Parity not in AC^0

SK Naseer

IIT Kanpur

April 7, 2014

Overview

1. Definitions
2. $PARITY \notin AC^0$
3. References

Definitions

Definition 1: (Boolean circuits)[ASB]

For every $n \in N$, an n-input single output Boolean circuit is a directed acyclic graph with n sources (vertices with no incoming edges) and one sink (vertex with no outgoing edges). All non-source vertices are called gates and are labeled with one of OR, AND, and NOT. The size of C, denoted by $|C|$, is the number of vertices in it.

Definitions

Definition 1: (Boolean circuits)[ASB]

For every $n \in N$, an n-input single output Boolean circuit is a directed acyclic graph with n sources (vertices with no incoming edges) and one sink (vertex with no outgoing edges). All non-source vertices are called gates and are labeled with one of OR, AND, and NOT. The size of C, denoted by $|C|$, is the number of vertices in it.

If C is a Boolean circuit, and $x \in \{0, 1\}$ is some input, then the output of C on x, denoted by $C(x)$, is defined in the natural way. More formally, for every vertex v of C we give it a value $val(v)$ as follows: if v is the i^{th} input vertex then $val(v) = x_i$ and otherwise $val(v)$ is defined recursively by applying v's logical operation on the values of the vertices connected to v. The output $C(x)$ is the value of the output vertex.

Definitions

Definition 2: (Circuit families and language recognition)[ASB]

Let $T : N \rightarrow N$ be a function. A $T(n)$-size circuit family is a sequence $\{C_n\}\ n \in N$ of Boolean circuits, where C_n has n inputs and a single output, and its size $|C_n| \leq T(n)$ for every n.

Definitions

Definition 2: (Circuit families and language recognition)[ASB]

Let $T : N \to N$ be a function. A $T(n)$-size circuit family is a sequence $\{C_n\}$ $n \in N$ of Boolean circuits, where C_n has n inputs and a single output, and its size $|C_n| \leq T(n)$ for every n.

We say that a language L is in SIZE($T(n)$) if there exists a $T(n)$-size circuit family $\{C_n\}$ $n \in N$ such that for every $x \in \{0, 1\}^n$, $x \in L \Leftrightarrow C_n(x) = 1$.

Definitions

Definition 3: (Class AC^0)

Class of all decision problems that are decided by circuit families of :

- polynomial size,

Definitions

Definition 3: (Class AC^0)

Class of all decision problems that are decided by circuit families of :

- polynomial size,
- constant depth,

Definitions

Definition 3: (Class AC^0)

Class of all decision problems that are decided by circuit families of :

- polynomial size,
- constant depth,
- unbounded fan-in

Definitions

Definition 4: (k-CNF)

A boolean formula that is an AND of OR's where each OR involves at most k variables.

Definitions

Definition 4: (k-CNF)

A boolean formula that is an AND of OR's where each OR involves at most k variables.

Definition 5: (k-DNF)

A boolean formula that is an OR of AND's where each AND involves at most k variables.

Definitions

Definition 6: (Random Restriction)

Let f is a function on n variables. A random resrtiction ρ is a partial assignment that assigns random values to $t < n$ randomly selected variables of f. We denote the random restriction of f under ρ by $f|_\rho$. That is, $f|_\rho$ takes an assignment τ to the variables not assigned by ρ as input, and outputs f applied to ρ and τ .

Theorem

Theorem 1([FSS81, Ajt83])

Let **PARITY** $= \{x \in \{0, 1\}^n : x$ has odd number of 1's$\}$.
Then **PARITY** $\notin AC^0$.

PARITY $\notin AC^0$

Proof Sketch [ASB]

The main tool in the proof of Theorem 1 is the concept of random restrictions. Let f be a function computable by a depth d circuit of polynomial size and suppose that we choose at random a vast majority (i.e., $n \quad n^{\epsilon}$ for some constant $\epsilon > 0$ depending on d) of the input variables and fix each such variable to be either 0 or 1 at random. By Hastad's switching lemma, it is clear that with positive probability, the function f subject to this restriction is constant (i.e., it is either always zero or always one). Since the parity function cannot be made a constant by fixing values to a subset of the variables, it follows that it cannot be computed by a constant depth polynomial-sized circuit.

Theorem

Theorem

Any AC^0 circuit of size S and depth d can be simplified so that:

1. All gates have fan-out 1; the circuit is a tree.

Theorem

Theorem

Any AC^0 circuit of size S and depth d can be simplified so that:

1. All gates have fan-out 1; the circuit is a tree.
2. All not gates are at the input level of the circuit; that is, the circuit has $2n$ input wires, where the extra n input wires are negations of original n input wires.

Theorem

Theorem

Any AC^0 circuit of size S and depth d can be simplified so that:

1. All gates have fan-out 1; the circuit is a tree.
2. All not gates are at the input level of the circuit; that is, the circuit has $2n$ input wires, where the extra n input wires are negations of original n input wires.
3. At each level of the tree there are either only AND gates or only OR gates. And no two consecutive levels have same type of gates.

Theorem

Theorem

Any AC^0 circuit of size S and depth d can be simplified so that:

1. All gates have fan-out 1; the circuit is a tree.
2. All not gates are at the input level of the circuit; that is, the circuit has $2n$ input wires, where the extra n input wires are negations of original n input wires.
3. At each level of the tree there are either only AND gates or only OR gates. And no two consecutive levels have same type of gates.
4. The bottom level gates have fan-in 1.

Theorem

Theorem

Any AC^0 circuit of size S and depth d can be simplified so that:

1. All gates have fan-out 1; the circuit is a tree.
2. All not gates are at the input level of the circuit; that is, the circuit has $2n$ input wires, where the extra n input wires are negations of original n input wires.
3. At each level of the tree there are either only AND gates or only OR gates. And no two consecutive levels have same type of gates.
4. The bottom level gates have fan-in 1.

Moreover the simplified circuit has size poly(S) and depth O(d).

Proof:**PARITY** $\notin AC^0$

- Assume that **PARITY** $\in AC^0$.

Proof:**PARITY** $\notin AC^0$

- Assume that **PARITY** $\in AC^0$.
- Then by definition, $\exists$ an AC^0 circuit of depth d which decides **PARITY**.

Proof:**PARITY** $\notin AC^0$

- Assume that **PARITY** $\in AC^0$.
- Then by definition, $\exists$ an AC^0 circuit of depth d which decides **PARITY**.
- Simplify the circuit using previous theorem.

Proof:**PARITY** $\notin AC^0$

- Assume that **PARITY** $\in AC^0$.
- Then by definition, $\exists$ an AC^0 circuit of depth d which decides **PARITY**.
- Simplify the circuit using previous theorem.
- Let n^b be the upper bound on the number of gates in the simplified circuit.

Proof:**PARITY** $\notin AC^0$

- Assume that **PARITY** $\in AC^0$.
- Then by definition, $\exists$ an AC^0 circuit of depth d which decides **PARITY**.
- Simplify the circuit using previous theorem.
- Let n^b be the upper bound on the number of gates in the simplified circuit.
- At each step, with high probability we reduce the depth of the circuit by 1 by randomly restricting some variables.

Proof:**PARITY** $\notin AC^0$

- Assume that **PARITY** $\in AC^0$.
- Then by definition, $\exists$ an AC^0 circuit of depth d which decides **PARITY**.
- Simplify the circuit using previous theorem.
- Let n^b be the upper bound on the number of gates in the simplified circuit.
- At each step, with high probability we reduce the depth of the circuit by 1 by randomly restricting some variables.
- We do this untill the depth of circuit becomes 2.

Proof:**PARITY** $\notin AC^0$

- Let n_i denote the number of unrestricted variables after step i.

Proof:**PARITY** $\notin AC^0$

- Let n_i denote the number of unrestricted variables after step i.
- We restrict n_i - $\sqrt{n_i}$ variables at step i+1.

Proof:**PARITY** $\notin AC^0$

- Let n_i denote the number of unrestricted variables after step i.
- We restrict n_i - $\sqrt{n_i}$ variables at step i+1.
- Since n_0 is n, we have $n_i = n^{\frac{1}{2^i}}$.

Proof:**PARITY** $\notin AC^0$

- Let n_i denote the number of unrestricted variables after step i.
- We restrict n_i - $\sqrt{n_i}$ variables at step i+1.
- Since n_0 is n, we have $n_i = n^{\frac{1}{2^i}}$.
- Let fan-in of bottom level after i^{th} step be atmost k_i.

Proof:**PARITY** $\notin AC^0$

- Let n_i denote the number of unrestricted variables after step i.
- We restrict n_i - $\sqrt{n_i}$ variables at step i+1.
- Since n_0 is n, we have $n_i = n^{\frac{1}{2^i}}$.
- Let fan-in of bottom level after i^{th} step be atmost k_i.
- Suppose that bottom level of circuit contains AND gates. Therefore the level above it contains OR gates.

Proof:**PARITY** $\notin AC^0$

- Let n_i denote the number of unrestricted variables after step i.
- We restrict n_i - $\sqrt{n_i}$ variables at step $i+1$.
- Since n_0 is n, we have $n_i = n^{\frac{1}{2^i}}$.
- Let fan-in of bottom level after i^{th} step be atmost k_i.
- Suppose that bottom level of circuit contains AND gates. Therefore the level above it contains OR gates.
- Observe that each OR gate computes a k_i-DNF.

Proof:**PARITY** $\notin AC^0$

- Let n_i denote the number of unrestricted variables after step i.
- We restrict n_i - $\sqrt{n_i}$ variables at step i+1.
- Since n_0 is n, we have $n_i = n^{\frac{1}{2^i}}$.
- Let fan-in of bottom level after i^{th} step be atmost k_i.
- Suppose that bottom level of circuit contains AND gates. Therefore the level above it contains OR gates.
- Observe that each OR gate computes a k_i-DNF.
- Apply switching lemma to the function computed by this gate.

Switching Lemma: Statement [ASB]

If f is a function that is expressible as a $k-$DNF and ρ is a random restriction that assigns random values to t randomly selected input bits, then $\forall s \geq 2$

$$\Pr_{\rho}[f|_{\rho} \text{ is not expressible as } s\text{-}CNF] \leq \left(\frac{(n-t)k^{10}}{n} \right)^{s/2} \tag{1}$$

Switching Lemma: Statement [ASB]

If f is a function that is expressible as a $k-$DNF and ρ is a random restriction that assigns random values to t randomly selected input bits, then $\forall s \geq 2$

$$\Pr_{\rho}[f|_{\rho} \text{ is not expressible as } s\text{-}CNF] \leq \left(\frac{(n-t)k^{10}}{n}\right)^{s/2} \tag{1}$$

Note that by applying this lemma to $\neg f$ we get the same result with the terms DNF and CNF interchanged.

Proof:**PARITY** $\notin AC^0$

- By switching lemma, with probability 1 - $\left(\frac{k_i^{10}}{n^{\frac{1}{2^{i+1}}}}\right)^{\frac{k_{i+1}}{2}}$, we can convert this k_i-DNF to k_{i+1}-CNF.

Proof:**PARITY** $\notin AC^0$

- By switching lemma, with probability 1 - $\left(\frac{k_i^{10}}{n^{\frac{1}{2^{i+1}}}}\right)^{\frac{k_{i+1}}{2}}$, we can convert this k_i-DNF to k_{i+1}-CNF.
- We want this probability to be atleast 1- $\frac{1}{10n^b}$ for any step i and for sufficiently large n.

Proof:**PARITY** $\notin AC^0$

- By switching lemma, with probability 1 - $\left(\frac{k_i^{10}}{n^{\frac{1}{2^{i+1}}}}\right)^{\frac{k_{i+1}}{2}}$, we can convert this k_i-DNF to k_{i+1}-CNF.
- We want this probability to be atleast 1- $\frac{1}{10n^b}$ for any step i and for sufficiently large n.
- Note that we are free to choose any $k_i \geq 2$.

Proof:**PARITY** $\notin AC^0$

- By switching lemma, with probability 1 - $\left(\frac{k_i^{10}}{n^{\frac{1}{2^{i+1}}}}\right)^{\frac{k_{i+1}}{2}}$, we can convert this k_i-DNF to k_{i+1}-CNF.
- We want this probability to be atleast 1- $\frac{1}{10n^b}$ for any step i and for sufficiently large n.
- Note that we are free to choose any $k_i \geq 2$.
- So we choose $k_i = 10b2^i$.

Proof:**PARITY** $\notin AC^0$

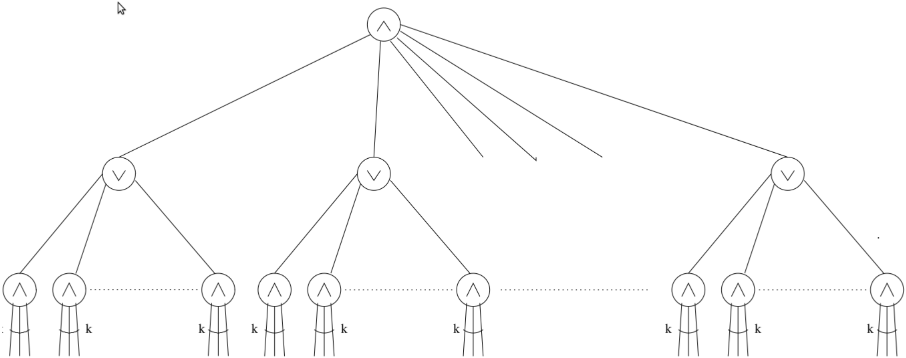

Figure: Circuit before Hastad switching transformation.[ASB]

Proof:**PARITY** $\notin AC^0$

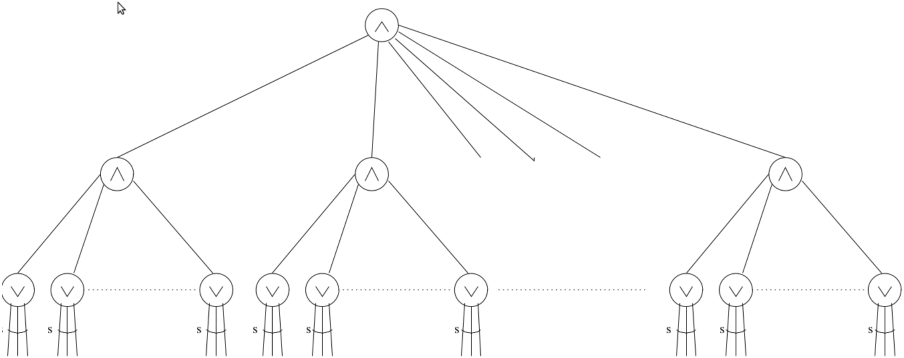

Figure: Circuit after Hastad switching transformation.[ASB]

Proof:**PARITY** $\notin AC^0$

- By switching lemma, with probability 1 - $\left(\frac{k_i^{10}}{n^{\frac{1}{2^{i+1}}}}\right)^{\frac{k_{i+1}}{2}}$, we can convert this k_i-DNF to k_{i+1}-CNF.
- we want this probability to be atleast 1- $\frac{1}{10n^b}$ for any step i and for sufficiently large n.
- Note that we are free to choose any $k_i \geq 2$.
- So we choose $k_i = 10b2^i$.

Proof:**PARITY** $\notin AC^0$

- By switching lemma, with probability 1 - $\left(\frac{k_i^{10}}{n^{\frac{1}{2^{i+1}}}}\right)^{\frac{k_{i+1}}{2}}$, we can convert this k_i-DNF to k_{i+1}-CNF.
- we want this probability to be atleast 1- $\frac{1}{10n^b}$ for any step i and for sufficiently large n.
- Note that we are free to choose any $k_i \geq 2$.
- So we choose $k_i = 10b2^i$.
- Since the top level gate of k_{i+1}-CNF is AND, and since gates can have unbounded fan-in, we can merge this AND gate with the AND gate above it reducing the depth of the circuit by 1.

Proof:**PARITY** $\notin AC^0$

- The symmetric reasoning applies in the case the bottom level contains OR gates. In this case we use the switching lemma to transform the k_i-CNF to k_{i+1}-DNF.

Proof:**PARITY** $\notin AC^0$

- The symmetric reasoning applies in the case the bottom level contains OR gates. In this case we use the switching lemma to transform the k_i-CNF to k_{i+1}-DNF.
- Note that we apply the lemma atmost once on each gate. And there are n^b gates.

Proof:**PARITY** $\notin AC^0$

- The symmetric reasoning applies in the case the bottom level contains OR gates. In this case we use the switching lemma to transform the k_i-CNF to k_{i+1}-DNF.
- Note that we apply the lemma atmost once on each gate. And there are n^b gates.
- By union bound , with probability $\frac{9}{10}$, if we apply this reduction $d-2$ times we get a circuit with depth 2.

Proof:**PARITY** $\notin AC^0$

- The symmetric reasoning applies in the case the bottom level contains OR gates. In this case we use the switching lemma to transform the k_i-CNF to k_{i+1}-DNF.
- Note that we apply the lemma atmost once on each gate. And there are n^b gates.
- By union bound , with probability $\frac{9}{10}$, if we apply this reduction $d-2$ times we get a circuit with depth 2.
- But this is either a k-CNF or a k-DNF where $k = k_{d-2}$.

Proof:**PARITY** $\notin AC^0$

- The symmetric reasoning applies in the case the bottom level contains OR gates. In this case we use the switching lemma to transform the k_i-CNF to k_{i+1}-DNF.
- Note that we apply the lemma atmost once on each gate. And there are n^b gates.
- By union bound , with probability $\frac{9}{10}$, if we apply this reduction $d-2$ times we get a circuit with depth 2.
- But this is either a k-CNF or a k-DNF where $k = k_{d-2}$.
- We can make such a formula constant by fixing atmost k variables.

Proof:**PARITY** $\notin AC^0$

- The symmetric reasoning applies in the case the bottom level contains OR gates. In this case we use the switching lemma to transform the k_i-CNF to k_{i+1}-DNF.
- Note that we apply the lemma atmost once on each gate. And there are n^b gates.
- By union bound , with probability $\frac{9}{10}$, if we apply this reduction $d-2$ times we get a circuit with depth 2.
- But this is either a k-CNF or a k-DNF where $k = k_{d-2}$.
- We can make such a formula constant by fixing atmost k variables.
- But the parity function can not be made constant under any restriction of less than n inputs.

Proof:**PARITY** $\notin AC^0$

- The symmetric reasoning applies in the case the bottom level contains OR gates. In this case we use the switching lemma to transform the k_i-CNF to k_{i+1}-DNF.
- Note that we apply the lemma atmost once on each gate. And there are n^b gates.
- By union bound , with probability $\frac{9}{10}$, if we apply this reduction $d-2$ times we get a circuit with depth 2.
- But this is either a k-CNF or a k-DNF where $k = k_{d-2}$.
- We can make such a formula constant by fixing atmost k variables.
- But the parity function can not be made constant under any restriction of less than n inputs.
- We get a contradiction. Therefore our assumption is wrong. Hence **PARITY** $\notin AC^0$.

References

[FSS81] M. Furst, J. Saxe, and M. Sipser (1981),
Parity, circuits, and the polynomial time hierarchy,
Mathematical Systems Theory 17:1327, 1984. Prelim version FOCS 81.

[Ajt83] M. Ajtai (1983),
$\sum$1 -formulae on finite structures,
Annals of Pure and Applied Logic 24:148.

[ASB] Arora, Sanjeev; Barak, Boaz (2009),
Computational Complexity: A Modern Approach, Cambridge, p. 248-249.

The End