CS320: Additional final review problems and solutions, Winter 2023

I advise you to try to solve these problems before reading their solutions. You learn more by solving on your own and learn most by trying, failing to solve and then reading the solution.

Note that problems 31–36 deal with Turing machines. They apply several different choices for the standard form of a Turing machine. All make use of a set of states, an input alphabet, a tape alphabet, a transition rule and an initial state. Some allow for accepting states and some allow the machine to attempt to move left of the start of the tape and crash (to reject an input string). In our test, every Turing machine will use the form given in the textbook with one initial, one accepting and one rejecting state. Also we will not allow the machine to try to move left of the start of the tape (instead the machine leaves the read/write head at the start of the tape). However, the analysis of the different types of Turing machines in these problems is very similar to the analysis of the Turing machines that we use. These old exam questions are useful to review any sort of Turing machine. Note also that the review problems use the term recursive where we say decidable and recursively enumearable where we say Turing recognizable.

Problem 1 Let L be the set of all strings over alphabet $\{a, b, c\}$ whose first letter occurs at least once again in the string.

Write a regular expression that defines L. If such a regular expression does not exist, prove it.

Answer:

$$\begin{gathered}\boldsymbol{a(b \cup c)^* a(a \cup b \cup c)^*} \\ \bigcup \\ \boldsymbol{b(a \cup c)^* b(a \cup b \cup c)^*} \\ \bigcup \\ \boldsymbol{c(a \cup b)^* c(a \cup b \cup c)^*}\end{gathered}$$

Problem 2 Write a regular expression that represents the set of all strings over alphabet $\{a, b, c\}$ that contain the substring ac and the substring bc. If such a regular expression does not exist, prove it.

Answer:

$$\begin{gathered}\boldsymbol{(a \cup b \cup c)^*\, ac\, (a \cup b \cup c)^*\, bc\, (a \cup b \cup c)^*} \\ \bigcup \\ \boldsymbol{(a \cup b \cup c)^*\, bc\, (a \cup b \cup c)^*\, ac\, (a \cup b \cup c)^*}\end{gathered}$$

Problem 3 Let L be the set of strings over alphabet $\{a, b, c\}$ with at most three a's.

(a) Write a regular expression that defines L. If such regular expression does not exist, prove it.

Answer:

$$\boldsymbol{(b \cup c)^*(\lambda \cup a)(b \cup c)^*(\lambda \cup a)(b \cup c)^*(\lambda \cup a)(b \cup c)^*}$$

(b) Is $\overline{L}$ (the complement of L) context-free? Explain your answer briefly.

Answer: Yes—language $\overline{L}$ is regular as the complement of a regular language; every regular language is context-free.

Problem 4 Let L be the set of strings over alphabet $\{a, b, c\}$ that have even length and contain exactly one c.

(a) Write a regular expression that defines L. If such regular expression does not exist, prove it.

Answer:

$$\begin{gathered}\boldsymbol{((a \cup b)(a \cup b))^* c(a \cup b)((a \cup b)(a \cup b))^*} \\ \bigcup \\ \boldsymbol{((a \cup b)(a \cup b))^*(a \cup b)c((a \cup b)(a \cup b))^*}\end{gathered}$$

(b) Write a regular expression that defines $\overline{L}$ (the complement of L). If such regular expression does not exist, prove it.

Answer:

$$\begin{gathered}\boldsymbol{((a \cup b \cup c)(a \cup b \cup c))^*(a \cup b \cup c)} \\ \bigcup \\ \boldsymbol{(a \cup b)^* \cup (a \cup b)^* c(a \cup b)^* c(a \cup b \cup c)^*}\end{gathered}$$

Problem 5 Let L_1 be the language defined over alphabet $\Sigma = \{a, b\}$ by the regular expression:

$$\boldsymbol{(a \cup bb)^*}$$

Let L_2 be the language generated by the context-free grammar $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b\}$, $V = \{S\}$, and the production set P is:

$$S \to aSbb \mid \lambda$$

1. Write a complete formal definition of a context-free grammar G_1 that generates language L_1. If such grammar does not exist, explain why.

   Answer: $G_1 = (V, \Sigma, P, S)$, where $\Sigma = \{a, b\}$,
   $V = \{S\}$, and the production set P is:

$$S \to SS \mid \lambda \mid a \mid bb$$

2. Write a complete formal definition of a context-free grammar G_2 that generates language L_2L_2. If such grammar does not exist, explain why.

   Answer: $G_2 = (V, \Sigma, P, T)$, where $\Sigma = \{a, b\}$,
   $V = \{S, T\}$, and the production set P is:

$$\begin{aligned} &T \to SS \\ &S \to aSbb \mid \lambda \end{aligned}$$

3. List six different strings that belong to $L_1 \setminus L_2$. If this is impossible, explain why.

   Answer:

$$a, bb, bba, aa, bbbb, bbaabbbbaaabbabb$$

4. List six different strings that belong to $L_2 \setminus L_1$. If this is impossible, explain why.

   Answer: Impossible, since $L_2 \setminus L_1 = \emptyset$. All strings in L_2 are of the form:

$$a^m(bb)^m \text{ for } m \geq 0$$

   and each can be expressed as a concatenation of strings a and bb. Language L_1 is exactly the set of all such concatenations.

5. List six different strings that belong to $L_2L_2 \setminus L_2$. If this is impossible, explain why.

   Answer:

$$\begin{aligned} &abbabb, abbaabbbb, aabbbbabb, \\ &aabbbbaabbbb, abbaaabbbbbb, aaabbbbbbabb \end{aligned}$$

Problem 6 Let L be the language generated by the context-free grammar $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c\}$, $V = \{S, A, B\}$, and P is:

$$\begin{aligned} &S \to AB \mid BA \\ &A \to ab \\ &B \to BB \mid \lambda \mid a \mid b \mid c \end{aligned}$$

(a) Write a regular expression that defines L. If such regular expression does not exist, prove it.

Answer:

$$\boldsymbol{ab(a \cup b \cup c)^* \cup (a \cup b \cup c)^* ab}$$

(b) Is $\overline{L}$ (the complement of L) context-free? Explain your answer.

Answer: Yes. The complement of a regular language is regular, so $\overline{L}$ is regular. Every regular language is context-free, so $\overline{L}$ is context-free.

Problem 7 **(a)** Let L be the language generated by the context-free grammar $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c\}$, $V = \{S, B, D, E\}$, and P is:

$$\begin{aligned} &S \to SS \mid \lambda \mid B \\ &B \to cD \mid bbD \mid aE \mid b \\ &D \to aD \mid cD \mid \lambda \\ &E \to EE \mid \lambda \mid b \end{aligned}$$

Write a regular expression that defines L. If such a regular expression does not exist, prove it.

Answer:

$$\boldsymbol{(c(a \cup c)^* \cup bb(a \cup c)^* \cup ab^* \cup b)^*}$$

(b) Let $\mathcal{R}$ be the class of languages that can be represented by a regular expression, and let $\mathcal{C}$ be the class of languages that can be represented by a context-free grammar. State the cardinalities of $\mathcal{R}$ and $\mathcal{C}$, and compare them.

Answer: Classes $\mathcal{R}$ and $\mathcal{C}$ have equal cardinalities; both of them are infinite and countable:

$$|\mathcal{R}| = |\mathcal{C}| = \aleph_0$$

Problem 8 **(a)** Let L be the set of all strings over alphabet $\{a, b\}$ that have the same symbol in the first and last positions.

Write a complete formal definition of a context-free grammar that generates L. If such grammar does not exist, prove it.

Answer: $G = \{V, \Sigma, P, S\}$, where: $\Sigma = \{a, b\}$,
$V = \{S, D\}$, and the production set P is:

$$\begin{array}{l} S \to aDa \mid bDb \mid a \mid b \mid \lambda \\ D \to DD \mid \lambda \mid a \mid b \end{array}$$

(b) Let L_1 be the set of all strings of odd length over alphabet $\{a, b\}$ that have the same symbol in the first, last, and middle positions.

Write a complete formal definition of a context-free grammar that generates L_1. If such grammar does not exist, prove it.

Answer: $G = \{V, \Sigma, P, S\}$, where: $\Sigma = \{a, b\}$,
$V = \{S, A, B, Z\}$, and the production set P is:

$$\begin{array}{l} S \to aAa \mid bBb \mid a \mid b \\ A \to ZAZ \mid a \\ B \to ZBZ \mid b \\ Z \to a \mid b \end{array}$$

(c) Let L_2 be the set of all strings of odd length over alphabet $\{a, b\}$ that have the same symbol in the first and middle positions.

Write a complete formal definition of a context-free grammar that generates L_2. If such grammar does not exist, prove it.

Answer: $G = \{V, \Sigma, P, S\}$, where: $\Sigma = \{a, b\}$,
$V = \{S, A, B, Z\}$, and the production set P is:

$$\begin{array}{l} S \to aAa \mid aAb \mid bBb \mid bBa \mid a \mid b \\ A \to ZAZ \mid a \\ B \to ZBZ \mid b \\ Z \to a \mid b \end{array}$$

Problem 9 Let L be the set of strings over alphabet $\{a, b, c\}$ in which no two adjacent symbols are equal.

(a) Write a complete formal definition of a context-free grammar that generates L. If such grammar does not exist, prove it.

Answer: $G = \{V, \Sigma, P, S\}$, where
$\Sigma = \{a, b, c\}$, $V = \{S, A, B, D\}$,
and the production set P is:

$$\begin{array}{l} S \to aA \mid bB \mid cD \mid \lambda \\ A \to bB \mid cD \mid \lambda \\ B \to aA \mid cD \mid \lambda \\ D \to aA \mid bB \mid \lambda \end{array}$$

(b) Write a complete formal definition of a context-free grammar that generates $\overline{L}$ (the complement of L). If such grammar does not exist, prove it.

Answer: $G = \{V, \Sigma, P, S\}$, where
$\Sigma = \{a, b, c\}$, $V = \{S, A, F\}$,
and the production set P is:

$$\begin{array}{l} S \to AFA \\ A \to AA \mid \lambda \mid a \mid b \mid c \\ F \to aa \mid bb \mid cc \end{array}$$

Problem 10 **(a)** Let:

$$L = \{a^i b^j c^k d^m \mid i = j \text{ and } j = 2k, \ i, j, k, m \geq 0\}$$

Write a complete formal definition of a context-free grammar that generates L. If such grammar does not exist, prove it.

Answer: L does not have a context-free grammar.
Observe that:

$$L = \{a^{2k}b^{2k}c^k d^m \mid k, m \geq 0\}$$

To prove that language L is not context-free, assume the opposite—that the Pumping Lemma holds for L.

Let ν be the constant as in the Pumping Lemma for L_2. Let $w = a^{2n}b^{2n}c^n$, such that $n > \nu$. (Note that $m = 0$ for this choice, and $d^0 = \lambda$.) Let $w = uvxyz$ be a decomposition of w, where vy is the pumping part.

Every substring of w that contains all the three letters (a, b, c) must contain the entire run of b's of length $2n$, plus some a's before this run of b's and some c's after it. Hence, every substring of w that contains all the three letters has length at least $2n + 2$. By the Lemma, it must be that $|vxy| < \nu < n < 2n + 2$. Thus, the pumping part vy is too short to contain all the three letters—it lacks at least one of them, say letter ξ, for some $\xi \in \{a, b, c\}$.

After the pumping is admitted, any word of the form uv^ixy^iz, for $i > 1$, is claimed to be in L. However, since the other two letters are pumped, while ξ is not pumped, such a word has a surplus of some of the other two letters, relative to ξ, thereby violating the pattern $a^{2k}b^{2k}c^k$—a contradiction.

(b) Is every countable language context-free? Explain your answer briefly.

Answer: No—every language is countable, but (infinitely and uncountably) many of them are not context-free.

Problem 11 **(a)** Let L_1 be a language over alphabet $\{a, b, c, d, e\}$, defined as follows:

$$L_1 = \{a^{2n}d^\ell b^m c^k d^{2m+1}e^{n+3} \mid k, n, m, \ell \geq 0\}$$

Write a complete formal definition of a context-free grammar G_1 that generates language L_1. If such grammar does not exist, explain why.

Answer: $G_1 = \{V, \Sigma, P, S\}$, where:
$\Sigma = \{a, b, c, d, e\}$, $V = \{S, A, B, D, E\}$, and P is:

$$\begin{aligned}
&S \to aaSe \mid Aeee \\
&A \to DB \\
&D \to dD \mid \lambda \\
&B \to bBdd \mid Ed \\
&E \to cE \mid \lambda
\end{aligned}$$

(b) Let L_2 be a language over alphabet $\{a, b, c, d, e\}$, consisting of those strings that have an even number of e's.

Write a complete formal definition of a context-free grammar G_2 that generates language L_2. If such grammar does not exist, explain why.

Answer: $G_2 = \{V, \Sigma, P, S\}$, where:
$\Sigma = \{a, b, c, d, e\}$, $V = \{S, Z, E\}$, and P is:

$$\begin{aligned}
&S \to Z \mid E \\
&Z \to ZZ \mid \lambda \mid a \mid b \mid c \mid d \\
&E \to EE \mid \lambda \mid ZeZeZ
\end{aligned}$$

Problem 12 Let:

$$L = \{a^i b^k c^{2i+1} d^{k+2} h^{2i} \mid i, k \geq 0\}$$

(a) Write a complete formal definition of a context-free grammar that generates L. If such grammar does not exist, prove it.

Answer: L does not have a context-free grammar. To prove that L is not context-free, assume the opposite—that the Pumping Lemma holds for the language L.

Let ν be the constant as in the Pumping Lemma for L. Let $w = a^i b^k c^{2i+1} d^{k+2} h^{2i}$, such that $i > \nu$ and $k > \nu$. Let $w = uvxyz$ be a decomposition of w, where vy is the pumping part.

Consider a non-empty, effectively "pumping" substring of the "pumping window", which is at least one of v, y. There are two cases. In the first case this "pumping" substring falls within one of the five segments containing a single letter: a^i, b^k, c^{2i+1}, d^{k+2}, h^{2i}. In this case, the "pumping" produces a surplus of occurrences of one letter, over against what should be the matching number of occurrences of another letter. To see this, observe that the number of occurrences of a, c, h is governed by i, while the number of occurrences of b, d is governed by k. In the second case, the "pumping" substring spans two of the five segments—it is too short to extend through as many as three of them. In this case, the "pumping" produces two kinds of letters that are out of sequence.

(b) Is L uncountable? Explain your answer briefly.

Answer: No—L is a subset of $\{a, b, c, d, h\}^*$, which is (infinite and) countable.

Problem 13 Let:

$$L = \{a^\ell b^j c^k d^m \mid m = 2k \text{ and } k = 2\ell,\ \ell, j, k, m \geq 0\}$$

(a) Write a complete formal definition of a context-free grammar that generates L. If such grammar does not exist, prove it.

Answer: Such grammar does not exist, because L is not context-free. Observe that:

$$L = \{a^\ell b^j c^{2\ell} d^{4\ell} \mid \ \ell, j \geq 0\}$$

To prove that L is not context-free, assume the opposite—that the Pumping Lemma holds for the language L.

Let ν be the constant as in the Pumping Lemma for L. Let:

$$w = a^m b^0 c^{2m} d^{4m} = a^m c^{2m} d^{4m}$$

where $m > \nu$. Let $w = uvxyz$ be a decomposition of w, where vy is the pumping part.

Observe that every substring of w that contains all the three letters (a, c, d) must contain the entire run of c's of length $2m$, plus some a's before this run of c's and some d's after it. Hence, every substring of w that contains all the three letters has length at least $2m + 2$. By the Lemma, it must be that $|vxy| < \nu < m < 2m + 2$. Thus, the pumping part vy is too short to contain all the three letters—it lacks at least one of them, say letter ξ, for some $\xi \in \{a, b, c\}$.

After the pumping is admitted, any word of the form uv^ixy^iz, for $i > 1$, is claimed to be in L. However, since the other two letters are pumped, while ξ is not pumped, such a word has a surplus of the other two letters, relative to ξ, thereby violating the pattern $a^m c^{2m} d^{4m}$—a contradiction.

(b) Write a regular expression that defines L. If such regular expression does not exist, prove it.

Answer: Such regular expression does not exist—L cannot be regular because it is not context-free, as is proved in the answer to part (a).

Problem 14 Let L be the set of all strings over alphabet $\{a, b, c\}$ whose length is even and two middle symbols are equal.

(a) Write a complete formal definition of a context-free grammar that generates L. If such a grammar does not exist, prove it.

Answer: $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c\}$,
$V = \{S, Z\}$, and P is:

$$S \to ZSZ \mid aa \mid bb \mid cc$$
$$Z \to a \mid b \mid c$$

(b) Draw a state-transition graph of a finite automaton M that accepts L. If such an automaton does not exist, prove it.

Answer: The required regular expression does not exist, since this language is not regular. To prove this, assume the opposite, that L is regular. Let η be the constant as in the Pumping Lemma for L. Let $m > \eta$; then the string:

$$(ac)^m bb(ac)^m$$

belongs to L, as its length is equal to $2(m + 1)$ and the two middle symbols are b.

In any "pumping" decomposition such that $(ac)^m bb(ac)^m = uvx$, we have: $|uv| \leq \eta < m$. Hence, the "pumping" substring v is contained entirely in the segment containing letters a and c. Let ℓ be the length of the "pumping" substring v. Recall that $\ell > 0$, since the "pumping" substring cannot be empty. Moreover, it has to be the case that $\ell \geq 2$, since ℓ has to be even, lest "pumping" once would produce a string of odd length, which is invalid. Hence we conclude that the "pumping" substring has one of the following two forms:

$$v = (ac)^k \text{ or } v = (ca)^k$$

for some $k = \ell/2 > 0$, whence the "pumped" part becomes:

$$v^i = (ac)^{ik} \text{ or } v^i = (ca)^{ik}$$

In both cases, letters a and c remain alternating, so that no two adjacent letters are equal in the entire segment to the left of the substring bb.

Let us "pump" once, to produce a word of the form:

$$(ac)^{m+k}bb(ac)^m \text{ where } k \geq 1$$

If $k = 1$, the two-letter substring in the middle of the word is cb; if $k > 1$, the two-letter substring in the middle of the word is either ca (if k is odd) or ac (if k is even.) In all cases, the two middle symbols are different, whence the contradiction.

Problem 15 Let L be the set of all strings over $\{a, b\}$ with twice as many a's as b's. Write a complete formal definition or a state-transition graph of a finite automaton M that accepts L. If such automaton does not exist, prove it.

Answer: Such finite automaton does not exist, since L is not a regular language. To prove this, assume the opposite. Let k be the constant as in the Pumping Lemma. Let $n > k$; then $a^{2n}b^n \in L$. In the "pumping" decomposition: $a^{2n}b^n = uvx$, we have: $|uv| \leq k < n < 2n$, hence the "pumping" substring v consists entirely of a's, say $v = a^j$. Recall that $j > 0$, since the "pumping" substring cannot be empty. By the pumping, every word of the form uv^ix, $i \geq 0$, belongs to L. However, such a word has exactly n occurrences of b and $2n + (i-1)j$ occurrences of a. Since $2n + (i-1)j > 2n$ whenever $i > 1$, this word has too many a's for its number of b's, and this is a contradiction.

Problem 16 Let L be the set of all strings over alphabet $\{a, b, c\}$ in which at least one of the letters appears at least twice.

(a) Write a complete formal definition of a context-free grammar G that generates L. If such a grammar does not exist, explain why.

Answer: $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c\}$, $V = \{S, X\}$, and the production set P is:

$$S \to XaXaX \mid XbXbX \mid XcXcX$$
$$X \to XX \mid \lambda \mid a \mid b \mid c$$

(b) Construct a state transition graph of a finite automaton that accepts L. If such an automaton does not exist, explain why.

Answer: See Figure 1.

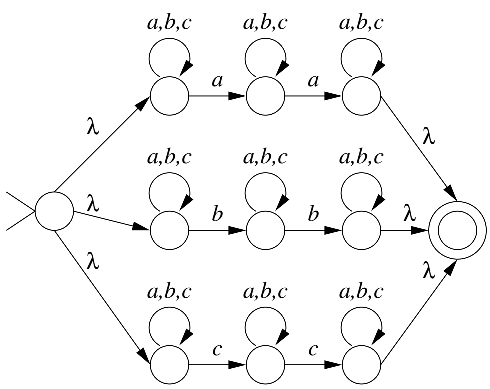

Figure 1:

Problem 17 Let:

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

and let L be the set of all strings over Σ ending with the substring $bacb$. In other words, precisely:

$$L = \{w \mid (\exists x \in \Sigma^*)(w = xbacb)\}$$

(a) Construct a state-transition graph of a finite automaton M that accepts L. If such automaton does not exist, prove it.

Answer: See Figure 2.

(b) Construct a state-transition graph of a <u>deterministic</u> finite automaton M' that accepts L. If such automaton does not exist, prove it.

Answer: See Figure 3.

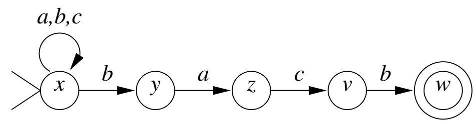

Figure 2:

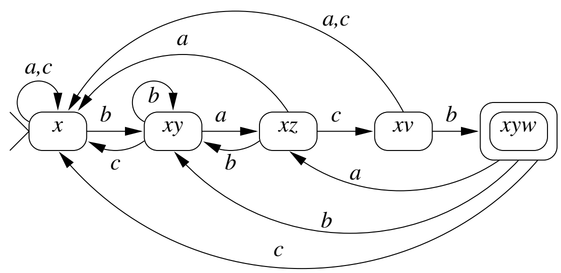

Figure 3:

Problem 18 **(a)** Let:

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

and let L_1 be the set of all strings over Σ in which every a is either immediately preceded or immediately followed by b.

Construct a state-transition graph of a finite automaton M_1 that accepts L_1. If such automaton does not exist, prove it.

Answer: See Figure 4.

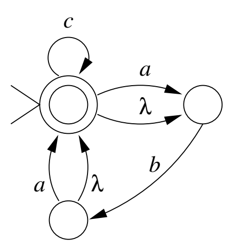

Figure 4:

(b) Let:

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

and let L_2 be the set of all strings over Σ with an even number of a's or an odd number of b's.

Write a regular expression that defines L_2. If such expression does not exist, prove it.

Answer:

$$\boldsymbol{((b \cup c)^* a (b \cup c)^* a (b \cup c)^*)^*}$$
$$\bigcup$$
$$\boldsymbol{(b \cup c)^*}$$
$$\bigcup$$
$$\boldsymbol{(a \cup c)^* b ((a \cup c)^* b (a \cup c)^* b)^* (a \cup c)^*}$$

Problem 19 Let L be the language accepted by the finite automaton $M = (Q, \Sigma, \delta, q, \{f\})$, where $\Sigma = \{a\}$, $Q = \{p, q, r, s, t, v, w, x, y, z, f\}$, and δ is given by the following table:

	a	λ
p	$\{z\}$	$\emptyset$
q	$\{t, r\}$	$\{s\}$
r	$\emptyset$	$\{q, t\}$
s	$\emptyset$	$\{w\}$
t	$\{z, y\}$	$\{p, w\}$
v	$\{x\}$	$\{r\}$
w	$\{y\}$	$\emptyset$
x	$\{p\}$	$\{v\}$
y	$\{p\}$	$\{f\}$
z	$\emptyset$	$\{v\}$
f	$\emptyset$	$\emptyset$

Compute the λ-closure of state v.

Answer:

$$\mathcal{C}(v) = \{v, r, q, t, s, p, w\}$$

Problem 20 Let M be the finite automaton represented by the state diagram on Figure 5, and let L be the language accepted by M.

Write a complete formal definition or a state-transition graph of a deterministic finite automaton M' that accepts L and show your work. If such automaton does not exist, prove it.

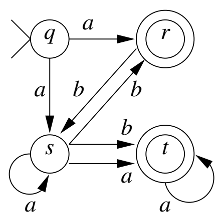

Figure 5:

Answer: There are no λ-transitions; therefore, the λ-closure of every state is the singleton containing that state. Hence, $q' = \{q\}$. Furthermore, the transition function δ of M and the input transition function t of M' are identical:

$t = \delta$	a	b
q	$\{r, s\}$	$\emptyset$
r	$\emptyset$	$\{s\}$
s	$\{s, t\}$	$\{r, t\}$
t	$\{t\}$	$\emptyset$

The transition function δ':

δ'	a	b
$\{q\}$	$\{r, s\}$	$\emptyset$
$\{r, s\}$	$\{s, t\}$	$\{r, s, t\}$
$\{s, t\}$	$\{s, t\}$	$\{r, t\}$
$\{r, s, t\}$	$\{s, t\}$	$\{r, s, t\}$
$\{r, t\}$	$\{t\}$	$\{s\}$
$\{t\}$	$\{t\}$	$\emptyset$
$\{s\}$	$\{s, t\}$	$\{r, t\}$
$\emptyset$	$\emptyset$	$\emptyset$

The set of states:
$Q' = \{\{q\}, \{r, s\}, \{s, t\}, \{r, s, t\}, \{r, t\}, \{t\}, \{s\}, \emptyset\}$.
The set of final states:
$F' = \{\{r, s\}, \{s, t\}, \{r, s, t\}, \{r, t\}, \{t\}\}$.
The state diagram of M' is given on Figure 6.

Problem 21 Let M be the finite automaton represented by the state diagram on Figure 7, and let L be the language accepted by M.

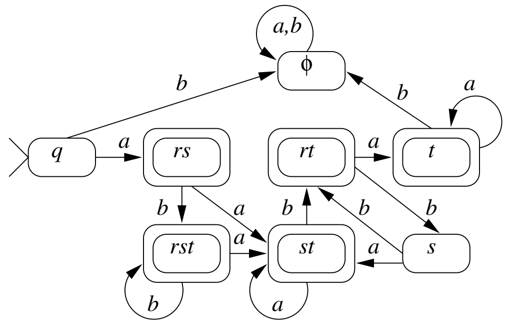

Figure 6:

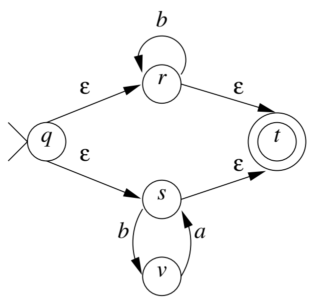

Figure 7:

(a) Is the finite automaton M deterministic? Justify briefly your answer.

Answer: No—for example, M has ϵ-transitions; there is no transition from state s on symbol a, etc.

(b) If M is not deterministic, construct a deterministic finite automaton M' that accepts L and show your work. If such an automaton M' does not exist, explain why.

Answer: Let $M' = (Q', \{a, b\}, \delta', q', F')$, where $Q' \in \mathcal{P}(Q)$.

Transition function of M:

δ	a	b	ϵ
q	$\emptyset$	$\emptyset$	$\{r, s\}$
r	$\emptyset$	$\{r\}$	$\{t\}$
s	$\emptyset$	$\{v\}$	$\{t\}$
t	$\emptyset$	$\emptyset$	$\emptyset$
v	$\{s\}$	$\emptyset$	$\emptyset$

ϵ-closure:

x	$\mathcal{C}(x)$
q	$\{q, r, s, t\}$
r	$\{r, t\}$
s	$\{s, t\}$
t	$\{t\}$
v	$\{v\}$

The initial state: $q' = \mathcal{C}(q) = \{q, r, s, t\}$.
The transition function δ':

δ'	a	b
$\{q, r, s, t\}$	$\emptyset$	$\{r, t, v\}$
$\{r, t, v\}$	$\{s, t\}$	$\{r, t\}$
$\{s, t\}$	$\emptyset$	$\{v\}$
$\{r, t\}$	$\emptyset$	$\{r, t\}$
$\{v\}$	$\{s, t\}$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$

The set of states:
$Q' = \{\{q, r, s, t\}, \{r, t, v\}, \{s, t\}, \{r, t\}, \{v\}, \emptyset\}$.
The set of final states:

$F' = \{\{q,r,s,t\},\{r,t,v\},\{s,t\},\{r,t\}\}$.
The state diagram of M' is given on Figure 8.

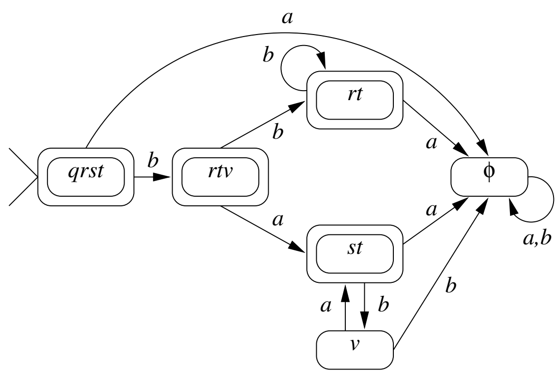

Figure 8:

Problem 22 Let M be the finite automaton represented by the state diagram given on Figure 9, and let L be the language accepted by M.

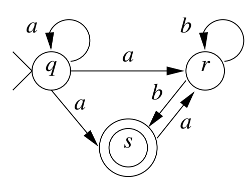

Figure 9:

Construct a deterministic finite automaton M' that accepts L and show your work. If such M' does not exist, explain why.

Answer: Let $M' = (Q', \{a,b\}, \delta', q_0', F')$, where $Q' \in \mathcal{P}(Q)$.

There are no λ-transitions; therefore, the λ-closure of every state is the singleton containing that state. Hence, $q' = \{q\}$. Furthermore, the transition function of M and the input transition function t of M' are identical:

t	a	b
q	$\{q,r,s\}$	$\emptyset$
r	$\emptyset$	$\{r,s\}$
s	$\{r\}$	$\emptyset$

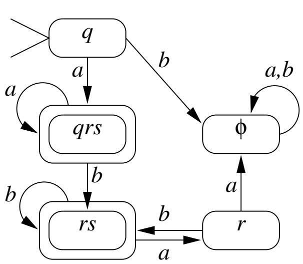

Figure 10:

The transition function δ':

δ'	a	b
$\{q\}$	$\{q,r,s\}$	$\emptyset$
$\{q,r,s\}$	$\{q,r,s\}$	$\{r,s\}$
$\{r,s\}$	$\{r\}$	$\{r,s\}$
$\{r\}$	$\emptyset$	$\{r,s\}$
$\emptyset$	$\emptyset$	$\emptyset$

The set of states:
$Q' = \{\{q\}, \{q, r, s\}, \{r\}, \{r, s\}, \emptyset\}$.
The set of final states:
$F' = \{\{q, r, s\}, \{r, s\}\}$.
The state diagram of M' is given on Figure 10.

Problem 23 Let M be the finite automaton represented by the state diagram on Figure 11, and let L be the language accepted by M.

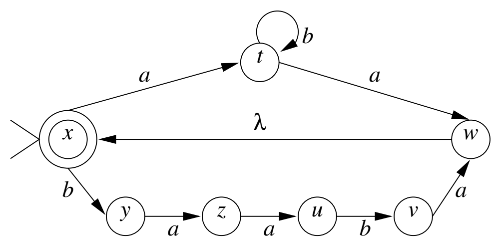

Figure 11:

Construct a regular expression that defines L and show your work. If such regular expression does not exist, prove it.

Problem 24 **(a)** Let L be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

where:

$$\begin{aligned} Q &= \{q, r\} \\ \Sigma &= \{a, b\} \\ \Gamma &= \{A\} \\ F &= \{r\} \end{aligned}$$

and the transition set δ is defined as follows:

$$\begin{aligned} &[q, a, \lambda, q, A] \\ &[q, \lambda, \lambda, r, \lambda] \\ &[r, b, A, r, \lambda] \end{aligned}$$

(Recall that M is defined so as to accept by final state and empty stack.)
Write a complete formal definition of a context-free grammar G that generates L. If such grammar does not exist, prove it.

Advice for Answer:

$$L = \{a^n b^n \mid n \geq 0\}$$

(b) Let L_1 be a language over alphabet $\{a, b\}$ defined as follows:

$$L_1 = \{a^m b^n \mid m \neq n, m \geq 0, n \geq 0\}$$

Write a complete formal definition of a context-free grammar G_1 that generates L_1. If such grammar does not exist, prove it.

Advice for Answer:

$$L_1 = L_< \cup L_>$$

where:

$$\begin{aligned} L_< &= \{a^m b^n \mid m < n, m \geq 0, n \geq 0\} \\ &= \{a^m b^{m+p} \mid m \geq 0, p > 0\} \\ L_> &= \{a^m b^n \mid m > n, m \geq 0, n \geq 0\} \\ &= \{a^{m+p} b^m \mid m \geq 0, p > 0\} \end{aligned}$$

(c) Is $\overline{L}$ recursive? Explain your answer.

Problem 25 Let L be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

where:

$$\begin{array}{l} Q = \{q, r, s, t\} \\ \Sigma = \{a, b\} \\ \Gamma = \{A\} \\ F = \{t\} \end{array}$$

and the transition function δ is defined as follows:

$$\begin{array}{l} [q, a, \lambda, r, A] \\ [r, a, A, s, \lambda] \\ [s, a, \lambda, q, A] \\ [t, b, A, t, \lambda] \\ [q, \lambda, \lambda, t, \lambda] \end{array}$$

(Recall that M is defined so as to accept by final state and empty stack.)

Write a complete formal definition of a context-free grammar that generates L. If such grammar does not exist, prove it.

Advice for Answer:

$$L = \{a^{3n}b^n \mid n \geq 0\}$$

Problem 26 Let L be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

where:

$$\begin{array}{l} Q = \{q, r, s, t\} \\ \Sigma = \{a, b, c\} \\ \Gamma = \{A, B\} \\ F = \{t\} \end{array}$$

and the transition function δ is defined as follows:

$$\begin{array}{l} [q, a, \lambda, q, A] \\ [r, b, A, r, \lambda] \\ [q, \lambda, \lambda, r, \lambda] \\ [r, \lambda, \lambda, s, \lambda] \\ [s, \lambda, \lambda, t, \lambda] \\ [s, c, \lambda, s, B] \\ [t, a, B, t, \lambda] \end{array}$$

(Recall that M is defined so as to accept by final state and empty stack.)

(a) Write a complete formal definition of a context-free grammar G that generates L. If such grammar does not exist, prove it.

Answer: M accpets the language:

$$L = \{a^m b^m c^k a^k \mid m, k \geq 0\}$$

whence the grammar: $G = \{V, \Sigma, P, S\}$, where:
$\Sigma = \{a, b, c\}$, $V = \{S, L, R\}$,
and the production set P is:

$$\begin{array}{l} S \to LR \\ L \to aLb \mid \lambda \\ R \to cRa \mid \lambda \end{array}$$

(b) Let $\mathcal{S}$ be a class of languages over alphabet $\{a, b, c\}$, defined as follows:

> Language L is a member of $\mathcal{S}$ if and only if L is accepted by some pushdown automaton, but there does not exist a context-free grammar that generates L.

What is the cardinality of $\mathcal{S}$? Explain your answer briefly.

Answer:

$$|\mathcal{S}| = 0, \text{ since } \mathcal{S} = \emptyset$$

Class $\mathcal{S}$ is empty, since every language accepted by a pushdown automaton is also generated by some context-free grammar. In fact, this context-free grammar is obtained by an algorithmic conversion of the original pushdown automaton.

Problem 27 Let L_3 be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, p, F)$$

where:

$$\begin{aligned} Q &= \{p, q, r, s\} \\ \Sigma &= \{0, 1\} \\ \Gamma &= \{A, T\} \\ F &= \{s\} \end{aligned}$$

and the transition function δ is defined as follows:

$$\begin{array}{l} [p, \lambda, \lambda, q, T] \\ [q, 1, T, s, \lambda] \\ [q, 1, \lambda, r, AAA] \\ [r, 0, A, r, \lambda] \\ [r, \lambda, T, q, T] \end{array}$$

(Recall that M is defined so as to accept by final state and empty stack.)

(a) Write a complete formal definition of a context-free grammar G that generates L_3. If such grammar does not exist, prove it.

Answer: M accepts **(1000)*1**, whence the grammar: $G = \{V, \Sigma, P, S\}$, where $\Sigma = \{0, 1\}$, $V = \{S, B\}$, and the production set P is:

$$\begin{array}{l} S \to B1 \\ B \to BB \mid \lambda \mid 1000 \end{array}$$

(b) Write a complete formal definition of a context-free grammar G that generates L_3^*. If such grammar does not exist, prove it.

Answer: $G = \{V, \Sigma, P, A\}$, where $\Sigma = \{0, 1\}$, $V = \{A, S, B\}$, and the production set P is:

$$\begin{array}{l} A \to AA \mid \lambda \mid S \\ S \to B1 \\ B \to BB \mid \lambda \mid 1000 \end{array}$$

(c) Is $\overline{L_3}$ (the complement of L_3) a context-free language? Prove your answer.

Answer: Yes—the regular expression for L_3 is given in the answer to part (a); the complement of every regular language is regular, and thereby context-free.

Problem 28 Let L be the language accepted by the pushdown automaton: $M = (Q, \Sigma, \Gamma, \delta, q, F)$, where $Q = \{q, r\}$, $\Sigma = \{a, b, c, d\}$, $\Gamma = \{A, B\}$, $F = \{r\}$, and the transition function δ is defined as follows:

$$\begin{array}{l} [q, a, \lambda, q, A] \\ [q, b, \lambda, q, B] \\ [q, c, \lambda, q, A] \\ [q, \lambda, \lambda, r, \lambda] \\ [r, d, A, r, \lambda] \\ [r, b, B, r, \lambda] \end{array}$$

(Recall that M is defined so as to accept by final state and empty stack.)

(a) Write a complete formal definition of a context-free grammar G that generates L. If such a grammar does not exist, prove it.

Answer: $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c, d\}$,
$V = \{S\}$, and P is:

$$S \to aSd \mid cSd \mid bSb \mid \lambda$$

(b) Write a complete formal definition of a context-free grammar G_1 that generates L^*. If such a grammar does not exist, prove it.

Answer: $G = (V, \Sigma, P, T)$, where $\Sigma = \{a, b, c, d\}$,
$V = \{S, T\}$, and P is:

$$T \to TT \mid \lambda \mid S$$
$$S \to aSd \mid cSd \mid bSb \mid \lambda$$

Problem 29 Let L be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

where:

$$Q = \{q, r\}$$
$$\Sigma = \{a, b, c\}$$
$$\Gamma = \{A, B, D, Z\}$$
$$F = \{q\}$$

and the transition function δ is defined as follows:

$$[q, \lambda, \lambda, r, ZABDA]$$
$$[r, a, A, r, \lambda]$$
$$[r, b, B, r, \lambda]$$
$$[r, c, D, r, \lambda]$$
$$[r, \lambda, Z, q, \lambda]$$

(Recall that M is defined so as to accept by final state and empty stack. Furthermore, if an arbitrary stack string, say $X_1 \ldots X_n \in \Gamma^*$, where $n \geq 2$, is pushed on the stack by an individual transition, then the leftmost symbol X_1 is pushed first, while the rightmost symbol X_n is pushed last.)

Write a regular expression that represents L. If such a regular expression does not exist, prove it.

Answer:

$$\boldsymbol{(acba)^*}$$

Problem 30 Let L be the language accepted by the pushdown automaton:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

where:

$$Q = \{q, r\}$$
$$\Sigma = \{a, b, c\}$$
$$\Gamma = \{A, B, D, Z\}$$
$$F = \{q\}$$

and the transition function δ is defined as follows:

$$[q, \lambda, \lambda, r, ZAB]$$
$$[q, \lambda, \lambda, r, ZDA]$$
$$[r, a, A, r, \lambda]$$
$$[r, b, B, r, \lambda]$$
$$[r, c, D, r, \lambda]$$
$$[r, \lambda, Z, q, \lambda]$$

(Recall that M is defined so as to accept by final state and empty stack. Furthermore, if an arbitrary stack string, say $X_1 \ldots X_n \in \Gamma^*$, where $n \geq 2$, is pushed on the stack by an individual transition, then the leftmost symbol X_1 is pushed first, while the rightmost symbol X_n is pushed last.)

Write a complete formal definition of a **regular** context-free grammar that generates L. If such a grammar does not exist, prove it.

Answer: The language L is represented by the regular expression:

$$\boldsymbol{(ba \cup ac)^*}$$

which corresponds to the finite automaton given on Figure 12, which in turn is converted to the regular grammar $G = (V, \Sigma, P, S)$, where $\Sigma = \{a, b, c\}$ is the set of terminals; $V = \{S, A, B\}$ is the set of variables; S is the start symbol, and the production set P is:

$$S \to bA \mid aB \mid \lambda$$
$$A \to aS$$
$$B \to cS$$

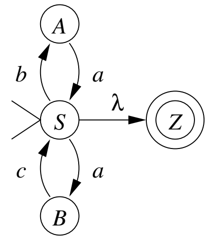

Figure 12:

Problem 31 Consider a Turing machine:

$$M = (Q, \Sigma, \Gamma, \delta, q_0)$$

such that:

$$Q = \{q_0, q_1, q_2\}$$
$$\Sigma = \{a, b, c\}$$
$$\Gamma = \{a, b, c, B\}$$

and δ is defined by the following transition set:

$$[q_0, a, q_1, a, R]$$
$$[q_0, b, q_0, b, R]$$
$$[q_0, c, q_0, c, R]$$
$$[q_0, B, q_0, B, R]$$
$$[q_1, a, q_2, a, R]$$
$$[q_1, b, q_1, b, R]$$
$$[q_1, c, q_1, c, R]$$
$$[q_1, B, q_1, B, R]$$

(where B is the designated blank symbol.)

(a) Let L be the set of those strings over Σ on which the Turing machine M halts. Draw a state transition graph of a finite automaton M_1 that accepts L. If such finite automaton M_1 does not exist, prove it.

Advice for Answer: The regular expression for L:

$$\boldsymbol{(b \cup c)^* \, a \, (b \cup c)^* \, a \, (a \cup b \cup c)^*}$$

(b) Is $\overline{L}$ (the complement of L) a recursive language? Explain your answer briefly.

Answer: Yes—L is regular, which is demonstrated by the regular expression constructed in part (a); $\overline{L}$ is regular because L is regular and the class of regular languages is closed under complement; every regular language is recursive as the decision procedure consists of a simulation of the finite automaton that accepts the language.

Problem 32 Consider the Turing machine:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

such that:
$Q = \{q, r, s, t, x, y, z\}$;
$\Sigma = \{a, b\}$;
$\Gamma = \{B, a, b\}$;

$F = \{t\}$;
and δ is defined by the following transition set:

$$\begin{array}{l} [q,a,r,a,R] \\ [q,b,s,b,R] \\ [r,a,r,a,R] \\ [r,b,r,b,R] \\ [r,B,x,B,L] \\ [s,a,s,a,R] \\ [s,b,s,b,R] \\ [s,B,y,B,L] \\ [x,a,t,a,R] \\ [x,b,z,B,R] \\ [y,b,t,b,R] \\ [y,a,z,B,R] \\ [z,B,z,B,R] \end{array}$$

(where B is the designated blank symbol.) Let L be the set of strings over Σ on which M does not halt.

(a) Write a regular expression that defines L. If such regular expression does not exist, prove it.

Answer:

$$\boldsymbol{a(a \cup b)^* b \ \cup \ b(a \cup b)^* a}$$

(b) Is L recursive? Explain your answer briefly.

Answer: Yes. L is regular, as witnessed by the regular expression constructed in the answer to part (a). Every regular language is recursive.

Problem 33 Write a complete formal definition of a Turing machine:

$$M = (Q, \Sigma, \Gamma, \delta, q_0)$$

over input alphabet $\{0,1\}$, such that M halts on every input, after making exactly 5 moves. If such machine does not exist, explain why.

Answer: $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$; $\Sigma = \{0,1\}$; $\Gamma = \{B,0,1\}$; and δ is defined by the following transition set:
$[q_0,0,q_1,0,R]$, $[q_0,1,q_1,1,R]$, $[q_0,B,q_1,B,R]$,
$[q_1,0,q_2,0,R]$, $[q_1,1,q_2,1,R]$, $[q_1,B,q_2,B,R]$,
$[q_2,0,q_3,0,R]$, $[q_2,1,q_3,1,R]$, $[q_2,B,q_3,B,R]$,
$[q_3,0,q_4,0,R]$, $[q_3,1,q_4,1,R]$, $[q_3,B,q_4,B,R]$,
$[q_4,0,q_5,0,R]$, $[q_4,1,q_5,1,R]$, $[q_4,B,q_5,B,R]$.

Problem 34 Consider the Turing machine:

$$M = (Q, \Sigma, \Gamma, \delta, p, F)$$

such that: $Q = \{p, q, s, t, e\}$; $\Sigma = \{a, b, c\}$;
$\Gamma = \{B, X, Y, Z, a, b, c\}$; $F = \{t\}$.
and δ is defined by the following transition set:

$$\begin{array}{l} [p,a,q,X,R] \\ [p,b,q,Y,R] \\ [p,c,q,Z,R] \end{array}$$

$$\begin{array}{l} [q,a,q,X,R] \\ [q,b,q,Y,R] \\ [q,c,q,Z,R] \\ [q,B,s,B,L] \end{array}$$

$$\begin{array}{l} [s,X,t,B,R] \\ [s,Y,s,B,R] \\ [s,Z,e,B,R] \end{array}$$

$$[e,B,e,B,R]$$

(where B is the designated blank symbol.)

Let L_1 be the set of strings *rejected* by M, and let L_2 be the set of strings on which M *diverges.*

(a) Write a regular expression that defines L_1. If such a regular expression does not exist, prove it.

Answer:

$$(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c})^* \boldsymbol{b} \ \cup \ \boldsymbol{\lambda}$$

(b) Write a regular expression that defines L_2. If such a regular expression does not exist, prove it.

Answer:

$$(\boldsymbol{a} \cup \boldsymbol{b} \cup \boldsymbol{c})^* \boldsymbol{c}$$

(c) Which (if any) of the two languages: L_1 and L_2 are recursive (i.e., decidable)? Explain your answer.

Answer: Both L_1 and L_2 are recursive. By the answer to part (a), they are both regular, and all regular languages are recursive.

Problem 35 Consider a Turing machine:

$$M = (Q, \Sigma, \Gamma, \delta, q)$$

such that:

$$Q = \{q_0, q_1\}$$
$$\Sigma = \{a, b, c\}$$
$$\Gamma = \{a, b, c, B\}$$

and δ is defined by the following transition set:

$$[q_0, a, q_0, a, R]$$
$$[q_0, b, q_0, b, R]$$
$$[q_0, c, q_1, c, R]$$
$$[q_0, B, q_0, B, R]$$

(where B is the designated blank symbol.)

(a) Write a complete formal definition of a Turing machine M_1 such that M_1 accepts η if M halts on η, and M_1 rejects η if M does not halt on η, for all $\eta \in \Sigma^*$. In short:

$$(M(\eta) \searrow) \ \rightarrow \ (M_1(\eta) \searrow \text{ and accepts })$$

and also:

$$(M(\eta) \nearrow) \ \rightarrow \ (M_1(\eta) \searrow \text{ and rejects })$$

If such Turing machine M_1 does not exist, prove it.

Answer: Observe that M halts on those input strings that contain c.

$$M_1 = (Q, \Sigma, \Gamma, \delta_1, q, F)$$

such that:

$$Q = \{q_0, q_1\}$$
$$\Sigma = \{a, b, c\}, \ \Gamma = \{a, b, c, B\}$$
$$F = \{q_1\}$$

and δ_1 is defined by the following transition set:

$$[q_0, a, q_0, a, R]$$
$$[q_0, b, q_0, b, R]$$
$$[q_0, c, q_1, c, R]$$

(where B is the designated blank symbol.)

(b) Is the language accepted by M recursive? Explain your answer.

(c) Is the language accepted by M recursively enumerable? Explain your answer.

Problem 36 Consider the Turing machine:

$$M = (Q, \Sigma, \Gamma, \delta, q, F)$$

such that:
$Q = \{q, r, s, t, v\}$;
$\Sigma = \{a, b\}$;
$\Gamma = \{B, a, b, \Psi\}$;
$F = \{t\}$;
and δ is defined by the following transition set:

$$[q, a, q, a, R]$$
$$[q, b, q, \Psi, R]$$
$$[q, B, r, B, L]$$

$$[r, a, r, a, L]$$
$$[r, \Psi, s, \Psi, L]$$

$$[s, a, s, a, L]$$
$$[s, \Psi, t, \Psi, L]$$

$$[t, a, t, a, L]$$
$$[t, \Psi, v, \Psi, R]$$

$$[v, a, v, a, R]$$
$$[v, \Psi, v, \Psi, R]$$
$$[v, B, v, B, R]$$

(where B is the designated blank symbol.)
M accepts by final state.

(a) Write a regular expression that defines the set of strings on which M diverges. If such regular expression does not exist, prove it.

Answer:

$$\boldsymbol{a^* b a^* b a^* b (a \cup b)^*}$$

(b) Write a regular expression that defines the set of strings on which M halts and accepts. If such regular expression does not exist, prove it.

Answer:

$$\emptyset$$

(c) Write a regular expression that defines the set of strings on which M halts and rejects. If such regular expression does not exist, prove it.

Answer:

$$\emptyset$$

(d) Write a regular expression that defines the set of strings on which M terminates abnormally (attempts to move the head to the left of the leftmost cell.) If such regular expression does not exist, prove it.

Answer:

$$\boldsymbol{a^* (b \cup \lambda) a^* (b \cup \lambda) a^*}$$

Problem 37 Let L be the language of strings over alphabet $\{a, b\}$ that contain at least three occurrences of letter a.

(a) Write a regular expression that defines L. If such regular expression does not exist, prove it.

(b) Describe a Turing machine M that decides the following problem:

INPUT: A representation of a Turing machine M.
QUESTION: Is $L(M) = L$?

If such Turing machine does not exist, prove it.

Advice for Answer: The property:

is defined by the regular expression:

$$\boldsymbol{b^*ab^*ab^*a(a \cup b)^*}$$

is non-trivial.

Problem 38 Let $\Sigma = \{a, b\}$. Construct a recursively enumerable language L over Σ that is not recursive, such that its complement $\overline{L}$ is recursively enumerable but not recursive. Explain your answer. If such a language does not exist, prove it.

Problem 39 Let L_1 be a recursively enumerable language which is not recursive; and let L_2 be a recursive language.

(a) Is $L_1 \setminus L_2$ a recursive language?
If your answer is "yes", prove it by describing an appropriate Turing machine. If your answer is "no", prove it by showing that such a Turing machine does not exist.

Answer: No. Otherwise, we could set:

$$L_1 = L_H$$
$$L_2 = \emptyset$$

where:

$$L_H = \{(M, w) \mid (M, w) \searrow\}$$

yielding:

$$L_1 \setminus L_2 = L_H$$

and claim that L_H is recursive, which is a contradiction.

(b) Is $L_1 \setminus L_2$ a recursively enumerable language?
If your answer is "yes", prove it by describing an appropriate Turing machine. If your answer is "no", prove it by showing that such a Turing machine does not exist.

Answer: Yes. There exists a machine M_1 that accepts L_1 and a machine M_2 that decides L_2. To accept $L_1 \setminus L_2$, simulate M_2 until it halts. If M_2 accepts, then reject. If M_2 rejects, then simulate M_1 and accept if and when M_1 accepts.

(c) Is $L_2 \setminus L_1$ a recursive language?
If your answer is "yes", prove it by describing an appropriate Turing machine. If your answer is "no", prove it by showing that such a Turing machine does not exist.

Answer: No. Otherwise, we could set:

$$L_2 = \Sigma^*$$
$$L_1 = L_H$$

and claim that:

$$\Sigma^* \setminus L_H = \overline{L_H}$$

is recursive. This is false, since $\overline{L_H}$ is not even recursively enumerable.

(d) Is $L_2 \setminus L_1$ a recursively enumerable language?
If your answer is "yes", prove it by describing an appropriate Turing machine. If your answer is "no", prove it by showing that such a Turing machine does not exist.

Answer: No—by the answer given in part (c).

Problem 40 Let L_1 be a recursively enumerable language, and let L_2 be a recursive language. Describe a Turing machine M that accepts $L_1 \setminus L_2$. If such M does not exist, explain why.