INTRODUCTION TO COMPLEXITY THEORY

(Sipser, Chapter 7)

SO FAR WE HAVE DEFINED A GENERAL MODEL OF COMPUTATION (TURING MACHINES) AND ARGUED THAT TM'S ARE POWERFUL ENOUGH TO CAPTURE ANY COMPUTATION. (CHURCH'S THESIS.) THE REASON FOR GOING TO ALL OF THIS TROUBLE IS SO THAT WE CAN RIGOROUSLY UNDERSTAND WHAT IS/ISN'T COMPUTABLE. (A LANGUAGE/FUNCTION IS COMPUTABLE IF SOME TM COMPUTES IT.) WE CHARACTERIZED ALL LANGUAGES OVER $\{0,1\}$ AS EITHER (i) RECURSIVE, (ii) RE BUT NOT RECURSIVE, OR (iii) NOT R.E.

FIRST WE SHOWED, USING DIAGONALIZATION, THAT L_D IS NOT RE. THIS ARGUMENT FORMALIZES THE FACT THAT MANY LANGUAGES CANNOT BE RE BECAUSE THE SET OF TM'S (over $\{0,1\}$) IS COUNTABLE WHEREAS THE SET OF ALL LANGUAGES (over $\{0,1\}$) IS UNCOUNTABLE.

THEN USING REDUCTIONS, WE SHOWED MANY EXAMPLES OF OTHER LANGUAGES THAT LIE IN (ii) OR (iii). MANY IMPORTANT PROBLEMS FOR COMPUTER SCIENCE, SUCH AS THE HALTING PROBLEM (HALT) ARE UNFORTUNATELY NOT RECURSIVE – i.e. THERE IS NO COMPUTER PROGRAM FOR SOLVING THEM.

KNOWING THAT A FUNCTION OR LANGUAGE IS RECURSIVE ISN'T GOOD ENOUGH IF THE RUNTIME IS HUGE $(>2^n)$. THIS BRINGS US TO THE STUDY OF COMPLEXITY THEORY WHICH IS THE STUDY OF THE INHERENT RESOURCES (TIME, SPACE, RANDOMNESS) NEEDED TO SOLVE PARTICULAR PROBLEMS.

TIME COMPLEXITY AND "O" NOTATION

DEF'N
LET M BE A TM (OVER Σ^*) WHERE M HALTS ON ALL INPUTS. THE TIME COMPLEXITY OF M IS A FUNCTION $f_M : \mathbb{N} \to \mathbb{N}$ WHERE

$$f_M(n) = \max \{ t_M(x) \mid x \in \Sigma^*, |x| = n \}$$

NUMBER OF STEPS UNTIL M HALTS ON X.

SINCE EXACT TIME COMPLEXITY IS USUALLY COMPLEX, IT IS CUSTOMARY TO ESTIMATE IT AS FOLLOWS.

DEF'N LET $f, g : \mathbb{N} \to \mathbb{R}^+$ ← REAL NUMBERS GREATER THAN 0
$f(n) = O(g(n))$ [READ: $f(n)$ IS BIG-O OF $g(n)$]
IF $\exists c, n_0$ SUCH THAT $\forall n \geq n_0$ $f(n) \leq c \cdot g(n)$.

EXAMPLES:

$$5n^3 + 2n + 300 = O(n^3)$$
$$5n^3 + 2n + 300 = O(n^4)$$
$$10 \log_2 n + 20 \log_2 \log_2 n = O(\log_2 n)$$

DEF'N LET $f, g : \mathbb{N} \to \mathbb{R}^+$. $f(n) = o(g(n))$
[READ: $f(n)$ IS LITTLE-O OF $g(n)$] IF $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$.

EXAMPLES:

$$\sqrt{n} = o(n)$$
$$n^2 \log n = o(n^2)$$

BIG-O: $f(n)$ IS AT MOST $g(n)$ IGNORING CONSTANT FACTORS
LITTLE-O: $f(n)$ IS STRICTLY LESS THAN $g(n)$, IGNORING CONSTANT FACTORS.

THE CLASS P

DEF'N LET $t: \mathbb{N} \to \mathbb{N}$.

$\text{TIME}(t(n)) = \{L \mid L$ IS A LANGUAGE OVER $\{0,1\}^*$ THAT IS DECIDED BY A $O(t(n))$ TIME TM$\}$

DEF'N P IS THE CLASS OF ALL LANGUAGES (OVER $\{0,1\}^*$) THAT ARE DECIDABLE IN POLYNOMIAL TIME.

$$P = \bigcup_k \text{TIME}(n^k)$$

LANGUAGES RECOGNIZABLE IN POLY TIME

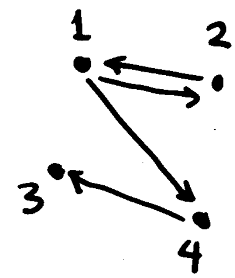
EXAMPLE 1

$\text{PATH} = \{\langle s,t,G\rangle \mid G$ IS A DIRECTED GRAPH THAT HAS A PATH FROM s to $t\}$

IS IN P, UNDER A REASONABLE ENCODING OF G, s, t

OUR ENCODING: G AS AN ADJACENCY MATRIX

g:



$\Rightarrow$ M

	1	2	3	4
1	0	1	0	1
2	1	0	0	0
3	0	0	0	0
4	0	0	1	0

REPRESENT '0' IN MATRIX BY 00, '1' BY 01.

ENCODE $\langle 3, 4, g\rangle$ BY:

$\underbrace{000}_{3}11\underbrace{0000}_{4}11\underbrace{0001\,00\,01\,11\;\;01\,00\,00\,00\,11\,00\,00\,00\,00\,11\,00\,00\,01\,00\;11}_{G}$

COMPUTE M^k: ENTRY $(i,j) \geq 1$ IFF THERE IS A PATH OF LENGTH k FROM i to j.

EXAMPLE 2

PRIME = $\{x \mid x \in \{0,1\}^*$ IS A BINARY ENCODING OF A PRIME NUMBER$\}$

IT IS NOT KNOWN WHETHER OR NOT PRIME $\in$ P.

EXAMPLE 3

RELPRIME = $\{\langle x,y\rangle \mid x,y \in \{0,1\}^*$ ARE RELATIVELY PRIME$\}$ IS IN P BY EXTENDED EUCLIDEAN ALGORITHM.

DEF'N FP = $\{f : \Sigma^* \to \Sigma^* \mid$ M COMPUTES f FOR SOME POLYNOMIAL TIME TM$\}$

FUNCTIONS SOLVABLE IN POLYTIME

Q: IS THE NOTION OF POLYNOMIAL TIME ROBUST??

YES, BUT NOT AS ROBUST AS THE NOTION OF COMPUTABLE.
DEPENDS ON ENCODING WHICH SHOULD BE EFFICIENT.
THE FOLLOWING THESIS EQUATES "EFFICIENTLY COMPUTABLE" WITH POLYTIME.

POLYTIME THESIS

IF A LANGUAGE/FUNCTION IS RECOGNIZED/COMPUTED BY SOME POLYTIME ALGORITHM/MACHINE, THEN THAT LANGUAGE/FUNCTION CAN BE COMPUTED BY A POLYTIME TM.

TRUE FOR ALL CURRENTLY REALIZABLE ALGORITHMS

POSSIBLE COUNTEREXAMPLE: QUANTUM COMPUTING.
CAN FACTOR IN POLYNOMIAL (QUANTUM) TIME

THE CLASS NP

DEF'N

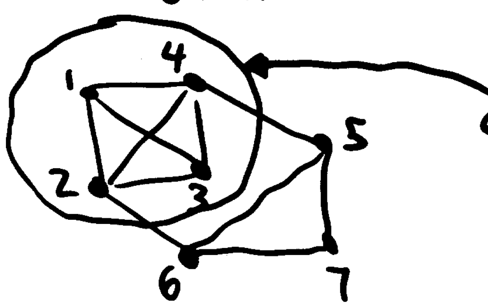
LET L BE A LANGUAGE OVER Σ^*.
$L \in NP$ IF THERE IS A 2-PLACE RELATION $R \subseteq \Sigma^* \times \Sigma^*$ SUCH THAT $R(w,c)$ IS COMPUTABLE BY A TM, V, WHERE V IS POLYTIME IN $|w|$, AND SUCH THAT

$$x \in L \Leftrightarrow \exists y \in \Sigma^* \text{ SUCH THAT } R(x,y)$$

V IS CALLED A POLYNOMIAL-TIME VERIFIER FOR L.

EXAMPLE 1 $L = \{\langle G,k,n\rangle \mid G$ IS AN UNDIRECTED GRAPH ON n VERTICES CONTAINING A CLIQUE OF SIZE $k\}$

G:



CLIQUE OF SIZE 4
SO $\langle G,4,7\rangle \in L$

VERIFIER $V(\langle G,k,n\rangle, y)$: VIEW y AS A STRING OF LENGTH n, WITH k 1's. y ENCODES A SUBSET OF k VERTICES IN G. ACCEPT $(\langle G,k,n\rangle, y)$ IFF G HAS A CLIQUE ON THE k VERTICES ENCODED BY y.

IN OUR EXAMPLE $(\langle G,4,7\rangle, 1111000)$ accepted by V BUT $(\langle G,5,7\rangle, y)$ REJECTED BY V FOR ALL y.