CSE 103	Computational Models	
Fall 2021	Prof. Fremont	Practice 1

Midterm Practice Problems

This set of problems tries to cover most of the material we've seen so far, so it is large. Feel free to jump around and find (sub)problems which you think will be most helpful. Each subproblem is a reasonable question to have on the midterm.

1. **Notation for sets and languages**
   Let $\Sigma = \{0, 1\}$. For each of the following statements, decide whether it is true or false.

   (a) $\Sigma^* - \{x_1 \dots x_n \mid n \geq 1, x_i \in \Sigma\} = \emptyset$

   (b) $\{x \in \Sigma^* \mid \exists k \geq 0.\ |x| = 2k\} = \{00, 01, 10, 11\}^*$

   (c) $|\Sigma^*| = |\{S \subseteq \Sigma\}|$

   (d) $|\Sigma^3 \times \Sigma^5| < |\Sigma^8|$

   (e) $\{xyz \mid x, y, z \in \Sigma \cup \{\varepsilon\}\} = \Sigma^{\leq 3}$

   (f) $|\mathscr{P}(X)| > |X|$ for all finite sets X.

2. **Working with a DFA (including converting to a regular expression)**
   Consider the DFA $M = (Q, \Sigma, \delta, q_0, F)$ where:

   - $Q = \{0, 1, 2, 3\}$
   - $\Sigma = \{1, 2\}$
   - $\delta(q, s) = \begin{cases} 1 & q = 0 \text{ and } s = 1 \\ 0 & s > q \\ 1 & q = 3 \text{ and } s = 1 \\ q - s & \text{otherwise} \end{cases}$
   - $q_0 = 3$
   - $F = \{0, 2, 3\}$

   (a) Draw M as a graph.

   (b) Which of the following strings does M accept: $\varepsilon, 1, 112, 111, 2221, 1112$? Give an accepting path for each accepted string.

   (c) What is $\hat{\delta}(q_0, x)$ for each of the strings x from part (b)? What about $\hat{\delta}(2, x)$?

   (d) What is the language $L(M)$ of M? (You may give your answer as a regular expression.)

   (e) Convert M into an equivalent regular expression, showing the sequence of GNFAs you get along the way. Confirm that the expression you get agrees with your answer to part (d).

3. **Designing DFAs**
   Draw DFAs recognizing each of the following languages:

(a) All strings over $\Sigma = \{a,b,c\}$ which contain at least 2 bs, and where the second b is immediately followed by a c.

(b) All strings of correctly-nested parentheses whose nesting depth never exceeds 2 (so for example $(()())()$ is fine but $(()(()))$ is not).

(c) All binary strings which contain an even number of 0s or an even number of 1s (or both).

4. Algorithms for DFAs

Give short descriptions of algorithms for each of the following problems. In each case, try to make your algorithm as fast as possible, and state its asymptotic runtime in terms of the size of its input (for a DFA D, measure its size $|D|$ as the total number of transitions, i.e. the number of states times the size of the alphabet).

(*Note:* You should not need to look up any algorithms; all of these problems can be solved using variations of graph algorithms and/or algorithms we've discussed in class or on the homework.)

(a) Decide whether there is a word $w \in L(D)$ containing the symbol s.

(b) Given a DFA D, find the longest word in $L(D)$ (if there is a tie, return any one of the longest words) or determine that $L(D)$ contains arbitrarily long words.

(c) Given a DFA D and a length ℓ, decide if $L(D)$ contains any word of length ℓ. (You should express the runtime of your algorithm in terms of ℓ as well as $|D|$.)

5. Working with an NFA

Consider the NFA $M = (Q, \Sigma, \delta, q_0, F)$ where:

- $Q = \{0,1,2,3\}$
- $\Sigma = \{1,2\}$
- $\delta(q,s) = \begin{cases} \{0\} & q = 3 \text{ and } s = \varepsilon \\ \{q' \in Q \mid q' - q \leq s\} & s \in \Sigma \\ \emptyset & \text{otherwise} \end{cases}$
- $q_0 = 0$
- $F = \{3\}$

(a) Draw M as a graph.

(b) What are the ε-closures $E(\{0\})$, $E(\{3\})$, $E(\{0,3\})$, and $E(\{1,2,3\})$?

(c) Which of the following strings does M accept: $\varepsilon, 1, 21, 11, 111, 1112$? Give an accepting path for each accepted string.

(d) What is $\hat{\delta}(q_0, x)$ for each of the strings x from part (b)? What about $\hat{\delta}(3, x)$?

(e) What is the language $L(M)$ of M?

(f) Use the subset construction to build a DFA equivalent to M (only drawing the states reachable from the start state). Confirm that it has the same language you found in part (e).

6. Designing NFAs

Draw NFAs recognizing each of the following languages:

(a) Binary strings where the last 1 is followed by at least two 0s.

(b) Binary strings containing either a pair of 0s with two symbols in between, or a pair of 1s with two symbols in between (e.g. 0100 and 1110011 are fine, but 101 and 0001110 are not).

(c) Strings over $\Sigma = \{a,b,c\}$ containing a pair of *a*s where the number of *b*s in between is a multiple of 3 (including zero). For example, *caca*, *ababcbba*, and *abababa* are fine (in the last case the pair of *a*s is the first and last *a*), but *a*, *aba*, and *abbbbaba* are not.

7. Closure properties of regular languages

For each of the following statements, give a short argument for why it is true or false.

(a) For any infinite sequence of regular languages $L_1, L_2, \ldots$ (i.e. we have a regular language L_i for each $i \geq 0$), the language $L = \cup_{i \geq 0} L_i$ is regular.

(b) If L is a regular language, then so is any subset $R \subseteq L$.

(c) If X, Y, and Z are regular languages over an alphabet Σ, then so is $\{w \in \Sigma^* \mid w \text{ is in exactly two of } X, Y, \text{ and } Z\}$.

(d) If L is a regular language, so is $\{w \in L \mid w \text{ contains every symbol of } \Sigma\}$.

8. More algorithms for finite automata

Give short descriptions of algorithms for each of the following problems. In each case, try to make your algorithm as fast as possible, and state its asymptotic runtime in terms of the size of its input (for an automaton M, measure its size $|M|$ as the total number of transitions, i.e. the number of states times the size of the alphabet).

(*Note:* You should not need to look up any algorithms; all of these problems can be solved using variations of graph algorithms and/or algorithms we've discussed in class or on the homework.)

(a) Given DFAs A and B, decide if there is any $w \in \Sigma^*$ such that both $w \in L(A)$ and $w \in L(B)$.

(b) Given DFAs A and B, decide whether $L(A) \subseteq L(B)$.

(c) Given a DFA D, decide whether there is any $w \in L(D)$ containing every symbol of Σ.

(d) Given a DFA D, decide whether $L(D)$ contains any word of even length.

(e) Given an NFA N and a word $w \in \Sigma^*$, decide whether *every* possible path for w in N ends in an accepting state.

9. Reading regular expressions

For each of the following pairs of regular expressions over $\Sigma = \{a,b,c\}$, state whether their languages are equal (=), one is a proper subset of the other ($\subset$), or if they are incomparable. If $L(R_1) \subset L(R_2)$, give an example of a string in $L(R_2)$ that is not in $L(R_1)$, or vice versa if $L(R_2) \subset L(R_1)$; if the languages are incomparable, give an example string from each language that is not in the other.

(a) $R_1 = (a^*b^*)^*$
$R_2 = (a^*b^*a^*)^*$

(b) $R_1 = aa^*$
$R_2 = \emptyset^*$

(c) $R_1 = (a \mid b \mid c \mid \varepsilon)^*$
$R_2 = (a \mid b \mid c \mid \emptyset)^*$

(d) $R_1 = (a \mid b)(b \mid c)$
$R_2 = \Sigma$

(e) $R_1 = (a \mid b \mid \varepsilon)(a \mid b \mid \varepsilon)$
$R_2 = aa \mid ab \mid ba \mid bb$

10. Writing regular expressions

Write regular expressions for each of the following languages over $\Sigma = \{a,b,c\}$:

(a) Strings that begin with two of the same symbol (e.g. *aabc*).

(b) Strings containing two as which have the substring cc somewhere in between (e.g. *aabccac* is fine but *acbca* is not).

(c) Strings where the number of bs is a multiple of 3 (including zero).

11. Converting regular expressions to NFAs

Convert the following regular expressions over the alphabet $\Sigma = \{a, b, c\}$ to NFAs:

(a) $(a^* b^*)^*$

(b) $(a \mid bc)a^*$

(c) $((a \mid \varepsilon)(b \mid \varepsilon)c)^*$

12. Non-regular languages

Prove that the following languages are not regular. If you like, you may refer to other languages shown to be non-regular in class or on the homework.

(a) $\{0^n 1^m 0^n \mid n, m \geq 0\}$

(b) All binary strings which are not palindromes (i.e., which are not equal to their reversal).

(c) All binary strings with an unequal number of 0s and 1s.

13. Working with a CFG (including parsing using the CYK algorithm)

Consider the CFG G with terminal symbols $\Sigma = \{a, b, c\}$, nonterminal symbols $V = \{S, T\}$, start symbol S, and the rules

$$
\begin{aligned}
S &\to aSbS \mid \varepsilon \mid T \\
T &\to c \mid \varepsilon
\end{aligned}
$$

(a) For each of the following strings, state whether it can be derived from G, giving a leftmost derivation if so: *ab*, *aab*, *aacbcb*, *acbab*.

(b) Is G ambiguous? If so, draw two different parse trees for some string in $L(G)$; otherwise argue why every string has a unique leftmost derivation.

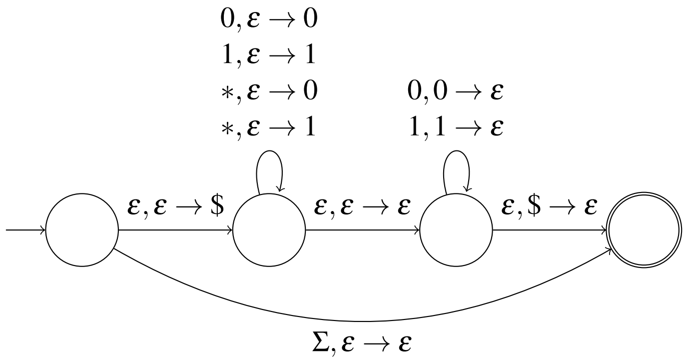
(c) Here is an equivalent grammar G' in Chomsky normal form:

$$
\begin{aligned}
S &\to XY \mid \varepsilon \mid c \\
X &\to AZ \mid a \\
Y &\to BZ \mid b \\
Z &\to XY \mid c \\
A &\to a \\
B &\to b
\end{aligned}
$$

Use the CYK algorithm to parse the string $w = aacbbc$ according to G'. Show the resulting table, and explain how it shows whether or not $w \in L(G')$.

14. Working with a PDA

Consider the following PDA with input alphabet $\Sigma = \{0, 1, *\}$ and stack alphabet $\Gamma = \{0, 1\}$:



What is the language of this PDA?

15. Designing a PDA

Let L be the language of binary strings with an equal number of 0s and 1s (so for example $\varepsilon, 01, 1010 \in L$ but $0, 11, 1000 \notin L$). Draw (as a graph) a PDA recognizing L.