Exponential Size Lower Bounds for Bounded-Depth Frege Proofs of the Pigeonhole Principle: Part I

Paul Beame
University of Washington
Seattle, WA 98195

The Depth of Proofs

Restrict to connectives $\bigvee$, $\neg$.
Results for other connectives easily derived.

Def: The *depth* of formula F (circuit C) = max # of runs of $\bigvee$ operators in F (C) on any path from a leaf (input) to root (output).

For Frege proof P, depth(P) = max$_{F \in P}$ depth(F)

Every formula F in Extended Frege proof P' corresponds to a *circuit* C_F whose inputs are the non-extension variables of P'. (Extension variables name certain internal gates of C_F.)

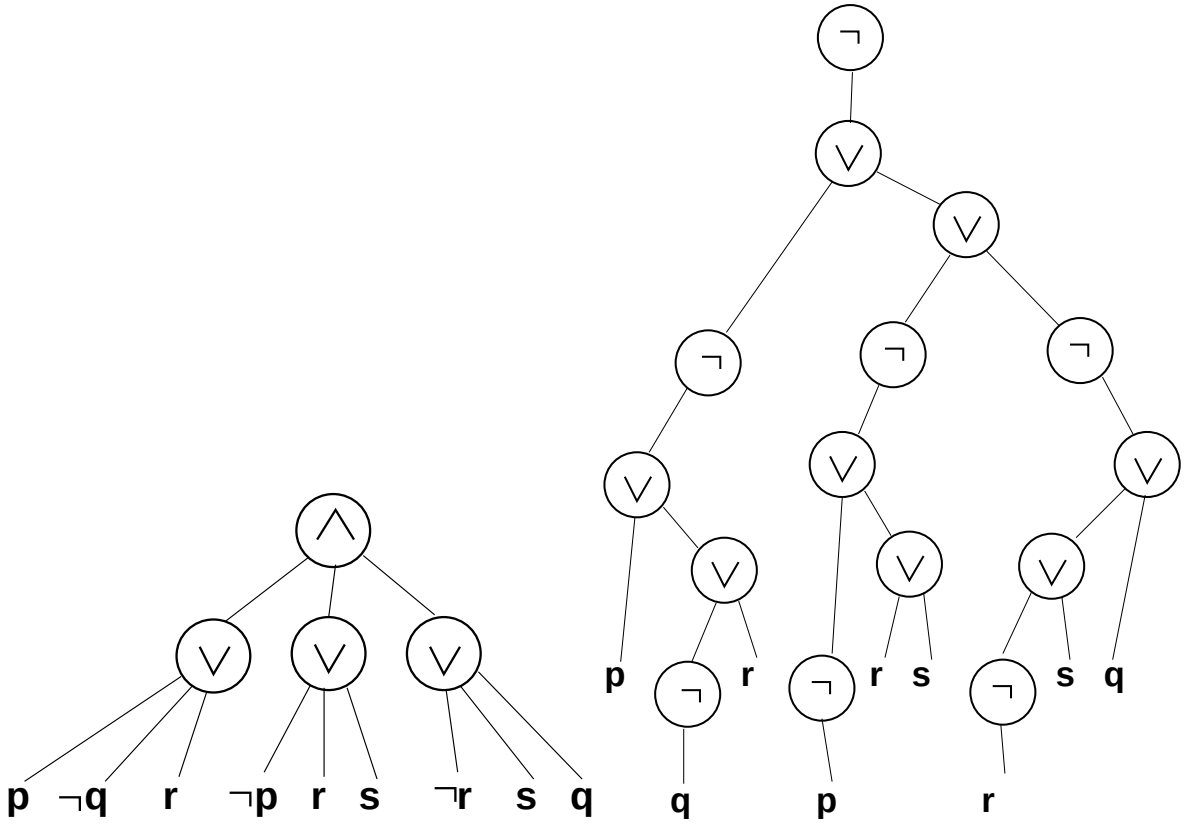
For $e\mathcal{F}$-proof P', depth(P') = max$_{F \in P'}$ depth(C_F).

Def: The *size* of a Frege or $e\mathcal{F}$- proof P = total number of distinct subformulas (sub-circuits) in P.

Use notation $\bigvee_{i=1}^{k} A_i$ to indicate arbirarily parenthesized $\vee$ formula in the A_i.

$\bigwedge_{i=1}^{k} A_i$ abbreviation for $\neg \bigvee_{i=1}^{k} \neg A_i$.

Example: Conjunctive and Disjunctive Normal Formulas have depth 2.



Size-depth lower bounds for Frege, $e\mathcal{F}$

Thm: (Pitassi, Beame, Impagliazzo 1992; Krajíček, Pudlák, Woods 1992)
Any depth d, Frege or $e\mathcal{F}$-proof of the propositional pigeonhole principle, PHP_n^{n+1}, requires size $\Omega(2^{n^{\epsilon_d}})$ where $\epsilon_d = 1/5^d$.

Cor: $PHP(f)$ is not provable in $S_2(f)$.

Cor: Polynomial-size $\mathcal{F}$ proofs of PHP_n^{n+1} require depth $\Omega(\log\log n)$.

Work builds on earlier superpolynomial size lower bounds for constant-depth Frege proofs of
PHP_n^{n+1} by (Ajtai 1988; Bellantoni, Pitassi, Urquhart 1991)

The restriction lower bound method

Introduced for bounded-depth circuit complexity by (Furst, Saxe, Sipser 1981; Ajtai 1983)

A *restriction* ρ for a domain of Boolean variables $X_I = \{x_i \mid i \in I\}$ is a partial assignment of values to the variables.

Formally, $\rho : X_I \to \{0, 1, *\}$ where $\rho(x_i) = *$ indicates that the variable x_i is not assigned a value.

Applying Restrictions:

If F is a function, formula or circuit, write $F\lceil_\rho$ for the result of substituting $\rho(x_i)$ for each x_i such that $\rho(x_i) \neq *$.

Why restrictions might help

Restrictions simplify functions, circuits, and formulas:

Given

$$F = \bigvee_i x_i \ \vee \ \bigvee_j \neg x_j,$$

assigning a single $\rho(x_i) = 1$ or $\rho(x_j) = 0$ makes $F\restriction_\rho$ a constant, i.e. wiping out F but only setting one variable.

Simplification substantially more than # of variables assigned.

Basic idea: To prove that small circuit C cannot compute function f, choose a restriction ρ so that $f\restriction_\rho$ is still a complicated function but $C\restriction_\rho$ is extremely simple and we can easily understand that $C\restriction_\rho$ cannot compute $f\restriction_\rho$.

Boolean decision trees

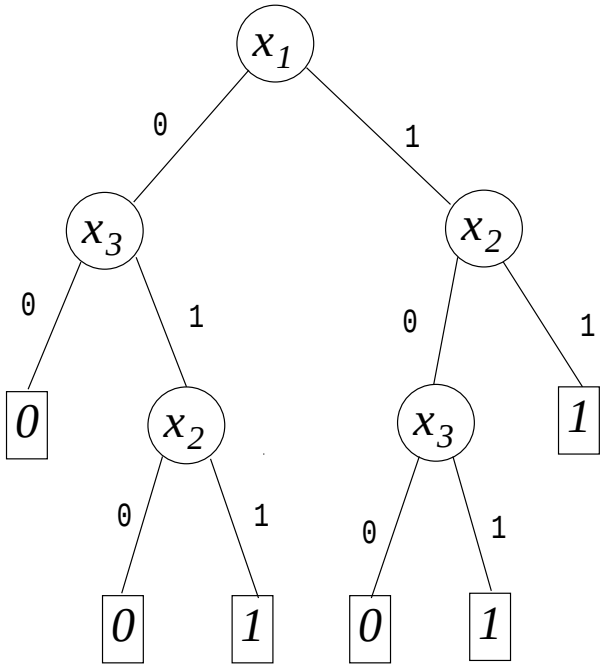
Def: A Boolean decision tree T is a binary rooted tree with each internal node labelled by some x_i and leaf nodes labelled 0 or 1. Edges out of each internal node are labelled 0 or 1.

Root-leaf path corresponds to a restriction ρ of input variables:

Outedge labelled b from node labelled x_i is taken iff $x_i \leftarrow b$ is in ρ

Tree T *computes* function f if every branch corresponding to ρ in T has leaf label $f\lceil_\rho$.

e.g. Compute $f(x) =$‘$x_1 + x_2 + x_3 \geq 2$’



Restriction for constant depth circuits

Def: An (S, d)-*circuit* will be a circuit of size $\leq S$ and depth $\leq d$.

To show no (S, d)-circuit C computes function f, find set $GR_{S,d}(f)$ of restrictions s.t.:

(a) For any (S, d)-circuit C there is a restriction $\rho \in GR_{S,d}(f)$ s.t. for each gate g of C, we can associate a small* height Boolean decision tree $T(g)$ such that $T(g)$ computes $g\lceil_\rho$

(b) For any $\rho \in GR_{S,d}(f)$, $f\lceil_\rho$ is not computed by *any* small* height Boolean decision tree

* relative to number of variables unset by ρ

Note: Condition (b) usually determines the *structure* of the restrictions as well as the number of unset variables.

Note: Original restriction arguments did not use small height decision tree property:

(Furst, Saxe, Sipser 1981; Ajtai 1983) used dependence on a small number of variables.

(Yao 1985; Håstad 1986) used small minterm property.

(Beame, Håstad 1987) used DNF formulas with small terms.

All such arguments can be converted to reason about small height decision trees and are often more natural that way.

Restrictions Applied to Frege Proofs

Introduced by (Ajtai 1988) to show $n^{\omega(1)}$ size bounds for constant-depth proofs of PHP_n^{n+1}.

Def: An (S,d)-*proof* will be a Frege proof of size $\leq S$ and depth $\leq d$.

To show no (S,d)-proof P proves tautology F, find set $GR_{S,d}(F)$ of restrictions such that:

(a) For any (S,d)-proof P there is a restriction $\rho \in GR_{S,d}(F)$ so that for each subformula A of P, we can associate a small height decision tree T_A such that:
(i) T_A approximates $A\lceil_\rho$
(ii) If A is a line of P then $T_A \approx$ "true"

(b) For any $\rho \in GR_{S,d}(F)$, and *any* small height decision tree T_F that approximates $F\lceil_\rho$ has $T_F \approx$ "false"

Note: Since both F and every line of P are tautologies, the functions that they compute are all identically 1.
$\Rightarrow$ The fact that there is only approximation rather than exact representation is critical.

Note: (Ajtai 1988) formalized his argument using non-standard integers and forcing.

(Bellantoni, Pitassi, Urquhart 1991) removed forcing and phrased the argument in terms of approximately correct inferences.

This formulation is a modification of the k-evaluations of (Krajíček, Pudlák, Woods 1992) using decision trees from
(Pitassi, Beame, Impagliazzo 1992).

Propositional Pigeonhole Principle

Let $D = [1, m]$, $R = [1, n]'$, $D \cap R = \emptyset$.
Propositional variables P_{ij}, $i \in D$, $j \in R$.
There are 3 natural variations of the Pigeonhole Principle:

"There's no total injective relation on $m \times n$"

$$rPHP_n^m = (\bigvee_{i \in D} \bigwedge_{j \in R} \neg P_{ij}) \vee \bigvee_{j \in R} \bigvee_{i \neq i' \in D} (P_{ij} \wedge P_{i'j}).$$

"There's no total 1-1 function from m to n:"

$$PHP_n^m = rPHP_n^m \vee \bigvee_{i \in D} \bigvee_{j \neq j' \in R} (P_{ij} \wedge P_{ij'}).$$

These two are polynomially equivalent as may be seen by setting:

$$P'_{ij} = P_{ij} \wedge \bigvee_{j' < j} \neg P_{ij'}.$$

However, the following variant does seem to be slightly weaker than PHP_n^m:

"There's no 1-1 onto function from m to n:"

$$\text{onto-}PHP_n^m = PHP_n^m \vee (\bigvee_{j \in R} \bigwedge_{i \in D} \neg P_{ij}).$$

Although we state the lower bounds for PHP_n^m the results apply to onto-PHP_n^m as well.

Restrictions for PHP_n^{n+1}

Let $D = [1, n+1]$, $R = [1, n]'$.
Assume that $D \cap R = \emptyset$.
Variables P_{ij}, $i \in D$, $j \in R$.

Restrictions: Partial 1-1 mappings, i.e. elements of
$M_n = \{\text{partial bipartite matchings on } D \times R\}$.

$\pi \in M_n \Rightarrow$ restriction ρ_π such that

$$\rho_\pi(P_{ij}) = \begin{cases} 1 & \text{if } \{i,j\} \in \pi \\ 0 & \text{if } \{i,j\} \notin \pi \text{ but } \pi \text{ touches } i \text{ or } j \\ * & \text{otherwise} \end{cases}$$

Abuse of Notation: Identify π and ρ_π.
Observe that applying π sets all variables whose endpoints are touched by π. Use $D\lceil_\pi$ and $R\lceil_\pi$ to denote the sets of points in D and R for which variables remain.

Matching Decision trees

A *matching decision tree over* $D \cup R$ is a rooted directed tree T whose:
Internal nodes are labelled by elements of $D \cup R$,
Leaves are labelled by 0 or 1 so that:

1. (a) If the root of T is labelled by $i \in D$ then for each $j \in R$ there is one out-edge from the root labelled $\{i, j\}$.

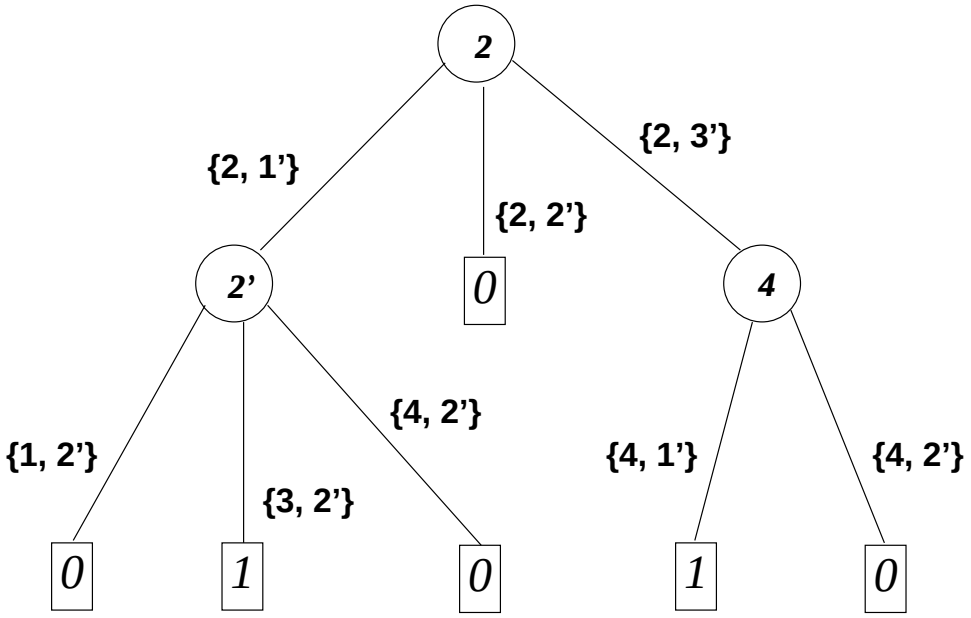
(b) If the root of T is labelled by $j \in R$ then for each $i \in D$ there is one out-edge from the root labelled $\{i, j\}$.

(c) There are no other out-edges from the root of T.

2. Let $T^{\{i,j\}}$ be the tree whose root is the node connected to the root of T by the edge labelled $\{i, j\}$. $T^{\{i,j\}}$ is a matching decision tree over $D' \cup R'$ where $D' = D \setminus \{i\}$ and $R' = R \setminus \{j\}$.

Example

$D = \{1, 2, 3, 4\},\ R = \{1', 2', 3'\}.$



$$\begin{aligned} \mathsf{br}_1(T) &= \{\ \{\ \{2,1'\},\ \{3,2'\}\ \},\\ &\qquad \{\ \{\{2,3'\},\ \{4,1'\}\ \}\ \} \end{aligned}$$

$$\begin{aligned} Disj(T) &= P_{21}P_{32} + P_{23}P_{41}\\ &= (P_{21} \wedge P_{32}) \vee (P_{23} \wedge P_{41}) \end{aligned}$$

Def: Define sets
$\text{br}(T) = \{\text{branches (root-leaf paths) in } T\}$.
$\text{br}_a(T) = \{\text{branches in } T \text{ with leaf label } a\}$.

The set of edge labels along any branch of T forms a partial matching. Identify a branch with its matching so $\text{br}(T)$ and $\text{br}_a(T)$ become sets of partial matchings.

For any matching decision tree T, let T^c be the same tree as T except that the leaf labels 0 and 1 are reversed,
i.e. $\text{br}_1(T^c) = \text{br}_0(T)$ and $\text{br}_0(T^c) = \text{br}_1(T)$.

Def: Matching decision tree T *represents* boolean function/formula/circuit f iff:

$\forall \pi \in \text{br}(T)$, $f\lceil_\pi \equiv$ the leaf label of π in T.

Note: Unlike ordinary Boolean decision trees, the fact that T represents f only determines the value of f on some inputs.

Intuition:

$T \approx "true"$ iff $\text{br}_1(T) = \text{br}(T)$,
i.e. every leaf of T has label 1.

$T \approx "false"$ iff $\text{br}_0(T) = \text{br}(T)$,
i.e. every leaf of T has label 0.

Compatibility Lemma

Definition: $\pi_1,\ \pi_2 \in M_n$ are *compatible* if $\pi_1 \cup \pi_2 \in M_n$. Write $\pi_1\pi_2$ for combined restriction.

Lemma: Let π be a matching and T be a matching decision tree over $D \cup R$ such that $|\pi| + height(T) \leq \min(|D|, |R|)$. Then

(i) there is a $\sigma \in \mathrm{br}(T)$ compatible with π.

(ii) the tree $T\lceil_\pi$ obtained by contracting all edges of T whose label is in π and deleting all edges of T (and their associated subtrees) whose labels are not compatible with π is a matching decision tree over $D\lceil_\pi \cup R\lceil_\pi$.

Proof: Part (ii): By induction on the height of T: The base case when T is a single labelled vertex is trivial.

If the label i of the root of T is touched by π then the remaining tree is $T^{\{i,j\}}\lceil_\pi$. We apply the inductive hypothesis to $T^{\{i,j\}}$ over $(D \setminus \{i\}) \cup (R \setminus \{j\})$ to obtain the desired result.

If the label $i \in D$ of the root of T is not touched by π then the tree $T\lceil_\pi$ consists of the root of T with an outedge labelled by $\{i,j\}$ for each
$j \in R\lceil_\pi$ and this reaches subtree $T^{\{i,j\}}\lceil_\pi$. Apply the inductive hypothesis to each such $T^{\{i,j\}}$ over $(D \setminus \{i\}) \cup (R \setminus \{j\})$ to obtain the desired result.

The case when the root label of T is $j \in R$ is similar.

Part (i): Any branch in T that is contracted to a branch in $T\lceil_\pi$ suffices. $\square$

Matching Disjunctions

A *matching term* A is $\bigwedge_{\{i,j\}\in\pi} P_{ij}$ for some matching $\pi \in M_n$.

A *matching disjunction* F is $\bigvee_i A_i$ where A_i are matching terms.

Def: Given matching decision tree T, the matching disjunction given by T is

$$Disj(T) = \bigvee_{\pi\in \mathsf{br}_1(T)} \bigwedge_{\{i,j\}\in\pi} P_{ij}$$

Note: (1) T represents $Disj(T)$.
(2) If T has height $\leq k$ then $Disj(T)$ has terms of size $\leq k$.
(3) $Disj(T)$ computes the minimal function forced to 1 by T.
(4) $Disj(T^c)$ is *not* equivalent to the complement of $Disj(T)$.

k-Evaluations

Def: Let Γ be a set of formulas closed under subformulas. A *k-evaluation* of Γ associates a matching decision tree T_A of height $\leq k$ with each formula $A \in \Gamma$ such that

(1) $T_0 = \bullet_0$, $T_1 = \bullet_1$, and $T_{P_{ij}}$ = tree of height 1 querying i that represents P_{ij}

(2) $T_{\neg A} = T_A^c$

(3) If the major connective of each A_i is not $\vee$

then $T_{\bigvee_{i \in I} A_i}$ represents $\bigvee_{i \in I} Disj(T_{A_i})$.

k-Evaluations and "Truth"

Def: The *size* of an axiom/rule in a Frege system $\mathcal{F}$ is the maximum number of distinct subformulas in it.

Lemma: Let P be a proof in Frege system $\mathcal{F}$ whose rules have size at most s. Suppose that $sk \leq n$ and let T be a k-evaluation of the set of subformulas of P . Then for any formula A in P, $T(A) \approx "true"$, i.e. $\text{br}_1(T_A) = \text{br}(T_A)$.

Proof: By complete induction on the number of inferences in P. (View axioms simply as inferences with empty antecedents.)

Consider an inference in P:

$$\frac{A_1(B_1/p_1,\cdots,B_m/p_m),\ldots,A_\ell(B_1/p_1,\cdots,B_m/p_m)}{A_0(B_1/p_1,\cdots,B_m/p_m)}$$

where the inference rule $\mathcal{R}$ is:

$$\frac{A_1(p_1,\cdots,p_m),\ldots,A_\ell(p_1,\cdots,p_m)}{A_0(p_1,\cdots,p_m)}$$

and assume that the tree for each
$A_i(B_1/p_1,\cdots,B_m/p_m)$ for $1 \leq i \leq \ell$ is $\approx$ "$true$".

Now show for $A_0(B_1/p_1,\ldots,B_m/p_m)$:

Let $\mathcal{A}$ be the set of distinct subformulas of $\mathcal{R}$
and let Γ be $\mathcal{A}(B_1/p_1,\cdots,B_m/p_m)$.
By assumption $|\Gamma| \leq s$,
say $\Gamma = \{A_0,\ldots,A_\ell,\ldots,A_j\}$ for $j < s$.

Let $\pi_0 \in \mathrm{br}(T_{A_0})$.
Since $sk \leq n$ we can apply Compatibility Lemma to inductively find $\pi_i \in \mathrm{br}(T_{A_i})$ compatible with $\pi_0 \cdots \pi_{i-1}$ for $1 \leq i \leq j$.
$\Rightarrow$ all the π_i are mutually compatible.
Let $\pi = \pi_0 \pi_1 \cdots \pi_j \in M_n$.

Observe that for any $A_i \in \Gamma$, $Disj(T_{A_i})\lceil_\pi$ is the constant 0 or 1 and define $V : \Gamma \to \{0, 1\}$ by $V(A) = Disj(T_{A_i})\lceil_\pi$.

By the definition of k-evaluations, V is a consistent truth evaluation and by assumption

$$V(A_1) = \cdots = V(A_\ell) = 1.$$

Since the rule $\mathcal{R}$ is sound it follows that $V(A_0) = 1$, i.e.

$$Disj(T_{A_0})\lceil_\pi = 1.$$

Since π extends branch π_0 of T_{A_0}, the leaf label of π_0 must be 1 as required. $\square$

PHP_n^{n+1} is approximately "false"

Recall that PHP_n^{n+1} is:

$$(\bigvee_{i \in D} (\neg \bigvee_{\ell \in R} P_{i\ell})) \vee \bigvee_{i \neq j \in D,\ \ell \in R} \neg(\neg P_{i\ell} \vee \neg P_{j\ell}).$$

Lemma If $k+1 \leq n$ and T is a k-evaluation of a set of formulas containing PHP_n^{n+1} then

$$T_{PHP_n^{n+1}} \approx "false"$$

i.e. every leaf of $T_{PHP_n^{n+1}}$ has label 0.

(The same applies to onto-PHP_n^{n+1}.)

Proof: By definition of a k-evaluation it is necessary and sufficient to show that $\mathrm{br}_1(T_A) = \emptyset$ for each disjunct A in PHP_n^{n+1}.

Case 1: $A = \neg(\neg P_{i\ell} \vee \neg P_{j\ell}) = \neg B$

Let $\pi \in \text{br}(T_A)$. It is also in $\text{br}(T_B)$. Since T_B represents $Disj(T_{\neg P_{i\ell}}) \vee Disj(T_{\neg P_{j\ell}})$, it suffices to show that π is compatible with some element in $\text{br}_1(T_{\neg P_{i\ell}})$ or in $\text{br}_1(T_{\neg P_{j\ell}})$.

By definition $T_{\neg P_{i\ell}}$ has height 1 with root label i and all its leaves are labelled 1 except the one below the out-edge with label $\{i, \ell\}$.

Since $k + 1 \leq n$, $T_{\neg P_{i\ell}}\lceil_\pi$ is well-defined
and consists of contractions of all branches compatible with π.
If π does not contain $\{i, \ell\}$ then some branch of $T_{\neg P_{i\ell}}$ other than $\{i, \ell\}$ remains and this has leaf label 1.

If π does contain $\{i, \ell\}$ then it does not contain $\{j, \ell\}$ and we apply the same argument to $T_{\neg P_{j\ell}}$.

Case 2: $A = \neg \bigvee_{\ell \in R} P_{i\ell}$

Similar to the previous case. Here, we show that $\pi \in \mathrm{br}(T_A)$ is compatible with some element of $\mathrm{br}_1(T_{P_{i\ell}})$ for some $\ell \in R$.

If π contains $\{i, j\}$ for some $j \in R$ then every branch in $T_{P_{ij}}$ compatible with π will be in $\mathrm{br}_1(T_{P_{ij}})$.

If π does not contain $\{i, \ell\}$ for every $\ell \in R$ then let $j' \in R$ be unmatched by π (such a j' must exist). Since π matches neither i nor j' and $k + 1 \leq n$, π is compatible with the $\{i, j'\}$ branch of $T_{P_{ij'}}$ which is what we need.

Case 3: The onto-PHP_n^{n+1} disjuncts are handled exactly as in Case 1. □

Building a k-evaluation

Given an (S, d)-proof P it is too hard in a single step to find a restriction ρ s.t. after ρ is applied a suitable k-evaluation exists for all the subformulas in P.

Instead, we inductively build restrictions and k-evaluations for all depth i subformulas in P for $i = 0, \ldots, d$. (The leaves have depth 0.)

More precisely we choose $\rho_0, \ldots, \rho_d = \rho$ so that after each ρ_i is applied, there is a k_i-evaluation of all subformulas of depth at most i in P and $|R\restriction_{\rho_i}| = n_i$ range vertices unset.

We will set the values of the k_i and n_i later.

Lemma: If Γ is a set of formulas closed under subformulas and T is a k-evaluation of Γ over vertex set $D \cup R$ and ρ is a restriction on S with $|\rho| + k \leq |R|$ then the map

$$T'_F = \begin{cases} T_F\lceil_\rho & \text{if } \mathsf{br}_1(T_F) \neq \emptyset \\ T_0 & \text{if } \mathsf{br}_1(T_F) = \emptyset \end{cases}$$

is a k-evaluation of Γ over $(D \cup R)\lceil_\rho$.

Proof: Note that:
(a) For any matching decision tree T and formula F, if T represents F over $D \cup R$ then $T\lceil_\rho$ represents $F\lceil_\rho$ over $(D \cup R)\lceil_\rho$.
(b) For any matching decision tree T,

$$Disj(T)\lceil_\rho = Disj(T\lceil_\rho).$$

¿From this the Lemma follows easily by induction.
(The extra condition when $\mathsf{br}_1(T_F) = \emptyset$ is to make sure that $T_{P_{ij}}\lceil_\rho = T_0$ when $P_{ij}\lceil_\rho = 0$.) $\square$

We now are ready to start the inductive argument. We only specify trees for unnegated formulas at each depth since negations do not add to depth and if we have a tree T_F then we easily have a tree $T_{\neg F} = T_F^c$.

Base Case: Let $k_0 = 1$, $\rho_0 = \emptyset$. For each literal P_{ij} create a tree of height 1 that queries i and has its only leaf label 1 on the node coming from edge labelled $\{i, j\}$.

For 0 and 1, create trees of height 0 consisting of a single node with the correct label.

Induction Step

Now suppose that after ρ_i is applied we have a k_i-evaluation of all the formulas up to depth i in P.

We wish to extend this to a k_{i+1}-evaluation of all formulas in P of depth up to $i+1$.

If $k_i \leq k_{i+1}$ the only formulas that could cause a problem are those of depth $i+1$.

Suppose that $A = \bigvee_j A_j$ where A is a depth i formula in P and each A_j has depth at most i in P. By the induction hypothesis we have decision trees T_{A_j} of height at most k_i for each of these A_j.

We seek a restriction π that applies to the unset variables of ρ_i so that there is a matching decision tree T_A of height at most k_i+1 that represents $\bigvee_j Disj(T_{A_j}\lceil_\pi) = \bigvee_j Disj(T_{A_j})\lceil_\pi$.

It is not obvious how to do this.

Worse...
we have to find a single π that does this for ALL depth $i+1$ formulas A in the proof simultaneously...

...so we use...

The Probabilistic Method

To show that an object with particular properties exists create a distribution of objects and prove that a randomly selected object according to this distribution has a positive probability of having the desired properties.

Then an object with the desired properties must exist in order to contribute to the positive probability.

In general this is non-constructive but in our case the distribution will be over restrictions so it will be constructive in the classical sense.

(Razborov 1993) shows that in our case it is also feasibly constructive in one sense.

Making it work

Observe that
(1) There are at most S formulas A at depth $i+1$ since we start with a proof of size $\leq S$
(2) $\bigvee_j Disj(T_{A_j})$ is a matching disjunction with term size at most k_i.

We will show that for *any* fixed matching disjunction F with term size at most k_i,
if we choose a restriction π *at random* from among the restrictions that leave n_{i+1} range vertices unset,
the probability that there does *not* exist a height k_{i+1} decision tree that represents $F\lceil_\pi$ is $< 1/S$

This is a form of "Switching Lemma."

Let P_{i+1} be the set of formulas of depth $i+1$ in P.

¿From facts (1) and (2) and the switching lemma.

$$\begin{aligned}
&\text{Prob}[\exists A \in P_{i+1} \text{ s.t. } \pi \text{ doesn't work for } A] \\
&\quad \leq \quad S \cdot \max_{A \in P_{i+1}} \text{Prob}[\pi \text{ doesn't work for } A] \\
&\quad < \quad S \cdot 1/S \\
&\quad = \quad 1
\end{aligned}$$

Therefore there is a restriction π that works for all the depth $i+1$ formulas A in P.
We fix $\rho_{i+1} = \rho_i \pi$ and continue.

To Be Continued

It remains to prove the Switching Lemma and to choose the appropriate values for k_i and n_i that make the proof go through.

The proof of the Switching Lemma and the computation of the values of k_i and n_i are the subject of Part II of this lecture.