1 YES/NO questions

1. For any binary relation $R \subseteq A \times A$, $R^*$ exists.
   \textbf{Justify:} definition \hspace{1cm} y

2. For any binary relation $R \subseteq A \times A$, $R^{-1}$ exists.
   \textbf{Justify:} The set $R^{-1} = \{(b, a) : (a, b) \in R\}$ always exists. \hspace{1cm} y

3. For any function $f$ from $A \neq \emptyset$ onto $A$, $f$ has property $f(a) \neq a$ for certain $a \in A$.
   \textbf{Justify:} $f(x) = x$ is always "onto". \hspace{1cm} n

4. All infinite sets have the same cardinality.
   \textbf{Justify:} $|N| < |2^N|$ by Cantor Theorem and $N, 2^N$ are infinite \hspace{1cm} n

5. Set $A$ is uncountable iff $R \subseteq A$ ($R$ is the set of real numbers).
   \textbf{Justify:} $R, 2^R$ are both uncountable and $R$ is not a subset of $2^R$ ($R \not\subseteq 2^R$) but $R \in 2^R$. \hspace{1cm} n

6. Let $A \neq \emptyset$ such that there are exactly 25 partitions of $A$. It is possible to define 20 equivalence relations on $A$.
   \textbf{Justify:} one can define up to 25 (as many as partitions) of equivalence classes \hspace{1cm} y

7. There is a relation that is equivalence and order at the same time.
   \textbf{Justify:} equality relation \hspace{1cm} y

8. Let $A = \{n \in N : n^2 + 1 \leq 15\}$. It is possible to define 8 alphabets $\Sigma \subseteq A$.
   \textbf{Justify:} $A$ has 4 elements, so we have $2^4 > 8$ subsets \hspace{1cm} y

9. There is exactly as many languages over alphabet $\Sigma = \{a\}$ as real numbers.
   \textbf{Justify:} $|\Sigma^*| = \aleph_0, |2^{\Sigma^*}| = |R| = C.$ \hspace{1cm} y

10. Let $\Sigma = \{a, b\}$. There are more than 20 words of length 4 over $\Sigma$.
    \textbf{Justify:} There are exactly $2^4 = 16$ words of length 4 over $\Sigma$ and 16 < 20. \hspace{1cm} n

11. $L^* = \{w_1...w_n : w_i \in L, i = 1, 2,..n, n \geq 1\}$. \hspace{1cm} n
    \textbf{Justify:} $n \geq 0$.
    \hspace{1cm} $L^+ = LL^*$.
    \textbf{Justify:} the problem is only with cases $e \in L$ or $e \not\in L$. When $e \in L$, then $e \in L^+$, and always $e \in L^*$, hence $e \in LL^*$.
    When $e \not\in L$, then $e \not\in L^+$, and always $e \in L^*$, hence $e \in LL^*$ and $L^+ \not= LL^*$ \hspace{1cm} n

12. $L^+ = L^* - \{e\}$. \hspace{1cm} n
    \textbf{Justify:} only when $e \not\in L$. When $e \in L$ we get that $e \in L^+$ and $e \not\in L^* - \{e\}$. 

13. If $L = \{w \in \{0, 1\}^*: w \text{ has an unequal number of } 0\text{'s and } 1\text{'s }\}$, then $L^* = \{0, 1\}^*$.

Justify: $1 \in L$, $0 \in L$ so $\{0, 1\} \subseteq L \subseteq \Sigma^*$, hence $\{0, 1\}^* \subseteq L^* \subseteq (\Sigma^*)^* = \Sigma^* = \{0, 1\}^*$ and $L^* = \{0, 1\}^*$.

14. For any languages $L_1$, $L_2$, $(L_1 \cup L_2) \cap L_1 = L_1$.

Justify: languages are sets and $(A \cup B) \cap A = A$.

15. For any languages $L_1$, $L_2$,

$$L_1^* = L_2^* \text{ if } L_1 = L_2$$

Justify: Consider $L_1 = \{a, e\}, L_2 = \{a\}$. Obviously, $L_1 \neq L_2$ and $L_1^* = L_2^*$.

16. For any languages $L_1$, $L_2$, $(L_1 \cup L_2)^* = L_1^*$.

Justify: languages are sets so it is true only when $L_1 \subseteq L_2$.

17. $((\emptyset^*) \cap a) \cup b^*$ describes a language with only one element.

Justify: $\emptyset \cup b^* = b^*$, $b^* \cap \{e\} = \{e\}$

18. $((\emptyset^* \cap a) \cup b^*) \cap a^*$ is a finite regular language.

Justify: $b^* \cap a^* = \{e\} = \emptyset^*$

19. $\{\{a\} \cup \{e\}\} \cap \{ab\}^*$ is a finite regular language.

Justify: $\{\{a\} \cup \{e\}\} \cap \{ab\}^* = \{a, e\} \cap \{ab\}^* = \{e\} = \emptyset^*$

20. Any regular language has a finite description.

Justify: by definition $L = \mathcal{L}(r)$ and $r$ is a finite string.

21. Any finite language is regular.

Justify: $L = \{w_1\} \cup \ldots \cup \{w_1\}$ and $\{w_1\}$ has a finite description $w_i$

22. Every deterministic automata is also non-deterministic.

Justify: $K \times \Sigma \subseteq K \times \Sigma \cup \{e\} \subseteq \Sigma^*$ and any function is a relation

The set of all configurations of any non-deterministic state automata is always non-empty.

Justify: $K \neq \emptyset$, because $s \in K$. If $\Sigma = \emptyset$ the set of all configuration of non-deterministic automata (book definition) is a subset of $K \times \emptyset \cup \{e\} \neq \emptyset$ as it always contains $(s, e)$. For the lecture definition, the set of all configuration is a subset of $K \times \Sigma^*$ and always $e \in \Sigma^*$ hence always $(s, e) \in K \times \Sigma^*$

23. Let $M$ be a finite state automaton, $L(M) = \{w \in \Sigma^*: (q, w) \xrightarrow{s,M} (s, e)\}$.

Justify: $L(M) = \{w \in \Sigma^*: \exists q \in F((s, w) \xrightarrow{s,M} (q, e))\}$

24. For any automata $M$, $L(M) \neq \emptyset$.

Justify: if $F = \emptyset$, $L(M) = \emptyset$

25. $L(M_1) = L(M_2)$ iff $M_1$, $M_2$ are deterministic.

Justify: Let $M_1$ be an automata over $\{a, b\}$ with with $\Delta = \{(q_0, ab, q_0)\}$, $F = \{q_0\}$, $s = q_0$ and let $M_2$ be an automata over $\{a, b\}$ with with $\Delta = \{(q_0, ab, q_0), (q_0, e, q_1)\}$, $F = \{q_1\}$, $s = q_0$.

$L(M_1) = L(M_2) = (ab)^*$ and both are non-deterministic
26. DFA and NDFA compute the same class of languages.
    Justify: basic theorem

27. Let $M_1$ be a deterministic, $M_2$ be a nondeterministic FA, $L_1 = L(M_1)$ and $L_2 = L(M_2)$ then there is a deterministic automaton $M$ such that $L(M) = (L^* \cup (L_1 - L_2)^*)L_1$
    Justify: the class of finite automata is closed under $\ast, \cup, -, \cap$

TWO DEFINITIONS OF NON DETERMINISTIC AUTOMATA

BOOK DEFINITION: $M = (K, \Sigma, \Delta, s, F)$ is non-deterministic when

$$\Delta \subseteq K \times (\Sigma \cup \{e\}) \times K$$

Observe that $\Delta$ is always finite because $K, \Sigma$ are finite sets.

LECTURE DEFINITION: $M = (K, \Sigma, \Delta, s, F)$ is non-deterministic when $\Delta$ is finite and

$$\Delta \subseteq K \times \Sigma^* \times K$$

Observe that we have to say in this case that $\Delta$ is finite because $\Sigma^*$ is infinite.

SOLVING PROBLEMS you can use any of these definitions.

2 Problems
In Problems 1-5 solutions you have to DRAW DIAGRAMS; do not need to LIST the Components

Problem 1 Let $L$ be a language defines as follows

$$L = \{w \in \{a, b\}^*: \text{every } a \text{ is either immediately proceeded or followed by } b\}.$$  

1. Describe a regular expression $r$ such that $L(r) = L$ (Meaning of $r$ is $L$).

Solution $L = (b \cup ab \cup ba \cup bab)^*$

2. Construct a finite state automata $M$, such that $L(M) = L$.

Solution

Components of $M$ are:

$$K = \{s\}, \{a, b\}, \ s, \ F = \{s\},$$

$$\Delta = \{(s, b, s), (s, ab, s), (s, ba, s), (s, bab, s)\}$$

Some elements of $L(M)$ are: $b, bb, baab, abab, abbbba, bbbabbbbabb$

Problem 2

1. Let $M = (K, \Sigma, \delta, s, F)$ be a deterministic finite automaton. Under exactly what conditions $e \in L(M)$?

Solution

$$e \in L(M) \text{ iff } s \in F.$$
2. Let \( M = (K, \Sigma, \Delta, s, F) \) be a non-deterministic finite automaton. Under exactly what conditions \( \epsilon \in L(M) \)?

**Solution** Now we have two possibilities: \( s \in F \) (computation of length 0) or there is a computation of length > 0 from \( (s, \epsilon) \) to \( (q, \epsilon) \) for \( q \in F \) when \( s \notin F \).

**Problem 3** Let

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

for \( K = \{q_0, q_1, q_2, q_3\} \), \( s = q_0 \)
\( \Sigma = \{a, b\} \), \( F = \{q_1, q_2, q_3\} \) and
\[\Delta = \{(q_0, a, q_1), (q_0, b, q_3), (q_1, a, q_2), (q_1, b, q_1), (q_3, a, q_3), (q_3, b, q_2)\}\]

1. List some elements of \( L(M) \).

**Solution** \( a, b, aa, bb, aba, abba \)

2. Write a regular expression for the language accepted by \( M \). Simplify the solution.

**Solution**
\[
L(M) = ab^* \cup ab^*a \cup ba^* \cup ba^*b = ab^*(e \cup a) \cup ba^*(e \cup b).
\]

3. Define a deterministic \( M' \) such that \( M \approx M' \), i.e. \( L(M) = L(M') \).

**Solution** We complete \( M \) do a deterministic \( M' \) by adding a TRAP state \( q_4 \) and put
\[\Delta' = \delta = \Delta \cup \{(q_2, a, q_4), (q_2, b, q_4), (q_4, a, q_4), (q_4, b, q_4)\}\]

**Justify** why \( M \approx M' \).

**Solution** \( q_4 \) is a trap state, it does not influence \( L(M) \).

**Problem 4** Let \( M \) be defined as follows

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

for \( K = \{q_0, q_1, q_2, q_3\} \), \( s = q_0 \)
\( \Sigma = \{a, b, c\} \), \( F = \{q_0, q_2, q_3\} \) and
\[\Delta = \{(q_0, abc, q_0), (q_0, a, q_1), (q_0, c, q_3), (q_1, bc, q_1), (q_1, b, q_2), (q_2, a, q_2), (q_2, b, q_3), (q_3, a, q_3)\}\]

Find the regular expression describing the \( L(M) \). Explain your steps. Does \( \epsilon \in L(M) \)?

**Solution**
\[
L = \alpha_1 \cup \alpha_2 \cup \alpha_3 \cup \alpha_4
\]

where
\[
\alpha_1 = (abc)^* - \text{loop on } q_0,
\alpha_2 = (abc)^*a(bc)^*ba^* - \text{path from } q_0 \text{ to } q_2,
\alpha_3 = (abc)^*a(bc)^*ba^*ba^* - \text{path from } q_0 \text{ to } q_3 \text{ via } q_2,
\alpha_3 = (abc)^*a^* - \text{path from } q_0 \text{ directly to } q_3
\]

This is not the only solution.
Observe that \( \epsilon \in L \) as \( q_0 \in F \) and also \( (q_0, \epsilon, q_3) \in \Delta \) and \( q_3 \in F \).
This is not the only solution.
We apply the "stretching" technique to \( M \) and the new \( M' \) is defined by the BOOK definition.

**Solution**

Solution We apply the "stretching" technique to \( M \) and the new \( M' \) is as follows.

\[
M' = (K \cup \{p_1, p_2, p_3\}, \Sigma, s = q_0, \Delta', F' = F)
\]

for \( K = \{q_0, q_1, q_2\}, \ s = q_0 \)
\( \Sigma = \{a, b\}, \ F = \{q_0, q_2, q_3\} \) and
\( \Delta' = \{(q_0, a, q_1), (q_0, c, q_3), (q_1, b, q_2), (q_2, a, q_2), (q_2, b, q_3), (q_3, a, q_3)\} \)
\( \bigcup\{(q_0, a, p_1), (p_1, b, p_2), (p_2, c, q_0), (q_1, b, p_3), (p_3, b, q_1)\} \).

**Problem 5** For \( M \) defined as follows

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

for \( K = \{q_0, q_1, q_2, q_3\}, \ s = q_0 \)
\( \Sigma = \{a, b\}, \ F = \{q_2\} \) and
\( \Delta = \{(q_0, a, q_3), (q_0, c, q_3), (q_1, b, q_1), (q_0, e, q_1), (q_1, a, q_2), (q_2, b, q_3), (q_2, e, q_3)\} \)

Write 2 steps of the general method of transformation the NDFA \( M \) defined above into an equivalent DFA \( M' \).

**Step 1:** Evaluate \( \delta(E(q_0), a) \) and \( \delta(E(q_0), b) \).

**Step 2:** Evaluate \( \delta \) on all states that result from step 1.

**Reminder:** \( E(q) = \{p \in K : (q, e) \underset{M}{\rightarrow} (p, e)\} \) and
\[
\delta(Q, \sigma) = \bigcup\{E(p) : \exists q \in Q, (q, \sigma, p) \in \Delta\}.
\]

**Solution Step 1:** First we need to evaluate \( E(q) \), for all \( q \in K \).

\[
E(q_0) = \{ q_0, q_1, q_3 \} = S, \ E(q_1) = \{ q_1 \}, \ E(q_2) = \{ q_2, q_3 \} \in F, \ E(q_3) = \{ q_3 \}
\]

\[
\delta(E(q_0), a) = \delta(\{q_0, q_1, q_3\}, a) = E(q_3) \cup E(q_2) \cup \emptyset = \{q_2, q_3\} \in F
\]
\[
\delta(E(q_0), b) = \delta(\{q_0, q_1, q_3\}, b) = E(q_1) \cup \emptyset \cup \emptyset = \{q_1\}
\]

**Solution Step 2:**

\[
\delta(\{q_2, q_3\}, a) = \emptyset \cup \emptyset = \emptyset
\]
\[
\delta(\{q_2, q_3\}, b) = E(q_3) \cup \emptyset = \{q_3\}
\]
\[
\delta(\{q_1\}, a) = E(q_2) = \{q_2, q_3\} \in F
\]
\[
\delta(\{q_1\}, b) = \emptyset
\]