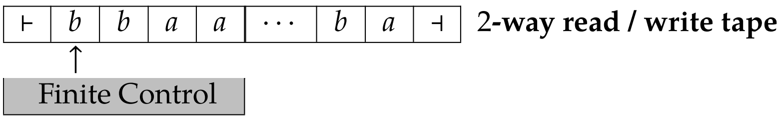
Tutorial 7
Computability Theory

Turing Machines, Equivalent Models and Unrestricted Grammars

1. Design a *total* Turing Machine (TM), $\mathcal{K}$, that accepts the language, $\mathcal{L}(\mathcal{K}) = \left\{0^{2^k} \,\middle|\, k \in \mathbb{N}\right\}$.

2. A linear bounded automaton (LBA) is exactly like a 1-tape TM, except that the input string $x \in \Sigma^*$ is enclosed in left and right endmarkers $\vdash$ and $\dashv$ which may not be overwritten. The machine is constrained never to move left of $\vdash$ or right of $\dashv$. It is allowed to read/write between these markers.



   (a) Give a rigorous formal definition of deterministic linearly bounded automata, including a definition of configurations and acceptance.

   (b) Let M be an LBA with state set Q of size k and tape alphabet Γ of size m. How many possible configurations are there on input x with $|x| = n$?

   (c) Argue that it is possible to detect in finite time whether an LBA loops on a given input. (*Hint*: Think of using Part (b).)

3. Show that the class of recursively enumerable sets is closed under union and intersection.

4. Prove that, a language L is recursive if and only if there is an enumeration machine enumerating the strings of L in a non-decreasing order of length (string of the same length may be arranged in lexicographic order). For example, strings of $\{0, 1\}^*$ would be arranged as $0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \ldots$.

5. Design an NTM to accept the language $\left\{wxyxz \mid w, x, y, z \in \{0, 1\}^*,\ |x| = 2024\right\}$.

6. A TM $\mathcal{M}$ has a two-way infinite tape. Initially, all cells on the tape are blank. Only one cell is storing the symbol #. The head of $\mathcal{M}$ is pointing to a blank. The task of $\mathcal{M}$ is to locate the cell storing #. Propose a strategy for doing this, (a) if $\mathcal{M}$ is a DTM, (b) if $\mathcal{M}$ is a NTM.

7. Consider the following languages:

   - $L_a = \left\{a^n w c^n \mid w \in \{a, b, c\}^*,\ n \geq 0, \text{ and } \#a(w) = n\right\}$
   - $L_b = \left\{a^n w c^n \mid w \in \{a, b, c\}^*,\ n \geq 0, \text{ and } \#b(w) = n\right\}$
   - $L_c = \left\{a^n w c^n \mid w \in \{a, b, c\}^*,\ n \geq 0, \text{ and } \#c(w) = n\right\}$
   - $L_d = \left\{w c^m d^n \mid w \in \{a, b\}^*, m = \#a(w), \text{ and } n = \#b(w)\right\}$
   - $L_e = \left\{c^m w d^n \mid w \in \{a, b\}^*, m = \#a(w), \text{ and } n = \#b(w)\right\}$
   - $L_f = \left\{w \in \{a, b, c\}^* \mid \#a(w) = \#b(w) = \#c(w)\right\}$
   - $L_g = \left\{ww \mid w \in \{a, b\}^*\right\}$
   - $L_h = \left\{a^i b^j c^k d^l \mid i = k \text{ and } j = l\right\}$

   where, $\#a(w)$, $\#b(w)$, and $\#c(w)$ denote the number of a's, b's, and c's in the string w.

   Answer the following.

(a) Design separate TMs to accept each of the above languages ($L_a - L_h$).

(b) Provide separate unrestricted grammars for each of the above languages ($L_a - L_h$).

(c) Can you mechanically construct the unrestricted grammars (separately) for each of the above languages ($L_a - L_h$) from their corresponding TMs designed in Part (a)?

8. Consider the unrestricted grammar over the singleton alphabet $\Sigma = \{a\}$, having the start symbol S, and with the following productions.

$$S \to AS \mid aT, \qquad Aa \to aaaA, \qquad AT \to T, \qquad T \to \epsilon$$

What is the language generated by this unrestricted grammar? Justify.

9. Prove that any grammar, defined over non-terminals N and terminals Σ, can be converted to an equivalent grammar with rules of the form $\alpha A \gamma \to \alpha \beta \gamma$ for $A \in N$ and $\alpha, \beta, \gamma \in (\Sigma \cup N)^*$.

10. A Jump Turing Machine (JTM) $\mathcal{J} = (Q, \Sigma, \Gamma, \delta, \vdash, \sqcup, s, t, r)$ is like a standard one-tape Turing Machine (TM) with the only exception that each transition of $\mathcal{J}$ is of the form $\delta(p, A) = (q, B, m)$, where $p, q \in Q$, and $A, B \in \Gamma$, and $m \in \mathbb{Z}$. This means that if the finite control of J is in the state p and the head of $\mathcal{J}$ scans the tape symbol A, then the state changes to q, the content of the tape cell is changed from A to B, and the head jumps by m (integer) cells relative to the current position. If $m = 0$, the head stays at the current cell. If $m > 0$, the head makes a right jump. If $m < 0$, the head makes a left jump with the understanding that if the head is at position i on the tape and $|m| > i$, then the head goes to the leftmost cell (which stores the left end-marker $\vdash$). Also assume that if A is $\vdash$, then $m \geq 0$. Prove that a JTM is equivalent to a TM.

11. Let $f : \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n^2\}$ be the bijective function $f(i, j) = (i-1)n + j$ for $1 \leq i, j \leq n$. Consider the language L that consists of all strings $x\#0^k$ of the following form.

- $x \in \{0, 1\}^*$, and $|x| = n^2$ for some integer $n \geq k \geq 1$. Denote $x = x_1 x_2 \ldots x_{n^2}$.
- For each i, j satisfying $1 \leq i < j \leq n$, we have $x_{f(i,j)} = x_{f(j,i)}$, and for each i in the range $1 \leq i \leq n$, we have $x_{f(i,i)} = 0$.
- There exists a set $S \subseteq \{1, 2, \ldots, n\}$ with $|S| \geq k$ such that for each $i, j \in S$, we have $x_{f(i,j)} = 0$.

Design a nondeterministic Turing machine (NTM) accepting L.

(*Hint*: Think of undirected graphs.)