CSE 303 FINAL SOLUTIONS

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

No justification- no credit

1. Set $A$ is uncountable iff $R \subseteq A$ ($R$ is the set of real numbers).

   Justify: $R, 2^R$ are both uncountable and $R$ is not a subset of $2^R$ ($R \not\subseteq 2^R$) but $R \in 2^R$.

   THIS is not the only example!

2. All infinite sets have the same cardinality.

   Justify: $|N| < |2^N|$ by Cantor Theorem and $N, 2^N$ are infinite

   THIS is not the only example!

3. $R^* = R \cup \{(a, b) : \text{there is a path from a to b}\}$.

   Justify: book definition

4. $(ab \cup a^*b)^*$ is a regular language.

   Justify: this is a regular expression

5. Let $\Sigma = \phi$, there is $L \neq \phi$ over $\Sigma$.

   Justify: $\emptyset^* = \{e\}$ and $L = \{e\} \subseteq \Sigma^*$

6. There are uncountably many languages over $\Sigma = \{a\}$

   Justify: $|\{a\}^*| = \aleph_0$ (infinitely countable) and by definition, $L \subseteq \Sigma^*$, and $|2^{\aleph_0}| = \mathfrak{c}$ and any set of cardinality $\mathfrak{c}$ is uncountable

   Hence there uncountably many languages over $\Sigma = \{a\}$

7. Let $R$ be a set of regular expressions and $L$ be a function that maps $R$ into set of all subsets of $\Sigma^*$. Then the following it true.

   $L \subseteq \Sigma^*$ is a regular language iff $L = \mathcal{L}$, for some $r \in R$

   Justify: definition of regular language

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

   Justify: this is part of a definition of $L(M)$, not $L^*$

9. $(a^*b \cup \phi^*)$ is a regular expression.

   Justify: definition

10. $\{a\}^*\{b\} \cup \{ab\}$ is a regular language

    Justify: 1. regular languages are closed under union, Kleene Star and concatenation , or

    2. This language is defined by a regular expression $a^*b \cup ab$

11. 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: $a, b$ are the only

12. $\Sigma = \{a\}$, there are $\mathfrak{c}$ (continuum) languages over $\Sigma$.

    Justify: $|(a)^*| = \aleph_0$ (infinitely countable) and by definition, $L \subseteq \Sigma^*$, and $|2^{\aleph_0}| = \mathfrak{c}$, or

    for SHORT: $|2^{\{a\}}| = \mathfrak{c}$

13. $L^* = L^+ \setminus \{e\}$

    Justify: $e \in L^*$ and $e \not\in L^+ \setminus \{e\}$
14. $L^* = \{ w_1 \ldots w_n, w_i \in L, i = 1, \ldots, n \}$. 
   **Justify:** $i = 0, 1, \ldots, n$.

15. 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

16. For any languages $L_1, L_2 \subseteq \Sigma^*$,
   if $L_1 \subseteq L_2$, then $(L_1 \cup L_2)^* = L_2^*$
   **Justify:** languages are sets, so $(L_1 \cup L_2) = L_2$ and also $(L_1 \cup L_2)^* = L_2^*$

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

18. $L(M) = \{ w \in \Sigma^* : (q, w) \vdash^*_M (s, e) \}$. 
   **Justify:** only when $s \in F$ and $q$ is initial state of $M$

19. For every deterministic automaton $M$, $L(M) \neq \emptyset$
   **Justify:** take $M$ with $\Sigma = \emptyset$

20. If $M$ is a nondeterministic FA, then $L(M) \neq \emptyset$
   **Justify:** take $M$ with $\Sigma = \emptyset$ or $F = \emptyset$

21. $L(M_1) = L(M_2)$ iff $M_1$ and $M_2$ are finite automata.
   **Justify:** take as $M_1$ any automata such that $L(M_1) \neq \emptyset$ and $M_2$ such that $L(M_2) = \emptyset$
   THIS is not the only example!

22. A language is regular iff $L = L(M)$ and $M$ is a deterministic automaton.
   **Justify:** Theorem: A language is regular iff $L = L(M)$ and $M$ is a finite automaton and
   Theorem: Deterministic and nondeterministic automata are equivalent

23. If $M$ is a deterministic automaton, then there is a nondeterministic $N$, such that $L(N) = L(M)$
   **Justify:** a function is a special relation, or
   Theorem: Deterministic and nondeterministic automata are equivalent

24. Any finite language is CF
   **Justify:** any finite language is regular and $RL \subseteq CFL$ or
   we construct a CF grammar with set of rules $\{ S \rightarrow \sigma : \sigma \in \Sigma \}$

25. Intersection of any two regular languages is a CF language.
   **Justify:** Regular languages are closed under intersection and $RL \subseteq CFL$

26. Union of a regular and a CF language is a CF language.
   **Justify:** $RL \subseteq CFL$ and FCL are closed under union

27. $L_1$ is regular, $L_2$ is CF and $L_1, L_2 \subseteq \Sigma^*$, then $L_1 \cap L_2 \subseteq \Sigma^*$ is a CF language
   **Justify:** theorem
28. If \( L \) is regular, then there is a PDA \( M \) such that \( L = L(M) \).
   **Justify:** Theorem: A language is regular iff \( L = L(M) \) and \( M \) is a finite automaton and FA is a PDA operating on an empty stack.

29. If \( L \) is regular, then there is a CF grammar \( G \), such that \( L = L(G) \).
   **Justify:** we proved that \( RL \subseteq CFL \).

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

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

32. Let \( \Sigma = \{ a \} \), then for any \( w \in \Sigma^* \), \( w^Rw \in \Sigma^* \).
   **Justify:** \( a^R = a \), \( (wa)^R = aw^R \) and by induction over length of \( w \), \( w^R = w \) for \( w \in \{ a \}^* \).

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

34. Regular grammar has only rules \( A \rightarrow xA, A \rightarrow x, x \in \Sigma^*, A \in V - \Sigma \).
   **Justify:** not only, \( A \rightarrow xB \) for \( B \neq A \) is also a rule of a regular grammar.

35. Let \( G = (\{S, (, )\}, \{(, )\}, R, S) \) for \( R = \{ S \rightarrow SS | (S) \} \)
   \( L(G) \) is regular.
   **Justify:** \( L(G) = \emptyset \) and hence regular.

36. \( L = \{ w \in \{a, b\}^* : w = w^R \} \) is regular.
   **Justify:** we use Pumping Lemma; while pumping the string \( a^kba^k \) with \( y \) containing only \( a \)'s we get that \( xy^2z \notin L \).

37. We can always show that \( L \) is regular using Pumping Lemma
   **Justify:** we use Pumping Lemma to prove (if possible) that \( L \) is not regular Does not ALWAYS work!

38. \( (p, e, \beta, (q, \gamma)) \in \Delta \) means: read nothing, move from \( p \) to \( q \)
   **Justify:** and replace \( \gamma \) by \( \beta \) on the top of the stack.

39. If \( L \) is not regular, then there is a CF grammar \( G \), such that \( L = L(G) \)
   **Justify:** \( L = \{ a^n b^n c^n : n \geq 0 \} \) in not regular and not CF.

40. There is countably many non CF languages over \( \Sigma \neq \phi \)
   **Justify:** there is countably many CF languages and there is uncountably many ( exactly \( c \)) languages over \( \Sigma \) so \( |\Sigma^* - CF| = c \), i.e. is uncountable.

41. Every subset of a regular language is a language.
   **Justify:** languages are sets and a subset of a set is a set.
42. Any regular language is accepted by some PD automata.
   \textbf{Justify:} \( RL \equiv FA, FA \subseteq PDA \) 

43. \( \Delta \subseteq (K \times \Sigma^* \times \Gamma^*) \times (K \times \Gamma^* \) is a transition relation of a pushdown automaton
   \textbf{Justify:} \( \Delta \) must be finite

44. Let \( M \) be a pushdown automaton
   \( L(M) = \{ w \in \Sigma^* : (s, w, e) \models^* M (f, e, e) \} \)
   \textbf{Justify:} \( f \in F \) and \( s \) the initial state

45. There is uncountably many regular languages
   \textbf{Justify:} there is countably many regular expressions

46. 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

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

48. A regular language is a CF language.
   \textbf{Justify:} Regular grammar is a special case of a context-free grammar

49. Every subset of a regular language is a regular language.
   \textbf{Justify:} \( L_1 = a^n b^n \) is a non-regular subset of a regular language \( L_2 = a^* b^* \)

\textbf{PART 2: PROBLEMS}

\textbf{QUESTION 1} Let \( \Sigma \) be any alphabet, \( L_1, L_2 \) two languages over \( \Sigma \) such that \( e \in L_1 \) and \( e \in L_2 \).
Show that
\[ (L_1 \Sigma^* L_2)^* = \Sigma^* \]
\textbf{Solution :} By definition, \( L_1 \subseteq \Sigma^* \), \( L_2 \subseteq \Sigma^* \) and \( \Sigma^* \subseteq \Sigma^* \). Hence
\[ (L_1 \Sigma^* L_2)^* \subseteq \Sigma^*. \]

We have to show that also
\[ \Sigma^* \subseteq (L_1 \Sigma^* L_2)^* . \]

Let \( w \in \Sigma^* \) we have that also \( w \in (L_1 \Sigma^* L_2)^* \) because \( w = ewe \) and \( e \in L_1 \) and \( e \in L_2 \).

\textbf{QUESTION 2}

Let \( L \) be a language defines as follows
\[ L = \{ w \in \{ a, b \}^* : every a is either immediately proceeded or followed by b \}. \]

\textbf{Part 1} Describe a regular expression \( r \) such that \( L(r) = L \) (Meaning of \( r \) is \( L \)).
\textbf{Solution} \( L = (b \cup ab \cup ba \cup aba)^* \)

\textbf{Part 2} Construct a \textit{finite state automata} \( M \), such that \( L(M) = L \).
\textbf{Draw} a diagram and list the components of \( M \).
List some elements of L(M)

Solution

Components of M are:

\[ K = \{ s \}, \{ a, b \}, s, F = \{ s \}, \]
\[ \Delta = \{ (s, b, s), (s, ab, s), (s, ba, s), (s, aba, s) \}. \]

Some elements of L(M) are: \( b, bb, baba, abab, abbbba, bbabbbabbbabbb \)

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

\[ V = \{ a, b, S, A \}, \quad \Sigma = \{ a, b \}, \]
\[ R = \{ S \rightarrow aS \mid A \mid e, A \rightarrow abA \mid a \mid b \}. \]

Part 1 Use the construction in the proof of L-GTheorem:

Language L is regular if and only if there exists a regular grammar G such that L = L(G)

to construct a finite automaton M, such that L(G) = L(M).

Solution We construct a non-deterministic finite automata

\[ M = (K, \Sigma, \Delta, s, F) \]

as follows:

\[ K = (V - \Sigma) \cup \{ f \}, \quad \Sigma = \Sigma, s = S, \quad F = \{ f \}, \]
\[ \Delta = \{ (S, a, S), (S, c, A), (S, e, f), (A, ab, A), (A, a, f), (A, b, f) \}. \]

Part 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.

Solution

The accepting computation is:

\[ (S, aaaababa) \vdash_M (S, aaaababa) \vdash_M (S, aaababa) \vdash_M (S, ababa) \vdash_M (A, ababa) \]
\[ \vdash_M (A, aba) \vdash_M (A, a) \vdash_M (f, e) \]

G derivation is:

\[ S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS \Rightarrow aaaaA \Rightarrow aaaaabA \Rightarrow aaaaababA \Rightarrow aaaaababa \]

QUESTION 4 Define a grammar G such that \( L(G) = \{ a^k b^j : k < j \} \)

Justify your construction

Solution 1
The grammar has the rules \( S \rightarrow AB, B \rightarrow b | bB, A \rightarrow e | aAb \)

**Justification**

the rule \( A \rightarrow e | aAb \) produces the same amount of a’s and b’s, the rule \( B \rightarrow bB \) adds only b’s

More formally, let’s look at the derivations

\[
S \Rightarrow AB \Rightarrow \ldots \Rightarrow a^n b^n B \Rightarrow \ldots \Rightarrow a^n b^n b^k \\
S \Rightarrow AB \Rightarrow \ldots \Rightarrow a^n b^n B \Rightarrow a^n b^n b
\]

we get \( a^n b^{n+k} \in L(G) \) and \( n < n + k \), and \( a^n b^{n+1} \in L(G) \) and \( n < n + 1 \)

**Solution 2**

The grammar has the rules \( S \rightarrow aSB | Sb | b \)

**QUESTION 5**

**Prove** that the Class of context-free languages is NOT closed under intersection

**Solution**:

\( L_1 = \{ a^n b^n c^m, n,m \geq 0 \} \) is CF, \( L_2 = \{ a^n b^n c^n, n,m \geq 0 \} \) is CF, but \( L_1 \cap L_2 = \{ a^n b^n c^n, n \geq 0 \} \) is not CF

**QUESTION 6** Construct a pushdown automaton \( M \) such that

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

**Draw a DIAGRAM**

**Write the components**

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

\( M \) components are

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

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

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

**Solution**

\[
S \quad ababa \ \ e \\
S \quad baba \ \ a \\
S \quad aba \ \ ba \\
f \quad ba \ \ ba
\]
QUESTION 7

Construct a PDA $M$, such that

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

Draw a DIAGRAM

Write the components

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 stock 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 stock 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 $bbaaaaa$.

Solution The accepting computation is:

$\vdash_M (s, bbaaaaa, e) \vdash_M (s, baaaaa, aa) \vdash_M (s, aaaaa, aaaa) \vdash_M (f, aaaa, aaaa)$

$\vdash_M (f, aaaa, aaaa) \vdash_M (f, aa, aa) \vdash_M (f, a, a) \vdash_M (f, e, e)$

Solution 2 $M = (K, \Sigma, \Gamma, \Delta, s, F)$ for

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

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

QUESTION 8 (Extra Credit)

Use closure under union for CF languages to show that $L = \{a^n b^n : n \neq m\}$ is CF language

Solution 1 We know that $L_1 = \{a^m b^n : m > n\}$ and $L_2 = \{a^m b^n : m < n\}$ are context-free languages (we constructed proper grammars for both of them). $L = L_1 \cup L_2$, hence $L$ is context free as the class of context free languages is closed under union.
Solution 2 Observe that $L_1 = \{a^m b^n : m > n\} = \{a\}^+ \{a^n b^n : n \in N\}$ We proved that $\{a^n b^n : n \in N\}$, is context free, $\{a\}^+$ is regular and hence CF and the class of context free languages is closed under concatenation, hence $L_1$ is also context free.

Similarly, $L_2 = \{a^m b^n : m < n\} = \{a^n b^n : n \in N\}\{b\}^+$, so $L_2$ is context free. $L = L_1 \cup L_2$, hence $L$ is context free as the class of context free languages is closed under union.