CSE303 PRACTICE FINAL
(5 extra points)

NAME **ID:**

Practice Final is DUE LAST DAY OF CLASSES. YOU DON'T NEED to solve the PUMPING LEMMA and Turing Machine Problems - they will NOT appear on the FINAL. I included them so show you the SOLUTIONS.

FOR FINAL study Practice Final (minus PUMPING LEMMA and Turing Machines) and Problems from $Q1 - Q4$, Practice $Q1 - Q4$, and Midterm and Practice midterm. I will choose some of these problems for your Final.

PART 1: Yes/No Questions Circle the correct answer to ALL questions. Write ONE-SENTENCE justification to **ten questions**.

1. There is a set A and an equivalence relation defined on A that is an order relation with 2 Maximal elements.
   Justify:
   y **n**
2. $(ab \cup a^*b)^*$ is a regular language.
   Justify:
   y **n**
3. Let $\Sigma = \phi$, there is $L \neq \phi$ over Σ.
   Justify:
   y **n**
4. A is uncountable iff $\mid A \mid = \mathbf{c}$ (continuum).
   Justify:
   y **n**
5. There are uncountably many languages over $\Sigma = \{a\}$.
   Justify:
   y **n**
6. Let RE be a set of regular expressions. $L \subseteq \Sigma^*$ is regular iff $L = L(r)$,
   $r \in RE$.
   Justify:
   y **n**

7. $L^* = \{w \in \Sigma^* : \exists_{q \in F} (s, w) \vdash_M^* (q, e)\}$.
Justify:

y **n**

8. $(a^*b \cup \phi^*)$ is a regular expression.
Justify:

y **n**

9. $\{a\}^*\{b\} \cup \{ab\}$ is a language (regular).
Justify:

y **n**

10. Let L be a language defined by $(a^*b \cup ab)$, i.e (shorthand) $L = a^*b \cup ab$. Then $L \subseteq \{a, b\}^*$.
Justify:

y **n**

11. $\Sigma = \{a\}$, there are **c** (continuum) languages over Σ.
Justify:

y **n**

12. $L^* = L^+ - \{e\}$.
Justify:

y **n**

13. $L^* = \{w_1 \ldots w_n, w_i \in L, i = 1, \ldots, n\}$.
Justify:

y **n**

14. For any languages $L_1, L_2, L_3 \subset \Sigma^* L_1, \cup (L_2 \cap L_3) = (L_1 \cup L_2) \cap (L_1 \cup L_3)$.
Justify:

y **n**

15. For any languages $L_1, L_2 \subset \Sigma^*$, if $L_1 \subseteq L_2$, then $(L_1 \cup L_2)^* = L_2^*$.
Justify:

y **n**

16. $((\phi^* \cap a) \cup (\phi \cup b^*)) \cap \phi^*$ represents a language $L = \{e\}$.
Justify:

y **n**

17. $L = ((\phi^* \cup b) \cap (b^* \cup \phi))$ (shorthand) has only one element.
Justify:

y **n**

18. $L(M) = \{w \in \Sigma^* : (q, w) \vdash_M^* (s, e)\}$.
Justify:

y **n**

19. If M is a FA, then $L(M) \neq \phi$.
Justify:

y **n**

20. If M is a nondeterministic FA, then $L(M) \neq \phi$.
Justify:

y **n**

21. $L(M_1) = L(M_2)$ iff M_1 and M_2 are finite automata.
Justify:

y **n**

22. A language is regular iff $L = L(M)$ and M is a deterministic automaton.
Justify:

y **n**

23. If L is regular, then there is a nondeterministic M, such that $L = L(M)$.
Justify:

y **n**

24. Any finite language is CF.
Justify:

y **n**

25. Intersection of any two regular languages is CF language.
Justify:

y **n**

26. Union of a regular and a CF language is a CF language.
Justify:

y **n**

27. L_1 is regular, L_2 is CF, $L_1, L_2 \subseteq \Sigma^*$, then $L_1 \cap L_2 \subseteq \Sigma^*$ is CF.
Justify:

y **n**

28. If L is regular, there is a PDA M such that $L = L(M)$.
Justify:

y **n**

29. If L is regular, there is a CF grammar G, such that $L = L(G)$.
Justify:

y **n**

30. $L = \{a^n b^n c^n : n \geq 0\}$ is CF.
Justify:

y **n**

31. $L = \{a^n b^n : n \geq 0\}$ is CF.
Justify:

y **n**

32. Let $\Sigma = \{a\}$, then for any $w \in \Sigma^*, w^R w \in \Sigma^*$.
Justify:

y **n**

33. $A \rightarrow Ax, A \in V, x \in \Sigma^*$ is a rule of a regular grammar.
Justify:

y **n**

34. Regular grammar has only rules $A \rightarrow xA, A \rightarrow x, x \in \Sigma^*, A \in V - \Sigma$.
Justify:

y **n**

35. Let $G = (\{S, (,)\}, \{(,)\}, R, S)$ for $R = \{S \rightarrow SS \mid (S)\}$. $L(G)$ is regular.
Justify:

y **n**

36. The grammar with rules $S \rightarrow AB, B \rightarrow b \mid bB, A \rightarrow e \mid aAb$ generates a language $L = \{a^k b^j : k < j\}$.
Justify:

y **n**

37. $L = \{w \in \{a, b\}^* : w = w^R\}$ is regular.
Justify:

y **n**

38. We can always show that L is regular using Pumping Lemma.
Justify:

y **n**

39. $((p, e, \beta), (q, \gamma)) \in \Delta$ means: read nothing, move from p to q.
Justify:

y **n**

40. $L = \{a^n b^m c^n : n, m \in N\}$ is CF.
Justify:

y **n**

41. If L is regular, then there is a CF grammar G, such that $L = L(G)$.
Justify:

y **n**

42. There is countably many non CF languages.
Justify:

y **n**

43. Every subset of a regular language is a regular language.
Justify:

y **n**

44. A parse tree is always finite.
Justify:

y **n**

45. Any regular language is accepted by some PD automata.
Justify:

y **n**

pfinal

46. Every subset of a regular language is a language.
Justify:

y **n**

47. A parse tree is always finite.
Justify:

y **n**

48. Parse trees are equivalence classes.
Justify:

y **n**

49. For some languages, all grammars are ambiguous.
Justify:

y **n**

50. A CF grammar G is called ambiguous if there is $w \in L(G)$ with at least two distinct parse trees.
Justify:

y **n**

51. A CF language L is inherently ambiguous iff all context-free grammars G, such that $L(G) = L$ are ambiguous.
Justify:

y **n**

52. Programming languages are sometimes inherently ambiguous.
Justify:

y **n**

53. The largest number of symbols on the right-hand side of any rule of a CF grammar G is called called a fanout and denoted by $\phi(G)$.
Justify:

y **n**

54. The Pumping Lemma for CF languages uses the notion of the fanout.
Justify:

y **n**

55. Any regular language is accepted by some PD automata.
Justify:

y **n**

56. Class of context-free languages is closed under intersection.
Justify:

y **n**

57. There is countably many non-regular languages.
Justify:

y **n**

58. Every subset of a regular language is regular.
Justify:

y **n**

59. A CF language is a regular language.
Justify:

y **n**

60. Class of context-free languages is closed under intersection.
Justify:

y **n**

61. Class of regular languages is closed under intersection.
Justify:

y **n**

62. A regular language is a CF language.
Justify:

y **n**

63. Turing Machines are as powerful as today's computers.
Justify:

y **n**

64. It is proved that everything computable (algorithm) is computable by a Turing Machine and vice versa.
Justify:

65. Church's Thesis says that Turing Machines are the most powerful.
Justify:

y **n**

66. Turing Machines can read and write.
Justify:

y **n**

67. A configuration of a Turing machine $M = (K, \Sigma, \delta, s, H)$ is any element of a set $K \times \Sigma^* \times (\Sigma^*(\Sigma - \{\#\}) \cup \{e\})$, where $\#$ denotes a blanc symbol.
Justify:

y **n**

68. A computation of a Turing machine can start at any position of $w \in \Sigma$.

69. A computation of a Turing machine can start at any state.
Justify:

y **n**

70. In Turing machines, words $w \in \Sigma^*$ can't contain blanc symbols.
Justify:

y **n**

71. A Turing machine M decides a language $L \subseteq \Sigma^*$, if for any word $w \in \Sigma^*$ the followong is true.
If $w \in L$, then M accepts w; and if $w \notin L$ then M rejects w.
Justify:

y **n**

PART 2: Problems

WRITE solutions to **TWO** problems of your choice. SOLVE all of them, a practice.

QUESTION 1 Let Σ be any alphabet, L_1, L_2 two languages over Σ such that $e \in L_1$ and $e \in L_2$. Show that

$$(L_1 \Sigma^{\star} L_2)^{\star} = \Sigma^{\star}$$

Solution :

QUESTION 2 Construct a non-deterministic finite automaton M, such that

$$L(M) = (ab)^{*}(ba)^{*}.$$

Draw a state diagram and specify all components K, Σ, Δ, s, F Justify your construction by listing strings accepted the state diagram of M.

State Diagram of M is:

Some elements of $L(M)$ as defined by the state diagram are:

Components of M are:

QUESTION 3 Given a **Regular grammar** $G = (V, \Sigma, R, S)$, where

$$V = \{a, b, S, A\}, \quad \Sigma = \{a, b\},$$

$$R = \{S \rightarrow aS \ |A \ |e, \quad A \rightarrow abA \ |a \ |b\}.$$

1. Construct a finite automaton M, such that $L(G) = L(M)$. You can draw a diagram.

2. Trace a transitions of M that lead to the acceptance of the string $aaaababa$, and compare with a derivation of the same string in G.

QUESTION 4 Construct a context-free grammar G such that

$$L(G) = \{w \in \{a,b\}^* : \; w = w^R\}.$$

Justify your answer.

QUESTION 5 Construct a **pushdown** automaton M such that

$$L(M) = \{w \in \{a,b\}^* : \; w = w^R\}$$

Components of M are:

Explain your construction. Write motivation.

Diagram of M is:

Trace a transitions of M that lead to the acceptance of the string *ababa*.

QUESTION 6 Construct a PDA M, such that

$$L(M) = \{b^n a^{2n} : n \geq 0\}.$$

Solution $M = \{K, \Sigma, \Gamma, \Delta, s, F\}$ for

Explain the construction. Write motivation.

Trace a transitions of M that leads to the acceptance of the string *bbaaaa*.

QUESTION 7

Use PUMPING LEMMA to prove that

$$L = \{ww : \quad w \in \{a, b\}^*\}$$

in NOT regular. Consider ALL cases.

Question 8 Use Pumping Lemma to prove that

$$L = \{a^{n^2} : n \geq 0\}\}$$

is not CF.

QUESTION 9 Here is the definition:

Let $L \subseteq \Sigma^*$. For any $x, y \in \Sigma^*$ we define an equivalence relation on Σ^* as follows.

$$x \approx_L y \quad iff \quad \forall z \in \Sigma^* (xz \in L \Leftrightarrow yz \in L).$$

Let now

$$L = (aab \cup ab)^*.$$

FIND all equivalence classes of $x \approx_L y$.

Write all definitions and show work.

Question 10 Show that the following language L in NOT CF.

$$L = \{w \in \{a, b, c\} : \ all\ accurences\ of\ a, b, c\ in\ w\ are\ different\}.$$