Lec 2/16

Friday, February 16, 2018 12:45

14. If M is NDTM then $L(M) = \{x \in \{0,1\}^* : M \text{ accepts } x\}$

15. The <u>time</u> taken by NDTM M on input x is

min $\{\#$ of steps taken by M on input x w/ guess $y : y \in \{0,1\}^*$, M accepts x w/ guess $y\}$

the minimum of $\emptyset$ is 0. time is 0 if x is not accepted. (Note guessing takes no time)

16. Let M be an NDTM. the time complexity of M is a function $T_M : \omega \to \omega$

defined by $T_M(n) = \max(\{\text{time}_M(x) : x \in \{0,1\}^n\} \cup \{1\})$.

17. An NDTM M is a polynomial-time NDTM if there is some $p(X) \in \mathbb{N}[X]$ s.t. $\forall n \in \mathbb{N}$, $T_M(n) \le p(n)$.

18. NP is the set of languages accepted by polynomial-time NDTMs.

<u>Lemma</u>: $P \subseteq NP$.

<u>Proof sketch</u>: Let $L \in P$. let M be a polynomial-time DTM which accepts L s.t. $T_M(n) \le q(n)$ $\forall n \in \omega$. Then an NDTM N which just does what M does regardless of guess has $T_N(n) \le T_M(n) \le q(n)$ as well so N is a p-t NDTM accepting L. □

[Actually this doesn't quite work: erase $p(n)$ bits of the guess on input $x \in \{0,1\}^n$, then simulate M.

[or take two steps left, reject if not blank, else simulate M on x.

<u>Lemma</u> every problem in NP is recursive.

Proof: Let a DTM simulate an NDTM by guessing in lex-order. There are only a finite # of possible guesses that could lead to acceptance, less than $p(n)$. So we simulate M on every guess of length $\le p(n)$ for $p(n)$ time. If it accepts, accept, if it doesn't, reject. $\square$

19. A DTM M computes a function $f: \{0,1\}^* \longrightarrow \{0,1\}^*$ if $(\forall x \in \{0,1\}^*)$ [M w/ input x eventually accepts w/ $f(x)$ on squares 1 through $|f(x)|$ and square $|f(x)|+1$ being blank.

20. Let $L_1, L_2 \subseteq \{0,1\}^*$. A polynomial transformation (reduction) from L_1 to L_2 is a function $f: \{0,1\}^* \longrightarrow \{0,1\}^*$ such that:

1) $\exists$ a poly-time DTM which computes f.

2) $\forall x \in \{0,1\}^*$, $x \in L_1 \Leftrightarrow f(x) \in L_2$.

We say $L_1 \le_p L_2$ if such a transformation exists.

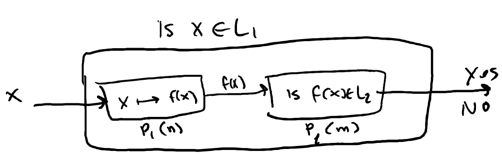
21. A language L is NP-complete if

1) $L \in NP$

2) $\forall L' \in NP$, $L' \le_p L$.

Lemma $(L_1 \le_p L_2$ and $L_2 \in P) \Longrightarrow L_1 \in P$.

Proof



$p_1(n) + p_2(|f(x)|) \le p_1(n) + p_2(p_1(n))$ polynomial.

Lemma $(L_1 \le_p L_2$ and $L_2 \le_p L_3) \Longrightarrow L_1 \le_p L_3$

Lemma $(L_1 \leq_p L_2 \text{ and } L_2 \leq_p L_3) \implies L_1 \leq_p L_3$

Proof