CSC 2401, Introduction to computational complexity
Lecture #17, Thursday, November 9, 2001
Lecturer: Allan Borodin
Scribe: Sotos Liaskos

1 Review

In the last lecture, we made an introduction to Circuit Complexity. In general, we consider circuits of AND and OR gates and negations. Without any significant loss of generality we can assume that negations apply only to the inputs. We have the following:

Theorem 1 *Every function $f : \{0,1\}^n \rightarrow \{0,1\}$ can be computed by a circuit of size:*

$$\frac{2^n}{n}(1 + \frac{1}{O(\sqrt{n})}) = \frac{2^n}{n}(1 + \epsilon)$$

Furthermore, $\exists f : \{0,1\}^n \rightarrow \{0,1\}$: every circuit computing f requires more than $\frac{2^n}{n}$ gates (”counting argument”). In fact, “most” $f : \{0,1\}^n \rightarrow \{0,1\}$ require $(1-\epsilon)\frac{2^n}{n}$ gates.

But: can we find an ”explicit” (or ”natural”) f_n requiring this many gates?

2 Uniform TM Time vs Non-Uniform Circuit Size

Definition 1 *For $L \subseteq \{0,1\}^*$ define $L|_n) = L_n = L \cap \{0,1\}^n$*

(We will prefer the notation L_n). It follows that:

If $L \in$ P, then $\exists$ a polynomial $p(n) : \forall n L_n$ is computed by a circuit of size $\leq p(n)$

In general:

if $L \in DTIME(t(n))$, then $\exists c : \forall n L_n$ is computed by a circuit of size $\leq c \cdot t(n) log t(n)$

... that is:

$L \in DTIME(t(n)) \Rightarrow \forall n L_n \in \cup_{c \geq 0} SIZE(c \cdot t(n) log(t(n)))$

The above represents the relationship between uniform TM time and non-uniform circuit size. We must emphasize the fact that the converse is not true. Below, we will use $L \in SIZE(t)$ to mean $\forall n L_n \in SIZE(t(n))$.

3 Applications

In the beginning days of P vs NP, many people thought that the best way to show that $P \neq NP$ was to find some $L \in NP$ such that $L \notin POLYSIZE = \cup_{c,k} SIZE(cn^k))$, and thus $L \notin P$. But there is no known example of an $L \in NP : L \notin SIZE(5n)$. Worse yet, we don't even know if, say, $NP \subseteq SIZE(O(n))$!!

Theorem 2 *If $P = NP$ then $\forall k, P \not\subseteq SIZE(O(n^k))$.*

(**Contrapositive:** If $NP \subseteq SIZE(n^k)$ for some k then $P \neq NP$.)

Proof:

$P = NP \Rightarrow PH = P$. I claim that $\forall k, \exists L \in PH : L_n \notin SIZE(O(n^k))$ for sufficiently large n.

Fact: By counting argument (or just by "padding" argument using the fact that $\frac{2^n}{n}(1+\epsilon)$ gates is better than $\frac{2^n}{n}$ gates): $L \in SIZE(n^{k+1})$ - $SIZE(O(n^k))$.

$L = \{x \mid \exists C\ \forall C' :$
1) $size(C) \leq n^{k+1}$ where $n = |x|$
2) $size(C') > size(C)$ *and* $size(C') \leq n^{k+1}$ *implies* $\exists C'' : size(C'') < size(C')$ *and* C'' *equivalent to* C'.
3) $\nexists C''\ size(C'') < size(C)$ *and* C'' *eqivalent to* C.
4) x is accepted by C.
$\}$

where "C' equivalent to C" means: $\forall y[C'(y) = C(y)]$

The above description shows that L is in the Polynomial Time Hierarchy.

Definition (review): $MaxC(n) = max\{size(C) : C$ computes an $f : \{0,1\}^n \rightarrow \{0,1\}$ and C is the smallest (in # of gates) such circuit computing $f\}$.

Furthermore, we know that:

$$\frac{2^n}{n} \leq MaxC(n) \leq \frac{2^n}{n}(1+\epsilon)$$

Theorem 3 *(Valentine Kabanets) $P = NP \Rightarrow \exists L \in E = \cup_c DTIME(C^n)$: $\forall n L_n$ requires size $Max(c^n)$.*

(**Contrapositive:** if $\forall L \in E,\ \exists n : L_n$ computable in size $\leq$ MaxC(n) - 1 $then P \neq NP$).

Fact: $\exists L \in ESPACE = \cup_{c>0} DSPACE(C^n) : \forall n : L^n$ requires size MaxC(n).

Proof:
$P = NP \Rightarrow PH = P \Rightarrow E^{PH} = E^P = E$
Claim: E^{PH} contains a language L: $\forall n L_n$ requires circuit size MaxC(n)

(Trick: use a long enough (exponential) query for the oracle)