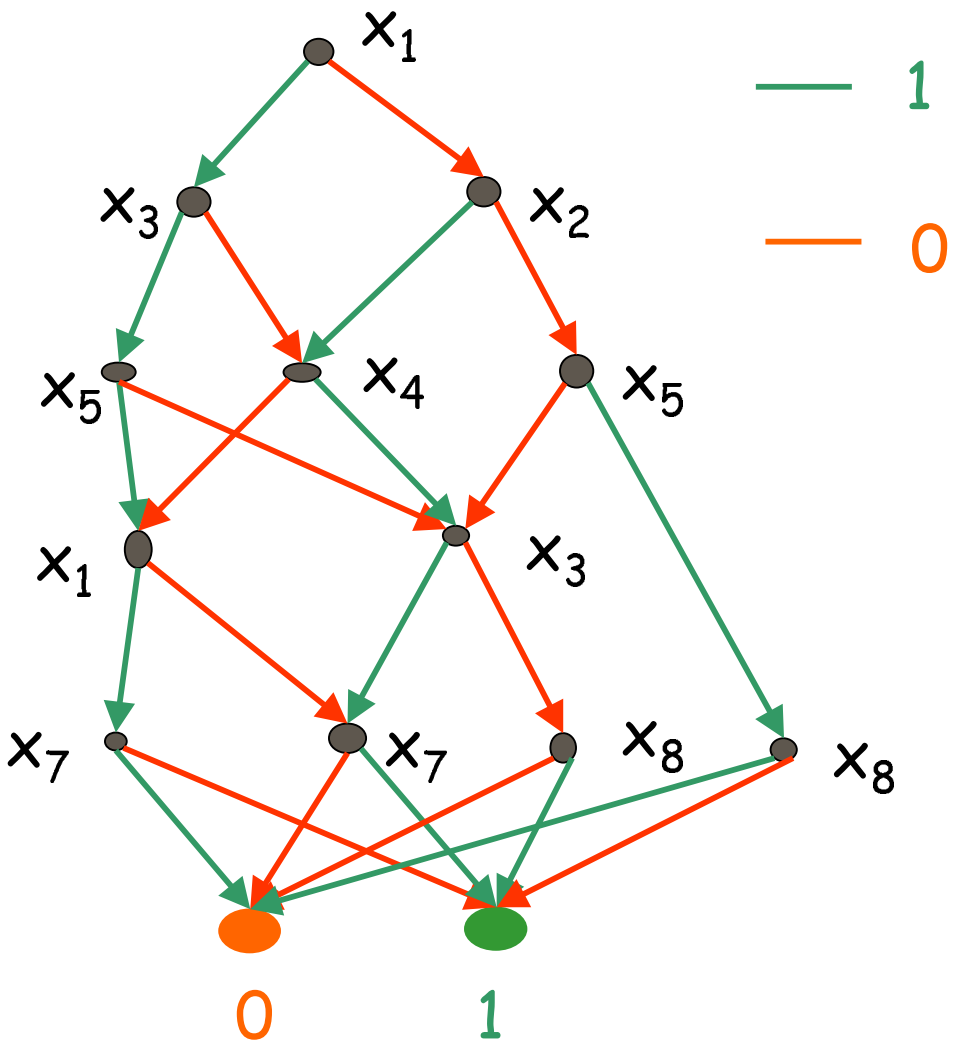
Super-linear Time-Space Tradeoff Lower Bounds for Randomized Computation

Paul Beame
University of Washington

with Mike Saks, Xiaodong Sun, Erik Vee

Branching Programs

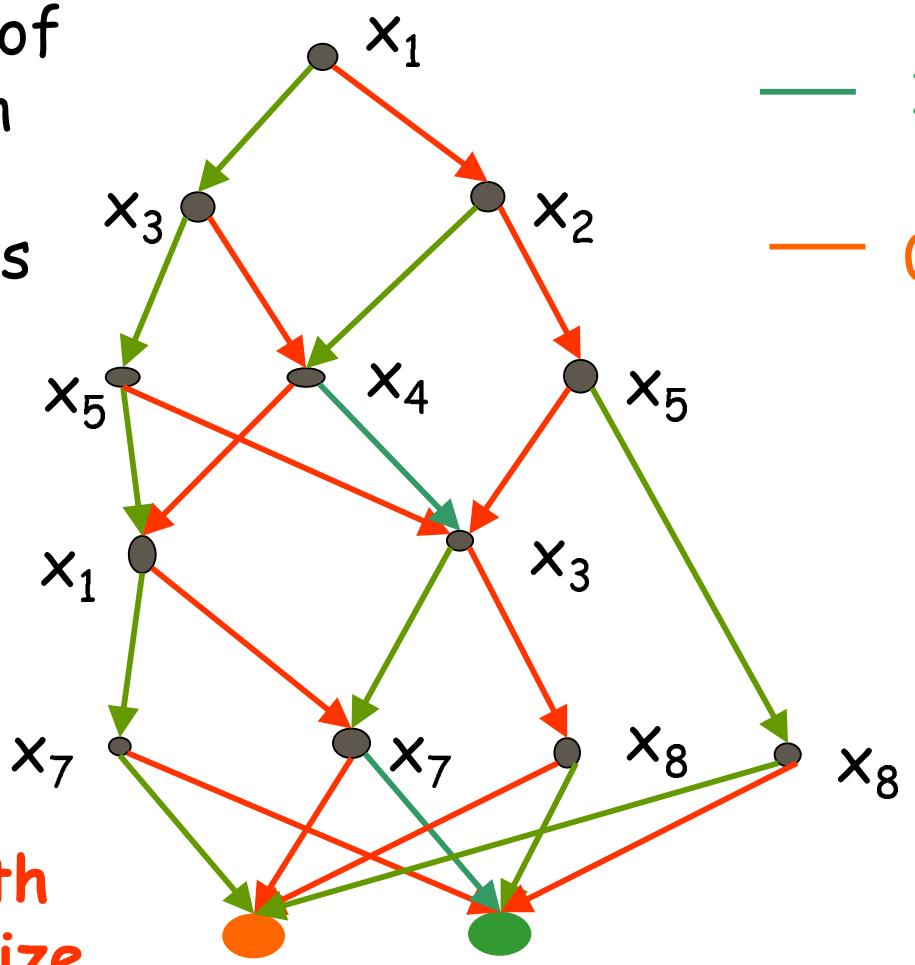
To compute $f:\{0,1\}^n \rightarrow \{0,1\}$ on input $(x_1,\ldots,x_n)$ follow path from source to sink



Branching Programs

Length = length of of longest path

Size = # of nodes



Time T = **Length**

Space S = $\log_2$**Size**

Branching Program Properties

- Models random-access
  - not just sequential access of Turing machines
- Multi-way version for x_i in arbitrary domain **D**
  - good for modeling RAM registers
- BPs can be **leveled** w.l.o.g.
- Restricted versions
  - **Oblivious** - same variable queried per level
  - **Read-k** - no variable queried **> k** times on any (consistent) path

Multi-output problems

- Sorting
  - $T S = \Omega (n^2/\log n)$ [Borodin-Cook 82]
  - $T S = \Omega (n^2)$ [Beame 89]
- Matrix-vector product
  - $T S = \Omega (n^3)$ [Abrahamson 89]
- Never more than $T S = \Omega (nm)$ where m is number of outputs
  - Says nothing about complexity of decision problems

Decision problems

- Many years of progress only on very restricted models
  - Constant-width branching programs
  - Read-once branching programs
  - Oblivious branching programs
  - Syntactic read-k branching programs
    - [Borodin-Razborov-Smolensky 89]
  - Semantic read-k branching programs ?
    - Nothing for k > 1
  - General branching programs?
    - Nothing

Uniform Tradeoffs

- **SAT** is not solvable using $O(n^{1-\varepsilon})$ space if time is $n^{1+o(1)}$. [Fortnow 97]
  - uses diagonalization
  - works for co-nondeterministic TM's
- **S= $O(n^{1-\varepsilon})$ implies T=$\Omega(n^{\phi})$ for SAT.**
  [Fortnow-van Melkebeek 00] improving [Lipton-Viglas 99]
  - up to $n^{o(1)}$ advice [Tourlakis 00]
- Problems in **P**? Non-uniform computation?

Non-uniform computation

- [Beame-Saks-Thathachar FOCS 98]
  - **Syntactic read-k branching programs exponentially weaker than semantic read-twice.**
  - **$f(x)$ = "$x^T M x = 0 \pmod q$"** $\quad x \in GF(q)^n$
    - ε **nloglog n** time implies Ω**(n log n)** space for $q \sim n$
  - **$f(x)$ = "$x^T M x = 0 \pmod 3$"** $\quad x \in \{0,1\}^n$
    - **1.017n** time implies Ω **(n)** space
    - first Boolean result above time **n** for general branching programs

Non-uniform computation

- [Ajtai STOC 99]
  - **0.5log n Hamming distance** for $x \in [1,n^2]^n$
    - **kn** time implies $\Omega(n \log n)$ space
    - follows from [Beame-Saks-Thathachar 98]
  - **element distinctness** for $x \in [1,n^2]^n$
    - **kn** time implies $\Omega(n)$ space
    - requires significant extension of techniques

A breakthrough

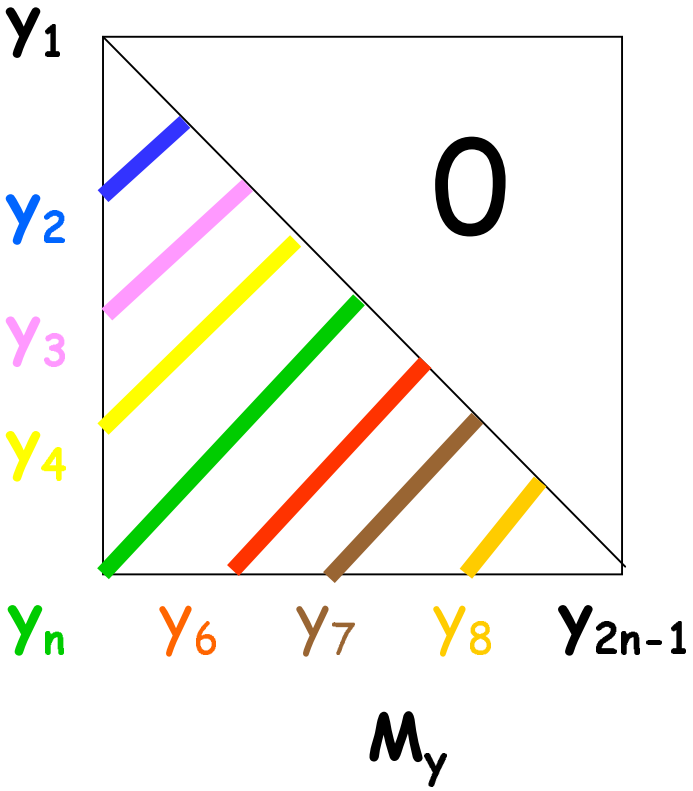
- [Ajtai FOCS 99]
  - **$f(x,y) =$ "$x^T M_y x \pmod 2$"**
    - **kn** time implies **$\Omega(n)$** space

$x \in \{0,1\}^n$

$y \in \{0,1\}^{2n-1}$

- First result for non-uniform Boolean computation showing
  - time **$O(n)$** $\Rightarrow$ space **$\omega(\log n)$**

Ajtai's Boolean function



$$f(x,y) = x^T M_y x \pmod 2$$

M_y is a modified Hankel matrix

Our new results

- Extension to ε-error randomized non-uniform algorithms
- Better time-space tradeoffs

$$T = \Omega(n\sqrt{\log/\log\log(n/S)})$$

- Apply to both element distinctness and $f(x,y) = "x^T M_y x \pmod 2"$

(m,α)-rectangles

- An **(m,α)-rectangle $R \subseteq D^X$** is a subset defined by

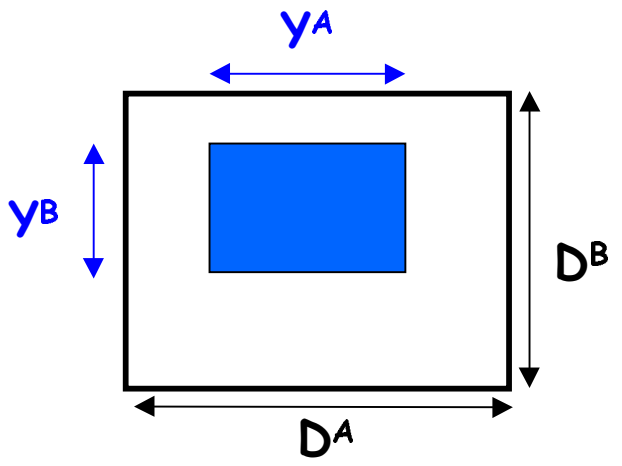
  disjoint sets $A,B \subseteq X$,

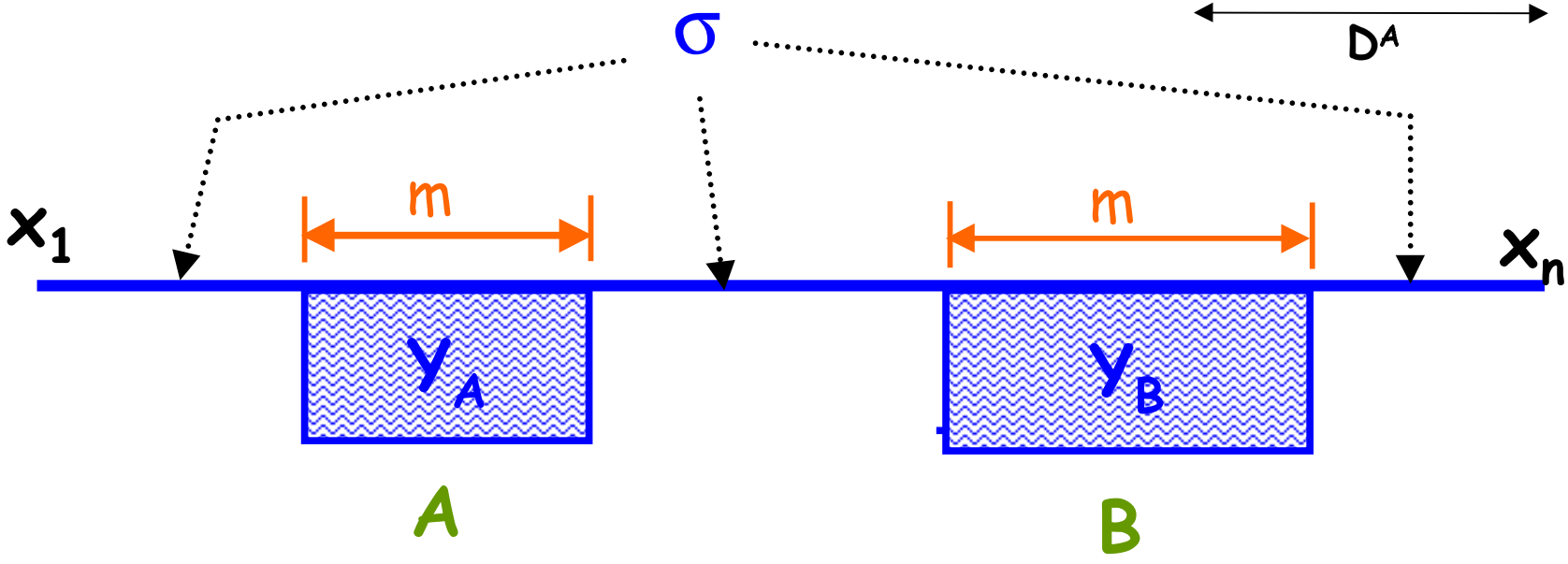
  $\sigma \in D^{\overline{A \cup B}}$

  $Y_A \subseteq D^A$, $Y_B \subseteq D^B$ such that

  - $R = \{ z \mid z_{\overline{A \cup B}} = \sigma,\ z_A \in Y_A,\ z_B \in Y_B \}$
  - $|A|, |B| \geq m$
  - $|Y_A|/|D^A|,\ |Y_B|/|D^B| \geq \alpha$

An (m,α)-rectangle





Y_A and Y_B each have density at least α

In general **A** and **B** may be interleaved in [1,n]

Key Lemma [BST 98]

- Let program **P** use
  - time **T = kn**
  - space **S**
  - accept fraction δ of its inputs in D^n
- then **P** accepts all inputs in some **(m,α)**-rectangle where
  - **m = βn**
  - α is at least $\delta\, 2^{-4(k+1)m - (S+1)r}$
  - $\beta^{-1} \sim 2^k$ and $r \sim k^2\, 2^k$

Improved key lemma [Ajtai 99 s]

- Let program **P** use
  - time **T = kn**
  - space **S**
  - accept fraction δ of its inputs in D^n
- then **P** accepts all inputs in some (m,α)-rectangle where
  - $m = \beta n$
  - α is at least $\delta\, 2^{-\beta^{1/50k} m - Sr}$
  - β^{-1} and **r** are constants depending on **k**

Proving lower bounds using the key lemmas

- Show that the desired function **f**
  - evaluates to **1** a large fraction of the time
  - is not the constant **1** on any large **(m, α)**-rectangle, where large is given by the lemma bounds

Our new key lemma

- Let program **P** use time **T = kn** space **S** and accept fraction δ of its inputs in D^n
- *Almost all inputs* **P** accepts are in **(m,**α**)**-rectangles accepted by **P** where
  - **m =** β**n**
  - α is at least $\delta\, 2^{-\beta^{1/4k^2}m - Sr}$
  - β^{-1} and **r** are $k^{\Theta(k^2)}$
- no input is in more than $O(k^2)$ rectangles

Proving lower bounds from our key lemma

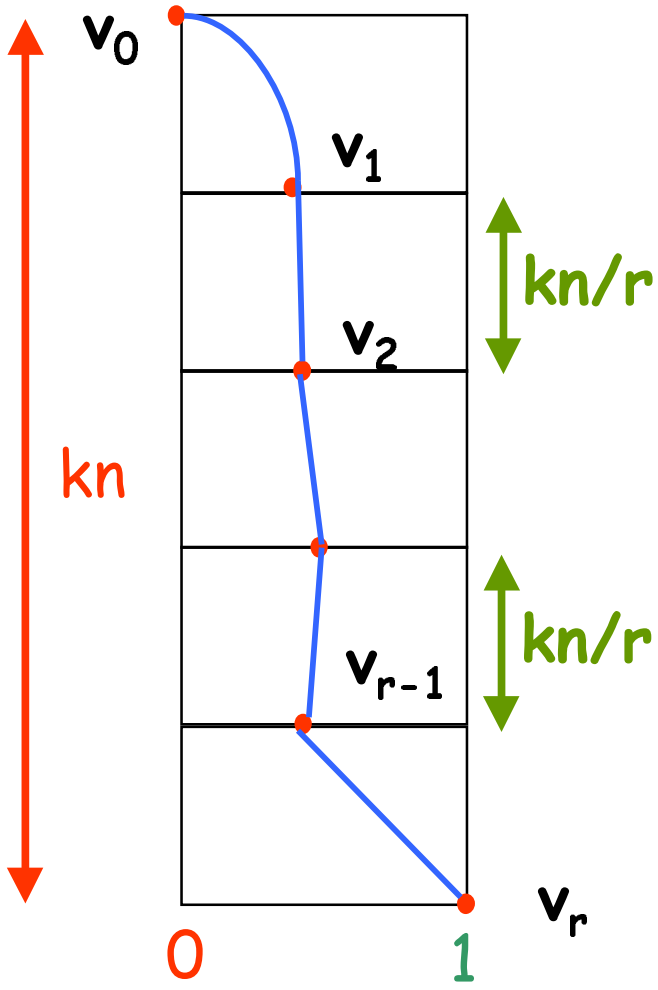
- Show that
  - for any large enough (m,α)-rectangle R

$$\Pr[f(x)=0 \mid x \in R] \geq \gamma$$

    and $\Pr[f(x)=1]=\delta$ is large
  - or … do the same for $\overline{f}$

- Gives space lower bound for $O(\gamma\delta/k^2)$-error randomized algorithms running in time kn

Proof ideas - Layering



$$f = \bigvee_{(v_1,\ldots,v_{r-1})} f_{(v_1,\ldots,v_{r-1})}$$

of $(v_1,\ldots,v_{r-1})$ is $2^{S(r-1)}$

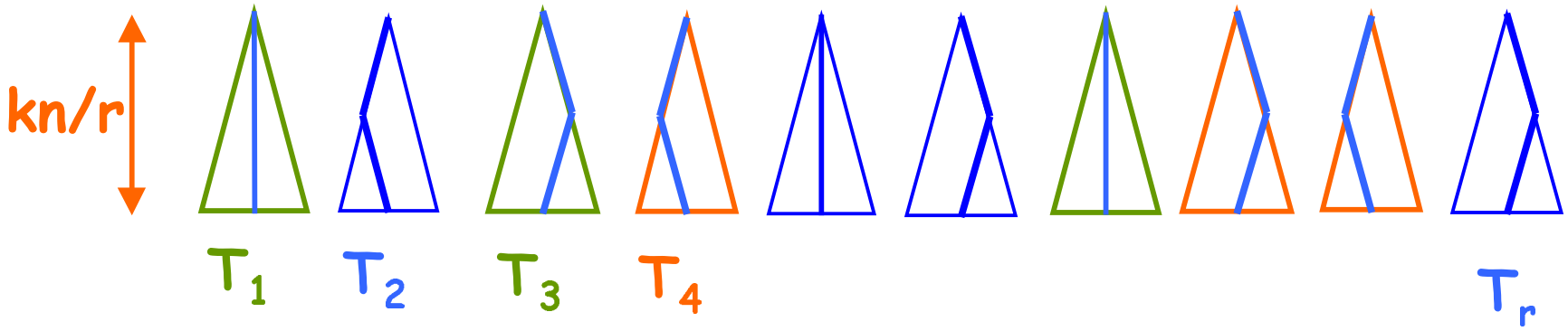
$$f_{(v_1,\ldots,v_{r-1})} = \bigwedge_{i=1}^{r} f_{v_{i-1}v_i}$$

$f_{v_{i-1}v_i}$ can be computed in **kn/r** height

(r,ε)-decision forest

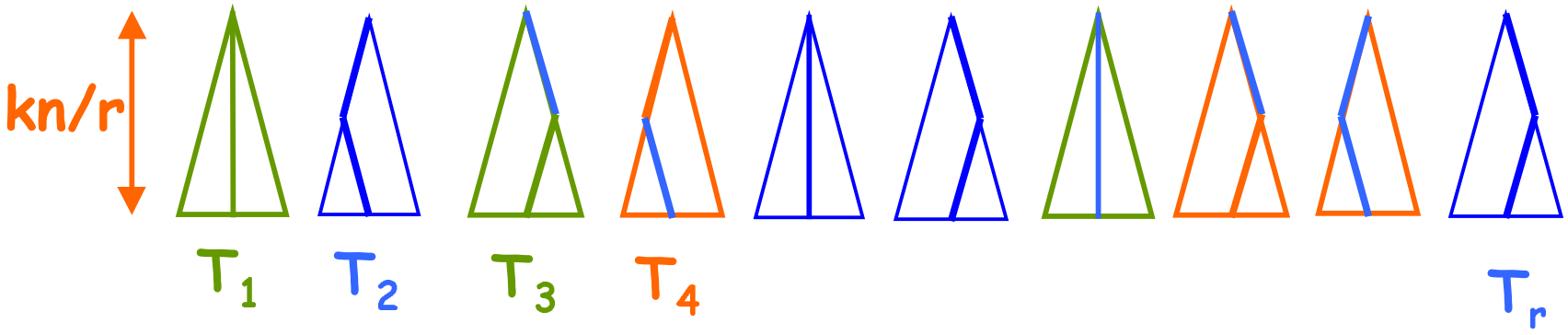
- The conjunction of **r** decision trees (BP's that are trees) of height ε**n**
  - Each $f_{(v_1,\ldots,v_{r-1})}$ is a computed by a **(r,k/r)**-decision forest
  - Only $2^{s(r-1)}$ of them
  - The various $f_{(v_1,\ldots,v_{r-1})}$ accept **disjoint** sets of inputs

Decision Forest



- Assume wlog **all** variables read on every input
- Fix an input **x** accepted by the forest
- Each tree reads only a small fraction of the variables on input **x**
- Fix two disjoint subsets of trees, **F** and **G**

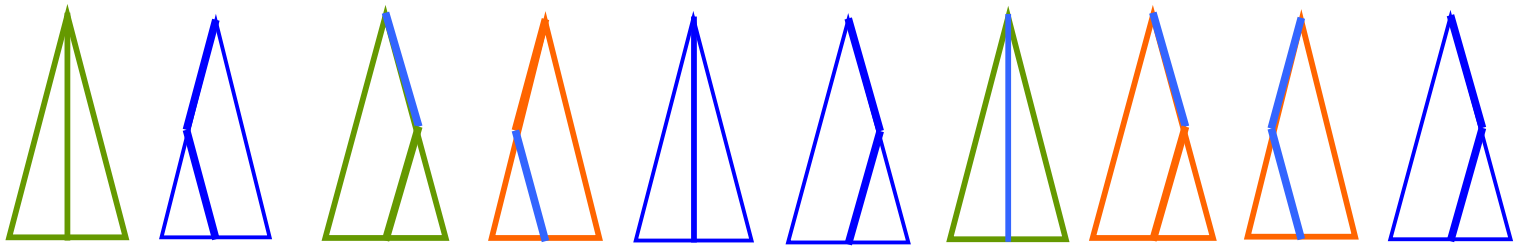
Core variables



- Can split the set of variables into
  - core(x,F)=variables read only in F
  - core(x,G)=variables read only in G
  - remaining variables
- General idea: use core(x,F) and core(x,G) as the basis for defining (**m**,α)-rectangles

Changing assignments on the cores

- For an assignment σ to **core(x,F)**, changing **x** by σ to get x^σ yields **core(x^σ,F)=core(x,F)**



  - **core(x^σ,F)** = vars not read outside **F** on input x^σ
  - vars read outside **F** on input x^σ are the same as those read outside **F** on input **x**

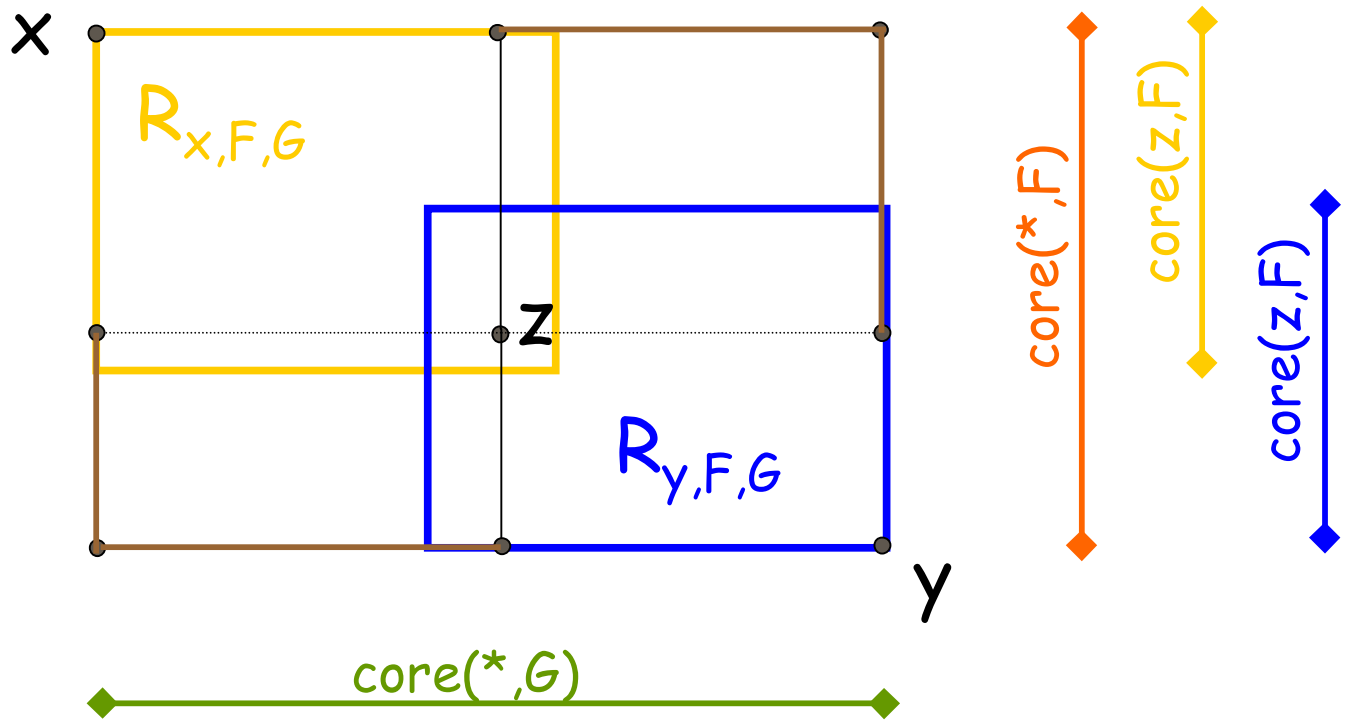
- What about ***core(x^σ,G)***?

Defining rectangles

- Y_F={σ on core(x,F)| forest accepts x^σ and core(x^σ,G)=core(x,G)}
- Y_G={τ on core(x,G)| forest accepts x^τ and core(x^τ,F)=core(x,F)}
- Claim: (x,core(x,F),core(x,G),Y_F,Y_G) defines an (m,α)-rectangle $R_{x,F,G}$ accepted by P
  - If forest accepts x, x^σ, x^τ then forest also accepts $x^{\sigma\tau}$
  - core($x^{\sigma\tau}$,F)=core(x,F) and core($x^{\sigma\tau}$,G)=core(x,G)

Partition into rectangles

- Claim: For fixed F,G, $\{R_{x,F,G}\}$ forms a partition of all accepted inputs



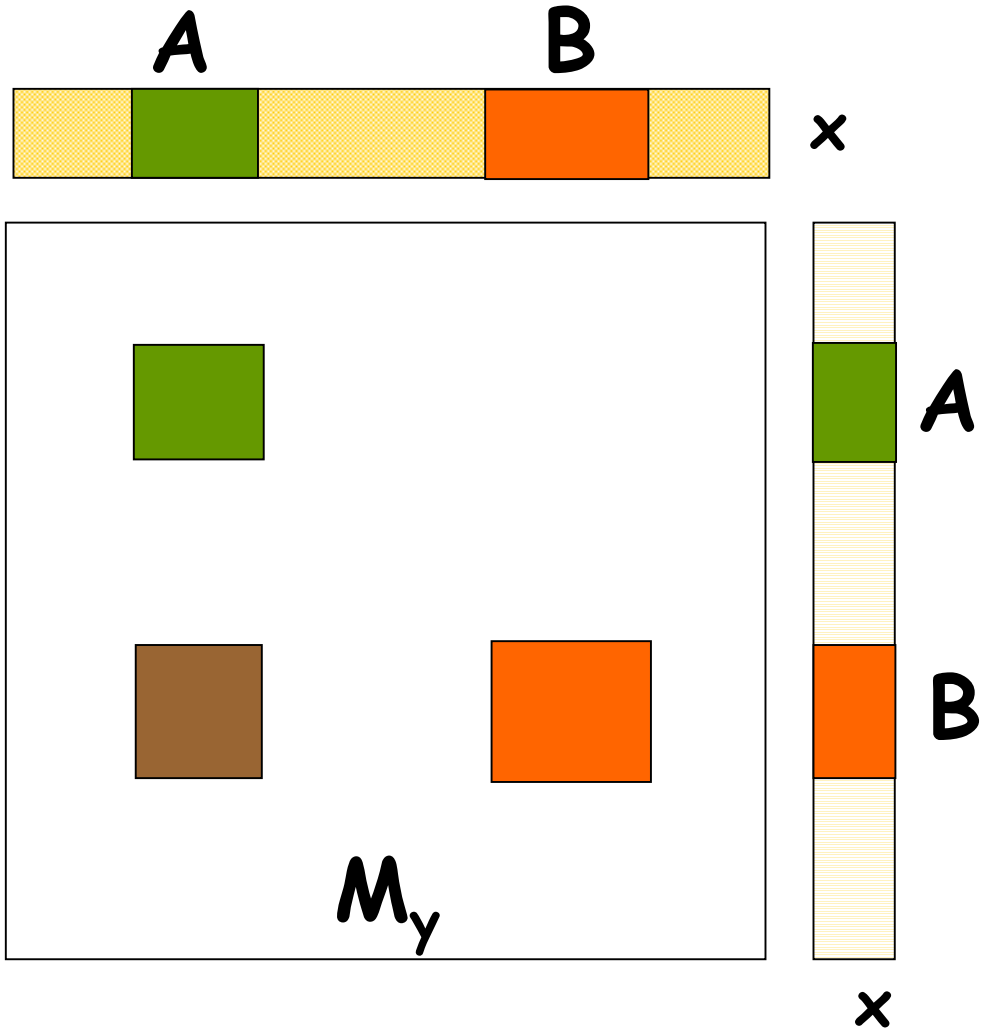
Few partitions suffice

- Only $8k^2$ pairs F,G suffice to cover almost all inputs accepted by P by large (m,α)-rectangles $R_{x,F,G}$
  - Choose F,G uniformly at random of suitable size, depending on access pattern of input
    - probability that F,G isn't good is tiny
    - one such pair will work for almost all inputs with the given access pattern
  - Only $8k^2$ sizes needed.

Access patterns

- Fix $x \in D^n$
- let ν_i = # of x_i read in j different trees
- choose F, G as random sets of qr different trees
  - $E[\,\mathbf{core}(x,F)\,] \sim \sum_j \nu_j q^j$
- Ajtai: categorize x by (t,q) s.t.
  - $\nu_i\, q^t$ dominates all ν_j, $j < t$
- Our results: categorize x by (t,q) s.t.
  - $\nu_t q^t >> \sum_{j \neq t} \nu_j q^j$

$x^T M_y x$ on an (m,α)-rectangle



For every σ on $\overline{AUB}$,

$$f(x_{AUB}, \sigma, y) = x_A^T M_{AB} x_B + g(x_A, y) + h(x_B, y)$$

Rectangles, rank, & rigidity

- largest rectangle on which $x_A^T M x_B$ is constant has $\alpha \geq 2^{-\text{rank}(M)}$
  - [Borodin-Razborov-Smolensky 89]
- **Lemma** [Ajtai 99] Can fix y s.t. every $\beta n \times \beta n$ minor M_{AB} of M_y has
  $\text{rank}(M_{AB}) \geq c\beta n/\log^2(1/\beta)$
  - improvement of bounds of [Beame-Saks-Thathachar 98] & [Borodin-Razborov-Smolensky 89] for Sylvester matrices

High rank implies balance

- For any rectangle $Y_A \times Y_B \subseteq \{0,1\}^A \times \{0,1\}^B$ with $\mu(Y_A \times Y_B) \geq |A||B|2^{3-\text{rank}(M)}$

  $$\Pr[\, x_A^T M x_B = 1 \mid x_A \in Y_A,\ x_B \in Y_B] \geq 1/32$$

  $$\Pr[\, x_A^T M x_B = 0 \mid x_A \in Y_A,\ x_B \in Y_B] \geq 1/32$$

  - derived from result for inner product in **r** dimensions
- So rigidity also implies balance for all large rectangles and so $T = \Omega(n\sqrt{\log/\log\log(n/S)})$
- Also follows for element distinctness
  - [Babai, Frankl, Simon FOCS 86]

Open Problems

- Improve this to
  - $T=\Omega(n\log(n/S))$?
  - $T=\Omega(n^2/S)$?
- Take better advantage of space bound?
  - now used only to count number of decision forests

Limitations of current method

- Need $n > T/r$ = decision tree height
  - else all functions trivial
  - so $r > T/n$
- A decision forest works on a 2^{-Sr} fraction of the accepted inputs
  - only place space bound is used
- So need $Sr < n$ else d.f. need only work on one input
  - implies $ST/n < n$, i.e. $T < n^2/S$