PART 1: Yes/No Questions  Circle the correct answer. Write ONE-SENTENCE justification.

1. There are uncountably many languages over $\Sigma = \{a\}$.
   Justify: $|\{a\}^*| = \aleph_0$ and $|2^{\{a\}^*}| = \mathfrak{c}$ and any set of cardinality $\mathfrak{c}$ is uncountable.

2. Let $\Sigma = \phi$, there is $L \neq \phi$ over $\Sigma$.
   Justify: $0^* = \{e\}$ and $L = \{e\} \subseteq \Sigma^*$

3. $L^* = \{w \in \Sigma^* : \exists q \in F(s, w) \vdash^*_M (q, e)\}$.
   Justify: this is definition of $L(M)$, not $L^*$

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

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

6. $\Sigma = \{a\}$, there are $\mathfrak{c}$ (continuum) languages over $\Sigma$.
   Justify: $|2^{\{a\}^*}| = \mathfrak{c}$

7. For any languages $L_1, L_2, L_3 \subseteq \Sigma^* L_1, \cup (L_2 \cap L_3) = (L_1 \cup L_2) \cap (L_1 \cup L_3)$.
   Justify: languages are sets

8. $L^* = L^+ - \{e\}$.
   Justify: only when $e \notin L$

9. $L = ((\phi^* \cup b) \cap (b^* \cup \phi))$ (shorthand) has only one element.
   Justify: $\{e, b\} \cap \{b\}^* = \{e, b\}$

10. If $M$ is a FA, then $L(M) \neq \phi$.
    Justify: take $M$ with $\Sigma = \phi$

11. If $M$ is a nondeterministic FA, then $L(M) \neq \phi$.
    Justify: take $M$ with $\Sigma = \phi$ or $F = \phi$
12. \( L(M_1) = L(M_2) \) iff \( M_1 \) and \( M_2 \) are finite automata.
   \textbf{Justify}: take as \( M_1 \) any automata such that \( L(M_1) \neq \emptyset \) and \( M_2 \) such that \( L(M_2) = \emptyset \).

13. If \( L \) is regular, then there is a finite \( M \), such that \( L = L(M) \).
   \textbf{Justify}: Main Theorem

14. Any finite language is CF.
   \textbf{Justify}: any finite language is regular and \( RL \subseteq CFL \)

15. \( L_1 \) is regular, \( L_2 \) is CF, \( L_1, L_2 \subseteq \Sigma^* \), then \( L_1 \cap L_2 \subseteq \Sigma^* \) is CF.
   \textbf{Justify}: theorem

16. Intersection of any two regular languages is CF language.
   \textbf{Justify}: Regular languages are closed under intersection and \( RL \subseteq CFL \)

17. Union of a regular and a CF language is a CF language.
   \textbf{Justify}: \( RL \subseteq CFL \) and FCL are closed under union

18. \( L = \{a^n b^n c^n : n \geq 0\} \) is CF.
   \textbf{Justify}: is not CF, as proved by Pumping Lemma for CF languages

19. If \( L \) is regular, there is a PDA \( M \) such that \( L = L(M) \).
   \textbf{Justify}: FA is a PDA operating on an empty stock

20. If \( L \) is regular, there is a CF grammar \( G \), such that \( L = L(G) \).
   \textbf{Justify}: \( RL \subseteq CFL \)

21. \( A \rightarrow Ax, A \in V, x \in \Sigma^* \) is a rule of a regular grammar.
   \textbf{Justify}: this is a rule of a left-linear grammar and we defined regular grammar as a right-linear

22. \( L = \{a^n b^n : n \geq 0\} \) is CF.
   \textbf{Justify}: \( L = L(G) \) for \( G \) with \( R = \{S \rightarrow aSb|e\} \)

23. Let \( \Sigma = \{a\} \), then for any \( w \in \Sigma^* \), \( w^Rw \in \Sigma^* \).
   \textbf{Justify}: \( a^R = a \) and \( w^R = w \) for \( w \in \{a\}^* \)

24. Let \( G = (\{S, (, )\}, \{(,), \}, R, S) \) for \( R = \{S \rightarrow SS \mid (S)\} \). \( L(G) \) is regular.
   \textbf{Justify}: \( L(G) = \emptyset \) and hence regular
25. \((p, \epsilon, \beta), (q, \gamma) \in \Delta\) means: read nothing, move from \(p\) to \(q\)

\textbf{Justify}: and replace \(\gamma\) by \(\beta\) on the top of the stack

26. \(L = \{a^n b^m c^n : n, m \in N\}\) is CF.

\textbf{Justify}: when \(n = m\) we get \(L = \{a^n b^n c^n : n \in N\}\) that is not CF

27. Every subset of a Context Free language is a language.

\textbf{Justify}: subset of a set is a set

28. A parse tree is always finite.

\textbf{Justify}: derivations are finite

29. Any regular language is accepted by some PD automata.

\textbf{Justify}: \(RL \equiv FA, FA \subseteq PDA\)

30. Every subset of a regular language is a regular language.

\textbf{Justify}: \(L = \{a^n b^n : n \geq 0\} \subseteq a^* b^*\) and \(L\) is not regular

31. A CF language is a regular language.

\textbf{Justify}: \(L = \{a^n b^n : n \geq 0\}\) is CF and not regular

32. A regular language is a CF language.

\textbf{Justify}: Regular grammar is a special case of a context-free grammar

33. A parse tree is always finite.

\textbf{Justify}: Any derivation of \(w\) in a CF grammar is finite.

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

\textbf{Justify}: definition

35. A CF language \(L\) is inherently ambiguous iff all context-free grammars \(G\), such that \(L(G) = L\) are ambiguous.

\textbf{Justify}: definition

36. Turing Machines can read and write.

\textbf{Justify}: by definition

37. 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.

\textbf{Justify}: a configuration is an element of a set \(K \times \Sigma^* \times (\Sigma^*(\Sigma - \{\#\}) \cup \{e\})\)

38. A computation of a Turing machine can start at any position of \(w \in \Sigma\).

\textbf{Justify}: by definition
39. Turing Machines are as powerful as today’s computers.
   \textbf{Justify:} thesis
   \textit{y}

40. It is proved that everything computable (algorithm) is computable by a Turing Machine and vice versa.
   \textbf{Justify:} this is Church - Turing Hypothesis, not a theorem
   \textit{n}

41. Church’s Thesis says that Turing Machines are the most powerful.
   \textbf{Justify:} We adopt a Turing Machine that halts on all inputs as a formal notion of ”an algorithm”.
   \textit{n}

\textbf{PART 2: PROBLEMS}

\textbf{QUESTION 1} Given a Regular grammar \( G = (V, \Sigma, R, S) \), where
\[ V = \{a, b, S, A\}, \quad \Sigma = \{a, b\}, \]
\[ R = \{ S \rightarrow aS | e, \quad A \rightarrow abA | a | b \}. \]

1. Construct a finite automaton \( M \), such that \( L(G) = L(M) \).
   \textbf{Solution} We construct a non-deterministic finite automata
   \[ M = (K, \Sigma, \Delta, s, F) \]
   as follows:
   \[ K = (V - \Sigma) \cup \{f\}, \quad \Sigma = \Sigma, \quad s = S, \quad F = \{f\}, \]
   \[ \Delta = \{(S, a, S), (S, e, A), (S, e, f), (A, ab, A), (A, a, f), (A, b, f)\} \]

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 \( G \).
   \textbf{Solution}
   The accepting computation is:
   \[ (S, aaaababa) \vdash_M (S, aaababa) \vdash_M (S, ababa) \vdash_M (A, ababa) \]
   \[ \vdash_M (A, aba) \vdash_M (A, a) \vdash_M (f, e) \]
   \[ G \text{ derivation is:} \]
   \[ S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS \Rightarrow aaaS \Rightarrow aaaA \Rightarrow aaabA \Rightarrow aaababA \Rightarrow aaaababa \]
QUESTION 2 Construct a context-free grammar $G$ such that

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

Justify your answer.

Solution $G = (V, \Sigma, R, S)$, where

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

$$R = \{ S \to aSa \mid bSb \mid a \mid b \mid e \}.$$  

Derivation example: $S \Rightarrow aSa \Rightarrow abSba \Rightarrow ababa$

$ababa^R = ((ab)a(ba))^R = (ba)^RaR(ab)^R = ababa.$

Observation 1 We proved in class that for any $x, y \in \Sigma^*$, $(xy)^R = y^Rx^R$.

From this we have that $(xyz)^R = ((xy)z)^R = z^R(xy)^R = z^Ry^Rx^R$

Grammar correctness justification: observe that the rules $S \to aSa \mid bSb \mid e$ generate the language $L_1 = \{ ww^R : w \in \Sigma^* \}$. With additional rules $S \to a \mid b$ we get hence the language $L = L_1 \cup \{ waw^R : w \in \Sigma^* \} \cup \{ wbw^R : w \in \Sigma^* \}$. Now we are ready to prove that

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

Proof Let $w \in L$, i.e. $w = xx^R$ or $w = xax^R$ or $w = xbx^R$. We show that in each case $w = w^R$ as follows.

c1: $w^R = (xx^R)^R = (x^R)^R x^R = x^R = w$ (used property: $(x^R)^R = x$).

c2: $w^R = (xax^R)^R = (x^R)^R a^Rx^R = xax^R = w$ (used Observation 1 and properties: $(x^R)^R = x$ and $a^R = a$).

c3: $w^R = (xbx^R)^R = (x^R)^R b^Rx^R = xbx^R = w$ (used Observation 1 and properties: $(x^R)^R = x$ and $b^R = b$).

QUESTION 3 Construct a pushdown automaton $M$ such that

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

Solution 1 We define $M$ as follows: $M = (K, \Sigma, \Gamma, \Delta, s, F)$
**M components** are

\[ K = \{ s, f \}, \Sigma = \{ a, b \}, \Gamma = \{ a, b \}, F = \{ f \} \]

\[ \Delta = \{ ((s, a, e), (s, a)), ((s, b, e), (s, b)), ((s, e, e), (f, e)), ((s, a, e), (f, a)), ((s, b, e), (f, b)), ((f, a, a), (f, e)), ((f, b, b), (f, e)) \} \]

Trace a transitions of \( M \) that lead to the acceptance of the string \( ababa \).

**Solution**

\[
\begin{align*}
S & \quad ababa \quad e \\
S & \quad baba \quad a \\
S & \quad aba \quad ba \\
f & \quad ba \quad ba \\
f & \quad a \quad a \\
f & \quad e \quad e 
\end{align*}
\]

**QUESTION 4** Construct a PDA \( M \), such that

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

**Solution** \( M = (K, \Sigma, \Gamma, \Delta, s, F) \) for

\[ K = \{ s, f \}, \Sigma = \{ a, b \}, \Gamma = \{ a \}, s, F = \{ f \}, \]

\[ \Delta = \{ ((s, b, e), (s, aa)), ((s, e, e), (f, e)), ((f, a, a), (f, e)) \} \]

Explain the construction. Write motivation.

**Solution** \( M \) operates as follows: \( \Delta \) pushes \( aa \) on the top of the stack while \( M \) is reading \( b \), switches to \( f \) (final state) non-deterministically; and pops \( a \) while reading \( a \) (all in final state). \( M \) puts on the stock two \( a \)'s for each \( b \), and then remove all \( a \)'s from the stack comparing them with \( a \)'s in the word while in the final state.

Trace a transitions of \( M \) that leads to the acceptance of the string \( bbaaaa \).

**Solution** The accepting computation is:

\[
\begin{align*}
(s, bbaaaa, e) \vdash_M (s, baaa, aa) & \vdash_M (s, aaaa, aaaa) \vdash_M (f, aaaa, aaaa) \\
\vdash_M (f, aaa, aaa) & \vdash_M (f, aa, aa) \vdash_M (f, a, a) \vdash_M (f, e, e)
\end{align*}
\]
Solution 2 \( M = (K, \Sigma, \Gamma, \Delta, s, F) \) for

\[
K = \{s, f\}, \Sigma = \{a, b\}, \Gamma = \{b\}, s, F = \{f\},
\]

\[
\Delta = \{((s, b, e), (s, b)), ((s, e, e), (f, e)), ((f, aa, b), (f, e))\}
\]