Harvard University Extension School
Computer Science E-121

Problem Set 5

Due October 28, 2016 11:59pm

Problem set by Walter Thornton

Collaboration Statement: I worked alone and only with course materials.

PROBLEM 1 (4+2 points, suggested length of 1 page)

(A) Transform the grammar $G = (V, \Sigma, R, S)$ where $V = \{S, A, B\}$, $\Sigma = \{a, b\}$, and $R = \{S \to bA \mid aB, A \to bAA \mid aS \mid a, B \to aBB \mid bS \mid b\}$ into an equivalent grammar in Chomsky normal form.
(B) Check whether the string $bbbaaaab$ is generated by the grammar, using the recognition algorithm for grammars in Chomsky normal form given in class. Show the complete filled-in matrix for the string and explain how you know that the string is or is not generated by the grammar.

Solution.

(A)
$\mathrm{S}_0 \to CA \mid DB$
$S \to CA \mid DB$
$A \to CF \mid DS \mid a$
$B \to CS \mid DG \mid b$
$C \to b$
$D \to a$
$F \to AA \; G \to BB$

(B)

S_0, S							
A	A						
S_0, S	F	F					
B	A	$\emptyset$	$\emptyset$				
G	S_0, S	F	$\emptyset$	F			
$\emptyset$	B	A	$\emptyset$	$\emptyset$	A		
G	G	S_0, S	F	F	F	S_0, S	
B, C	B, C	B, C	A, D	A, D	A, D	A, D	$B.C$
b	b	b	a	a	a	a	b

The string bbbaaaab is accepted. We know this because the start state is in the top left hand corner; it is the result of a bottom up derivation.

PROBLEM 2 (3+3 points, suggested length of 1/2 page)

(A) Suppose we have a finite set of TMs, $\{M_i | i \in \mathbb{N} \text{ and } i < N\}$ for some finite N, over some alphabet Σ with the following properties:

- $\forall i \neq j,\ L(M_i) \cap L(M_j) = \varnothing$
- $\bigcup_i L(M_i) = \Sigma^*$

State whether the language accepted by each TM must be recursive, and prove why or why not.

(B) Let $\Sigma = \{a\}$. Define a TM such that given a string of n a's, the TM computes a string of $3n$ a's, i.e. 3 times the length. You do not have to provide the full tuple but a sufficient description, i.e. Q, Γ, δ

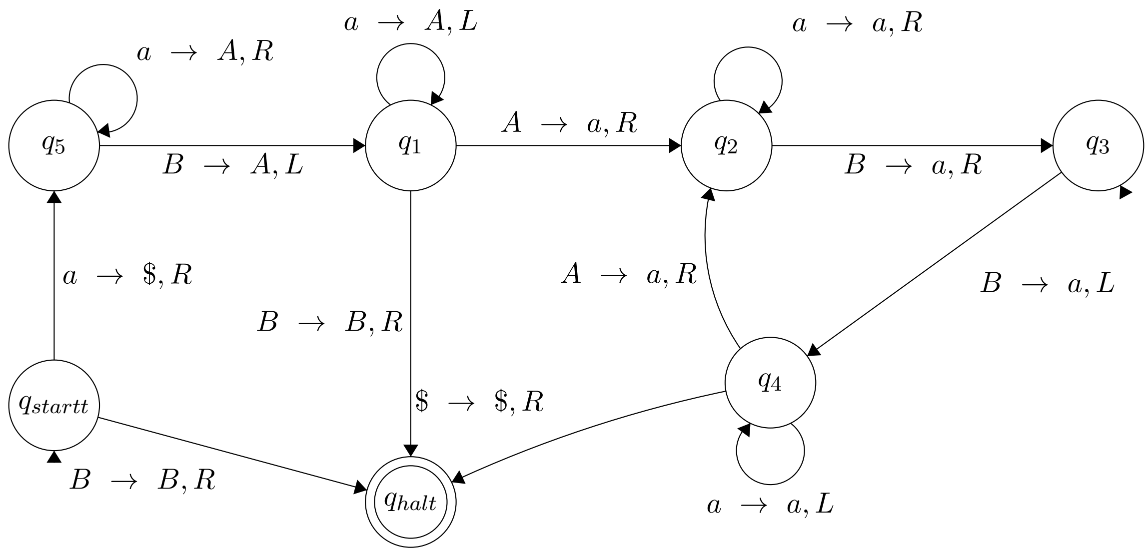
Solution.

(A)

The language accepted by each TM must be recursive.

A language is recursive if there is some Turing machine that accepts every string in the language and rejects every other string, over the same alphabet, that is not in the language.
Since $\bigcup_i L(M_i) = \Sigma^*$ and the language of each Turing machine is disjoint from one another, we can construct a finite number of Turing machines that each accept one language. Since every language is represented in Σ^*, then there is some Turing machine that accepts it, and all others will reject or loop. Given that one of these finite number of TMs accept each L, then both L and $\overline{L}$ are recursively enumerable.
If if both L and $\overline{L}$ are r.e., then L is recursive.
(Theorem 4.22)
Therefore the language accepted by each TM must be recursive.

(B) $Q = \{q_{start}, q_1, q_2, q_3, q_4, q_5, q_{halt}\}$
$\Sigma = \{a\}$
$\Gamma = \{a, A, B, \$\}$ where B is the blank space character.
δ is defined by the following state diagram



PROBLEM 3 (4+2 points, suggested length of 2/3 a page)

Consider a Turing machine M with input alphabet Σ_0 and tape alphabet Γ_0 (where $|\Gamma_0| = m \geq 2$) deciding a language L.

(A) Show how to construct a Turing machine with input alphabet Σ_0 and tape alphabet $\Gamma = \Sigma_0 \cup \{\sqcup, 0\}$ that decides L while only ever writing $\sqcup$ or 0 to the tape, by converting its input into some encoding over $\{\sqcup, 0\}$ and then processing that encoding. Please provide an implementation-level description.

(B) Compare your constructed machine's and M's time to halt. That is, if M takes N steps to halt on input w, approximately how many steps, as a function of N, m, and perhaps other parameters of M, will your transformed machine take? Your answer should not be longer than 3 sentences. (You should ignore the process of converting the input into the encoding over $\{\sqcup, 0\}$; so focus only on the running time of processing the string after encoding it.)

Solution.

(A) 1. encode the L(M) as a distinct number of 0s for each Σ of L(M) followed by $\sqcup$ 2. Use the encoding of M as the input for TMM_2. 3. TMM_2 reads a number of 0s, moving right until $\sqcup$. 4. If it is in the language, accept,else reject

(B) n^3

PROBLEM 4 (5 points, suggested length of 1/2 page)

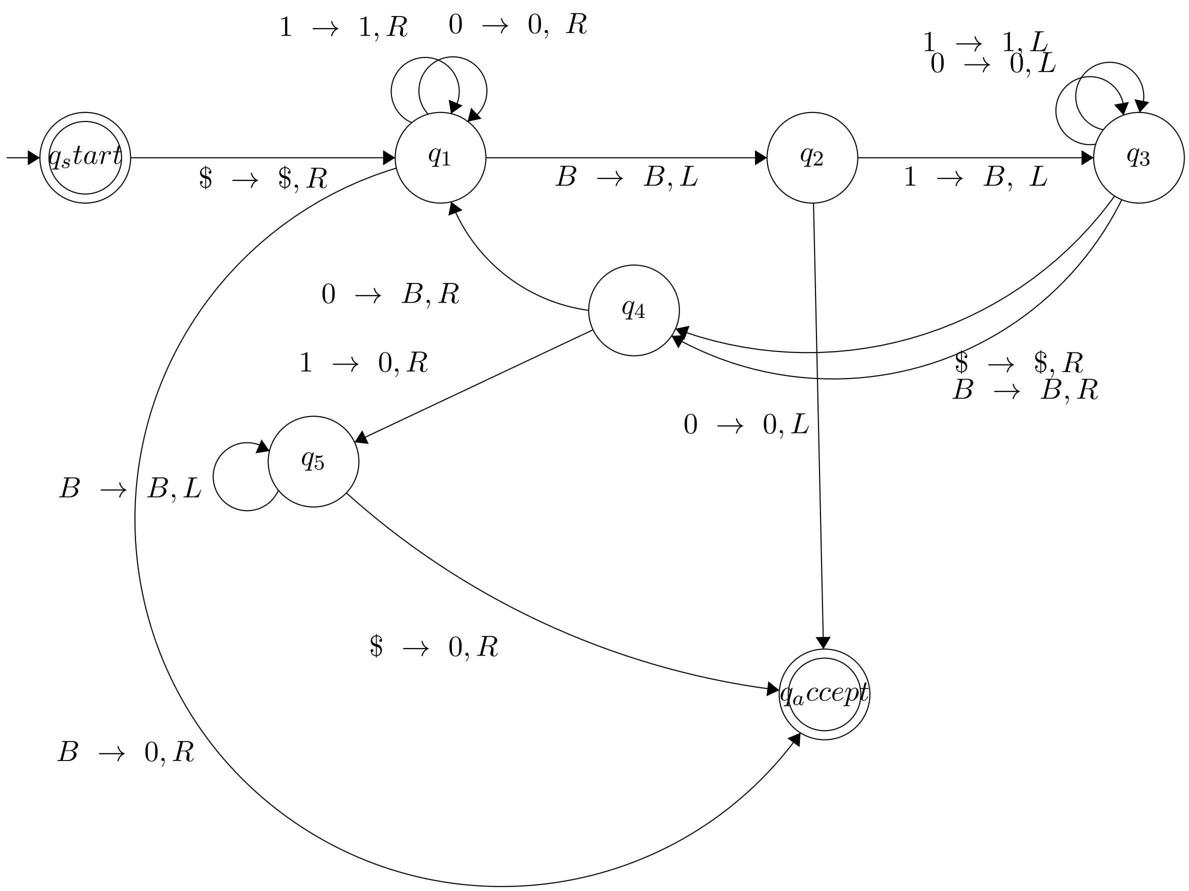
For any $x, y \in \mathbb{N}$ we define x *monus* y to be

$$x \dot{-} y = \begin{cases} x - y & \text{if } y \leq x \\ 0 & \text{if } y > x \end{cases}$$

Define a deterministic, one-head, one-tape Turing machine M using the input alphabet $\Sigma = \{0, 1, \$\}$, with a formal description and 7-tuple, which computes $f(\$0^n1^m) = \$0^{n \dot{-} m}$. In other words, if we input the string $\$0^n1^m$ for any $n, m \in \mathbb{N}$, your machine should end up in the accept state with just $\$0^{n \dot{-} m}$ written on the tape. (The \$ at the left is to make your life easier, to detect the start of the tape. We don't care what your TM does with strings that are not in the form $\$0^n1^m$.) In a few sentences, briefly explain how your Turing machine works. [*Note*: If your machine has more than 10 states, you will be penalized.]

Solution.

$M = \{\{q_{start}, q_1, q_2, q_3, q_4, q_5, q_{accept}, q_{reject}\}, \{\$, 0, 1\}, \{\$, 0, 1, B\}, \delta, q_{start}, q_{accept}, q_{reject}\}$
where B is the blank space character.
and δ is defined by the following state diagram



The TM scans over the tape crossing off one 1 for each 0, what remains is either a surplus of 1s or 0s. If there are 1s remaining they are overwritten by a single 0 and blanks. The number of 0s remaining is the computation.

PROBLEM 5 (6 points, suggested length of 1-2 paragraphs)

Let L_1 be a language. Prove that L_1 is r.e. if and only if there exists some recursive language L_2 such that $L_1 = \{x : \text{there exists } y \text{ such that } \langle x, y\rangle \in L_2\}$. (Hint: Imagine y gives you some information about an accepting computation on x, if one exists.)

(Note: Recall that the $\langle\rangle$ notation signifies that you are representing the contents within the bracket as a string, since Turing Machines take strings as input.)

Solution.

Construct a TM that recognizes A_{TM} where $\langle M, w\rangle$
Simulate M on w, if M accepts, accept, if M enters the reject state, reject
Since M is decidable, from Theorem 4.11, the A_{TM} is r.e and therefore L$_2 is r.e.$