1 Finite Automata

A1: For the following, build a deterministic FA. The alphabet is \( \{0, 1\} \). The empty string is not in the language. If the string starts with a 0, then the string has length at least 3. If the string starts with a 1, then the last bit must be a 0.

A2: Consider the following language \( L \) with alphabet \( \{0, 1, 2\} \).
Every string in \( L \) starts with a 2, ends with a 1, and contains an even number of 0’s. For example, both 201100201 and 211 are in the language.
Give both a DFA and an RE for \( L \).

A3: Let \( M \) be the language described as follows. The alphabet is \( \{0, 1\} \).
The empty string is in \( M \). If a string starts with a 0, then it ends with a 1. If a string starts with a 1, then it does not contain 00 as a substring.
Give both a DFA and an RE for \( M \).

A4: Describe in succinct English what languages the following two FAs accept:

A5: Give an RE for the set of all binary strings that do not contain 101 as a substring.
A6: Consider the following FA.

(a) Which 3-bit strings does it accept?
(b) Describe in English the language accepted by the machine.

A7: Consider the following DFA.

(a) List all 4-bit strings accepted by the machine.
(b) Give an RE for this language of the machine.
(c) Describe in English the language of this machine. (Hint: consider the strings as binary.)

A8: Let $J$ be the language given by the RE $(a + ba)^*$. 
(a) List all strings of length 4 in $J$.
(b) Give an NFA that accepts $J$. 
2 Regular Languages

B1: (a) For the leftmost NFA below, use the subset construction to give a DFA that is equivalent to it.

(b) For the rightmost NFA below, use the GFA approach to construct an RE that is equivalent to it.

![NFA Diagram]

B2: Suppose language $L$ is accepted by FA $M$. Define $L^E$ as the set of all strings in $L$ that have even length. Explain how to construct an FA that accepts $L^E$.

B3: Apply the subset construction to produce a DFA that accepts the same language as the following NFA.

![NFA Diagram]

B4: (a) Given language $L$, the language $L^F$ is defined as the set of all final segments of strings in $L$. For example, if $L = \{\text{DO, GOOD}\}$, then $L^F = \{\text{DO, 0, \varepsilon, GOOD, OOD, OD, D}\}$. Provide an algorithm that converts an FA for language $L$ to one for language $L^F$.

(b) For a language $L$, let $L^o$ be the set of all strings obtained from all strings of $L$ by omitting one symbol from the string. For example, if $L$ contains dog, then $L^o$ contains og, dg, and do. Show how to build an FA for $L^o$ given an FA for $L$.

(c) Given a string $z$, the sampling of $z$ is the string consisting of every alternate character in $z$, starting with the first. For example, the sampling of Goddard is Gdadr. Describe an algorithm that converts a DFA for language $L$ to an FA for the set of all samplings of strings in $L$.

B5: (a) Give an example of nonregular languages $L_1$ and $L_2$ such that $L_1 \cup L_2$ is regular.

(b) Give an example of a regular language such that any DFA accepting it requires at least 100 states.
B6: Explain how to tell whether a language is $\Sigma^*$ or not:
(a) Given a DFA for the language.
(b) Given an RE for the language.

B7: Give a biggest possible set of **pairwise distinguishable** strings with respect to:
(a) the set of all binary strings containing 110 as a substring.
(b) the set of all binary strings not containing 000 as a substring.
(c) the set of all binary strings starting with 011.

B8: Use the **Pumping Lemma** to show that the following languages is not regular:
$\{0^n1^n0^n : n \geq 10 \}$.

B9: Consider the language $B = \{0^n1^n : n \geq 0 \}$ and let $C$ be a subset of $B$. Use the Pumping Lemma to show that if $C$ is infinite, then $C$ is not regular.

B10: Let $L$ be the language corresponding to the RE $(0+1)^*(00+11)(0+1)^*$. Find a set of maximum size of strings that are pairwise distinguishable with respect to $L$. 
3 Grammars and PDAs

C1: Let \( L \) be the language \( \{0^i1^j : i < j \} \).

(a) Give a CFG for \( L \)

(b) Give the diagram for a PDA for \( L \)

C2: Let \( D \) be the following language \( \{0^a1^b2^c : b = a + c \} \). For example, 0001112 is in the language. Give a context-free grammar for \( D \).

C3: Consider the following CFG:

\[
S \rightarrow SSS \mid bS \mid a
\]

(a) Give a derivation tree for the string ababa.

(b) Explain in English what language this grammar generates:

C4: Consider the following language \( L \) with alphabet \( \{0, 1\} \).

The empty string is not in \( L \). If the string is nonempty and has even length, then it is in \( L \). If the string has odd length, then it is in \( L \) if and only if it is a palindrome.

(a) Give a CFG for \( L \).

(b) Give a PDA for \( L \).

C5: Give a grammar for the set of all binary strings with equal 0s and 1s.

C6: Explain in English what language the following PDA accepts.
C7: Consider the following PDA.

(a) Give the shortest string accepted by this PDA.
(b) Give two more strings accepted by this PDA.
(c) Describe in succinct English the language of this PDA.

C8: Consider the following PDA with alphabet \{0, 1\}.

(a) List one binary string of length 4 that this PDA accepts.
(b) List one binary string of length 4 that this PDA does not accept.
(c) Give a precise short English summary description of the language of this PDA.
4 Context-Free Languages

D1: Give an example of:
   (a) a language accepted by a PDA but not by any deterministic PDA.
   (b) a language accepted by a deterministic PDA but not by any NFA.
   (c) a non-context-free language whose complement is context-free.

D2: (a) Define a regular grammar.
   (b) Give a regular grammar for the language of the following FA.

D3: For the following grammar:

\[
\begin{align*}
S & \rightarrow AS \mid SB \mid a \\
A & \rightarrow BA \\
B & \rightarrow \varepsilon
\end{align*}
\]

(a) Does this grammar generate the empty string?
(b) Is this grammar in Chomsky Normal Form?
(c) Is the language of this grammar finite?

D4: Give one example of each of the following, or state that it does not exist.
   (a) A context-free language $A$ such that $A^*$ is context-free.
   (b) A context-free language $B$ such that $B^*$ is not context-free.
   (c) A non-context-free language $C$ such that $C^*$ is context-free.
   (d) A non-context-free language $D$ such that $D^*$ is not context-free.
   (e) Languages $E_1$ and $E_2$ such that $E_1$ and $E_2$ are context-free but $E_1 \cap E_2$ is not.
   (f) Languages $F_1$ and $F_2$ such that $F_1$ and $F_2$ are not context-free but $F_1 \cap F_2$ is.

D5: (a) Give an unambiguous regular grammar for the set of all binary strings ending in 00.
   (b) Give an example of a regular grammar that is ambiguous.

D6: For a string $w$ from the alphabet \{a, b, c\}, the splonk of $w$ is obtained by replacing every a by 0, every b by 1, and every c by 01. For example, if $w$ is abbc, then the splonk of $w$ is 01101. For a language $L$ with alphabet \{a, b, c\}, let $L^s$ be the set of splonks of all strings in $L$.

Show, by means of an algorithm, that if $L$ is context-free then so is $L^s$. 
D7: Let $B$ denote the set of all binary palindromes and $E$ denote the set of all strings with equal numbers of 0s and 1s. Use the Pumping Lemma to show that the intersection $B \cap E$ is not context-free.

D8: Explain TWO things wrong with the following “proof” that the language $L$ of $0^n1^n$ is not context-free.

Assume $L$ is not context-free.
Consider the string $z = 0^{999}1^{999}$.
Then $z$ is in $L$.
Split $z$ as $z = uvwxy$.
Assume $v = 01$ and $x = \varepsilon$.
Then $uv^2wx^2y$ is not in $L$.
This is a contradiction of the Pumping Lemma, and so $L$ is not context-free.

D9: Show that the following language is not context-free: $\{www : w \in \{a,b\}^* \}$
5 Turing Machines

E1: Draw a TM for the language \( \{0^m1^n : n \leq m \leq 2n \} \).

E2: Draw a TM for the function \( f(x) = 3x \) on binary input.
For example, if the input is 101, then the output is 1111.

E3: Consider the following TM. What language does it accept?

E4: Draw a TM that accepts the set of binary strings that have length a multiple of 3 and have equal numbers of 0's and 1's.

E5: Consider the following TM.

(a) Give one string of length 4 that it accepts.
(b) Give one string of length 4 that it rejects.
(c) What is the language of this TM?
E6: Consider the following nondeterministic TM.

(a) Give one string of length 3 the machine accepts.
(b) Give one string of length 3 the machine rejects.
(c) Give a succinct English-ish description of the language of the TM.
6 Decidability and Undecidability

F1: A wombat is a deterministic model of computation. A wombat is like a TM in that it has finite memory and an infinite tape with a single head, on which the head can move both ways. However, the tape is read-only. Show that the acceptance problem for wombats is decidable.

F2: Given an NFA $M$, one can create a new NFA $M^*$ by interchanging the accept and reject states. (That is, changing every accept state to a reject state and every reject state to an accept state.) Define an NFA $M$ as happy if $M$ and $M^*$ accept the same language. Show that it is decidable to determine if an NFA is happy.

F3: It is known that the problem of determining if a PDA accepts every string is undecidable.
(a) Explain why this means that it is undecidable to determine if two PDAs accept the same language.
(b) Let $N = \{ \langle G \rangle : G$ is CFG that does NOT generate all strings$\}$. Show that $N$ is r.e.
(c) Is the complement of $N$ r.e.?

F4: State whether each of the following is true or false. In each case give a brief justification.
(a) The following is recursive: the set of all encodings of TMs that do not accept their own encoding.
(b) The following is recursive: the set of all encodings of FAs that do not accept their own encoding.
(c) The set of all C++ programs is countable.
(d) The set of all infinite subsets of the reals is uncountable

F5: Consider any r.e. language $L$. Show that $L$ is recursive if and only if $L \leq \overline{L}$ (that is, it reduces to its complement).

F6: Give an example of:
(a) a language that is r.e. but not recursive
(b) a language that is recursive but not r.e.
(c) an infinite countable set
(d) an uncountable set
(e) a language accepted by an LBA that does not have a context-free grammar

F7: A 2FA is like a deterministic FA except that it has two heads and these heads can move in both directions on the read-only tape (but not off the tape). Show that the Halting Problem for a 2FA is decidable.
1 Finite Automata

A1: 

A2: \(2(0(1+2)^*0+1+2)^*1\)

A3: \(\varepsilon + 0(0+1)^*1 + 1(1+01)^*(0+\varepsilon)\)

A4: (a) All binary strings ending in 10
(b) All strings with an even number of each of 0’s, 1’s and 2’s

A5: \(0^*(11^*000^*)^*(1^* + 11^*0)\)
A6: (a) 000, 010, 011, 001
    (b) All binary strings with a 0 in the third-to-last position

A7: (a) 0000, 0011, 0110, 1001, 1100, 1111
    (b) \((0 + 1(01^*0)^*1)^*\)
    (c) The empty string and all strings when interpreted as binary numbers that are a multiple of 3.

A8: (a) aaaa, baaa, abaa, aaba, baba
2 Regular Languages

B1: (a) 

(b) Several answers including: \((b^*a + bab^*a)^*b\)

B2: Use the product construction with the two-state DFA accepting all string of even length. If \(M\) is the FA for \(L\), then the FA for \(L_E\) is as follows. For every state \(q\) in \(M\), create two states labeled \(q^e\) and \(q^o\). The start state is \(q^e_0\) where \(q_0\) was the start state of \(M\); the accept state are all states \(q^e\) where \(q\) is an accept state of \(M\). And for each transition \(q_i \rightarrow q_j\) of \(M\) add two transitions \(q_i^e \rightarrow q_j^o\) and \(q_i^o \rightarrow q_j^e\).

B3: 

B4: (a) Add a new start state with \(\varepsilon\)-transitions to all other states.

(b) Add a separate copy of the automaton, say \(L'\). Then for every non-\(\varepsilon\)-transition in \(L\), say \(x \rightarrow y\) on \(c\), add an \(\varepsilon\)-transition from \(x\) to the copy of \(y\) in \(L'\). Finally, make all states in \(L\) reject.

(c) For every state, say \(x\), create two copies, say \(x_1\) and \(x_2\). For every original transition \(x \rightarrow y\) on \(c\), add a transition from \(x_1\) to \(y_2\) on \(c\) and a transition from \(x_2\) to \(y_1\) on \(\varepsilon\).

B5: (a) Let \(L_1\) be any nonregular language and let \(L_2\) be its complement.

(b) \(\{0^{100}\}\)

B6: (a) A DFA accepts every string if and only if every state is an accept state.

(b) Convert the RE to a DFA (via an NFA) and then use the above test.
B7: (a) \{\varepsilon, 1, 11, 110\}
(b) \{\varepsilon, 0, 00, 000\}
(c) \{\varepsilon, 0, 01, 011, 1\}

B8: Suppose the language, call it $L$, were regular. Let $k$ be the number of states of a DFA for $L$. Consider the string $z = 0^k1^k0^k$—this is in $L$. Split $z = uvw$ according to the Pumping Lemma. Then, because $|uv| \leq k$, it follows that $v$ is always a string of 0’s. Thus, $uv^2w \notin L$, a contradiction of the Pumping Lemma.

B9: Let $C$ be an infinite subset of $B$ and suppose that $C$ is regular. Then $C$ is accepted by some DFA with $k$ states. Since $C$ is infinite, there exists some $\ell \geq k$ such that $z = 0^\ell 1^\ell$ is in $C$. By the Pumping Lemma one can write $z = uvw$ such that $v$ is nonempty and $uv^2w$ is in $C$. But $uv^2w$ either has the wrong format or does not have equal numbers of 0s and 1s, a contradiction. Therefore $C$ is not regular.

B10: \{\varepsilon, 0, 1, 00\}
3 Grammars and PDAs

C1: (a) \( S \rightarrow 0S1 \mid S1 \mid 1 \)

(b)

C2: \( S \rightarrow XY \)
\( X \rightarrow 0X1 \mid \varepsilon \)
\( Y \rightarrow 1Y2 \mid \varepsilon \)

C3: (a)

(b) All strings of a’s and b’s ending in a and having an odd number of a’s.

C4: (a)

\( S \rightarrow X \mid Y \)
\( X \rightarrow 00X \mid 01X \mid 10X \mid 11X \mid 00 \mid 01 \mid 10 \mid 11 \)
\( Y \rightarrow 0Y0 \mid 1Y1 \mid 1 \mid 0 \)

(b)
C5: $S \rightarrow 0S1 \mid 1S0 \mid SS \mid \varepsilon$

C6: Balanced brackets/parentheses where both left brackets are considered the same and both right brackets are considered the same.

C7: (a) 01010  
(b) 0110110 or any string starting 01010  
(c) All strings that contain two equal-length blocks of 1s surrounded by 0s.

C8: (a) 0000  
(b) 0011  
(c) Binary strings of even length with first and last bit the same.
4 Context-Free Languages

D1: (a) palindromes without middle marker
   (b) $0^n1^n$
   (c) $0^n1^n2^n$

D2: (b) 
   
   $A \rightarrow 0C \mid 1B$
   $B \rightarrow 0D \mid 1A \mid 0$
   $C \rightarrow 0A \mid 1D \mid 1$
   $D \rightarrow 0B \mid 1C$

D3: (a) No
   (b) No
   (c) Yes: the language contains just one string: it is \{a\}.

D4: (a) Any context-free language e.g. $\Sigma^*$
   (b) Does Not Exist
   (c) For example, $C = \{0, 1, 2\} \cup \{0^n1^n2^n : n \geq 0\}$: here $C^* = \Sigma^*$
   (d) $0^n1^n2^n$
   (e) $0^n1^n2^m$ and $0^n1^m2^m$
   (f) Take any two non context-free languages whose alphabets are disjoint.

D5: (a) 
   $S \rightarrow 0S \mid 1S \mid 0T$
   $T \rightarrow 0$

   (b) 
   $S \rightarrow 0S \mid 0T \mid 0$
   $T \rightarrow 0S$

D6: Take the CFG for $L$. Then in every production, replace $a$ by 0, $b$ by 1, and $c$ by 01.

D7: Assume $B \cap E$ is context-free. Let $k$ be the constant of the Pumping Lemma. Let $z = 0^k1^k0^k$. Note $z \in L$.
   Consider split $z = uwx$. Since $|vwx| \leq k$, if $v$ or $x$ contains 0’s, then these 0’s are from only one block of 0’s, and so $z^{(2)} = uv^2wx^2y$ is not a palindrome. On the other hand, if $v$ and $x$ contain only 1’s, then $z^{(2)} = uv^2wx^2y$ has more 1’s than 0’s.
   A contradiction.

D8: 1) We should assume the language IS context-free.
   2) We did not guarantee $z$ is long enough for the Pumping Lemma to apply.
   3) We did not consider all possibilities for $v$. 
D9: Let $L$ stand for the language. Suppose that $L$ is context-free. Let $k$ be the constant of the pumping lemma. Set $z = a^k b^k a^k b^k a^k b^k$. Say one writes $z = uvwxy$. The pumping lemma claims that $z^{(i)} \in L$ for all $i$; but this does not hold for $z^{(0)}$. For, since $|vwx| \leq k$, at least one block of $a$’s is undisturbed as one block of $b$’s. So the result when split in three does not have three identical pieces. This is a contradiction. Therefore, $L$ is not context-free.
5 Turing Machines

E1:

E2:

E3: The set of all binary strings where the blocks of the same character all have the same length, starting with a 0 and finishing with a 1.

E4: The simplest idea is to first check that the length is a multiple of 3. Then feed it into a machine that checks for equal 0’s and 1’s.
E5: (a) aaaa  
     (b) bbbb  
     (c) all strings of a’s and b’s such that every block of b’s has odd length.

E6: (a) 101  
     (b) 111  
     (c) all binary strings with a block of 0’s of odd length surrounded by 1’s.
6 Decidability and Undecidability

F1: Let $q$ be the number of states of the TM. If the head stays on the input or within $q$ cells of the input, then there is a finite number of configurations and so if it computes too long we can stop it. If the head goes more than $q$ cells away from the input, then I claim the TM is stuck in an infinite loop: for, the last $q + 1$ steps must have had a repeated state, and every time the TM was reading a blank. So then too we can stop it.

F2: Take $M$, create $M_S$, test for equivalence of NFAs.

F3: (a) Suppose we can decide if two PDAs accept the same language. Then on input PDA $M$, build a simple PDA $N$ that accepts all strings of that alphabet. Then test whether $M$ and $N$ accept the same strings. This answers the question of whether $M$ accepts all strings. Which contradicts what is known.

(b) Take $N$ and convert to Chomsky Normal Form. Create all strings in increasing order of length, checking each to see if generated by $N$. Stop and accept if find a string not generated by $N$.

(c) No.

F4: (a) False. ($S_{tm}$ is not even r.e.)

(b) True. (Since FAs are guaranteed to halt, we can simulate them.)

(c) True. (Since a program can be represented by a finite string.)

(d) True.

F5: Assume that $L$ is recursive (but neither empty nor $\Sigma^*$). Then $L$ is accepted by some TM $M$ that always halts. Let $x$ be any string in $L$, let $y$ be any string in $\bar{L}$. Then the reduction is: “On input $w$, run $M$ on $w$. If $M$ accepts $w$ then output $y$, else output $x$.”

Assume that $L$ is the language of TM $M$ and that $L$ reduces to $\bar{L}$ by reduction $f$. Then the decider for $L$ is: “On input $w$, calculate $f(w)$. Run $M$ on both $w$ and $f(w)$ in parallel. If $M$ accepts $w$ then accept; if $M$ accepts $f(w)$ then reject.” This works, since if $w \in L$, then $M$ will halt and accept $w$; and if $w \notin L$, then $f(w) \in L$, so $M$ will halt and accept $f(w)$.

F6: (a) Acceptance problem for TM.

(b) Does not exist.

(c) The integers

(d) The real numbers

(e) $0^n1^n2^n$

F7: Since the input is read-only, the current configuration of the 2FA can be given by specifying the state and the positions of the two heads. Suppose the input has length $n$ and the number of states is $q$. Then there are at most $qn^2$ configurations. In particular, if the machine runs for more than $qn^2$ steps, then some configuration must have recurred, and so the deterministic machine is stuck in an infinite loop. So we can decide if the machine will halt or not by running it for this long.