COMP2270/6270 – Theory of Computation
Tenth Week

School of Electrical Engineering & Computing
The University of Newcastle

Exercise 1) Construct a standard 1-tape Turing machine M to compute each of the following functions:

a) The function sub_3, which is defined as follows:

$$sub_3(n) = \begin{cases} n-3 & \text{if } n > 2 \\ 0 & \text{if } n \leq 2. \end{cases}$$

Specifically, compute sub_3 of a natural number represented in binary. For example, on input `10111`, M should output `10100`. On input `11101`, M should output `11010`. (Hint: you may want to define a subroutine.)

b) Multiplication of two unary numbers. Specifically, given the input string $<x>$`;`$<y>$, where $<x>$ is the unary encoding of a natural number x and $<y>$ is the unary encoding of a natural number y, M should output $<z>$, where z is the unary encoding of xy. For example, on input `111;1111`, M should output `111111111111`.

Exercise 2) Define a Turing Machine M that computes the function f: $\{\texttt{a}, \texttt{b}\}^* \rightarrow N$, where:

$f(x)$ = the unary encoding of $max(\#_{\texttt{a}}(x), \#_{\texttt{b}}(x))$.

For example, on input `aaaabb`, M should output `1111`. M may use more than one tape. It is not necessary to write the exact transition function for M. Describe it in clear English.

Exercise 3) Encode the following Turing Machine as an input to the universal Turing machine:

$M = (K, \Sigma, \Gamma, \delta, q_0, \{h\})$, where:

$K = \{q_0, q_1, h\}$,

$\Sigma = \{\texttt{a}, \texttt{b}\}$,

$\Gamma = \{\texttt{a}, \texttt{b}, \texttt{c}, \square\}$, and

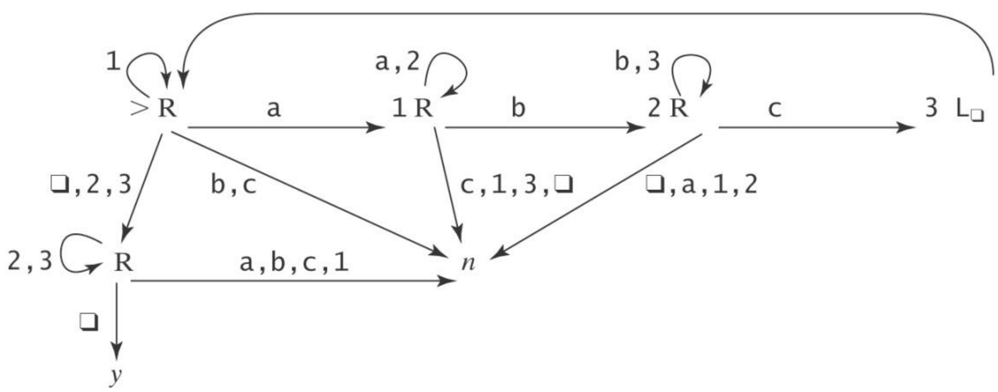
δ is given by the following table:

q	σ	$\delta(q, \sigma)$
q_0	A	$(q_1, \texttt{b}, \rightarrow)$
q_0	B	$(q_1, \texttt{a}, \rightarrow)$
q_0	❑	$(h, ❑, \rightarrow)$
q_0	C	$(q_0, \texttt{c}, \rightarrow)$
q_1	A	$(q_0, \texttt{c}, \rightarrow)$
q_1	B	$(q_0, \texttt{b}, \leftarrow)$
q_1	❑	$(q_0, \texttt{c}, \rightarrow)$
q_1	C	$(q_1, \texttt{c}, \rightarrow)$

Exercise 4) What is the minimum number of tapes required to implement a universal Turing machine?

Exercise 5) In Example 17.9, we showed a Turing machine that decides the language W`c`W. If we remove the middle marker `c`, we get the language WW. Construct a Turing machine *M* that decides WW. You may exploit nondeterminism and/or multiple tapes. It is not necessary to write the exact transition function for *M*. Describe it in clear English.

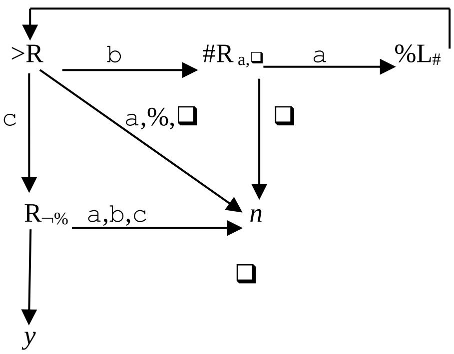
Exercise 6) Consider the following Turing Machine *M*, taken from the book, that decides the language $A^nB^nC^n = \{a^n b^n c^n : n \geq 0\}$:



Modify *M* so that it accepts $\{a^n b^n c^{2n} : n \geq 1\}$.

Exercise 7) Consider a three-tape Turing machine *M*, where $\Gamma_M = \{❑, \mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Suppose that we want to simulate *M* with a one-tape Turing machine *T* using the technique described in Section 17.3.1. How large must Γ_T be?

Exercise 8) Give a clear formal description of language accepted by each of these Turing machines: $\Sigma M = \{a, b, c\}$. $M =$



REFERENCES

[1] Elaine Rich, Automata Computatibility and Complexity: Theory and Applications, Pearson, Prentice Hall, 2008.