigma_m..\sigma_k$ to $\rho\pi_1...\pi_{m-1}\pi_m\sigma_{m+1}..\sigma_k$ using the knowledge of π_m and σ_m. Finally, given all the values of the π_m we can reconstruct ρ.

Now we compute the value $|S|/|\mathcal{R}_n^l|$:
$|\mathcal{R}_n^l| = \binom{n}{l} 2^{n-l}$ so

$$\frac{|\mathcal{R}_n^{l-s}|}{|\mathcal{R}_n^l|} = \frac{l^{(s)}}{(n-l+s)^{(s)}} \cdot 2^s \leq \frac{(2l)^s}{(n-l)^s}.$$

Applying the bounds we obtain

$$\begin{aligned} \frac{|S|}{|\mathcal{R}_n^l|} &\leq \frac{|\mathcal{R}_n^{l-s}|}{|\mathcal{R}_n^l|} \cdot |stars(r,s)| \cdot 2^s \\ &\leq \left(\frac{4lr}{(n-l)\ln 2} \right)^s \\ &= \left(\frac{4pr}{(1-p)\ln 2} \right)^s \end{aligned}$$

for $l = pn$. For $p < 1/7$ this is at most $(7pr)^s$. □

5 Lower bound for AC_0-Frege proofs of PHP_n^{n+1}.

We have discussed PHP_n^{n+1} problem in details in our third and fourth lectures.

In circuit complexity, for each gate g of a given circuit, we define decision trees $T(g)$ that precisely computed each $g\lceil_\rho$ in the circuit. But in case of proof complexity if we define a decision tree for each formula (or subformula) that appears in the proof, this cannot possibly work because every formula in the proof is a tautology and hence computes the constant function 1. So we use a different notion of decision trees that approximate each formula such that the bigger the proof the worse approximation we get.

Here for proving lower bound of PHP_n^{n+1} we will use matching decision trees. We will explain it elaborately in the next lecture.