Assignment 1 Solutions

1. The following problems each have a short answer, perhaps just a few sentences and maybe an equation or two. Try to make your answers clear and to the point—and choose the simplest answer whenever possible.

(a) Suppose that \( A \) and \( B \) are nonempty sets for which \( B \subseteq A \), and assume that there exists an onto function \( f : \mathbb{N} \rightarrow A \). Prove that there exists an onto function of the form \( g : \mathbb{N} \rightarrow B \). What do you conclude from this fact about any subset of a countable set?

(b) Prove that there are countably many finite languages over any alphabet \( \Sigma \).

(c) For any alphabet \( \Sigma \) and any language \( A \subseteq \Sigma^* \) we define \( \text{Prefix}(A) \) to be the language containing all prefixes of strings in \( A \):

\[
\text{Prefix}(A) = \{ x \in \Sigma^* : \text{there exists } v \in \Sigma^* \text{ such that } xv \in A \}.
\]

(This operation is discussed in the notes for Lecture 6, although you do not need to make use of anything covered in that lecture to answer this question.) Give an example of a nonregular language \( A \subseteq \{0, 1\}^* \) for which \( \text{Prefix}(A) \) is regular.

Solution. (a) Let \( y \in B \) be any fixed element in \( B \); such an element exists by the assumption that \( B \) is nonempty. Define \( g : \mathbb{N} \rightarrow B \) as follows:

\[
g(n) = \begin{cases} 
  f(n) & \text{if } f(n) \in B \\
  y & \text{if } f(n) \notin B.
\end{cases}
\]

It is evident that \( g(n) \in B \) for every \( n \in \mathbb{N} \). For any element \( x \in B \), one has \( x \in A \) by the assumption that \( B \subseteq A \), and therefore \( x = f(n) \) for some \( n \in \mathbb{N} \) by the assumption that \( f \) is onto. As it must hold that \( g(n) = f(n) = x \) in this situation, it follows that \( g \) is onto, as required. From this fact we conclude that every nonempty subset of a countable set is also countable.

(b) Here are two possible solutions to this problem (but of course you were only expected to provide one answer):

The first solution uses the fact (discussed in class) that there are countably many regular languages over any alphabet \( \Sigma \). Because every finite language is regular (e.g., you can easily obtain a regular expression for a given finite language), one has that the set of finite languages is a subset of the set of regular languages—so the fact that there are countably many finite languages over \( \Sigma \) follows.

The second solution argues directly that there are countably many finite languages over an alphabet \( \Sigma \). One way to argue this is to observe that for every \( n \in \mathbb{N} \), there are only finitely many languages over \( \Sigma \) that have at most \( n \) strings and are such that each string in the language has length at most \( n \). (Note that we are using \( n \) to limit two separate aspects of these languages: the number of elements and the length of each element. This sort of trick will often be used later in the course.) If we list all of the languages obtained in this way for increasing values of \( n \), we obtain a countable sequence of finite languages. As every finite language appears somewhere in the sequence, we conclude that there are countably many finite languages.
There are many such examples. One is the language of all palindromes,
\[ A = \{ w \in \{0,1\}^* : w = w^R \}, \]
which was proved to be nonregular in Lecture 5. It is not hard to see that \( \text{Prefix}(A) = \{0,1\}^* \) in this case, which is regular.

2. Suppose \( A \subseteq \{0,1\}^* \) is a given regular language, over the binary alphabet.
   (a) Define a language \( B \subseteq \{0,1\}^* \) as follows:
   \[ B = \{ u \in \{0,1\}^* : u0 \in A \}. \]
   In words, \( B \) is the language containing exactly those strings that can be obtained by choosing a string in \( A \) that ends with 0 and then deleting that ending 0 (leaving the rest of the string). Prove that \( B \) is regular.
   (b) Define a language \( C \subseteq \{0,1\}^* \) as follows:
   \[ C = \{ \sigma u : u \in \{0,1\}^*, \sigma \in \{0,1\}, \text{ and } u\sigma \in A \}. \]
   In words, \( C \) is the language containing exactly those strings that can be obtained by choosing a nonempty string in \( A \), then moving the last symbol of that string to the beginning of the string. Prove that \( C \) is regular.
   Hint: the fact proved by a correct answer to part (a) can be helpful when solving this problem.

Solution. (a) Let \( \Sigma = \{0,1\} \) and let \( M = (Q, \Sigma, \delta, q_0, F) \) be a DFA such that \( L(M) = A \). Such a DFA exists under the assumption that \( A \) is regular. Now define
   \[ G = \{ q \in Q : \delta(q,0) \in F \}, \]
   and define a new DFA \( K = (Q, \Sigma, \delta, q_0, G) \). The DFA \( K \) is the same as \( M \) except that we’ve changed its accept states: a state \( q \in Q \) is an accept state of \( K \) if and only if reading the symbol 0 while in this state would lead to an accept state of \( M \). A string \( u \) is accepted by \( K \) if and only if \( u0 \) is accepted by \( M \), and therefore \( L(K) = B \). From this we conclude that \( B \) is regular.
   (b) First let us define two languages:
   \[ B_0 = \{ u \in \{0,1\}^* : u0 \in A \}, \]
   \[ B_1 = \{ u \in \{0,1\}^* : u1 \in A \}. \]
   We have already proved \( B_0 \) is regular, because \( B_0 \) is the same as \( B \) from part (a). The language \( B_1 \) is also regular, based on exactly the same argument (but replacing the symbol 0 by 1). Of course the finite languages \( \{0\} \) and \( \{1\} \) are also regular, and because the regular languages are closed under concatenation and union, we have that \( \{0\}B_0 \cup \{1\}B_1 \) is regular. Because
   \[ C = \{0\}B_0 \cup \{1\}B_1, \]
   we therefore have that \( C \) is regular, as required.
3. Let $\Sigma$ be any alphabet and let $A, B \subseteq \Sigma^*$ be given regular languages.

(a) Define a language $C \subseteq \Sigma^*$ as follows:

$$C = \{uxv : u, x, v \in \Sigma^*, uv \in A, \text{ and } x \in B\}.$$ 

In words, $C$ is the language of all strings that can be obtained by first choosing a string from $A$ and then inserting anywhere into that string any one string chosen from $B$. Prove that $C$ is regular.

(b) Define a language $D \subseteq \Sigma^*$ as follows:

$$D = \{uv : u, x, v \in \Sigma^*, uv \in A, \text{ and } x \in B\}.$$ 

In words, $D$ is the language of all strings that can be obtained by first choosing a string from $A$ and then removing from that string any one substring that is contained in $B$. Prove that $D$ is regular.

Hint: you do not actually need to use the assumption that $B$ is regular to conclude that $D$ is regular.

**Solution.** (a) We will describe two different solutions to this part of the problem. The first solution constructs an NFA $N$ that recognizes $C$.

The idea behind this NFA construction is that it runs in 3 stages, corresponding to the substrings $u, x, \text{ and } v$ in the definition of $C$, nondeterministically guessing when to transition between the phases. To define this NFA precisely, let us first take

$$M_A = (Q, \Sigma, \delta, q_0, F) \quad \text{and} \quad M_B = (R, \Sigma, \mu, r_0, G)$$

to be DFAs such that $L(M_A) = A$ and $L(M_B) = B$. Our NFA $N$ is to be defined as follows:

$$N = (S, \Sigma, \eta, (1, q_0), H),$$

where

$$S = \{(1, q) : q \in Q\} \cup \{(2, q, r) : q \in Q, \ r \in R\} \cup \{(3, q) : q \in Q\},$$

the transition function $\eta$ is defined as

$$\eta((1, q), \sigma) = \{(1, \delta(q, \sigma))\} \quad \text{for all } q \in Q \text{ and } \sigma \in \Sigma$$

$$\eta((1, q), \varepsilon) = \{(2, q, r_0)\} \quad \text{for all } q \in Q$$

$$\eta((2, q, r), \sigma) = \{(2, q, \mu(r, \sigma))\} \quad \text{for all } q \in Q, \ r \in R, \text{ and } \sigma \in \Sigma$$

$$\eta((2, q, r), \varepsilon) = \{(3, q)\} \quad \text{for all } q \in Q \text{ and } r \in G$$

$$\eta((2, q, r), \varepsilon) = \varnothing \quad \text{for all } q \in Q \text{ and } r \in Q \setminus G$$

$$\eta((3, q), \sigma) = \{(3, \delta(q, \sigma))\} \quad \text{for all } q \in Q \text{ and } \sigma \in \Sigma$$

$$\eta((3, q), \varepsilon) = \varnothing \quad \text{for all } q \in Q$$

and the set of accept states is defined as

$$H = \{(3, q) : q \in F\}.$$ 

In the first stage, which corresponds to states of the form $(1, q)$ for $q \in Q$, the NFA $N$ simply mimics $M_A$. In the second stage, which corresponds to states of the form $(2, q, r)$ for $q \in Q$ and
r \in R$, the NFA $N$ mimics $M_B$, but throughout this entire stage it must remember the state $q \in Q$ that $M_A$ was in when the first stage ended. Finally, the third stage picks up the computation of $M_A$ where it left off, starting from the state $q \in Q$ that was stored during the second stage, with this being possible if and only if $M_B$ is in an accept state.

It is fairly straightforward to argue that $L(N) = C$. If we have a string $uxv$ for $uv \in A$ and $x \in B$, then $N$ can be made to accept by following an $\varepsilon$-transition between the first and second stages after reading $u$ and following an $\varepsilon$-transition between the second and third stages after reading $x$. On the other hand, any string accepted by $N$ can be decomposed into substrings $u, x,$ and $v$ matching the definition of $C$ by splitting the input string in the natural way according to when $\varepsilon$-transitions between the three stages were taken.

Because there exists an NFA $N$ such that $C = L(N)$, we conclude that $C$ is a regular language.

An alternative solution to part (a) uses a very useful technique that you can also find in the notes for Lecture 6 (where it is used for a different problem). First, let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA for $A$. For each pair of states $(p, q) \in Q \times Q$, define a new DFA $M_{p,q} = (Q, \Sigma, \delta, p, \{q\})$, as well as a language $A_{p,q} = L(M_{p,q})$.

The language $A_{p,q}$ contains precisely those strings that cause $M$ to move from state $p$ to state $q$. It holds that each $A_{p,q}$ is regular, as we have a DFA $M_{p,q}$ that recognizes $A_{p,q}$, as just described.

Now, let us observe that

$$C = \bigcup_{p \in Q} A_{q_0,p}BA_{p,q}.$$

This fact can be argued as follows. First, every string in $C$ takes the form $uxv$ for $uv \in A$ and $x \in B$, and by setting $p = \delta^*(q_0, u)$ and $q = \delta^*(q_0, uv) \in F$ we find that $u \in A_{q_0,p}$, $x \in B$, and $v \in A_{p,q}$, and therefore

$$C \subseteq \bigcup_{p \in Q} A_{q_0,p}BA_{p,q}.$$

On the other hand, if a string is contained in $A_{q_0,p}BA_{p,q}$ for some choice of $p \in Q$ and $q \in F$, then it must be possible to write that string as $uxv$ for $u \in A_{q_0,p}$, $x \in B$, and $v \in A_{p,q}$, and we see that this string is contained in $C$ because

$$uv \in A_{q_0,p}A_{p,q} \subseteq A_{q_0,q} \subseteq A.$$

Therefore

$$\bigcup_{p \in Q} A_{q_0,p}BA_{p,q} \subseteq C.$$

Finally, we conclude that $C$ is regular because $B$ is regular, each $A_{p,q}$ is regular, and the regular languages are closed under concatenations and finite unions.
(b) We will use a similar methodology to the second solution to part (a). (An NFA construction would also be possible.)

Let \( M = (Q, \Sigma, \delta, q_0, F) \) be a DFA for \( A \), and each pair of states \((p, q) \in Q \times Q\), define a DFA

\[
M_{p,q} = (Q, \Sigma, \delta, p, \{q\}),
\]

as well as a language

\[
A_{p,q} = L(M_{p,q}),
\]

exactly as before.

For every pair of states \((p, q) \in Q \times Q\) it is either possible or not possible to move from state \( p \) to the state \( q \) in \( M \) by reading some string in \( B \). Let us take \( T \) to be the set of all pairs for which this is possible:

\[
T = \{ (p, q) \in Q \times Q : A_{p,q} \cap B \neq \emptyset \}.
\]

Equivalently, \((p, q) \in T\) if and only if there exists \( x \in B \) such that \( \delta^*(p, x) = q \). It is important to note that it is not necessary that we describe a method for determining whether or not a pair \((p, q)\) is or is not in \( T \)—all that we need to know is that \( T \) is a well-defined set and that it is finite (which is obvious because \( Q \times Q \) is finite). Notice that we did need to use the condition that \( B \) is regular, just as the hint suggested.

To complete the proof, we observe that

\[
D = \bigcup_{(p,q) \in T} A_{q_0,p} A_{q,r},
\]

and then conclude that \( D \) is regular because the regular languages are closed under concatenations and finite unions.

If you were to claim that the above equality it is true without a proof, that would be an acceptable answer—it is pretty routine to show that it is true. For the sake of these solutions, let us take a look at how to do this in greater detail. Again we can do this by proving two inclusions.

First, if we consider any string in \( D \), we know that it must be possible to write this string as \( uv \), where there must also exist \( x \in B \) so that \( u x v \in A \). Define \( p = \delta^*(q_0, u) \) (so that \( u \in A_{q_0,p} \)), \( q = \delta^*(p, x) \), and \( r = \delta^*(q, v) \) (so that \( v \in A_{q,r} \)). It therefore holds that \( u v \in A_{q_0,p} A_{q,r} \). Moreover, we know that \((p, q) \in T\) because \( x \in B \) satisfies \( q = \delta^*(p, x) \), and \( r \in F \) because \( M_A \) accepts \( u x v \).

It follows that

\[
D \subseteq \bigcup_{(p,q) \in T} A_{q_0,p} A_{q,r}.
\]

On the other hand, suppose we have strings \( u \in A_{q_0,p} \) and \( v \in A_{q,r} \), for some choice of states \( p, q, r \in Q \) satisfying \((p, q) \in T\) and \( r \in F \). Because \((p, q) \in T\), there must exist \( x \in B \) such that \( \delta^*(p, x) = q \). It therefore holds that \( \delta^*(q_0, u x v) = r \), which implies that \( M_A \) accepts \( u x v \) because \( r \in F \). By the definition of \( D \), we therefore have that \( u v \in D \), so we have proved

\[
\bigcup_{(p,q) \in T} A_{q_0,p} A_{q,r} \subseteq D,
\]

as required.
4. Let $\Sigma = \{0\}$, and consider the nonregular language

$$A = \left\{ 0^{n^2} : n \in \mathbb{N} \right\}.$$

Suppose further that $B \subseteq \Sigma^*$ is a language for which $B \subseteq A$. Prove that $B$ is regular if and only if $B$ is finite.

**Solution.** We know that every finite language is regular, so it suffices to prove that if $B$ is infinite, then $B$ is nonregular. The proof will use the pumping lemma.

Assume toward contradiction that $B$ is regular. There must therefore exist a pumping length $m \geq 1$ for $B$. Let $n$ be any positive integer for which it holds that $n \geq m$ and

$$0^{n^2} \in B.$$

There must exist such a choice of $n$ under the assumption that $B$ is infinite—$B$ is a subset of $A$, so if there were no such choice of $n$, then $B$ would necessarily be finite.

Now, it holds that

$$\left|0^{n^2}\right| = n^2 \geq m.$$

and so the pumping lemma implies that it is possible to write

$$0^{n^2} = xyz$$

for some choice of $x, y, z \in \Sigma^*$ satisfying $y \neq \epsilon$, $|xy| \leq m$, and $xy^iz \in B$ for all $i \in \mathbb{N}$. By the first two requirements, we must have $y = 0^k$ for some choice of $k \in \{1, \ldots, m\}$. Taking $i = 2$, we find that

$$xyyz = 0^{n^2+k} \in B.$$

However, this is not possible. Because $1 \leq k \leq m \leq n$, we must have

$$n^2 < n^2 + k < n^2 + 2n + 1 = (n + 1)^2.$$

As $n^2 + k$ lies properly between the two consecutive perfect squares $n^2$ and $(n + 1)^2$, it cannot be a perfect square, in contradiction with the assumption that $B \subseteq A$. Having obtained a contradiction, we conclude that $B$ is nonregular, as required.