A circuit family is a sequence \( C = (C_1, C_2, C_3, \ldots) \) of Boolean circuits \( C_i \) where \( C_i \) takes \( i \) inputs. The language decided by a circuit family \( C \) is \( L(C) = \{ x : C_{|x|}(x) = 1 \} \). \( \text{P/poly} \) is the class of languages that can be decided by a circuit family of polynomial size, that is, where \( |C_n| \leq \text{poly}(n) \).

1. Show that \( \text{P} \subseteq \text{P/poly} \).

2. Show that there are uncomputable languages in \( \text{P/poly} \). Conclude that \( \text{P} \neq \text{P/poly} \).

3. Definition: A circuit family \( C \) is \( \text{P}-\text{uniform} \) if there is a polynomial-time Turing machine that, on input \( 1^n \), outputs a description of the circuit \( C_n \).

   Show that \( \text{P}-\text{uniform} \text{P/poly} \) is equal to \( \text{P} \).

4. Given a class \( \mathcal{C} \) of languages and a function \( f : \mathbb{N} \rightarrow \mathbb{N} \), we define “\( \mathcal{C} \) with \( f \)-bounded advice”, denoted \( \mathcal{C}/f \), as the class of languages \( L \) such that there exists \( L' \in \mathcal{C} \) and there exist strings \( a_1, a_2, a_3, \ldots \) (“a” for “advice”) with \( |a_n| \leq f(n) \) such that for all \( x \),

   \[ x \in L \iff (x, a_{|x|}) \in L'. \]

   In other words, there is a single advice string \( a_n \) that helps \( L' \) decide membership in \( L \) for all strings \( x \) of length \( n \).

   Prove that \( \text{P/poly} \) (defined in terms of circuits as above) is equal to the union of advice classes \( \bigcup_k \text{P/O}(n^k) \). (Hence the notation “\( \text{P/poly} \)”.)

5. A language \( L \) is (polynomially) sparse if there is a polynomial \( p \) such that the number of strings in \( L \) of length \( \leq n \) is at most \( p(n) \).

   (a) Show that all sparse languages are in \( \text{P/poly} \).
(b) Show that $\mathsf{P}/\mathsf{poly} = \mathsf{P}^{\mathsf{SPARSE}}$, that is, $\mathsf{P}/\mathsf{poly}$ is the class of languages $L$ such that there is some sparse language $S$ and $L$ reduces to $S$ by a polynomial-time oracle Turing machine (denoted $L \leq^p_T S$).

6. Show that $\mathsf{P} \neq \mathsf{P}/\mathsf{O}(\log n)$, by showing that the latter has uncomputable languages.

7. (a) Show that search reduces to decision for SAT: there is a function in $\mathsf{FP}^{\mathsf{NP}}$ that, given a Boolean formula $\phi$, either outputs a satisfying assignment to $\phi$ (if one exists), or correctly reports that no satisfying assignments exist.

(b) Despite Question 6, show that $\mathsf{NP} \subseteq \mathsf{P}$ iff $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{O}(\log n)$.

(c) What can you say if $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$?

8. It is natural to wonder whether uncomputable languages are the only thing standing in the way of $\mathsf{P}$ being equal to $\mathsf{P}/\mathsf{poly}$. Here, show that’s not the case, i.e., that $\mathsf{P}/\mathsf{poly} \cap \mathsf{COMP} \neq \mathsf{P}$, i.e., that there are computable languages in $\mathsf{P}/\mathsf{poly}$ that aren’t in $\mathsf{P}$. Hint: Pick a hard but computable language, far outside of $\mathsf{P}$, and encode it in unary. You may assume the Time Hierarchy Theorem: if $T(n) \log T(n) < o(T'(n))$, then $\mathsf{DTIME}(T(n)) \subsetneq \mathsf{DTIME}(T'(n))$. How large must $T'$ be to get this to work against $\mathsf{P}$?

**Resources**

- Sipser §9.3
- Arora & Barak §6.1
- Du & Ko §6.2
- Homer & Selman §8.1
- Hemaspaandra & Ogihara *Complexity Theory Companion* p. 276
- Wigderson §5.2.1.
- Moore & Mertens §6.5