CPSC 313 (Winter 2014) L01

University of Calgary
Faculty of Science
Final Examination

April 29, 2014.
Time available: 180 minutes.
No books or calculators are permitted (two lettersized sheets of notes are permitted).

Write answers in this booklet only.
Do **not** open this exam until you are told to do so.

Name: ______________________________________

There are 6 (six) problems in total. There are 39 points and 3 bonus points in total. Full answer is **38** points.

For your convenience, the last page (page 18) contains a collection of information.

You may use the blank pages at the end of the exam if you need more space for your answers. Please indicate clearly when your answers are continued on these pages.

Please note that points may be deducted for untidy handwriting.

Problem	Possible score	Score
1	20	
2	4	
3	4	
4	4	
5	4	
6	3 + 3 bonus	
Total	39 + 3 bonus	

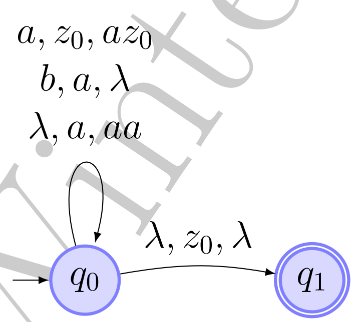
1. Short answers (20 points)

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 $\boldsymbol{r}_1 = \boldsymbol{b}^*\boldsymbol{ab}^*\boldsymbol{ab}^*$ and $\boldsymbol{r}_2 = (\boldsymbol{ab}+\boldsymbol{ba})^*$. 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.



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.

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
```

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.

8. [**3** points] 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

$\square$	b	b	a	a	b	$\square$
	q_0					

and it ends in the following final configuration

$\square$	a	a	b	b	a	$\square$
	q_f					

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_1			
q_2			
q_3			
q_4			
q_5			
q_6			
q_f			

9. [**3** points] 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.

2. Language operations (4 points)

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 $L_1 \cap L_2 = \emptyset$.

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

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

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 $L_1 \setminus L_2 = \emptyset$.

3. Reversible grammars (4 points)

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 $\textsc{Pal} = \{w \in \{a, b\}^* \mid w = w^R\}$. Justify your answer.

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

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

4. True or False (4 points)

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 (4 points)

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	$\left\{\langle M_1, M_2, w\rangle \mid \begin{array}{l}\text{either TM } M_1 \text{ halts on } w \\ \text{or TM } M_2 \text{ halts on } w\end{array}\right\}$				
2	$\{\langle M, w\rangle \mid \text{TM } M \text{ loops on } w\}$				
3	$\left\{\langle M\rangle \mid \begin{array}{l}\text{whenever } M \text{ halts, it halts after an} \\ \text{even number of steps}\end{array}\right\}$				
4	$\{w \mid w = \langle M\rangle \text{ for some TM } M\}$				

6. Recursive enumerable (3 points + 3 bonus points)

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.

2. Show that Halt $\leq$ Double.

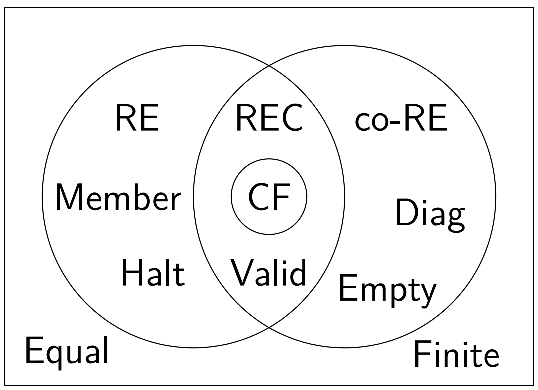
(This page is intentionally left blank)

(This page is intentionally left blank)

	$x \in L$	$x \notin L$	
$L \in \mathsf{REC}$	halts & accepts	halts & rejects	Recursive
$L \in \mathsf{RE}$	halts & accepts	—	Recursive enumerable
$L \in \textsf{co-RE}$	—	halts & accepts	

$L \in \mathsf{RE} \Leftrightarrow \overline{L} \in \textsf{co-RE}$
$L \in \mathsf{REC} \Leftrightarrow L \in \mathsf{RE}$ and $\overline{L} \in \mathsf{RE}$
$L \in \mathsf{RE} \Leftrightarrow \exists$ unrestricted grammar

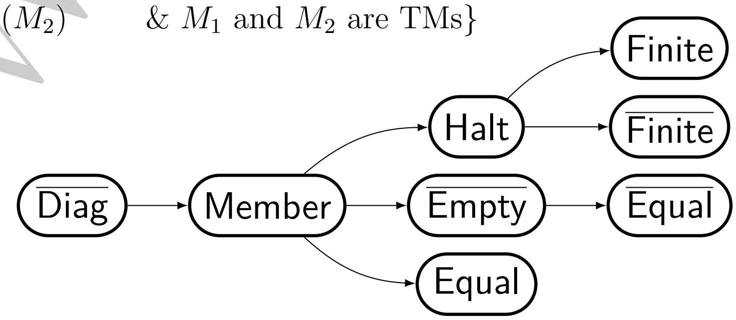
L Recursive $=$ L Decidable $=$ χ_L Computable
L Not recursive $=$ L Undecidable $=$ χ_L Noncomputable



	REG	CF	REC	RE
$L_1 \cup L_2$	✓	✓	✓	✓
$L_1 \cap L_2$	✓	(1)	✓	✓
$\overline{L}$	✓	(2)	✓	(4)
$L_1 \setminus L_2$	✓	(3)	✓	(5)
$L_1 L_2$	✓	✓	✓	✓
L^*	✓	✓	✓	✓
L^R	✓	✓	✓	✓
prefix(L)	✓	✓	✓	✓
$h(L)$	✓	✓	(6)	✓
λ-free $h(L)$	✓	✓	✓	✓

Diag $= \{\langle M \rangle \mid M \notin L(M)$ & M is a TM $\}$
Member $= \{\langle M, w \rangle \mid w \in L(M)$ & M is a TM $\}$
Halt $= \{\langle M, w \rangle \mid M$ halts on input w & M is a TM $\}$
Empty $= \{\langle M \rangle \mid L(M) = \emptyset$ & M is a TM $\}$
Finite $= \{\langle M \rangle \mid L(M)$ is finite & M is a TM $\}$
Equal $= \{\langle M_1, M_2 \rangle \mid L(M_1) = L(M_2)$ & M_1 and M_2 are TMs$\}$

$\overline{\text{Diag}} \leq_m$ Member
Member $\leq_m \overline{\text{Empty}}$
Halt $\leq_m \overline{\text{Finite}}$
Halt $\leq_m$ Finite
Member $\leq_m$ Equal
Member $\leq_m$ Halt
$\overline{\text{Empty}} \leq_m \overline{\text{Equal}}$



Language $A \subseteq \Sigma_A^*$ **reduces to** language $B \subseteq \Sigma_B^*$ if there is a computable function $f : \Sigma_A^* \to \Sigma_B^*$ such that $w \in A \Leftrightarrow f(w) \in B$ for all strings $w \in \Sigma_A^*$.

If A reduces to B, denoted $A \leq_m B$, then problem A is at most as difficult as problem B.
$(A \leq_m B$ and $A \notin \mathsf{RE}) \Rightarrow B \notin \mathsf{RE}$ (If problem A is hard, so is B.)
$(A \leq_m B$ and $B \in \mathsf{RE}) \Rightarrow A \in \mathsf{RE}$ (If problem B is easy, so is A.)
$(A \leq_m B) \Leftrightarrow (\overline{A} \leq_m \overline{B})$ Transitivity: $\big(A \leq_m B$ and $B \leq_m C\big) \Rightarrow A \leq_m C$

******** PH/PH ********