Lecture I
THE TURING MODEL (II)

§1. SPACE COMPLEXITY

We continue our discussion of the Turing model in chapter 2.

SPACE COMPLEXITY needs a simple variation of multitape TM.

To motivate this, suppose space is the expected definition. Then palindromes need at least linear space (why?)

But consider the following 3-tape TM that seems to use only log space! On tape 2, we keep an index $0 \le i \le n$ (where n is the input length). On tape 3, we have another index j $(0 \le j \le i)$. Let the input be w. Basically, we use i to indicate that we want check that $w[i] = w[n-i+1]$. We use j to keep track of our position from one end of the input tape.

Hence we define the following class of multitape TM (also known as an **offline multitape TM**):

- Tape 1 is read only. This is sometimes called a "2-way input tape" since the input head can move back and forth.
- The head on tape k (the last tape) is not allowed to move left. Hence it is essentially "write-only". The contents of this tape is considered the output when the machine halts.
- When we count space, we only count the number of cells that were used in tapes 2 to $k-1$.

Then we define $SPACE(f)$ to be the class of languages decided by some offline multitape TM that uses space at most $f(n)$.

E.g. It follows from the motivating example that the palindrom language belongs to $SPACE(\log n)$.

The following is easy, using super symbol tricks.

Theorem: for all $f(n)$ and $\epsilon > 0$, $SPACE(f) \subseteq SPACE(\epsilon.f)$.

§2. RAM

Random Access Machines (RAM) is another natural model of computation that matches modern computers even more closely than Turing machines.

It turns out that TMs and RAMs can mutually simulate each other within a quadratic time factor.

This is another vindication of the Church-Turing Thesis. It is also an illustration of the "polynomial thesis" that says that all reasonable general models of computation has time complexity that are within a quadratic factor of Turing machine time.

§3. Nondeterminism

We introduce the notion of nondeterminism.

$NTIME(f)$ is the class of languages decided by a nondeterministic TM in time $\leq f(n)$. The most famous example of nondeterministic classes is the following

$$NP = \cup_{k>0} NTIME(n^k).$$

Theorem: $NTIME(f) \subseteq \cup_{d>1} TIME(d^f)$

The proof is a direct simulation of all computation paths of a nondeterministic machine that decides in time f.