Weekly Challenge 15: Complexity Classes

CS 212 Nature of Computation
Habib University
Ali Muhammad Asad - aa07190

Fall 2023

1. Checking Primality

Explain succinctly why the language recognized by the following Turing machine, M, does not belong to P. Assume the input to be a binary representation of a number.

M = On input n:

1. Check if 2 divides n, if so *reject*.
2. Repeat Step 1 for all numbers less than n. That is, check if 3 divides n. If so *reject*, otherwise check if 4 divides n, if so *reject*, and so on.
3. If all numbers less than n have been checked, *accept*.

Solution: `P` is the class of languages that are decidable in polynomial time on a single tape deterministic Turing Machine. $\texttt{P} = \bigcup_k \text{TIME}(n^k)$

Essentially, the language of M, $L(M)$, is the set of the binary representation of all prime numbers. It takes as input n which is the binary representation of a number x, and rejects if x is not prime, and accepts if x is prime.

The algorithm used by M is quite a straightforward, brute-force approach. The time complexity of this approach, intuitively, is $O(n)$ in terms of the number of divisions M performs. However, considering the length of the input (the number of bits representing n), let's say k bits, n is at most 2^k. So in terms of the input size that M has to compute on, the compelexity is proportional to 2^k, and thus is in $O(2^k)$ which is exponential. Therefore, the language of M is not polynomial, and hence M does not belong to `P`.

■

**Note that the complexity of deciding whether a number is a Prime or not is in polynomial time. But this particular algorithm on this particular machine is not.*