1. (a) Consider Boolean functions on \( n \) variables, but which only depend on the first \( m \) variables (for some fixed \( m \leq n \)). That is, functions of the form \( f(\bar{x}) \) such that \( f(x_1, \ldots, x_m, x_{m+1}, \ldots, x_n) = f(x_1, \ldots, x_m, x'_m+1, \ldots, x'_n) \), regardless of the values of \( x'_m+1, \ldots, x'_n \). Give an upper bound on the size of circuit needed to compute such functions. \textit{Hint}: Use DNF. Your upper bound should depend only on \( m \), not on \( n \).

(b) How many such functions are there? Your answer should depend only on \( m \), not on \( n \).

(c) Find a value of \( m \), as a function of \( n \) and \( k \) (that is, \( m = m(n, k) \)), such that all Boolean functions that depend only on their first \( m \) variables can be computed by circuits of size at most \( n^{k+1} \). Try to make \( m \) as large as possible subject to this condition.

(d) Fix \( k \geq 1 \). How many Boolean circuits are there using AND, OR, NOT gates, which take \( n \) inputs, and have size at most \( n^k \)?

(e) Fix \( k \geq 1 \). Using the value of \( m \) from part (c), compare the count from part (b) with the count from part (d) to conclude that there exist Boolean functions computable by circuits of size \( n^{k+1} \) but not of size \( n^k \). (If you can’t get \( n^{k+1} \) vs \( n^k \), see if you can get your counting arguments to work to show the existence of a function computable by circuits of size \( n^{3k} \) but not of size \( n^k \).)

2. (Kannan’s Theorem)

(a) Fix \( k \geq 1 \). Try to write down the statement “There is a polynomial-size circuit \( C \) that computes a function that isn’t computable by
any circuit of size at most \( n^k \), using as few quantifier alternations as possible. (It is possible to do with at most 4 quantifier alternations, but even if you do more that is fine, as long as it’s a fixed number.)

(b) Use your answer from the previous part to build a language in \( \text{PH} \) that is not computable by circuits of size \( n^k \). \textit{Hint:} You need to make sure that on all inputs of a given length \( n \), the \textit{same} circuit \( C \) is chosen by the existential quantifier. One way to do this is to enforce that \( C \) is the circuit whose description is lexicographically first, among circuits satisfying the property from part (a).

(c) Use the preceding part to show that in fact there is a language \( L_k \in \Sigma_2^P \) such that \( L_k \) is not computable by circuits of size \( n^k \), as follows. If \( \text{NP} \not\subseteq \text{P/poly} \), then we are done (why?). If \( \text{NP} \subseteq \text{P/poly} \), then what can we say about \( \text{PH} \)? \textit{(Hint: Combine part (b) with the Karp--Lipton Theorem.)}

3. \textit{(Shannon’s Theorem)} Using similar counting as in Question 1 (but with \( m = n \)), show that most \( n \)-variable Boolean functions cannot be computed by circuits smaller than size \( 2^n/(10n) \) (the value of 10 is not crucial—if you can do it with 1000 instead of 10 that’s fine—but you will need some constant \( > 1 \) in the denominator to get the counting to work out).

\textbf{Resources}

- \textbf{Arora & Barak} Section 6.3 for Shannon’s Theorem, Section 6.4 for the circuit size hierarchy theorem (a tighter version of what is asked in Question 1 above)
- \textbf{Homer & Selman} Proposition 8.1 for counting Boolean circuits of a given size
- Du & Ko Theorem 6.1 gives Shannon’s Theorem.
- \textbf{Lecture notes by Paul Beame} on Karp–Lipton and Kannan’s Theorems are pretty good, and in line with how we’ve been covering them in class.