Lecture I
THE TURING MODEL (I)

§1. SIMPLE TURING MACHINES

This basically follows chapter 2 of the text.

The basic notion of Turing machines as a 4-tuple

$$M = (K, \Sigma, \delta, s)$$

interpreted appropriately. We will, however, called this **simple Turing machine** (sTM). The k-string version is more usually called **multitape Turing machine** (mTM) or simply "Turing machine" (TM).

Another conventions simplifies our notation: instead of (K, Σ, δ, s), we note that δ already implies K and Σ. We also fix a "universal" s_0 as our start state. Hence a sTM is given by δ, its transition table!

We use M to (1) decide languages (2) accept languages. This gives rise to recursive languages and r.e. languages. Clearly, recursive languages are r.e.

EXAMPLE: Palindromes can be decided by sTM in $O(n^2)$ time.

§2. MULTITAPE TURING MACHINES

Papadimitriou call them k-string Turing machines, a non-standard name. We prefer "multitapeTM".

TIME COMPLEXITY will be defined using mTM's.

Thus $TIME(f)$ are the languages decided in time f. Also, $P = \cup_{k>0} TIME(n^k)$.

Theorem: If M is a mTM which accepts in time f, then there is a sTM that accepts $L(M)$ in time $O(f^2)$.

Theorem: (Linear Speedup) $TIME(f) \subseteq TIME(\varepsilon f + n)$.

SPACE COMPLEXITY needs a simple variation of mTM. Suppose Space is just the expected definition. Then palindromes need at least linear space. But consider the following 2-tape TM that seems to use only log space!