COMS 3003A

Project 1

21 March, 2025

This project contributes 10% to your final mark
Due Date: 6 April 2025

Design and implement two C++ programs, an encoder, and then a simulator for Turing Machines. The submission is a single `.cpp` file for each question.

1. **(30 points)** Implement a program that takes as input an encoding table $f : \Sigma_1 \to \Sigma_2^*$, and a string $s_1 s_2 ... s_n \in \Sigma_1^*$, which then outputs $f(s_1)f(s_2)...f(s_n) \in \Sigma_2^*$.

   - The input of f must be in the following format: $\langle f \rangle :=$ "s_1=w_1|s_2=w_2|...|s_k=w_k|", where $s_i \in \Sigma_1$ and $w_i \in \Sigma_2^*$ eg. $f(0) = 00$, $f(1) = 10$ and, $f(\#) = 11$ has $\langle f \rangle :=$"0=00|1=10|#=11|".
   - A symbol s is represented by a single UTF-8 / ASCII character.
   - Your program will not be tested on strings that have invalid symbols i.e. if f is defined for $\Sigma_1 \to \Sigma_2^*$, you do not need to handle $f(s)$ where $s \notin \Sigma_1$
   - Sample inputs are in table 1.

Sample Input	**Sample Output**
a=00|b=01|c=10| abc	000110
0=00|1=01|#=10|;=11| 00#00#01#00#1;	000010000010000110000010011 1

Table 1: Sample input for Submission 1 - Encoder

2. **(70 points)** Implement a program that takes as input the binary-encoded transition table of a two-way-tape Turing Machine $\langle\delta\rangle$, where $M = \langle Q, \Gamma, \Delta, q_{\text{init}}, q_{\text{accept}}, q_{\text{reject}}, \delta\rangle$, and some string $x \in \Gamma^*$, and then outputs the computation result of $M(x)$. Should the simulated computation encounter an error in the machine, it must terminate and output "Error" instead.

   - Direction instructions L and R are encoded as $\langle L\rangle = 0$ and $\langle R\rangle = 1$ respectively.
   - A transition $\delta(q_i, s_k) = (q_j, s_l, D)$ is encoded as $\langle\delta(q_i, s_k)\rangle := E(b(i)\#b(k)\#b(j)\#b(l)\#\langle D\rangle\,;)$ where $b(n)$ is the binary representation of $n \in \mathbb{N}$, and mapping $E : \{0, 1, \#, ;\} \to \{0, 1\}^*$ is

$$E(s) = \begin{cases} 00, & \text{if } s = 0 \\ 01, & \text{if } s = 1 \\ 10, & \text{if } s = \# \\ 11, & \text{if } s =; \end{cases}$$

   - The transition table encoding $\langle\delta\rangle$ is therefore the concatenation of each transition's encoding.
   - For convenience, you can get a $\langle\delta\rangle$ without mapping E applied by going onto the simulator at https://tm.seagrass.co.za, and exporting the transition table by going to `File > Export > Transition Table`, enabling state counter, symbol counter, direction as natural with $L = 0$ and $R = 1$, and using base 2.
   - For $\langle\delta\rangle$ to be a sufficient encoding of M (why?), you must make the following assumptions:
     (a) $\Delta = \{\sqcup\} \cup \Gamma$ and $s_0 = \sqcup$
     (b) $q_{\text{init}} = q_0$
     (c) $q_{\text{accept}} = q_1$
     (d) $q_{\text{reject}} = q_2$
   - The input alphabet Γ is equal to or a subset of $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, but ordered such that the i-th symbol is the number i ($s_i = i$).
   - Your program will only be tested on strings that represent a Turing Machine and on unencoded input strings from Γ^*, however, it may be tested on a non-deterministic / incomplete Turing Machine.
   - Sample input and output is provided in table 2, where the input machine cases are

$$\langle\delta\rangle = E(0\#0\#10\#0\#1;0\#1\#0\#10\#1;0\#10\#1\#10\#1;)$$

     and

$$\langle\delta\rangle = E(0\#0\#10\#0\#1;0\#10\#1\#10\#1;)$$

Sample Input	**Sample Output**
00100010010010001001110010011000100100100111001001001001100100100111 0	Reject
00100010010010001001110010011000100100100111001001001001100100100111 1	Accept
0010001001001000100111001001001001100100100111 0	Error

Table 2: Sample Input for Submission 2 - Simulator