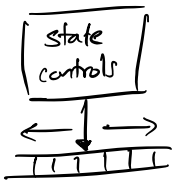
COMS W3261 – Lecture 9, Part 1:

Recognizing, Deciding, & Enumerating.

Teaser: Is the language of palindromes over the alphabet $\{0,1\}$ Turing-recognizable?



$L = \{ ww^R, w0w^R, w1w^R \mid w \in \{0,1\}^* \}$.

Implementation-level description of a TM M_1 for L:

M_1 = "On an input s, we

(0) accept if the tape contains 0 or 1 symbols (shuttle back and forth to read string length.)

(1) "Remember" the leftmost symbol and erase it.

(2) Traverse the tape and check if the rightmost symbol matches the leftmost symbol.

- If yes – erase the rightmost, go back to the leftmost uncrossed symbol, and repeat from step 0.
- If no – reject."

✓

0 ... 0

0 ... 1

✗

Announcements: HW #5 due 8/2/21 @ 11:59 PM EST.

See Ed for information about the final (pinned, ask questions)

Readings: Sipser 3.1 (TMs)
Sipser 3.2 (Variations on TMs, Multitape, Nondeterministic, Enumerators.
Sipser 3.3 ("From Turing Machines to Algorithms")

Today: 1. Review of TMs
2. Variant TMs
3. TMs → more general notions of 'algorithm'

1. Some TM examples (implementation level.)

Example.

Goal: Deciding $C = \{a^i b^j c^k \mid ij = k \text{ and } i, j, k \geq 1\}$.

Deciding: YES if in the language and NO otherwise
(Recognizing: YES if in the language, may loop or reject otherwise)

M_3 = "On input string w:

1. Scan left to right to ensure we have a string matching $a^+b^+c^+$. (Reject if not.)
2. Return to the leftmost square.
3. Cross off the first a. Then, shuttle back and forth between bs and cs, crossing off one c for each b crossed off. (Reject if we run out of cs.)
4. Restore (uncross) all the b's and return to step 2.
5. Once all a's are crossed, accept if no c's remain uncrossed.

Example: aaa bb cccccc
aaa bb cccccc
/ / / / / / /

(Multiplication on TMs).

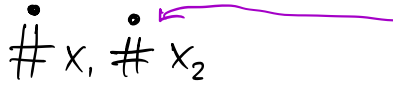
Example 2. (Element Distinctness).

Goal: Decide $E = \{\#x_1 \#x_2 \#x_3 \cdots \#x_\ell \mid \text{each } x_i \in \{0,1\}^* \text{ and } x_i \neq x_j \text{ for } i \neq j\}$

Idea: Compare x_1 with all x_i, $i > 1$, reject if we find a match. Then compare x_2 with all x_i, $i > 2$, and so on.

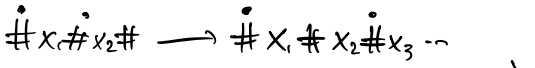
M_4 = "On input w:

1. Check to make sure the string is in the right format, reject if not.
2. Accept if there is <2 inputs in the right format.
3. Otherwise, mark the first two # like this:



this is OK because we can have $\#, \# \in \Gamma$.

4. Scan the two strings to the right of the marked # and reject if they match.
5. If possible, move the right mark to the right and repeat step 4.



If not possible (right mark is on the last #), move the left mark forward and the right mark back to the next # after the left mark.

Finally, if we've marked the last two #'s, accept.

Now: know how to nest two loops
know how to multiply
know how to check string matching and distinctness,
...

Next: discuss variants on TMs.