6.045 Problem Set 1

BY CHENLIN WANG

chenlin520.com

1. Recall the protocol by which Alice commits herself to a bit $x \in \{0, 1\}$ without revealing x to Bob. Namely, Alice first chooses two large random prime numbers P and Q, one of which ends in 7 if and only if $x = 1$. She then computes their product $N = \mathrm{PQ}$ and sends N to Bob,but keeps the factors P and Q to herself. To reveal the value of x later, Alice sends P and Q to Bob, whereupon Bob checks that:

   i. P and Q encode the claimed value x,

   ii. P and Q are indeed prime numbers

   iii. $\mathrm{PQ} = N$

   Suppose Bob forgets to check that P and Q are prime. Dose the protocol still work correctly. and if not, what can go wrong?

   Answer. Alice could cheat if Bob does not check the wheth P and Q are prime. For example, Alice could send Bob the product of three large prime numbers, which end in 3, 7 and 7. If Bob sends 1, then Alice could multiply the first two numbers and then sends back two numbers ending in 1 and 7. If Bob sends 0, then Alice could multiply the last two numbers and then sends back two numbers ending in 1 and 9. Thus, either ways, Alice could win.

2. Recall Euclid's algorithm for computing $\gcd(a, b)$ for positive integers $a \geq b$, which is given by the following recursive pseudocode:

   ```
   if b divides a then return b
   else return gcd(b, a(mod b))
   ```

   Show that, if initialized on n-bit integers $a \geq b$, Euclid's algorithm halts afiter at most $2n$ iterations.

   Proof. Suppose that a and b are n and m bits long ($n \geq m$). Thus, if a is not divided by b, then the next divisor will has at most m bits if $m < n$ or $n - 1$ bits if $m = n$. Thus, when it comes down to 1 bits, it takes at most $2n$ steps. □

3. Show that any language L containing only finitely many strings is regular.

Proof. We shall prove it by the following:

i. Any language that contains only single string is regular.

   The deterministic finite automata regarding a single-string language is simple to construct. Take $|s|+2$ states: one for initial state and one for dead state, and others are linked with the letters from the string in order. The only accepted state is the the last state on the line. Any other letters not in the order will lead to the dead states.

ii. Any union of two regular languages is regular.

   Thus, we could break the finitely many strings into single-string language and union them all together to construct the language we hope to have.

□

4. Show that, if L_1 and L_2 are any two regular languages, then $L_1 \cap L_2$ is also a regular language.

Proof. Suppose the L_1 and L_2 could be writen as deterministic finite automatas as D_1: $(Q_1, \Sigma_1, \delta_1, p_1^0, F_1)$ and D_2: $(Q_2, \Sigma_2, \delta_2, p_2^0, F_2)$. We now construct a new DFA D_3 as:

$$\begin{aligned} Q_3 &= Q_1 \times Q_2 \\ \Sigma_3 &= \Sigma_1 \times \Sigma_2 \\ \delta_3((p_1, p_2), \omega) &= (\delta_1(p_1, \omega), \delta_2(p_2, \omega)) \\ p_3^0 &= (p_1^0, p_2^0) \\ F_3 &= \{(f_1, f_2) |\, f_1 \in F_1,\, f_2 \in F_2\} \end{aligned}$$

We now show that it is the machine that describe the language $L_1 \cap L_2$. If any string s that is accepted by the above D_3, it is accepted by both D_1 and D_2; for any string that is in $L_1 \cap L_2$, it should be accepted by both D_1 and D_2, thus it should also be accepted by D_3. Therefore complete the proof. □

5. Let $L = \{x \in \{a, b\}^* : x \text{ does not contain two consecutive } b's\}$. Write regular expression for L.

Answer. $(a^* | (\mathrm{ba})^*)^*(b|\epsilon)$

6. Let $L \subseteq \{a, b\}^*$ be the language consisting all *palindromes*: that is, strings like *abba* that are the same backwards and forwards. Using the pigeonhole principle, show that L is not regular.

Proof. We could prove it by controdiction. Suppose that there is a DFA with n states that characterise the *palindromes*. So we let use the *palindromes* in the form of $a^i b a^i$. As i could go beyond n, the *pigeonhole principle* states that there exists $j, k \geqslant n$ so that a^j and a^k both stays at the same state. And as the $a^j b a^j$ and $a^k b a^k$ are both accepted by the DFA, the string $a^j b a^k$ is also accepted by the DFA. Thus we have the controdiction. $\square$

7. Concatenation of regular languages

a) Let $L \subseteq \{a, b, c\}^*$ be the language consisting of all strings ω that can be expressed as $\omega_1 \circ \omega_2$, where ω_1 contains even number of b's, ω_2 contains a number of c's that is divisible by 3, and $\circ$ denotes string concatenation. Show that L is regular, by constructing an NDFA that recognizes L.

Answer.

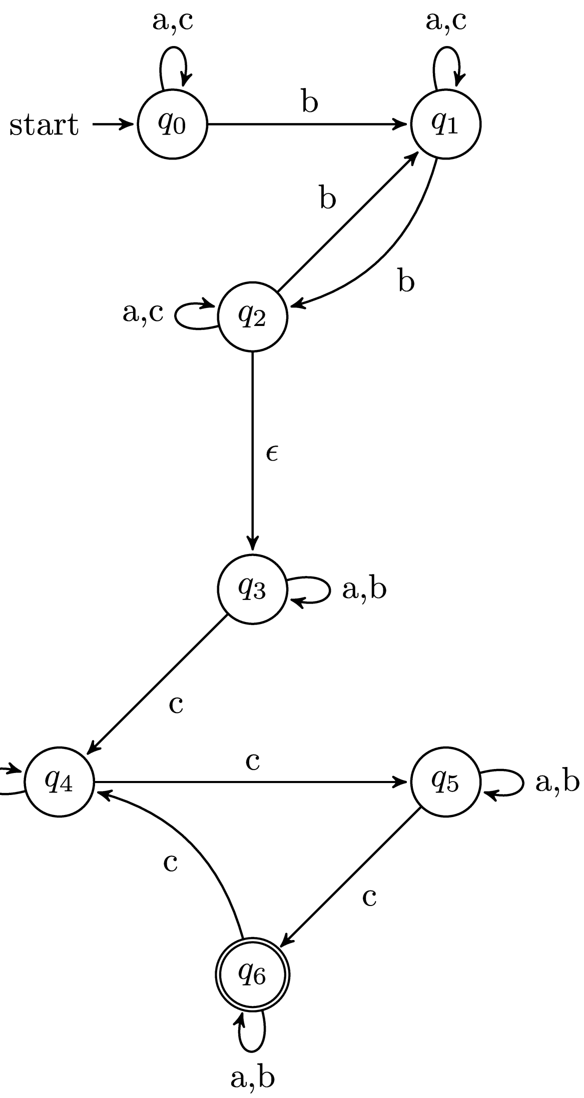

Figure 1. Nondeterministic Finite Automata

b) Let $L \subseteq \{a, b\}^*$ be the language consisting of all strings ω that can be expressed as $\omega_1 \circ \omega_2$, where ω_1 contains an even number of b's and ω_2 contains a number of b's that is divisible by 3. Construct a DFA that recognizes L.

Answer.

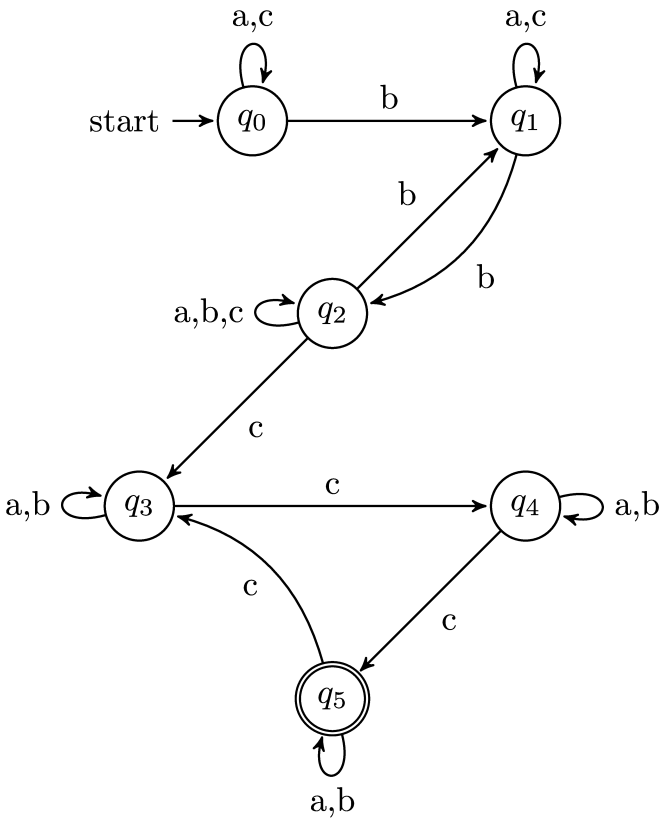

Figure 2. Deterministic Finite Automata

c) Generalize part a) to show that, if L_1 and L_2 are *any* two regular languages, then

$$L = \{\omega_1 \circ \omega_2 | \omega_1 \in L_1, \omega_2 \in L_2\}$$

is also a regular language.

Answer. We could just construct a NFA from the two NFAs of the substrings ω_1 and ω_2. Start from the two NFAs, we just simply add ϵs from the final states of the ω_1 to the initial states of ω_2, thus, finishing the construction.