Hardness of Function Composition for Semantic Read once Branching Programs

Jeff Edmonds, Venkatesh Medabalimi, Toniann Pitassi

June 23, 2018
P: Polynomial time computable functions.

L: Functions computable in logarithmic space.

\[ L \subset P \]
**Branching Programs**

\[ f(x_1, x_2, \ldots, x_n) \rightarrow \{0, 1\} \quad x_i \in \{0, 1\}, \forall i \in [n] \]

**Definition**

Deterministic Branching program

- DAG with a source node and two sinks, 1-sink (for accept) and 0-sink (for reject).

- Each non-sink node is labeled by some \( x_i \), outdegree 2 with an edge each for \( x_i = 0 \) and \( x_i = 1 \).
**Non-det Branching Programs**

**Definition**

Non-deterministic Branching program (NBP)

- allow unlabelled guessing nodes and arbitrary out-degree.

The size of a NBP = number of labelled nodes.
NBP computing \( f : \{0,1\}^n \rightarrow \{0,1\} \)

\[ f(u) = 1 \iff \exists \text{ a path from source to accept node that is consistent with input } u. \]
**Program Size and Space Complexity of computing f**

- \( BP(f_n) = \min_{B \in \text{BP computing } f_n} \text{size } (B) \)

- \( S(f_n) = \min_{T \in \text{non-uniform TMs computing } f_n} \text{space complexity } (T) \)

- \( \log(P(f_n)) \approx S(f_n) \quad \text{[Cobham ‘66]} \)
It is easy to show functions with high $BP(f_n)$ exist.
Big Picture

- It is easy to show functions with high $BP(f_n)$ exist.
- Can we show that some function in $P$ requires exponential size $BP$?
  - amounts to showing $L \subset \textbf{P}$.
<table>
<thead>
<tr>
<th>Formulas</th>
<th>Branching Programs</th>
<th>Circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L(f)$</td>
<td>$\geq$</td>
<td>$BP(f)$</td>
</tr>
</tbody>
</table>

$\Omega(n^3)$, $\Omega(n^2 \log 2^n)$, $\Omega(n)$
### Formulas

<table>
<thead>
<tr>
<th>Formulas</th>
<th>Branching Programs</th>
<th>Circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L(f)$ $\geq$ $BP(f)$ $\geq$ $\frac{1}{3}C(f)$</td>
<td>$\Omega \left( \frac{n^2}{\log^2 n} \right)$</td>
<td>$\Omega(n)$</td>
</tr>
</tbody>
</table>

- **Random Restrictions**: $\Omega(n^3)$
- **Nechiporuk**: $\Omega \left( \frac{n^2}{\log^2 n} \right)$
- **Gate Elimination**: $\Omega(n)$

**Open Problems**

- **BPs and other Computation Models**
- **Formulas Branching Programs Circuits**
- **L** $(f)$ $\geq$ $BP(f)$ $\geq$ $\frac{1}{3}C(f)$
- **Random Restrictions**: $\Omega(n^3)$
- **Nechiporuk**: $\Omega \left( \frac{n^2}{\log^2 n} \right)$
- **Gate Elimination**: $\Omega(n)$
Bounded Width: same as $\text{NC}^1$, Barrington's characterization.
Restricted Branching Programs

- Bounded Width: same as $\text{NC}^1$, Barrington's characterization.
- Length Restricted: give Time-Space tradeoffs.
**Time-Space Tradeoffs**

- $t \leq cn \implies s = 2^{\Omega(n)}$  
  Jukna’09
**Time-Space tradeoffs**

- \( t \leq cn \implies s = 2^{\Omega(n)} \)  
  \( \text{Jukna'09} \)

- culmination results by Ajtai '99 and Beame, Jayram, Saks '01
**Time-Space tradeoffs**

- \( t \leq cn \implies s = 2^{\Omega(n)} \) \hspace{1cm} \text{Jukna'09}

- culmination results by Ajtai ‘99 and Beame, Jayram, Saks ‘01

- We look at:
  
  
  time-space tradeoffs
  
  for
  
  *iterated function composition*.
**Read Once**

- **Syntactic** read once: Along any path from source to sink any variable appears at most once.

![Diagram of a read-once branching program]

**Syntactic** read once: Along any path from source to sink any variable appears at most once.
**Read Once**

- **Syntactic** read once: Along any path from source to sink any variable appears at most once.

- **Semantic** read once: Along any consistent path from source to sink no variable is read more than once.
The *Exact Perfect matching* function ($EPM_n$): accept a matrix iff it is a permutation matrix.

Jukna and Razborov ‘98 showed

**Theorem**

*Every syntactic read once NBP computing $EPM_n$ must have size $2^\Omega(n)$.*
**Theorem (Jukna)**

$EPM_n$ can be solved by a semantic read once NBP of size $O(n^3)$.

$$
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 
\end{bmatrix}
$$
Theorem (Jukna)

\( EPM_n \) can be solved by a semantic read once NBP of size \( O(n^3) \).

\[
\begin{bmatrix}
  0 & 0 & 1 \\
-1 & 0 & 0 \\
  0 & 1 & 0 \\
\end{bmatrix}
\]
THEOREM (JUKNA)

$EPM_n$ can be solved by a semantic read once NBP of size $O(n^3)$. 

$$
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
$$
**Theorem (Jukna)**

\( EPM_n \) can be solved by a semantic read once NBP of size \( O(n^3) \).

\[
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix}
\]

Sees only 1s
**Theorem (Jukna)**

$EPM_n$ can be solved by a semantic read once NBP of size $O(n^3)$.

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Sees only 1s  Sees only 0s
**Theorem (Jukna)**

\( EPM_n \) can be solved by a semantic read once NBP of size \( O(n^3) \).

\[
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\]

Sees only 1s  Sees only 0s
**Embedded Rectangles**

\[ C_{red} \times \{w\} \times C_{blue} \subseteq [D]^n \]

\[
\begin{align*}
C_{red} & \subseteq D^A \\
A & \subset [n] \\
C_{blue} & \subseteq D^B \\
B & \subset [n]
\end{align*}
\]

\[
\begin{bmatrix}
012210101 \\
012010201 \\
101012202
\end{bmatrix} 
\times \{10210001\} 
\times 
\begin{bmatrix}
012001012 \\
201101002 \\
110020120 \\
210200120
\end{bmatrix}
\]
**Embedded Rectangles**

$$C_{\text{red}} \times \{w\} \times C_{\text{blue}} \subseteq [D]^n$$
KRW Conjecture

- An approach to separating $\text{NC}^1$ from $\text{NC}^2$.
- KRW conjecture that for every random $f$ and $\forall g$,
  \[ D(fog) \geq \epsilon D(f) + D(g). \]
- KRW conjecture on formula size of a composed function $fog$.
  \[ L(fog) \approx L(f)L(g) \]
How does space compose?
**Figure**: $\text{TEP}_d^h$ that is height 4, degree 2.
Is $BP(TEP_h^d) = \Omega(k^h)$??
Is $BP(TEP^h_2) = \Omega(k^h)$? 

$\Rightarrow \text{L} \subset \text{P}$
$K^{-\epsilon}$ density

$K^{1-\epsilon}$

$Tree_{\mathcal{F},\epsilon}(\cdot)$
Theorem

For any $h$, and $k$ sufficiently large, there exists $\epsilon$ and $\vec{F}$ such that any $k$-ary nondeterministic semantic read-once branching program for ternary $Tree_{\vec{F},\epsilon}$ requires size at least

$$\left(\frac{k}{\log k}\right)^h.$$
Black White pebbling Upperbound

Hardness of Function Composition for Semantic Read once Branching Programs

Jeff Edmonds, Venkatesh Medabalimi, Toniann Pitassi

Motivation
Branching Programs
Lower Bounds against Function Composition
Proof Overview
∃ A special query state for each input
Special low entropy node in the Tree
Two-way Product Sets
Conclusion
Open Problems
\[ \frac{1}{2} (d - 1) h + 1 \] PEBBLES AT THIS MOMENT
Guess the remaining siblings

Motivation

Branching Programs

Lower Bounds against Function Composition

Proof Overview

- A special query state for each input
- Special low entropy node in the Tree

Two-way Product Sets

Conclusion

Open Problems
Infer the root

∃A special query state for each input
Special low entropy node in the Tree
Two-way Product Sets
Conclusion

Open Problems
Hardness of Function Composition for Semantic Read once Branching Programs

Jeff Edmonds, Venkatesh Medabalimi, Toniann Pitassi

Motivation
Branching Programs
Lower Bounds against Function Composition
Proof Overview
∃ A special query state for each input
Special low entropy node in the Tree
Two-way Product Sets
Conclusion
Open Problems
Verify Guesses
Definition

A Latin Cube is a function $f : [k]^3 \rightarrow [k]$ such that $f$ is invertible in each of its coordinates. Equivalently, every element of $[k]$ appears exactly once along every row, column and leg in the cube $[k]^3$. 

Lower bound uses Invertible functions
**Definition**

Any element in $[k]$ appears at most 4 times along any row, column or leg in the cube $[k]^3$. 

**4-invertible function, $f : [k]^3 \rightarrow [k]$**
Proof Overview

A small BP for a $Tree_{F,\epsilon}$  $\implies$  $Tree_{F,\epsilon}$ accepts a large rectangle of inputs over its leaves
Proof Overview

- A small BP for a $Tree_{F,ε}$ $\implies$ $Tree_{F,ε}$ accepts a large rectangle of inputs over its leaves

- A large rectangle over leaves $\implies$ $\exists$ a special node $v^*$ in the tree whose $F_{v^*}$ can be described in few bits.
Proof Overview

- A small BP for a $Tree_{F,\epsilon}$ $\implies$ $Tree_{F,\epsilon}$ accepts a large rectangle of inputs over its leaves

- A large rectangle $\implies$ $\exists$ a special node $v^*$ in the tree whose $F_{v^*}$ can be described in few bits.

- Show that the distribution on $\tilde{F}$ is rich or sufficiently random looking that one cannot save these bits.
FOR EVERY ACCEPTING INPUT ∃ A SPECIAL QUERY STATE:

A white subtree has at least a fraction of leaf node which are white

A red subtree has at least a fraction of leaf node which are red

∃ A special query state for each input

Special low entropy node in the Tree

Two-way Product Sets

Conclusion
Few special states \( \implies \exists \) a large embedded rectangle over leaves

- Choose a popular labelled path down the tree.
- Choose a popular red variable for the first red-subtree. Prune the input set. Continue to choose \( h \) red variables one for each red-subtree. Similarly for each blue-subtree.
- Fix the remaining variables in \([n]-\text{Red-Blue}\) to the most popular projection ‘\( w \)’.
∃ A LARGE EMBEDDED RECTANGLE OVER THE LEAVES.

\[ B_h(x, y) = F_h(A_h(x), B_{h-1}(x, y)C_h(y)) \]

\[ B_i(x, y) = F_i(A_i(x), B_{i-1}(x, y)C_i(y)) \]

Motivation
Branching Programs
Lower Bounds against Function Composition
Proof Overview
∃ A special query state for each input
Special low entropy node in the Tree
Two-way Product Sets
Conclusion
Open Problems
∃ A node v* at which leaves in both red and blue trees take a lot of values.
∃ A node \( v^* \) at which both red and blue child take a lot of values.
∃ A NODE V* WITH LOW ENTROPY ON A TWO-WAY PRODUCT SET

\[ B_h(x, y) = F_h(A_h(x), B_{h-1}(x, y)C_h(y)) \]

\[ B_i(x, y) = F_i(A_i(x), B_{i-1}(x, y)C_i(y)) \]

Motivation
Branching Programs
Lower Bounds against Function Composition
Proof Overview
∃ A special query state for each input
Special low entropy node in the tree
Two-way Product Sets
Conclusion
Open Problems
Two-way Product Set at $v^*$, in $F_{v^*}()$
Two-way Product Set at \( v^* \), in \( F_{v^*}(\cdot) \)

- \( A, C \subset [k] \)
- \( |A| = |C| = r << k \)
- \( S_r = \{(x, Q(x, y), y) | x \in A, y \in C\} \)  \( |S| = r^2 \)
Entropy of spread on Two-way Product Set can’t be low

A, C ⊂ [k] \quad |A| = |C| = r << k

S_r = \{(x, B(x, y), y) \mid x \in A, y \in C\} \quad |S| = r^2

∀ Two-way Product Sets S_r and target set T_\epsilon

\Pr_{f \sim U(All 4-invertible cubes)}[f(S_r) \subseteq T_\epsilon] \leq \frac{1}{k^\epsilon r^2}
**Figure**: This figure depicts a label $L_{\vec{F}}$ associated with a problem instance $Tree_{\vec{F}}$.
\exists \text{many } \vec{F} \text{ that remain unaccounted without such a special label.}
Open Problems

- More general time space tradeoffs for composition.
- Exponential lower bound for boolean semantic NBPs for some problem in P.
- Super-quadratic lower bound for BPs via understanding composition fog where g is element distinctness.
Thank You!