1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time.

   (i) F Every subset of a regular language is regular.
   Every language is the subset of some regular language.

   (ii) F The complement of a CFL is always a CFL.

   (iii) T The class of context-free languages is closed under union.

   (iv) F The class of context-free languages is closed under intersection.

   (v) T The set of binary numerals for multiples of 23 is regular.
   The set of numerals (of any base, not just 2) for the members of any arithmetic sequence is a regular language.

   (vi) T The set of binary numerals for prime numbers is in P–TIME.
   The base doesn’t matter, as long as its at least 2. (This excludes unary (caveman) numerals.) This is a fact that was proven only recently, by Manindra Agrawal, N. Kayal, and N. Saxena, and published in 2004, but I believe the result leaked out earlier. Before then, the correct answer to this question would have been O.

   (vii) F Every PDA is equivalent to some DPDA.
   See Problem 2 below.

   (viii) T Every language is countable.
   There are only countably many strings over any given alphabet.

   (ix) T The set of languages over the binary alphabet is countable.
   Let Σ be any alphabet. Then Σ∗ is the set of all strings over Σ, which is infinite and countable. But Cantor proved that, for any set S, the set 2S has more elements than S. The set of all languages over any alphabet Σ is 2Σ∗, which is then not countable.

   (x) O P = NP.
   Solve this and you will be really famous.

   (xi) T The complement of any P–TIME language is P–TIME.
   If a machine decides a language L, it (by switching the 0 and 1 outputs) decides the complement of L in the same number of steps. This rule does not hold for acceptance.

   (xii) O The complement of any NP language is NP.
   If P = NP, then the answer is true, otherwise it is false, so it’s open.
The complement of any decidable language is decidable.

If a machine decides a language $L$, it (by switching the 0 and 1 outputs) decides the complement of $L$.

The complement of any undecidable language is undecidable. Let $L'$ be the complement of $L$. If $L$ is undecidable and $L'$ is decidable, this violates the answer to the previous question.

2. Give an unambiguous CFG which generates a language not accepted by any DPDA.

There are many correct answers, but I believe the one given here is simplest.

$$S \rightarrow aSa$$
$$S \rightarrow bSb$$
$$S \rightarrow \lambda$$

3. Suppose $L$ is a problem such that you can check any suggested solution in polynomial time. Which one of these statements is certainly true?

(i) $L$ is $\mathcal{P}$.

(ii) $L$ is $\mathcal{NP}$.

(iii) $L$ is $\mathcal{NP}$-complete.

Only the second one. If $\mathcal{P} = \mathcal{NP}$, all three statements are equivalent, hence true.

4. $L$ be the language of all binary strings in which each 0 is followed by 1. Draw a DFA which accepts $L$.

5. Consider the NFA $M$ pictured below. Construct a minimal DFA equivalent to $M$.

\begin{center}
\begin{tikzpicture}[->,>=stealth,shorten >=1pt,auto,node distance=2.8cm,semithick]
  \node[state] (A) {$0$};
  \node[state] (B) [right of=A] {$1$};
  \node[state] (C) [below of=A] {$2$};

  \path
    (A) edge [loop above] node {1} ()
    (A) edge node {0} (B)
    (B) edge [loop below] node {1} ()
    (B) edge node {0} (A)
    (A) edge node {a,b} (C)
    (C) edge node {a} (A)
    (C) edge node {b} (B)
    (C) edge [loop below] node {\lambda} ()
  ;
\end{tikzpicture}
\end{center}
6. Let $G_1$ be the CF grammar given below. Prove that $G_1$ is ambiguous by giving two different parse trees for the string $iiwaea$.

$$
\begin{align*}
S & \rightarrow a \\
S & \rightarrow wS \\
S & \rightarrow iS \\
S & \rightarrow iSeS \\
\end{align*}
$$

1. $S \rightarrow a$
2. $S \rightarrow wS$
3. $S \rightarrow iS$
4. $S \rightarrow iSeS$

7. The CNF grammar $G_2$, given below, is equivalent to the grammar $G_1$ given in Problem 6.

Use the CYK algorithm to prove that $iiwaea$ is generated by $G_2$.

$$
\begin{align*}
S & \rightarrow a \\
S & \rightarrow WS \\
W & \rightarrow w \\
S & \rightarrow IS \\
S & \rightarrow AB \\
A & \rightarrow IS \\
B & \rightarrow ES \\
E & \rightarrow e \\
\end{align*}
$$

8. Give a grammar, with at most 3 variables, for the language accepted by the following NFA.

$$
\begin{align*}
S & \rightarrow aS|bS|aA|cA \\
A & \rightarrow aB|bB|cB \\
B & \rightarrow \lambda \\
\end{align*}
$$

You actually need only two variables. Do you see how?
9. Give a regular expression for the language accepted by the following NFA

\[(a + b)(b(a + b) + a + ba^*b)^*\]

10. Let \( L \) be the language consisting of all strings over \( \{a, b\} \) which have equal numbers of each symbol. Give a CFG for \( L \).

There are many solutions. The grammar below is, I believe, the simplest. It is ambiguous, but \( L \) does have an unambiguous CFG.

\[
S \rightarrow aSbS \\
S \rightarrow bSaS \\
S \rightarrow \lambda
\]

11. Design a DPDA which accepts the language described in Problem 10.