Additional Practice Problems

1. True or False. If the question is currently open, write “O” or “Open.”
   (i) _______ Every subset of a regular language is regular.
   (ii) _______ There exists some proposition which is true but which has no proof.
   (iii) _______ The set of all binary numerals for prime numbers is a regular language.
   (iv) _______ For any deterministic finite automaton, there is always a unique minimal non-deterministic finite automaton equivalent to it.
   (v) _______ It can always be decided, in polynomial time, whether two given regular expressions are equivalent.
   (vi) _______ The halting problem is decidable.
   (vii) _______ The complement, over the binary alphabet, of every regular binary language is regular.
   (viii) _______ The union of any two regular languages is regular.
   (ix) _______ The Kleene closure of any regular language is regular.
   (x) If \( w \) is a string, let \( w^R \) denote the reverse of \( w \). For example, if \( w = \text{said} \) then \( w^R = \text{di}a\text{s} \). If \( L \) is a language \( L^R = \{w^R : w \in L\} \).
        _______ If \( L \) is regular, then \( L^R \) is regular.
   (xi) A string \( w \) is called a palindrome if \( w^R = w \). For example, \( \text{bob} \) is a palindrome.
        _______ The set of all palindromes over the binary alphabet is a regular language.
   (xii) _______ We say a binary string \( w \) is balanced if \( w \) has the same number of 1’s as 0’s.
        _______ The set of balanced binary strings is a regular language.
   (xiii) _______ The intersection of any two regular languages is regular.
   (xiv) _______ The language of all regular expressions over the binary alphabet is a regular language.
   (xv) _______ There is no computer program that decides whether two given C++ programs are equivalent.
   (xvi) _______ If anyone ever proves that \( P = \text{NP} \), then all one-way encoding systems will be insecure.
   (xvii) _______ The set of all binary numerals for prime numbers is a \( \mathcal{P} \)-time language.

2. Draw the state diagram for a minimal DFA that accepts the language described by the regular expression \( a^*b^* \).

3. Give a regular grammar for the language accepted by the NFA shown in Figure 1.
4. Construct a minimal DFA equivalent to the NFA shown in Figure 1.

5. Let $L$ be the language of all binary numerals for positive integers equivalent to 2 modulo 3. Thus, for example, the binary numerals for 2, 5, 8, 11, 14, 17 . . . are in $L$. We allow a binary numeral to have leading zeros; thus (for example) 001110 $\in L$, since it is a binary numeral for 14. Draw a minimal DFA which accepts $L$.

6. Give a grammar (not a regular grammar) for the language $L = \{a^n b^n : n \geq 0\}$

7. Name a class of machines that accepts the class of regular languages.
8. Give a regular grammar for the language accepted by the NFA shown below, and draw a state diagram for a minimal equivalent DFA.