Computer Science Theory
COMS W3261
Lecture 23

Alexander Roth

2014 – 11 – 26

Problems

1. Consider the Language $L_\infty = \{M | M$ is a TM that runs forever on all inputs $\}$. Is L_∞ is RE? True or False.

Solution: False, we can reduce this to L_{ne} which is the not empty language. This is RE, but not decidable. We can construct M' that simulates M on any given input. If M accepts M' halts, otherwise loops forever. M' is in the complement of M. $\overline{L}$ is RE and not recursive, which implies $\overline{L}$ is not RE.

2. $L_{nsub} = \{(M, D) \,|\, L(M) \not\subseteq L(D)\}$. L_{nsub} is RE?

Solution: True, this language is recursively enumerable. Given a DFA D and an input string w, we must decide whether D rejects w. We create a verifier $V(D, w)$ that verifies D on input string w. If D rejects w, we check if M accepts w. Nondeterministically guess every single w. Run it against the verifier, three cases. Accepted by $L(D)$, rejected by both $L(D)$ and $L(M)$, rejected by $L(D)$ and accepted by $L(M)$. There may be a w such that $w \notin L(D)$ but $w \in L(M)$.

3. DNF-TAUT $\in$ CO-NP?

Solution: True

4. DNF-TAUT $\in$ CO-NP? Is it CONP-hard?

Solution: L is C-hard. $\forall L' \in$ CONP. $L' \leq_P$ DNF-TAUT. True, reductions hold even if you complement the language. $L_c \leq_P L'_c$. $L \leq_P L$. If L is NP-hard, then L^c is CONP-hard. $\forall M \in \mathcal{NP}$, $M \leq_P L$ and $M^c \leq_P L^c$. $\{\phi | \exists X \phi(x) = 0\} = L^c$. CNF-SAT $\leq_P L^c$.

5. $L = \{(m, k) | m > k \exists i < k | i$ is a prime factor of $M\}$. Is $L \in \mathcal{NP}$? Is $L \in$ CONP?

Solution: True, suppose there is a language $PR = \{m|m\text{is a prime number}\}$. $PR \in \mathcal{P}$. We can guess all numbers between 2 and $k-1$. Construct a verifier for this problem, $V(m,i)|1 < i < k$. Our V must test if this is a prime factor. If $PR(i)$ accepts, then it's prime. Then do m/i, if it is an integer add it to the set. $L^c = \{(m,k)|\forall i, 1 < i < k$ where i is not a prime factor of $M\}$. Nondeterministically guess a set of prime numbers, from 1 to k. See if it divides m, and if contains all primes, then L^c is in $\mathcal{NP}$.