| Recent advances in Complexity CIS 6930/CIS 4930 | October 8, 2002 |
|---|---|

# Lecture 13

Lecturer: Dr. Meera Sitharam — Scribe: Erwin Jansen

## 1 Introduction

In today's lecture we continued with the proof that the $clique_{k,n}$ function requires a *monotone* circuit of size at least $n^{\Omega(\sqrt{k})}$. The $clique_{k,n}$ function outputs a one only when there exists a clique of size $k$.

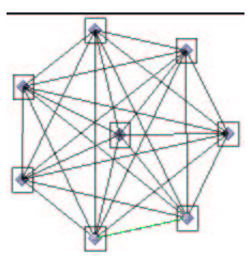


**Figure 1**: Positive graph

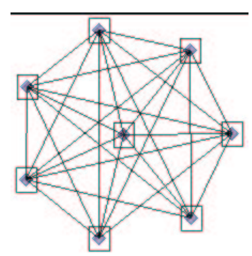


**Figure 2**: Negative graph

The idea behind the proof is that we want to show that the class of small monotone functions does not contain the function clique. See figure 3. The class of monotone functions is however very diffuse and it is hard to get a grasp on this set. In order to get a bit of a grip on this class we are going to look at a *dense* subset of this class. In our case this subset will consist of *nice* approximator circuits. If there were a monotone circuit that computes $clique_{k,n}$, then you can always find a small circuit, an approximator, in the neighborhood that does at least as well as $clique_{k,n}$. Now we are going to show that these approximators can't do $clique_{k,n}$ very well on a particular domain. For the domain we will choose those graphs for which it is very hard to distinguish between a $clique_{k,n}$ or not. These graphs will be positive and negative test graphs:

Positive test

**Definition 1 (Positive test graph).** *A positive test graph is a graph that barely has clique's, and will have only one $k$ clique.*

Negative test

**Definition 2 (Negative test graph).** *A negative test graph is a graph that has many $(k-1)$ clique's but no $k$ clique. This graph is constructed by coloring $n$ vertices with $k-1$ colors. Every vertex is assigned a color with equal probability.Now we have $\frac{n}{k-1}$ vertices with the same color. After coloring we connect all pairs of vertices with distinct colors.*

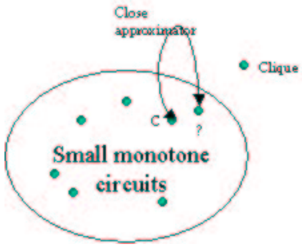



**Figure 3**: Approximators and *Clique*

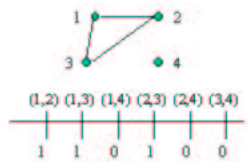



**Figure 4**: Clique indicator

We can consider a positive test graph to be the *minimal* graph for which the $clique_{k,n}$ function will return a one. Negative test graphs on the other hand are the *maximal* graphs for which the $clique_{k,n}$ function will return a zero.

## 2 Constructing approximator circuit

The first thing we will do is get an idea of how we are going to construct these approximator circuits. Approximator circuits are going to be constructed using so called clique indicators. A clique indicator is a function that returns a 1 if it has a clique on its inputs:

Clique Indicator

**Definition 3 (Clique Indicator).** *A clique indicator $\lceil X \rceil$ over the vertices in $X$, is a function of $\binom{n}{2}$ variables that is 1 if the associated graph contains a clique on the vertices $X$ and 0 otherwise.*

Notice that we can write $clique_{k,n}$ as an or of clique indicators. Every clique indicator is of size $k$ and we have $\binom{n}{k}$ of these clique indicator. In other words, our clique function is rather large.

**Example 1 (Clique Indicator).** *If we look at figure 4 we see a negative test graph with the proper encoding. Now $\lceil\{2,3,4\}\rceil = 0$, $\lceil\{1,2\}\rceil = 1$ and $\lceil\{1,2,3\}\rceil = 1$.*

Based upon the clique indicator we can define what an approximator is. An approximator is an OR of clique indicators that do not exceed a fixed size.

Approximator

**Definition 4 (Approximator).** *An $(l,m,p)$-approximator, or simply approximator is an or of clique indicators such that:*

$$\bigvee_{1\leq i\leq m} \lceil X_i \rceil \qquad\qquad \forall_{1\leq i\leq m}\langle |X_i| \leq l\rangle$$

We have not yet defined where the values $m, p$ stand for, but we will see later on that we can use these parameters to establish our proof of the lowerbound. Now we are going to construct the approximator circuits by induction over the monotone circuit $C$ that is supposed to compute $clique_{n,k}$ as follows:

**Base case:** An input variable is of the form $x_{i,j}$ where $i, j$ are different vertices (see also figure 4). But this is equivalent to $\lceil\{i,j\}\rceil$, hence every input variable is a clique indicator.

**Induction:** Take an $\vee, \wedge$ gate in $C$ and convert this into an approximator.

Since the induction step is not trivial we will look at in more detail. We will first look at the $\vee$ gate and after that we will examine the construction of an $\wedge$ gate

## 2.1 Approximating at an $\vee$ gate

Suppose we are converting an $\vee$ gate in our circuit $C$. Let $A$ and $B$ the functions that feed into this $\vee$ gate. By the induction hypothesis these are both approximators. Hence $A = \vee_{i=1}^{r}\lceil X_i \rceil$ and $B = \vee_{i=1}^{s}\lceil Y_i \rceil$, where both $s, r \leq m$. We could simply construct a big approximator of these two by taking an or. We would obtain something like $\vee_{i=0}^{i \leq r+s}\lceil Z_i \rceil$, where $Z = \{X_1, ..., X_r, Y_1, ..., Y_s$. However this approximator could have the a size of at most $r + s = 2m$. Which exceeds our restriction that we can have at most $m$ distinct sets.

Clearly we need to cut down in the size of our approximator. In order to do that we are going to replace several clique indicators from $A$ and $B$ with their *common* part. To do this we will introduce the concept of a *sunflower*:

Sunflower

**Definition 5 (Sunflower).** *A* sunflower *is a collection of distinct sets $Z_1, ..., Z_p$, called* petals *such that the intersection $Z_i \cup Z_j$ is the same for every pair of distinct indices $i$ and $j$, i.e.:*

$$\forall_{1 \leq i \leq p} \forall_{1 \leq j \leq p} \forall_{1 \leq k \leq p} \langle Z_i \cup Z_j = Z_i \cup Z_k \rangle$$

*We call $Z_i \cup Z_j$ the center of the sunflower.*

The reason for the name sunflower comes from the visualization of the intersection. If we look at the figure on the right we see a sunflower of four sets. It is not hard to imagine that the more sets we have, the more the figure will look like a sunflower.

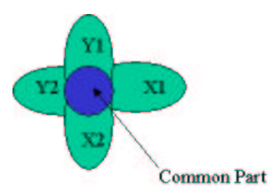



**Figure 5**: A sunflower

If we have such a sunflower we can perform an operation called *plucking*. Plucking a collection of sets $Z_0, ..., Z_n$ is nothing more than replacing the sets $Z_0, ..., Z_n$ with their common center.

We are going to apply this idea to our approximators. We are going to pluck the the vertex sets $\{X_1, ..., X_r, Y_1, ...Y_s\}$ until we cannot pluck it anymore. It is easy to see that we can have at most $2m$ plucking operations. The following lemma will be of use:

**Lemma 6.** *Let $\mathcal{L}$ be a collection of sets each of cardinality at most $l$. If $|\mathcal{L}| > (p-1)^l \cdot l!$, then the collection contains a sunflower with $p$ petals*

Excercise

*Proof.* The proof can be found in [1] and is left as a reading exercise. □

Using this lemma we can finally get an idea of the value's $m$ and $p$. Setting the value's $m = (p-1)^l \cdot l!$, we can assure that after the plucking procedure we have at most $m$ vertex sets. Later on we will relate the value's $m, p$ to $n, k$ of the clique function. After plucking we get the approximator we are looking for:$\bigvee_{i \leq i \leq m}\lceil Z_i \rceil$.

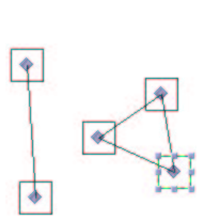


**Figure 6**: Two approximators

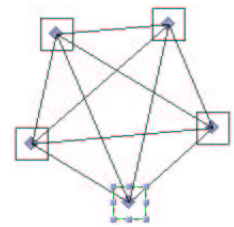


**Figure 7**: Combined approximator

## 2.2 Approximating at an $\wedge$ gate

Suppose we are converting an $\wedge$ gate in our circuit $C$. Let $A$ and $B$ the functions that feed into this $\wedge$ gate. By the induction hypothesis these are both approximators. Hence $A = \vee_{i=1}^{r} \lceil X_i \rceil$ and $B = \vee_{i=1}^{s} \lceil Y_i \rceil$, where both $s, r \leq m$. Converting an and works by applying the following distributive law: $\vee_{i=1}^{r} \vee_{j=1}^{s} (\lceil X_i \rceil \wedge \lceil Y_j \rceil)$. There is however a problem by applying this law. First of all $\lceil X_i \rceil \wedge \lceil Y_j \rceil$ is not a clique indicator. If we look at figure 6 for example we could get a combined clique indicator of figure 7, which adds a lot of edges!. Notice that this is a very coarse approximator.

Another problem is that we can have as many as $m^2$ terms!. To overcome these problems we are going to apply the following steps.

**Replacement:** We replace all the clique approximators $\lceil X_i \rceil \wedge \lceil Y_j \rceil$ by $\lceil X_i \cup Y_j \rceil$. This can introduce sets that are bigger than $l$.

**Throwing away:** We throw away all the indicators who have a size of the set $|X_i \cup Y_j| > l$. This is good in the sense that this operation is likely to throw away those indicators that were bad approximators to begin with. If we look at figure 6 once more and assume that these sets are disjoint then the resulting union is very large. The approximator in figure 7 is a poor approximator anyway.

**Plucking:** If we put two or's together we could obtain a set as large as $m^2$. To deal with this double or, we are going to perform a plucking operation on the remaining indicators.

Applying these steps assures that we end up with an approximator of the proper size.

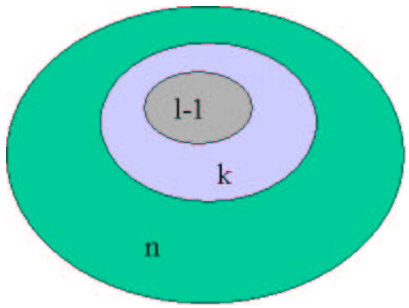
# 3 Proving the first lemma

**Lemma 7.** *Every approximator circuit either is identically 0 or outputs 1 on at least:*

$$\left(1 - \frac{\binom{l}{2}}{k-1}\right) \cdot (k-1)^n$$

*of the negative test graphs.*

*Proof.* Let $\hat{C}$ be the approximator circuit. If $\hat{C}$ is identically 0 then the first part of the lemma holds. Now the case where one of the indicators turns on. In other words $\hat{C} \geq \lceil X_1 \rceil(x))$. A negative test graph is rejected by the clique indicator $\lceil X_1 \rceil$ if and only if $\lceil X_1 \rceil = 0$. An output of 0 on $\lceil X_1 \rceil$ means that at least two




Here $n$ denote all the vertices in the graph and $k$ is the clique on this graph. All the $l+1$ vertices are within the clique $k$

**Figure 8**: $l+1$ in $k$ in $n$

vertices on $X_1$ don't have an edge. That will happen only if they are colored the same. The probability that a fixed pair has the same color is $\frac{1}{k-1}$ and there are $\binom{l}{2}$ of these pairs. Now the probability that $\lceil X_1 \rceil = 1$ is $\geq \frac{(1-\binom{l}{2})}{k-1}$. We have $(k-1)^n$ possible colorings of the graph hence there are $(k-1)^n$ negative test graphs. Therefor we conclude that the number of negative test graphs for which $\lceil X_1 \rceil = 1$ is at least $\left(1 - \frac{\binom{l}{2}}{k-1}\right) \cdot (k-1)^n$ $\square$

**Lemma 8.** *For every monotone circuit $C$, the number of positive test graphs for which the inequality $C \leq \hat{C}$ does not hold is at most size$(C) \cdot m^2 \cdot \binom{n-l-1}{k-l-1}$.*

*Proof.* Let $A = \vee_{i=1}^{r} \lceil X_i \rceil$ and $B = \vee_{i=1}^{r} \lceil Y_i \rceil$ be two approximators. Now when we find an approximator for an $\vee$ we only do plucking. When we are plucking we are replacing a large clique indicator $\lceil X_i \rceil$ by a small clique indicator $\lceil Z_i \rceil$. This procedure can only increase the number of accepted graphs. So it can only go wrong with an $\wedge$ gate. The *replacement* procedure will keep the output $\hat{\wedge} \geq \wedge$ as we have seen before in figure 6 and 7. We have seen that *plucking* does not create any problems either. The only step where it can go wrong is the *throwing away* stage, where we throw away those clique indicators $\lceil X_i \cup Y_j \rceil$ for which $|X_i \cup Y_i| \geq l+1$. This is only a problem if the clique indicator we are throwing away output a value of 1. So the $l+1$ (or more) vertices must all be in the clique, as depicted in figure 8. Now the number of graphs such that we are throwing away these $l-1$ vertices is $\binom{n-l-1}{k-l-1}$. This can happen at most *size*$(C)$ times if all the gates in $C$ are $\wedge$ gates. It is left as an exercise to prove where the value $m^2$ stems from. $\square$

exercise

**Exercise 1.** *Extends the proof to include the value $m^2$.*

# References

[1] P. Erdos and R. Rado. Intersection theorems for systems of sets. *Journal London Math Society*, 35:85–90, 1960.