Lower bounds for monotone circuits

B Pratheek
Neha Jawalkar

December 27, 2020

Outline

Monotone circuits and functions

Monotone Circuit

A Boolean circuit is *monotone* if it contains only AND and OR gates, and no NOT gates.

Monotone circuits and functions

Monotone Circuit

A Boolean circuit is *monotone* if it contains only AND and OR gates, and no NOT gates.

Monotone Function

A function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if for every $x \preceq y$, we have $f(x) \leq f(y)$. We denote $x \preceq y$ if every bit that is 1 in x is also 1 in y.

Monotone circuits and functions

Monotone Circuit

A Boolean circuit is *monotone* if it contains only AND and OR gates, and no NOT gates.

Monotone Function

A function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if for every $x \preceq y$, we have $f(x) \leq f(y)$. We denote $x \preceq y$ if every bit that is 1 in x is also 1 in y.

Note that every monotone circuit computes a monotone function and every monotone function can be computed by a monotone circuit.

Cliques

Clique

A *clique* in a graph $G = (V, E)$ is a subset of vertices of G, say C such that every pair of vertices in C are adjacent to each other.

History

- The first monotone circuit lower bound was shown by Razborav in 1985.
- It was improved upon by Andreev and further by Alon and Boppana. The result and proof we are going to see today is due to Alon and Boppana.
- Razborov, in another work in 1985 showed that the monotone circuit complexity is not polynomially related to the non-monotone circuit complexity.
- This was taken further by Eva Tardos who showed the existence of a language whose monotone circuit complexity is exponentially larger than its non-monotone circuit complexity.

Notations

- We have an undirected graph G with n vertices.
- Our goal is to study circuits that detect cliques of size k in G.
- The input to the circuit has $\binom{n}{2}$ bits and is of the form of $x_{\{i,j\}}$, where each bit is an indicator representing if there is an edge between node i and node j.

Theorem

CLIQUE

Denote by $CLIQUE_{k,n} : \{0,1\}^{\binom{n}{2}} \to \{0,1\}$ the function that on input an adjacency matrix of an n-vertex graph G outputs 1 if and only if G contains a k-vertex clique.

Theorem

CLIQUE

Denote by $CLIQUE_{k,n} : \{0,1\}^{\binom{n}{2}} \to \{0,1\}$ the function that on input an adjacency matrix of an n-vertex graph G outputs 1 if and only if G contains a k-vertex clique.

Theorem

There exists a constant ϵ such that for every $k \leq n^{1/4}$, there is no monotone circuit of size less than $2^{\epsilon\sqrt{k}}$ that computes $CLIQUE_{k,n}$.

High level idea

- A monotone circuit can be approximated well using an OR of Clique Indicators (a special kind of circuit for detecting cliques). We use the sunflower lemma to prove this.
- A circuit formed using a small number of Clique Indicators errs on input graphs of two specific varieties.

Distributions

Positive-test graphs

We call as *positive-test* graphs, graphs that are a exact k-clique with no other edges present.

Distributions

Positive-test graphs

We call as *positive-test* graphs, graphs that are a exact k-clique with no other edges present.

Negative-test graphs

We call as *negative-test* graphs, graphs generated by the following procedure: choose a function $c : [n] \to [k-1]$ at random, and place and edge between u and v, if $c(u) \neq c(v)$.

Approximators

Clique Indicators

A clique indicator (CI) for a set $X \subseteq [n]$, is Boolean function, that on input graph G returns 1 if X is a clique in G, and 0 otherwise.
A clique indicator for X is denoted by, $\lceil X \rceil$

Approximators

Clique Indicators

A clique indicator (CI) for a set $X \subseteq [n]$, is Boolean function, that on input graph G returns 1 if X is a clique in G, and 0 otherwise.
A clique indicator for X is denoted by, $\lceil X \rceil$

Approximator

An approximator is a disjunction (OR) of at most m clique indicators.
An approximator A can be expressed as,

$$A = \bigvee_{i=1}^{m} \lceil X_i \rceil$$

Sunflower Lemma

Sunflower

A collection of sets $z_1, \ldots, z_p$, is called a sunflower if the pairwise intersection of any two of the sets is always constant.

Sunflower Lemma

Sunflower

A collection of sets $z_1, \ldots, z_p$, is called a sunflower if the pairwise intersection of any two of the sets is always constant.

If $z_1, \ldots, z_p$ is a sunflower, then there exists a set z_0 such that

$$z_0 = z_i \cap z_j, \quad \forall i \neq j$$

The intersection is called the center or core of the sunflower, and the sets $z_i, i \neq 0$ are called petals of the sunflower.

Sunflower Lemma

Sunflower

A collection of sets $z_1, \ldots, z_p$, is called a sunflower if the pairwise intersection of any two of the sets is always constant.

If $z_1, \ldots, z_p$ is a sunflower, then there exists a set z_0 such that

$$z_0 = z_i \cap z_j, \quad \forall i \neq j$$

The intersection is called the center or core of the sunflower, and the sets $z_i, i \neq 0$ are called petals of the sunflower.

Sunflower Lemma

If Z is a collection of sets, each of size at most l. If $|Z| > (p-1)^l l!$, for some $p > 1$, then Z contains a sunflower with p petals.

Sunflower Lemma

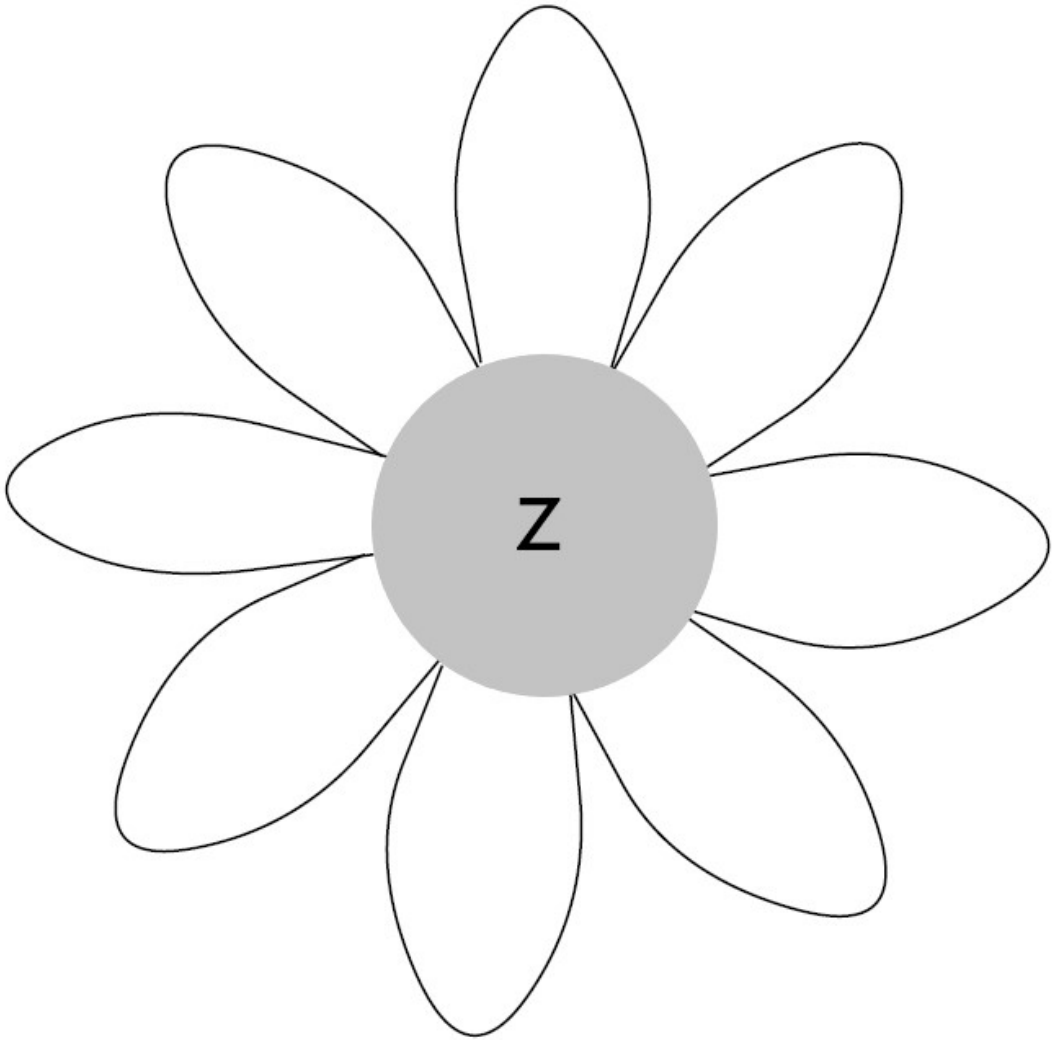


Image credits: Extremal Combinatorics by Stasys Jukna

Construction

Let C be a monotone circuit. Let s be the size of the circuit C.

Construction

Let C be a monotone circuit. Let s be the size of the circuit C.

Approximator for C, call it C' is constructed as follows. Our aim is to get an approximator for C made of at most m clique indicators.

Construction

Let C be a monotone circuit. Let s be the size of the circuit C.

Approximator for C, call it C' is constructed as follows. Our aim is to get an approximator for C made of at most m clique indicators.

Traverse the circuit C such that the inputs to a particular node are considered for approximation before the node itself is approximated.

Construction

Let C be a monotone circuit. Let s be the size of the circuit C.

Approximator for C, call it C' is constructed as follows. Our aim is to get an approximator for C made of at most m clique indicators.

Traverse the circuit C such that the inputs to a particular node are considered for approximation before the node itself is approximated.

This can be achieved by ordering the gates in C. The circuit C can be viewed as a sequence of s monotone function (one per gate), $c_1, \ldots, c_s$. Each c_i is either an input node, or it is either an AND or OR of two other nodes, c_a and c_b such that $a, b < i$.

Construction

Each approximator is made up of at most m clique indicators, each clique indicator of size at most l.

Construction

Each approximator is made up of at most m clique indicators, each clique indicator of size at most l.

For the sunflower lemma to work, we must have l, p and m set appropriately.

Construction

Each approximator is made up of at most m clique indicators, each clique indicator of size at most l.

For the sunflower lemma to work, we must have l, p and m set appropriately.
We set $l = \sqrt{k}$, $p = 10\sqrt{k}logn$, and $m = (p-1)^l l!$.

Construction

Each approximator is made up of at most m clique indicators, each clique indicator of size at most l.

For the sunflower lemma to work, we must have l, p and m set appropriately.
We set $l = \sqrt{k}$, $p = 10\sqrt{k}logn$, and $m = (p-1)^l l!$.

There are three possibilities for any node.

- Leaf node
- AND gate
- OR gate

Construction

Case 1: Leaf node.

Construction

Case 1: Leaf node.

The node being approximated, call it c, is a leaf node.

Construction

Case 1: Leaf node.

The node being approximated, call it c, is a leaf node.

In this case, the approximator for c, is the input at node itself.

Construction

Case 2: OR gate

Construction

Case 2: OR gate

The node being approximated, $c = c_a \vee c_b$.
Let $A = \bigvee_{i=1}^{m} \lceil X_i \rceil$ and $B = \bigvee_{i=1}^{m} \lceil Y_i \rceil$ be the approximation of c_a and c_b respectively.

Construction

Case 2: OR gate

The node being approximated, $c = c_a \vee c_b$.
Let $A = \bigvee_{i=1}^{m} \lceil X_i \rceil$ and $B = \bigvee_{i=1}^{m} \lceil Y_i \rceil$ be the approximation of c_a and c_b respectively.

The approximator for c be $A \sqcup B$

$$\bigvee_{i=1}^{m} \lceil X_i \rceil \vee \bigvee_{i=1}^{m} \lceil Y_i \rceil$$

Construction

Case 2: OR gate

The node being approximated, $c = c_a \vee c_b$.
Let $A = \bigvee_{i=1}^{m} \lceil X_i \rceil$ and $B = \bigvee_{i=1}^{m} \lceil Y_i \rceil$ be the approximation of c_a and c_b respectively.

The approximator for c be $A \sqcup B$

$$\bigvee_{i=1}^{m} \lceil X_i \rceil \vee \bigvee_{i=1}^{m} \lceil Y_i \rceil$$

If the number of distinct clique indicators in the approximation for c is less than m, we are done. Else, find a sunflower in the set of clique indicators and replace the p sunflower petals with the core. Repeat till the number of clique indicators is less or equal to m.

Construction

Case 3: AND gate

Construction

Case 3: AND gate

The node being approximated, $c = c_a \wedge c_b$.
Let $A = \bigvee_{i=1}^{m} \lceil X_i \rceil$ and $B = \bigvee_{i=1}^{m} \lceil Y_i \rceil$ be the approximation of c_a and c_b respectively.

Construction

Case 3: AND gate

The node being approximated, $c = c_a \wedge c_b$.

Let $A = \bigvee_{i=1}^{m} \lceil X_i \rceil$ and $B = \bigvee_{i=1}^{m} \lceil Y_i \rceil$ be the approximation of c_a and c_b respectively.

The approximator for c be $A \sqcap B$

$$\bigvee_{i=1}^{m} \bigvee_{j=1}^{m} \lceil X_i \cup Y_j \rceil$$

Construction

Case 3: AND gate

The node being approximated, $c = c_a \wedge c_b$.
Let $A = \bigvee_{i=1}^{m} \lceil X_i \rceil$ and $B = \bigvee_{i=1}^{m} \lceil Y_i \rceil$ be the approximation of c_a and c_b respectively.
The approximator for c be $A \sqcap B$

$$\bigvee_{i=1}^{m} \bigvee_{j=1}^{m} \lceil X_i \cup Y_j \rceil$$

Discard any clique indicator, $\lceil X_i \cup Y_j \rceil$ with $|X_i \cup Y_j| > l$. If there are more that m distinct clique indicators then find a sunflower in the set of clique indicators and replace the sunflower petals with the core. Repeat till the number of clique indicators is less or equal to m.

Next steps

- Subgoal-1: Show that any approximator C' either makes a lot of errors (outputs 0) on positive test graphs, or makes a lot of errors (outputs 1) on negative test graphs.
- Subgoal-2: Bound the number of mistakes C' makes on positive and negative test graphs. Since C is always correct on both positive and negative test graphs, the mistakes C' makes must be explained away by the errors of approximation. We show that that this implies that C must be large.

Subgoal-1

Lemma 1

Let $A = \bigvee_{i=1}^{m'} \lceil X_i \rceil$ be an approximator. Then either

(i) A outputs 0 on all graphs or

(ii) A outputs 1 (accepts) more than $\left(1 - \frac{\binom{l}{2}}{k-1}\right)(k-1)^n$ negative test graphs.

Subgoal-1

Lemma 1

Let $A = \bigvee_{i=1}^{m'} \lceil X_i \rceil$ be an approximator. Then either

(i) A outputs 0 on all graphs or

(ii) A outputs 1 (accepts) more than $\left(1 - \frac{\binom{l}{2}}{k-1}\right)(k-1)^n$ negative test graphs.

Proof. If $m' = 0$, then A is the empty disjunction (in other words, 0). If $m' \geq 1$, consider just some $\lceil X_j \rceil$, $|X_j| \leq l$.
$\lceil X_j \rceil$ rejects a negative test graph iff it has no clique on X_j. The number of negative test graphs that have no clique on X_j is at most $\binom{|X_j|}{2}(k-1)^{n-1}$ (counting). The bound follows.

Bounding error: strategy

Lemma

The number of false negatives (positives) generated by C' (relative to C) is at most $\sum_{i=1}^{size(C)} e_i$, where e_i is number of false negatives (positives) generated by the approximator of gate i relative to gate i.

Proof. By induction.

- Base case. Replacing input gates introduces no error.
- Assume that everything until gate $(i-1)$ has been replaced by approximators. Also assume the inductive hypothesis holds, i.e. the number of false negatives produced by the intermediate circuit C_{i-1} is $\leq \sum_{j=1}^{i-1} e_j$.

  Now replace gate i by the corresponding approximator.

Bounding error: strategy

- The resulting circuit C_i will produce a 0 when C_{i-1} produces a 1 only when the approximator of gate i produces a 0 when gate i produces a 1.
- This is because the outputs of of gate i and its approximator are being pushed into a monotone circuit (gates are replaced bottom up, so everything above gate i is yet to be approximated).
- Thus the number of false negatives being generated by C_i (relative to C) is at most the number of false negatives generated by C_{i-1} relative to C plus the number of false negatives generated by the approximator of gate i relative to i. This completes the proof.

Subgoal-2, part i

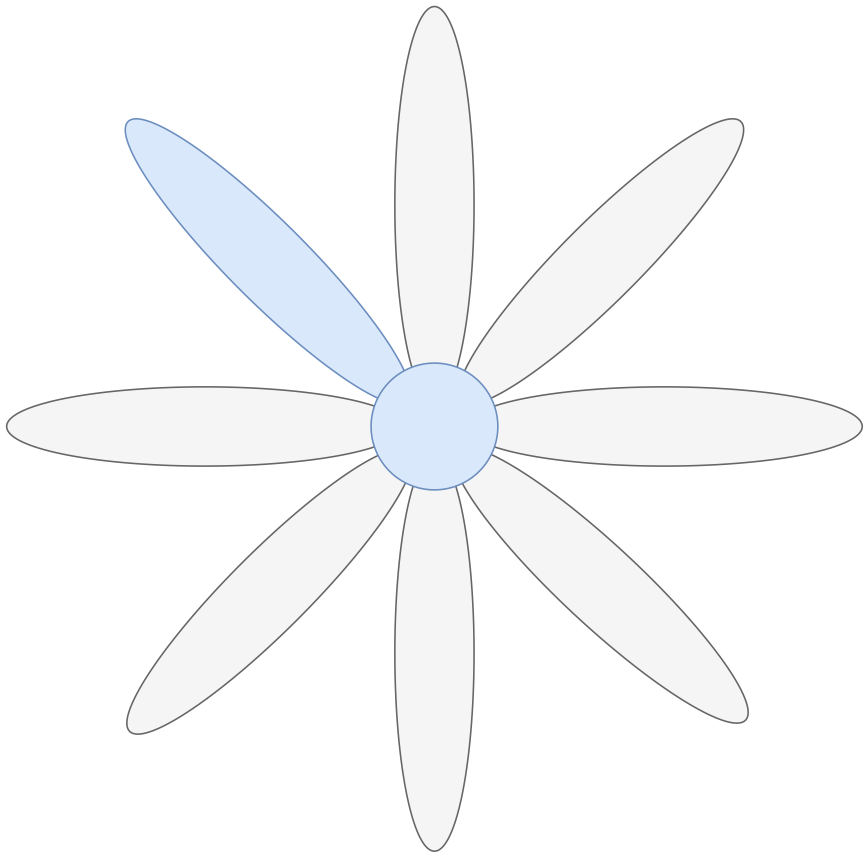
Lemma 2

The number of false negatives generated by C' relative to C on positive test graphs is at most $size(C)m^2\binom{n-l-1}{k-l-1}$.

Proof. We'll show that each gate of C contributes no more than $m^2\binom{n-l-1}{k-l-1}$ false negatives on positive test graphs.

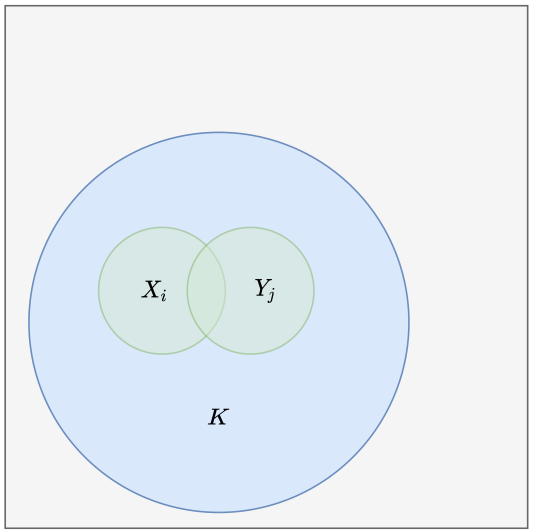
Subgoal-2, part i

- Input gates. Here the approximation introduces no error.
- $\vee$ gates.
  1. The disjunction of a sunflower has the same truth value as the center (empty center gets value 1).



Subgoal-2, part i

- $\wedge$ gates.
  1. Going from $\lceil X_i \rceil \wedge \lceil Y_j \rceil$ to $\lceil X_i \cup Y_j \rceil$ does not introduce false negatives on positive test graphs.
     - $\lceil X_i \rceil = 1 \implies X_i \subseteq K$
     - $\lceil Y_j \rceil = 1 \implies Y_j \subseteq K$
     - Together, these imply that $X_i \cup Y_i \subseteq K$.
     - $X_i \cup Y_i \subseteq K \implies \lceil X_i \cup Y_j \rceil = 1$
  2. However, discarding sets of large cardinality might introduce false negatives. We bound this number as follows: let $|X_i \cup Y_j| = l' > l$. Then discarding $\lceil X_i \cup Y_j \rceil$ introduces at most $\binom{n-l'}{k-l'} \leq \binom{n-(l+1)}{k-(l+1)}$ false negatives on positive test graphs. The bound follows from the fact that there are at most m^2 $\lceil X_i \cup Y_j \rceil$.



Subgoal-2, part ii

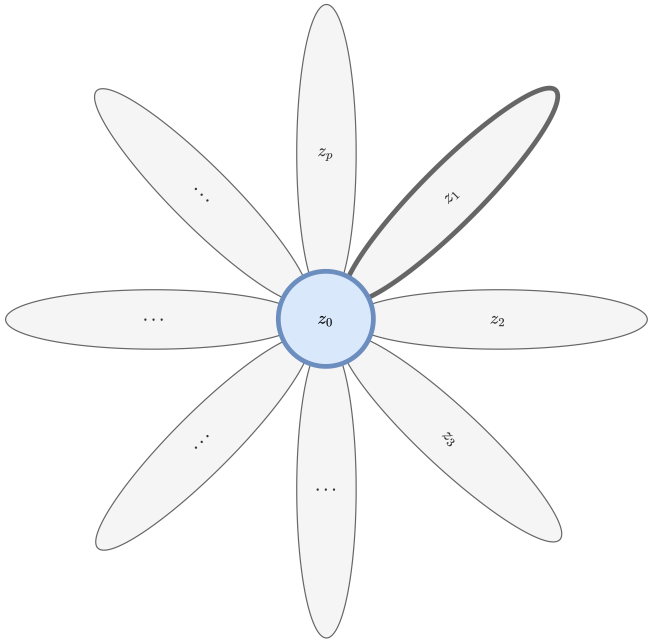
Lemma 3

The number of false positives generated by C' relative to C on negative test graphs is at most $size(C)m^2\left(\frac{\binom{l}{2}}{k-1}\right)^p(k-1)^n$.

Proof. The argument is similar to the last lemma.

Subgoal-2, part ii

- Input gates. No error.
- At $\vee$ gates, replacing sunflowers by their center might introduce false positives. We bound this number as follows (on negative test graphs):
- Let $z_1, \ldots, z_p$ be a sunflower.



Subgoal-2, part ii

- We want to bound the number of negative test graphs rejected by $\vee_{i=1}^{p} \lceil z_i \rceil$, but accepted by $\lceil z_0 \rceil$, where z_0 is the center of the sunflower.
- Argue using counting. For every petal, pick a pair of vertices that collide (are assigned the same value under c). This yields at most $\binom{l}{2}$ choices per petal, and $\binom{l}{2}^p$ choices for p petals.
- Assign values to all the vertices. This can be done in $(k-1)^{n-p}$ ways (since there are p pairs of colliding vertices and we pick only one value for each colliding pair).
- Thus a single sunflower contributes at most $\binom{l}{2}^p (k-1)^{n-p}$ false positives.
- There can be at most $2m$ sunflowers per $\vee$ gate (since there are at most $2m$ clique indicators), so an $\vee$ gate contributes at most $2m\binom{l}{2}^p (k-1)^{n-p}$ false positives.

Subgoal-2, part ii

- $\wedge$ gates.
  1. $\lceil X_i \rceil \wedge \lceil Y_j \rceil$ going to $\lceil X_i \cup Y_j \rceil$ will not introduce false positives.
  2. Discarding $\lceil X_i \cup Y_j \rceil$, $|X_i \cup Y_j| > l$ will not introduce false positives since we are discarding clauses from a disjunction.
  3. Replacing sunflowers by their center might introduce false positives, but this number has been bounded. It is $\leq m^2 \left(\frac{\binom{l}{2}}{k-1} \right)^p (k-1)^n$.

Putting it all together

Theorem

There exists a constant ϵ such that for every $k \leq n^{1/4}$, there is no monotone circuit of size less than $2^{\epsilon\sqrt{k}}$ that computes $CLIQUE_{k,n}$.

Let C be a circuit for $CLIQUE_{n,k}$.
Let $l = \sqrt{k}, p = \lceil 10klogn \rceil, m = (p-1)^l l!$.
By Lemma 1, we have two possible situations:

1. C' rejects all positive test graphs. In this case
$$\binom{n}{k} \leq size(C)m^2\binom{n-l-1}{k-l-1}$$
2. C' accepts at least $\left(1 - \frac{\binom{l}{2}}{k-1}\right)(k-1)^n$ negative test graphs. Then
$$\left(1 - \frac{\binom{l}{2}}{k-1}\right)(k-1)^n \leq size(C)m^2\binom{n-l-1}{k-l-1}$$

Algebra for situation 1

$$m = (p-1)^l l! = (p-1)^{\sqrt{k}}(\sqrt{k})!$$
$$\leq (10\sqrt{k}\log n)^{\sqrt{k}}\ \sqrt{k}^{\sqrt{k}}$$
$$= k^{\sqrt{k}}(10\log n)^{\sqrt{k}}$$

$$\binom{n}{k} \leq size(C)m^2\binom{n-l-1}{k-l-1}$$

$$\frac{\binom{n}{k}}{\binom{n-l-1}{k-l-1}} = \frac{n}{k}\frac{n-1}{k-1}\cdots\frac{n-l}{k-l} \geq \left(\frac{n-\sqrt{k}}{k}\right)^{\sqrt{k}}$$

Algebra for situation 1

$$m^2 \leq k^{2\sqrt{k}}(10 \log n)^{2\sqrt{k}}$$

Therefore,

$$size(C) \geq \left(\frac{n-\sqrt{k}}{k}\right)^{\sqrt{k}} \frac{1}{k^{2\sqrt{k}}(10 \log n)^{2\sqrt{k}}}$$

$$size(C) \geq \frac{(n-\sqrt{k})^{\sqrt{k}}}{k^{3\sqrt{k}}(10 \log n)^{2\sqrt{k}}}$$

Algebra for situation 1

Since $k \leq n^{1/4}$, we have

$$size(C) \geq \frac{(n-\sqrt{k})^{\sqrt{k}}}{n^{3/4\sqrt{k}}\left(10logn\right)^{2\sqrt{k}}}$$

Algebra for situation 2

In the other situation, we have

$$\left(1-\frac{\binom{l}{2}}{k-1}\right)(k-1)^n \leq size(C)m^2\left(\frac{\binom{l}{2}}{k-1}\right)^p (k-1)^n$$

Since $l=\sqrt{k}$,

$$\frac{\binom{l}{2}}{k-1} < \frac{1}{2}$$

$$size(C)m^2\frac{1}{2^p} \geq \frac{1}{2}$$

Hence,

$$size(C) \geq \frac{2^{p-1}}{m^2} \geq \frac{2^{10\sqrt{k}logn}}{k^{2\sqrt{k}}\left(20\log n\right)^{2\sqrt{k}}} = \frac{n^{10\sqrt{k}}}{n^{\sqrt{k}/2}\left(20\log n\right)^{2\sqrt{k}}}$$