THEOREM 0.1. *$\forall k$, $\exists A \in \sum_2^p$ such that A does not have circuits of size n^k.*

Before we prove this theorem, some lemmas are needed:

LEMMA 0.1. *There is a set B such that $\forall n$, $B \subseteq \{w$: w is among the first n^k strings of length n $\}$ and B requires circuits of size $> n^{k/2}$.*

Proof. The number of subsets of the first n^k strings of length n is 2^{n^k}. However, the number of circuits of size $n^{k/2}$ is less than 2^{n^k}. So the lemma follows.

■

LEMMA 0.2. *$\exists A$ of this form in $\sum_4^p$.*

Proof. On input x of length n, guess a subset $B \subseteq \{$ the first n^{2k} strings of length n $\}$. $\forall$ check (universally check) that for all circuits C of size n^k, there exists an input y of length n such that $C(y) = 1$ if and ony if $y \notin B$ and $\forall B'$ that lexicographically precede B there is a circuit C' of size n^k such that for all x of length n $C'(x) = 1$ iff $y \in B'$

■

The proof of the theorem:

Proof. There are two cases:

Case 1. $SAT \notin P/poly$; then $SAT \in \sum_2^p$ and does not have circuit of size n^k.

Case 2. $SAT \in P/poly$; then $PH = \sum_2^p$, and the lemmas give the result.

■

COROLLARY 0.1. *There are sets in $EXSPACE$ that require circuits of size $2^{n/2}$*

Open Question: (1) $EXP \subseteq P/Poly$?; (2) $NEXP \subseteq P/Poly$?.

Now we

are starting our next topic: *Probabilistic Turing Machines.*

A *probabilistic Turing machine* is a Turing machine augmented with the ability to generate an unbiased coin flip in one step. It corresponds to a randomized algorithm. On any input x, the output is a random variable (or map).

The motivation for stuying this special machine is that, in practice, there are many problems for which we know efficient randomized algorithms, but for which no polynomial-time deterministic algorithms are known.

Related to the probabilistic Turing machine, we can define several complexity classes. We are going to discuss some relationships between these classes.

(1) **PP** = { A : $\exists$ a polynomial time probabilistic Turing machine M such that $x \in A$ if and only if $Prob(M(x) = 1)) > 1/2$ }

(2) **BPP** = {A:$\exists$ a polynomial time probabilistic Turing machine M such that $x \in A$ if and only if $Prob(M(x) = 1)) > 3/4$ and $x \notin A$ if and only if $Prob(M(x) = 1)) < 1/4$ }

(3) **RP** = {A : A is accepted by an NP-machine M and $x \in A$ if and only if $Prob(M(x) = 1)) > 3/4$ }

Based on our observations, we can easily show: (1) $NP \subseteq PP$; (2) $PP = Co - PP$. Also we can have a fact about BPP: $A \in BPP$, then $\forall k$, $\exists$ a probabilistic polynomial-time Turing machine accpeting A with error probability less than $1/2^{n^k}$. In fact, on input x, we make m (for our convenience, we just assume m is an even integer) independent trials and take a majority vote, the probability of error is

$$\sum_{j=0}^{m/2} \binom{m}{j} (3/4)^j (1/4)^{m-j}$$

which is less than

$$(3/8)^{m/2}$$

In fact, $(3/4)^j(1/4)^{m-j} \leq (3/16)^{m/2}$ for any $j = 0, ..., m/2$.