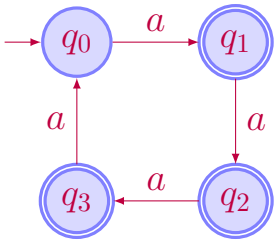
CPSC 313 (Winter 2014) L01

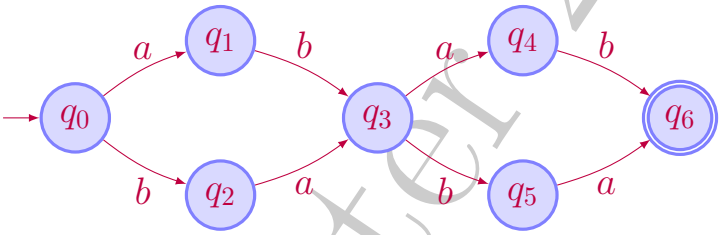
Final Sketch Solutions

1. Short answers

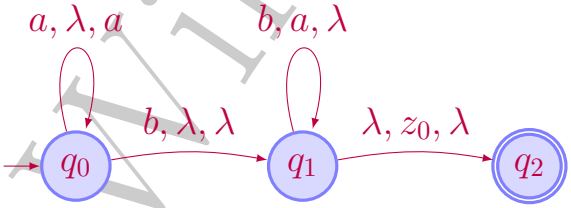
1. Give an NFA accepting the regular language $L = \{a^i \mid i \text{ is } \textbf{not} \text{ divisible by } 4\}$. Briefly justify.



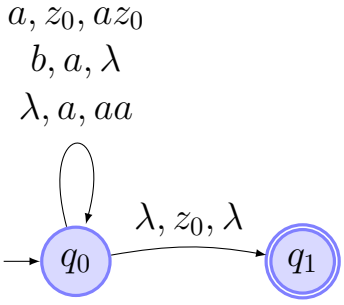
2. Give an NFA for the language $L = L(\boldsymbol{r}_1) \cap L(\boldsymbol{r}_2)$ where $\boldsymbol{r}_1 = \boldsymbol{b}^*\boldsymbol{a}\boldsymbol{b}^*\boldsymbol{a}\boldsymbol{b}^*$ and $\boldsymbol{r}_2 = (\boldsymbol{a}\boldsymbol{b}+\boldsymbol{b}\boldsymbol{a})^*$. Briefly justify.



3. Give a PDA for the language $L = \{a^i b^j \mid i \geq 0, j = i + 1\}$. Briefly justify.



4. Give a regular expression for the language accepted by the following PDA. Briefly justify.



$\boldsymbol{r} = (\boldsymbol{a}\boldsymbol{b}^+)^*$

5. Give a context-free grammar for the language $L = \{a^i b^j c^k \mid i + j = k \text{ and } i, j, k \geq 0\}$. Briefly justify.

$S \to aSc \mid T; \quad T \to bTc \mid \lambda$

6. The following pseudo-code prints a string w over the alphabet $\{a, b, c, d, e\}$. Express w as a regular expression, using the variables k, m, and n. Briefly justify.

```
main(k,m,n);
for i ← 1 to k do
    print a;
    for j ← 1 to m do
        print b;
        for ℓ ← 1 to n do
            print c;
        end
        print d;
    end
    print e;
end
```

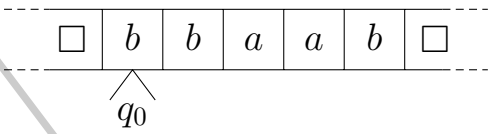
$\boldsymbol{r = (a(bc^n d)^m e)^k}$

7. Given a Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$, we construct a modified Turing Machine M' by moving the tape head of M' by **two** cells positions whenever the tape head of M moves one cell position. That is, we replace each left-going rule $\delta(q_1, a) = (q_2, b, L)$ by $\delta'(q_1, a) = (q_2, b, LL)$, and each right-going rule $\delta(q_1, a) = (q_2, b, R)$ by $\delta'(q_1, a) = (q_2, b, RR)$. Rules that stay are kept un-changed, and everything else is also the same.

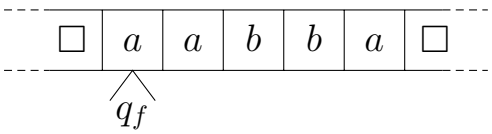
   If M accepts the language $L(M)$, then what language does M' accept? Justify your answer.

   $L(M') = \{w \in \Sigma^* \mid \text{ODD}(w) \in L(M)\}$ where $\text{ODD}(w) = w_1 w_3 w_5 \cdots w_{n'}$ is every other symbol of w, and where n' is n for odd n, and $n - 1$ for even n.

8. We want to construct a Turing Machine that replaces a's by b's, and b's by a's. E.g. if the input is $w = bbaab$, the Turing Machine starts in the initial configuration



   and it ends in the following final configuration



   That is, the Turing Machine ends in state q_f with its head pointing at position 1 and with all a's and b's interchanged.

   Give the transition function δ of a Turing Machine that given any input string $w \in \{a, b\}^*$ interchanges the symbols a and b in w. You do not have to use all states given below.

	a	b	$\square$
q_0	(q_0, b, R)	(q_0, a, R)	$(q_1, \square, L)$
q_1	(q_1, a, L)	(q_1, b, L)	$(q_f, \square, R)$
q_f	$--$	$--$	$--$

9. We are given $\langle M, w\rangle$, the description a Turing Machine M and an input string $w \in \Sigma^*$.

   Let $L = L(M)$ be the language accepted by M. We now define a new Turing Machine M' that does as follows on an input $z \in \Sigma^*$:

   (a) First M' checks if $z = w$.

   (b) If $z = w$, then M' halts and rejects.

   (c) If $z \neq w$, then M' simulates M on $\boldsymbol{w}$.

   (d) If the simulation halts and accepts, then M' outputs "accept."

   (e) If the simulation halts and rejects, then M' outputs "reject."

   Characterize the language $L(M')$ accepted by M'. Justify your answer.

   If $w \in L(M)$, then $L(M') = \Sigma^* \setminus \{w\}$
   If $w \notin L(M)$, then $L(M') = \emptyset$

2. Language operations

Answer True or False to each of the following four statements. If you answer "True," **no** justification is required. If you answer "False," give the correct answer and give a brief justification. The alphabet is $\{a, b, c\}$ in questions 1 and 3, and it is $\{a, b, c, d\}$ in questions 2 and 4.

1. If $L_1 = \{a^i b^i \mid i \geq 0\}$ and $L_2 = \{a^i b^i c^i \mid i \geq 0\}$, then $L_1 \cap L_2 = \emptyset$.

   False. $L_1 \cap L_2 = \{\lambda\}$

2. If $L_1 = \{a^i b^i \mid i \geq 0\}$ and $L_2 = \{c^i d^i \mid i \geq 0\}$, then $L_1 L_2 = \{a^i b^i c^i d^i \mid i \geq 0\}$.

   False. $L_1 L_2 = \{a^i b^i c^j d^j \mid i, j \geq 0\}$

3. If $L = \{a^i b^i c^i \mid i \geq 0\}$ then $\overline{L} = \{a^i b^j c^k \mid (i \neq j) \text{ or } (i \neq j) \text{ or } (j \neq k)\}$.

   False. $\overline{L} = L_{\text{order}} \cup L_{\text{count}}$ where
   $L_{\text{order}} = \{w \in \{a, b, c\}^* \mid w \text{ contains } ba \text{ or } ca \text{ or } cb \text{ as a substring}\}$ and
   $L_{\text{count}} = \{a^i b^j c^k \mid (i \neq j) \text{ or } (j \neq k)\}$.

4. If $L_1 = \{a^i b^i c^i \mid i \geq 0\}$ and $L_2 = \{a^i b^i c^i d^i \mid i \geq 0\}$, then $L_1 \setminus L_2 = \emptyset$.

   False. $L_1 \setminus L_2 = \{a^i b^i c^i \mid i \geq 1\}$

3. Reversible grammars

We say a rule is **reversible** if it is on the form

$$A \to x \qquad \text{with} \qquad x = x^R,$$

where the right hand side $x \in (V \cup T)^*$ reads the same when reversed. For instance the rules $A \to ABBA$ and $B \to pop$ are reversible, while $C \to aBc$ and $D \to aA$ are not. A context-free grammar $G = (V, T, S, P)$ is **reversible** if all its rules are reversible.

1. Give a reversible context-free grammar that generates $\text{PAL} = \{w \in \{a, b\}^* \mid w = w^R\}$. Justify your answer.

   $S \to aSa \mid bSb \mid a \mid b \mid \lambda$

2. Give a context-free language L that can **not** be generated by a reversible grammar. Justify your answer.

   $L = \{ab\}$

3. Give a reversible context-free grammar for $L = \{a, b\}^*$. Justify your answer.

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

4. True or False

Answer True or False. No justification required. The languages in questions 3 and 4 are over the alphabet $\Sigma = \{a, b\}$. In questions 1 and 8, $\boldsymbol{r}$ and $\boldsymbol{a}$ denote a regular expression.

	Question	True	False
1	If $\boldsymbol{r} = \boldsymbol{r}^*$ then $\boldsymbol{rr} = \boldsymbol{r}$	✓	
2	All non-recursive languages are infinite	✓	
3	$L = \{a^i b^j \mid i \neq j\}$ is context-free	✓	
4	The language $L = \{\}$ is co-recursive enumerable	✓	
5	The grammar $S \to abSc \mid abS \mid Sc \mid \lambda$ generates a regular language	✓	
6	It is easy to decide if a finite automata (FA) accepts a finite language or not	✓	
7	$\{\}^* = \{\}$		✓
8	$(\boldsymbol{aa} + \boldsymbol{aaa})^* + \boldsymbol{a} = \boldsymbol{a}^*$	✓	

5. Language classification

Answer **exactly one** of four possible answers REC, RE, co-RE, None to each of the following languages. Answer REC if the language is recursive, answer RE if the language is recursive enumerable but not recursive, answer co-RE if the language is co-recursive enumerable but not recursive, and answer None if the language is neither of the first three possible answers. No justification required.

	Question	REC	RE	co-RE	None
1	$\{\langle M_1, M_2, w\rangle \mid \text{either TM } M_1 \text{ halts on } w \text{ or TM } M_2 \text{ halts on } w\}$				✓
2	$\{\langle M, w\rangle \mid \text{TM } M \text{ loops on } w\}$			✓	
3	$\{\langle M\rangle \mid \text{whenever } M \text{ halts, it halts after an even number of steps}\}$			✓	
4	$\{w \mid w = \langle M\rangle \text{ for some TM } M\}$	✓			

6. Recursive enumerable

Define the language

$$\mathsf{Double} = \{\langle M, x\rangle \mid x \in L(M) \textbf{ and } \lambda \in L(M)\}$$

1. Show that the language Double is recursive enumerable.

   Ask $\langle M, x\rangle \in$ Halt and $\langle M, \lambda\rangle \in$ Halt. Accept if both accept, and reject if either reject. Since Halt $\in$ RE and RE is closed under intersection, Double $\in$ RE.

2. Show that Halt $\leq$ Double.

   Given an input $\langle M, w\rangle$ to Halt, we reduce to an input $\langle M', w'\rangle$ to Double. We set $w' = w$. Our definition of the Turing Machine M' depends on whether w is λ or not.

   (a) If $w = \lambda$, we set $M' = M$.

   (b) If $w \neq \lambda$, we let M' be a Turing Machine that accepts if its input is λ, and that simulates M if its input is different from λ. That is, M' does as M on all inputs, except it has been modified to always accept the input λ.

   Lastly, we make all states in M' final so halting implies accepting.

   As for the proof of correcness, in the first case that $w = \lambda$, if M halts on λ (and hence also halts on w), M' also halts on λ (and hence w') and hence accepts λ (and hence also w').

   In the second case that $w \neq \lambda$, M' always accepts λ, and it accepts w' if and only if M halts on w.

   We have shown that $\langle M, w\rangle \in$ Halt if and only if $\langle M', w'\rangle \in$ Double.