CS 2429 - Propositional Proof Complexity

Lecture #10: 21 November 2002

Lecturer: Toniann Pitassi

Scribe Notes by: Periklis A. Papakonstantinou

In the previous lecture we presented the first part of the proof that for the AC_0-Frege proof systems (also called bounded depth Frege proof systems) there exists an exponential lower bound with respect to the proof size. In order to establish this result we started to prove that every PHP_n^{n+1} proof requires exponential proof size. The attack to this problem uses the following tools: (a)"translation" of the switching lemma from the circuit complexity to the proof complexity context and (b) an interpretation from model/proof theory. From model/proof theory we apply the idea of interpreting each formula in a local fashion that it is consistent with the negation of the pigeonhole principle derived formula. In this lecture we define and sketch the proof of the switching lemma.

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

1 Overview

There is a difference between the use of the decision trees in circuit complexity lower bound proof and their use in proof complexity. The problem is that we cannot straightforward relate a decision tree with each formula or subformula (as we did with the gates of the circuits), because each formula/subformula is a tautology and hence computes the constant 1. So, subsequently, we are going to define the notion of *Matching Decision Trees*. The high level of this proof follows:

- Define matching decision trees and relate them with the restrictions. Show how to "combine" matching decision trees.
- Prove a variation of the *switching lemma* we demonstrated in the previous lecture. This switching lemma concerns the proof complexity. Specifically it concerns the pigeonhole principle where we will use new distributions of the restrictions.

In this lecture we sketch the proof of the switching lemma and we give the basic intuition behind the semantics we are going to use.

2 Matching Decision Trees and Restrictions

We begin with some definitions, and then we are going to prove a few properties for them.

Definition

1. A restriction ρ is a matching from D to R.
2. A matching term is the associated set of literals (i.e. P_{32}, P_{45}, P_{54}).

Definition A matching *covers* a pigeon at the hole i if some edge in matching mentions i.

Definition A *matching disjunction* is an unbounded disjunction of *matching terms*. An *r-disjunction* is a matching disjunction where all terms have size at most r. For example $P_{11}P_{22} \vee P_{34}P_{21} \vee P_{54}P_{11}$ is a 2-disjunction.

Definition Let t be a matching term and let ρ be a matching restriction. Then $t|_\rho$ is defined as follows: If there exists a variable $P_{i,j}$ that occurs in t and either ρ maps i to some $j' \neq j$ or ρ maps some $i' \neq i$ to j, then $P_{i,j}|_\rho = 0$ and therefore $t|_\rho = 0$. Otherwise for every variable $P_{i,j}$ in t such that t contains $P_{i,j}$, $P_{i,j}$ is set to 1 by ρ.

Let $F = C_1 \vee C_2 \vee \ldots \vee C_n$ be a matching disjunction and ρ a matching restriction. Then $F|_\rho$ is another matching disjunction obtained by applying ρ to each term one at a time. If any term is set to 1 under ρ, then $F|_\rho = 1$, and if all terms are set to 0 under ρ, then $F|_\rho = 0$.

For example, let $F = P_{17}P_{38} \vee P_{16}P_{27} \vee P_{49}P_{56} \vee P_{16}P_{59}$, and let ρ be the restriction that maps pigeon 2 to hole 8, and pigeon 3 to hole 7. Then : $F\restriction_\rho = P_{27} \vee P_{59}$.

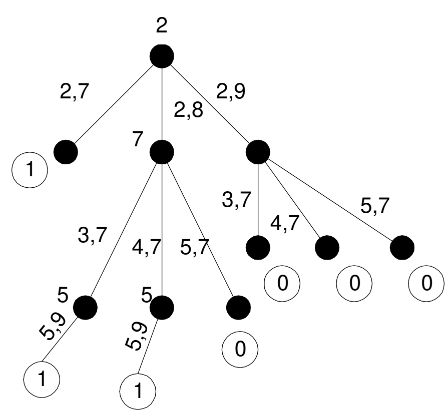

Figure 1: A canonical matching decision tree for $P_{2,7} \vee P_{5,9}$.

Definition A *matching tree* T over $S = D \cup R$ is a tree satisfying the following conditions:

1. The nodes of the tree, other than the leaves, are labeled with vertices of S.
2. If a node of a T is labeled with a vertex $i \in S$ then the edges leading out of the vertex are labeled with distinct pairs of the form $\{i, j\}$, where $j \in R$ if $i \in D$ or $j \in D$ if $i \in R$.
3. No vertex or edge label is repeated on a branch of T.

4. If p is a vertex of T then the edge labels on the path from the root of T to p determine a matching $\pi(p)$ between D and R.

Definition The *branches of* T $Br(T)$ is the set of all matching terms/restrictions associated with the paths of T. We distinguish between the paths having as leaves 1 and 0. Then, $Br_1(T) \cup Br_0(T) = Br(T)$.

Formally, the above definition corresponds to $Br(T) = \{\pi(l) | l \text{ is a leaf of } T\}$.
If M is the set of matchings, then T is *complete* for M if for every vertex p in T labeled with a vertex $i \in S$, the set of matchings $\{\pi(q) | q \text{ is a child of } p\}$ consists of all matchings of M of the form $\pi(q) \cup \{\{i, j\}\}$.

Definition If F is a matching disjunction and T is a matching decision tree then T *represents* F if for every $\pi \in B_r(T), F \lceil_\pi = 1$. If π is labeled 1 in T and $F \lceil_\pi = 0$ if π is labeled by 0 in T.

Below we provide the inductive definition of a *canonical matching tree.*

Definition Let $F = C_1 \vee \ldots \vee C_m$ be a matching disjunction over S. The *canonical matching decision tree for* F *over* S, $Tree_S F$, is defined inductively as follows:

1. If $F \equiv 0$ then $Tree_S(F)$ is a single node labeled 0. If $F \equiv 1$ then $Tree_S(F)$ is a single node labeled 1.
2. If $F \not\equiv 1$ and $F \not\equiv 0$, then let C be the first matching term in F such that $C \not\equiv 0$. Then $Tree_S(F)$ is constructed as follows:
   (a) Construct the full matching tree for the vertices that are associated with variables occurring in C.
   (b) Replace each leaf l of the previously constructed full matching tree by the canonical matching decision tree $Tree_{S\lceil_{\pi(l)}}(F \lceil \pi(l))$.

See Figure 1 for an example of a construction of a canonical decision tree associated with a matching disjunction.

Important Remark: The same tree can represent a lot of formulae. The reason is that we do not have values for the hole truth assignment.

We schematically sketch the definition of a restriction applied to a complete matching decision tree: Let T be a complete matching decision tree and ρ a restriction. Consider the tree of figure 1. Then $T \lceil_\rho$ is another matching decision tree obtained as in figure 2. You can observe that the restriction *shrinks the tree.* The new (derived under the restriction) tree is the one of figure 3. The X's means that the variable labelling this edge is set to either 0 or 1 by the restriction. For example, the edge labelled "3,7" is set to 0, and the edge labelled "4,7" is set to 1. The new derived tree is obtained by pruning the original tree in the natural way starting at the root: (ii) if there is an edge from the root to a vertex v that is labelled 1, we replace the whole tree with the subtree rooted at v; (ii) otherwise, for any edge from the root to a vertex v labelled by 0, we remove this edge as well as the subtree rooted at v.

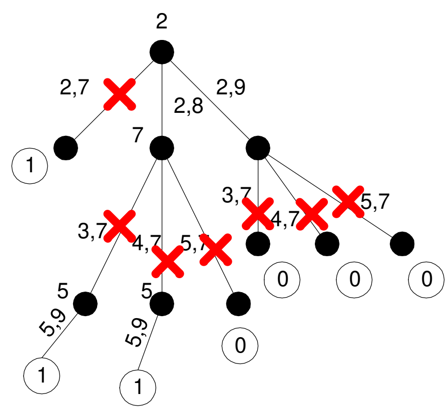

Figure 2: We apply the restriction $\rho : \mathbf{4} \to \mathbf{7}$

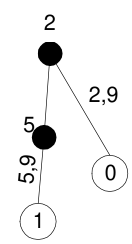

Figure 3: The resulting tree, after the application of the restriction (see figure 2).

Definition Let T be a matching decision tree, then T^c is the tree T with the leaf labels complemented (just change the leaves from 0 to 1 and vice versa).

Definition $Disj(T) = t_1 \vee \ldots \vee t_m$, where $\{t_1, \ldots, t_m\} = Br_1(T)$.

Lemma 2 *Let T be a matching decision tree, ρ a restriction.*

1. $Disj(T) \upharpoonright_\rho = Disj(T \upharpoonright_\rho)$.
2. *If T is complete for D, R then $T \upharpoonright_\rho$ is complete for $D \upharpoonright_\rho, R \upharpoonright_\rho$.*
3. $(T \upharpoonright_\rho)^c = T^c \upharpoonright_\rho$
4. *If l is a leaf in $T \upharpoonright_l$ then there is a leaf l' in T with the same label as l so that $\pi(l') \subseteq \pi(l) \cup \rho$, where π is a restriction or a matching term.*
5. *If T represents F, then $T \upharpoonright_\rho$ represents $F \upharpoonright_\rho$.*

From the above lemma the "most important" part for our proof is the (5).

3 Evaluations and the Matching Switching Lemma

Let $\mathcal{P}$ be a small (that is 2^{n^s}, where $s < \frac{1}{5^d}$) depth of Frege proof of PHP_n^{n+1}. $\mathcal{P} = F_1, F_2, \ldots, F_m$, where $F_m = PHP_n^{n+1}$ bounded depth Frege proof. Let $\mathcal{R}$ be their set of *all* subformulae occurring in $\mathcal{P}$ (think of it of the corresponding way, of as having many circuits).

Definition $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \ldots \cup \mathcal{R}_d$, where $\mathcal{R}_i$ is the depth[1] i subformulae in $\mathcal{R}$.

Let $M_n^\rho = \{\rho \in M_n | \mathcal{R} \restriction_\rho = l\}$. We intuitively define which is bad: "you" are bad if the corresponding canonical tree has height bigger than s. Hence:
$Bad_n^l(F, s) = \{\rho \in M_n^l |\ height(Tree_{S\restriction_\rho}(F \restriction_\rho)) \geq s\}$

Lemma 3 *Let F be an r-disjunction over D, R, where $|D| = n+1, |R| = n$, $l \geq 10, \rho = \frac{l}{n}$. If $r \leq l$ and $p^4n^3 \leq 1/10$ then:*

$$\frac{|Bad_n^l(F, 2s)|}{|M_n^l|} \leq (11p^4n^3r)^s$$

The proof goes like the one we have seen in the previous lecture (lecture 9). Now we are going to put everything together:

Definition Let $\mathcal{R}$ be as defined previously. A k-evaluation T is an assignment of a complete matching decision trees $T(A)$ to formulae A in $\mathcal{R}$ such that:

1. $T(A)$ has depth less or equal to k.
2. $T(1)$ is the single node labeled 1, and $T(0)$ is the single node labeled 0.
3. $T(P_{ij})$ is the full tree for i, j over D, R (i.e. the canonical tree for P_{ij}).
4. $T(\neg A) = T(A)^c$
5. If A is a disjunction $A = A_1 \vee A_2 \vee \ldots \vee A_k$, then $T(A)$ represents $\vee_{i \in I} Disj(T(A_i))$.

This is one of the basic concepts of the proof. Semantically there is some connection with the tree and the representation as it is shown in figure 4. For example if all leaves are labeled to 1 we have a "kind" of "tautology". However this notion of "tautology" is **not preserved** under (sound) inferences. This is the key idea of the lower bound argument.

This connection can be demonstrated by the example of figure 5 (Attention! This is not a formula which corresponds to the PHP; we provided it here only to exemplify things). We will talk about this "semantic connection" in the next lecture.

[1] For a definition of circuit depth see at lecture 9

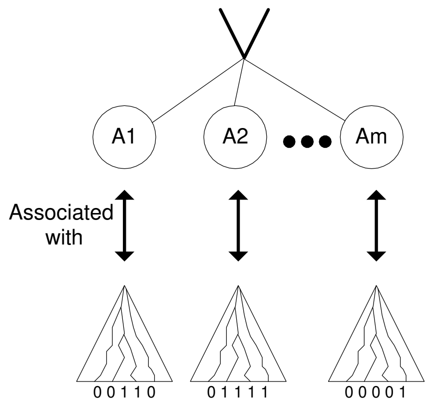

Figure 4: XXX

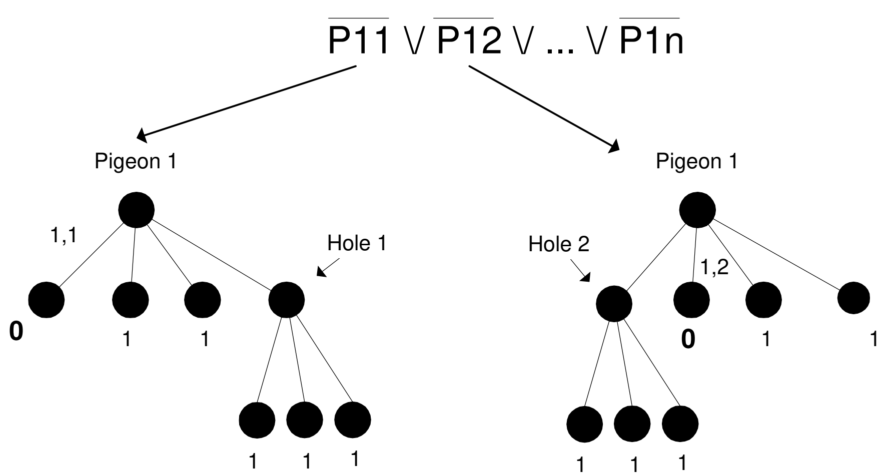

Figure 5: XXX