Lecture 6

Further discussion of regular languages

This is the last lecture to be devoted to regular languages, but we will refer back to regular languages frequently and relate them to various computational models and classes of languages throughout the course. For the most part we will use this lecture to relate some of the different concepts we have already discussed, introduce a few new concepts along the way, and go over some examples of problems concerning regular languages.

6.1 Other operations on languages

We've discussed some basic operations on languages, including the regular operations (union, concatenation, and star) and a few others (such as complementation and intersection). There are many other operations that one can consider—you could probably sit around all day thinking of increasingly obscure examples if you wanted to—but for now we'll take a look at just a few more.

Reverse

Suppose Σ is an alphabet and $w \in \Sigma^*$ is a string. The *reverse* of the string w, which we denote by w^{R}, is the string obtained by rearranging the symbols of w so that they appear in the opposite order. As we observed in the previous lecture, the reverse of a string may be defined inductively as follows:

1. If $w = \varepsilon$, then $w^{\mathrm{R}} = \varepsilon$.
2. If $w = \sigma x$ for $\sigma \in \Sigma$ and $x \in \Sigma^*$, then $w^{\mathrm{R}} = x^{\mathrm{R}}\sigma$.

Now suppose that $A \subseteq \Sigma^*$ is a language. We define the *reverse* of A, which we denote by A^{R}, to be the language obtained by taking the reverse of each element of A. That is, we define

$$A^{\mathrm{R}} = \bigl\{w^{\mathrm{R}} \,:\, w \in A\bigr\}. \tag{6.1}$$

You can check that the following identities hold that relate the reverse operation to the regular operations:

$$(A \cup B)^{\mathrm{R}} = A^{\mathrm{R}} \cup B^{\mathrm{R}}, \quad (AB)^{\mathrm{R}} = B^{\mathrm{R}} A^{\mathrm{R}}, \quad \text{and} \quad (A^*)^{\mathrm{R}} = (A^{\mathrm{R}})^*. \tag{6.2}$$

A natural question concerning the reverse of a languages is this one:

> If a language A is regular, must its reverse A^{R} also be regular?

The answer to this question is "yes." Let us state this fact as a proposition and then consider two ways to prove it.

Proposition 6.1. *Let Σ be an alphabet and let $A \subseteq \Sigma^*$ be a regular language. The language A^{R} is regular.*

First proof. There is a natural way of defining the reverse of a regular expression that mirrors the identities (6.2) above. In particular, if S is a regular expression, then its reverse regular expression can be defined inductively as follows:

1. If $S = \varnothing$ then $S^{\mathrm{R}} = \varnothing$.
2. If $S = \varepsilon$ then $S^{\mathrm{R}} = \varepsilon$.
3. If $S = \sigma$ for some choice of $\sigma \in \Sigma$, then $S^{\mathrm{R}} = \sigma$.
4. If $S = (S_1 \cup S_2)$ for regular expressions S_1 and S_2, then $S^{\mathrm{R}} = (S_1^{\mathrm{R}} \cup S_2^{\mathrm{R}})$.
5. If $S = (S_1 S_2)$ for regular expressions S_1 and S_2, then $S^{\mathrm{R}} = (S_2^{\mathrm{R}} S_1^{\mathrm{R}})$.
6. If $S = (S_1^*)$ for a regular expression S_1, then $S^{\mathrm{R}} = \bigl((S_1^{\mathrm{R}})^*\bigr)$.

It is evident that $\mathrm{L}(S^{\mathrm{R}}) = \mathrm{L}(S)^{\mathrm{R}}$; for any regular expression S, the reverse regular expression S^{R} matches the reverse of the language matched by S.

Now, under the assumption that A is regular, there must exist a regular expression S such that $\mathrm{L}(S) = A$, because every regular language is matched by some regular expression. The reverse of the regular expression S is S^{R}, which is also a valid regular expression. The language matched by any regular expression is regular, and therefore $\mathrm{L}(S^{\mathrm{R}})$ is regular. Because $\mathrm{L}(S^{\mathrm{R}}) = A^{\mathrm{R}}$, we have that A^{R} is regular, as required. □

Second proof (sketch). (We'll consider this as a proof "sketch" because it just summarizes the main idea without covering the details of why it works.) Under the assumption that A is regular, there must exist a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $\mathrm{L}(M) = A$. We can design an NFA N such that $\mathrm{L}(N) = A^{\mathrm{R}}$, thereby implying that A^{R} is regular, by effectively running M backwards in time (using the power of nondeterminism to do this because deterministic computations are generally not reversible).

Here is the natural way to define an NFA N that does what we want:

$$N = (Q \cup \{r_0\}, \Sigma, \mu, r_0, \{q_0\}), \tag{6.3}$$

where it is assumed that r_0 is not contained in Q (i.e., we are letting N have the same states as M along with a new start state r_0), and we take the transition function μ to be defined as follows:

$$\begin{aligned} \mu(r_0, \varepsilon) &= F, \qquad \mu(r_0, \sigma) = \varnothing, \\ \mu(q, \varepsilon) &= \varnothing, \qquad \mu(q, \sigma) = \{p \in Q : \delta(p, \sigma) = q\}, \end{aligned} \tag{6.4}$$

for all $q \in Q$ and $\sigma \in \Sigma$.

The way N works is to first nondeterministically guess an accepting state of M, then it reads symbols from the input and nondeterministically chooses to move to a state for which M would allow a move in the opposite direction on the same input symbol, and finally it accepts if it ends on the start state of M.

The most natural way to formally prove that $\mathrm{L}(N) = \mathrm{L}(M)^{\mathrm{R}}$ is to refer to the definitions of acceptance for N and M, and to check that a sequence of states satisfies the definition for M accepting a string w if and only if the reverse of that sequence of states satisfies the definition of acceptance for N accepting w^{R}. □

Symmetric difference

Given two sets A and B, we define the *symmetric difference* of A and B as

$$A \triangle B = (A \setminus B) \cup (B \setminus A). \tag{6.5}$$

In words, the elements of the symmetric difference $A \triangle B$ are those objects that are contained in either A or B, but not both. Figure 6.1 illustrates the symmetric difference in the form of a Venn diagram.

It is not hard to conclude that if Σ is an alphabet and $A, B \subseteq \Sigma^*$ are regular languages, then the symmetric difference $A \triangle B$ of these two languages is also regular. This is because the regular languages are closed under the operations union, intersection, and complementation, and the symmetric difference can be described in

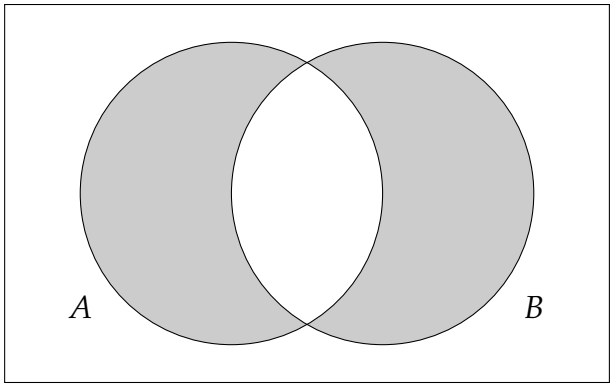

Figure 6.1: The shaded region denotes the symmetric difference $A \triangle B$ of two sets A and B.

terms of these operations. More specifically, if we assume that A and B are regular, then their complements $\overline{A}$ and $\overline{B}$ are also regular; which implies that the intersections $A \cap \overline{B}$ and $\overline{A} \cap B$ are also regular; and therefore the union $\big(A \cap \overline{B}\big) \cup \big(\overline{A} \cap B\big)$ of these two intersections is regular as well. Observing that we have

$$A \triangle B = \big(A \cap \overline{B}\big) \cup \big(\overline{A} \cap B\big), \tag{6.6}$$

we see that the symmetric difference of A and B is regular.

Prefix, suffix, and substring

Let Σ be an alphabet and let $w \in \Sigma^*$ be a string. A *prefix* of w is any string you can obtain from w by removing zero or more symbols from the right-hand side of w; a *suffix* of w is any string you can obtain by removing zero or more symbols from the left-hand side of w; and a *substring* of w is any string you can obtain by removing zero or more symbols from either or both the left-hand side and right-hand side of w. We can state these definitions more formally as follows: (i) a string $x \in \Sigma^*$ is a *prefix* of $w \in \Sigma^*$ if there exists $v \in \Sigma^*$ such that $w = xv$, (ii) a string $x \in \Sigma^*$ is a *suffix* of $w \in \Sigma^*$ if there exists $u \in \Sigma^*$ such that $w = ux$, and (iii) a string $x \in \Sigma^*$ is a *substring* of $w \in \Sigma^*$ if there exist $u, v \in \Sigma^*$ such that $w = uxv$.

For any language $A \subseteq \Sigma^*$, we will write $\text{Prefix}(A)$, $\text{Suffix}(A)$, and $\text{Substring}(A)$ to denote the languages containing all prefixes, suffixes, and substrings (respectively) that can be obtained from any choice of a string $w \in A$. That is, we define

$$\text{Prefix}(A) = \big\{x \in \Sigma^* \,:\, \text{there exists } v \in \Sigma^* \text{ such that } xv \in A\big\}, \tag{6.7}$$

$$\text{Suffix}(A) = \big\{x \in \Sigma^* \,:\, \text{there exists } u \in \Sigma^* \text{ such that } ux \in A\big\}, \tag{6.8}$$

$$\text{Substring}(A) = \big\{x \in \Sigma^* \,:\, \text{there exist } u, v \in \Sigma^* \text{ such that } uxv \in A\big\}. \tag{6.9}$$

Again we have a natural question concerning these concepts:

> If a language A is regular, must the languages $\text{Prefix}(A)$, $\text{Suffix}(A)$, and $\text{Substring}(A)$ also be regular?

The answer is "yes" as the following proposition states.

Proposition 6.2. *Let Σ be an alphabet and let $A \subseteq \Sigma^*$ be a regular language over the alphabet Σ. The languages $\text{Prefix}(A)$, $\text{Suffix}(A)$, and $\text{Substring}(A)$ are regular.*

Proof. Because A is regular, there must exist a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $\text{L}(M) = A$. Some of the states in Q are *reachable* from the start state q_0, by following zero or more transitions specified by the transition function δ.[1] We may call this set R, so that

$$R = \bigl\{q \in Q \,:\, \text{there exists } w \in \Sigma^* \text{ such that } \delta^*(q_0, w) = q\bigr\}. \tag{6.10}$$

Also, from some of the states in Q, it is *possible to reach* an accept state of M, by following zero or more transitions specified by the transition function δ. We may call this set P, so that

$$P = \bigl\{q \in Q \,:\, \text{there exist } w \in \Sigma^* \text{ and } r \in F \text{ such that } \delta^*(q, w) = r\bigr\}. \tag{6.11}$$

(See Figure 6.2 for a simple example illustrating the definitions of these sets.)

First, define a DFA $M_{\text{Prefix}} = (Q, \Sigma, \delta, q_0, P)$. In words, M_{Prefix} is the same as M except that its accept states are all of the states in M from which it is possible to reach an accept state of M. It holds that $\text{L}(M_{\text{Prefix}}) = \text{Prefix}(A)$, and therefore $\text{Prefix}(A)$ is regular.

Next, define an NFA $N_{\text{Suffix}} = (Q \cup \{r_0\}, \Sigma, \eta, r_0, F)$, where the transition function η is defined as

$$\begin{aligned} &\eta(r_0, \varepsilon) = R, \\ &\eta(q, \sigma) = \{\delta(q, \sigma)\} \quad \text{(for each } q \in Q \text{ and } \sigma \in \Sigma\text{)}, \end{aligned}$$

and η takes the value $\varnothing$ in all other cases. In words, we define N_{Suffix} from M by adding a new start state r_0, along with ε-transitions from r_0 to every reachable state in M. It holds that $\text{L}(N_{\text{Suffix}}) = \text{Suffix}(A)$, and therefore $\text{Suffix}(A)$ is regular.

Finally, define an NFA $N_{\text{Substring}} = (Q \cup \{r_0\}, \Sigma, \eta, r_0, P)$, for η defined precisely as above. It holds that $\text{L}(N_{\text{Substring}}) = \text{Substring}(A)$ and therefore $\text{Substring}(A)$ is regular. □

[1] If you were defining a DFA for some purpose, there would be no point in having states that are not reachable from the start state—but there is nothing in the definition of DFAs that forces all states to be reachable.

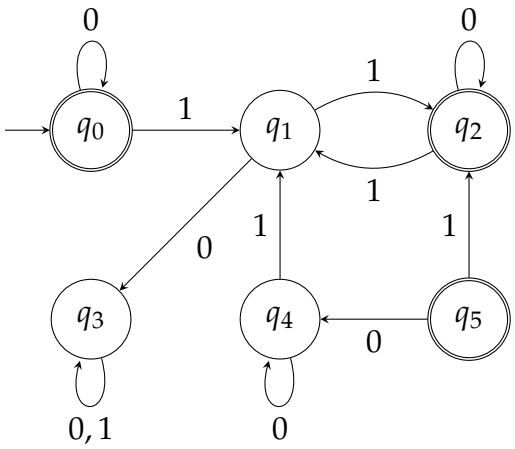

Figure 6.2: An example of a DFA M. In this case, the set R of reachable states is $R = \{q_0, q_1, q_2, q_3\}$ while the set P of states from which it is possible to reach an accepting state of M is $P = \{q_0, q_1, q_2, q_4, q_5\}$.

6.2 Example problems concerning regular languages

We will conclude with a few other examples of problems concerning regular languages along with their solutions.

Problem 6.1. Let $\Sigma = \{0, 1\}$ and let $A \subseteq \Sigma^*$ be an arbitrarily chosen regular language. Prove that the language

$$B = \{uv : u, v \in \Sigma^* \text{ and } u\sigma v \in A \text{ for some choice of } \sigma \in \Sigma\} \tag{6.12}$$

is regular. (Note that the language B can be described in intuitive terms as follows: it is the language of all strings that can be obtained by choosing a nonempty string w from A and deleting one symbol of w.)

Solution. A natural way to solve this problem is to describe an NFA for B, based on a DFA for A, which must exist by the assumption that A is regular. This will imply that B is regular, as every language recognized by an NFA is necessarily regular.

Along these lines, let us suppose that

$$M = (Q, \Sigma, \delta, q_0, F) \tag{6.13}$$

is a DFA for which it holds that $A = \mathrm{L}(M)$. Define an NFA

$$N = (P, \Sigma, \eta, p_0, G) \tag{6.14}$$

as follows. First, we will define

$$P = \{0,1\} \times Q, \tag{6.15}$$

and we will take the start state of N to be

$$p_0 = (0, q_0). \tag{6.16}$$

The accept states of N will be

$$G = \{(1,q) \,:\, q \in F\}. \tag{6.17}$$

It remains to describe the transition function η of N, which will be as follows:

(i) $\eta((0,q),\sigma) = \{(0,\delta(q,\sigma))\}$ for every $q \in Q$ and $\sigma \in \Sigma$.

(ii) $\eta((0,q),\varepsilon) = \{(1,\delta(q,0)),(1,\delta(q,1))\}$ for every $q \in Q$.

(iii) $\eta((1,q),\sigma) = \{(1,\delta(q,\sigma))\}$ for every $q \in Q$ and $\sigma \in \Sigma$.

(iv) $\eta((1,q),\varepsilon) = \varnothing$ for every $q \in Q$.

The idea behind the way that N operates is as follows. The NFA N starts in the state $(0,q_0)$ and simulates M for some number of steps. This is the effect of the transitions listed as (i) above. At some point, which is nondeterministically chosen, N follows an ε-transition from a state of the form $(0,q)$ to either the state $(1,\delta(q,0))$ or $(1,\delta(q,1))$. Intuitively speaking, N is reading nothing from its input while "hypothesizing" that M has read some symbol σ (which is either 0 or 1). This is the effect of the transitions listed as (ii). Then N simply continues simulating M on the remainder of the input string, which is the effect of the transitions listed as (iii). There are no ε-transitions leading out of the states of the form $(1,q)$, which is why we have the values for η listed as (iv).

If you think about the NFA N for a moment or two, it should become evident that it recognizes B.

Alternative Solution. Here is a somewhat different solution that may appeal to some of you. Part of its appeal is that it illustrates a method that may be useful in other cases. In this case we will also discuss a somewhat more detailed proof of correctness (partly because it happens to be a bit easier for this solution).

Again, let

$$M = (Q, \Sigma, \delta, q_0, F) \tag{6.18}$$

be a DFA for which $\mathrm{L}(M) = A$. For each choice of $p, q \in Q$, define a new DFA

$$M_{p,q} = (Q, \Sigma, \delta, p, \{q\}), \tag{6.19}$$

and let $A_{p,q} = \mathrm{L}(M_{p,q})$. In words, $A_{p,q}$ is the regular language consisting of all strings that cause M to transition to the state q when started in the state p. For any choice of p, q, and r, we must surely have $A_{p,r}A_{r,q} \subseteq A_{p,q}$. Indeed, $A_{p,r}A_{r,q}$ represents all of the strings that cause M to transition from p to q, touching r somewhere along the way.

Now consider the language

$$\bigcup_{(p,\sigma,r)\in Q\times\Sigma\times F} A_{q_0,p}A_{\delta(p,\sigma),r}. \tag{6.20}$$

This is a regular language because each $A_{p,q}$ is regular and the regular languages are closed under finite unions and concatenations. To complete the solution, let us observe that the language above is none other than B:

$$B = \bigcup_{(p,\sigma,r)\in Q\times\Sigma\times F} A_{q_0,p}A_{\delta(p,\sigma),r}. \tag{6.21}$$

To prove this equality, we do the natural thing, which is to separate it into two separate set inclusions. First let us prove that

$$B \subseteq \bigcup_{(p,\sigma,r)\in Q\times\Sigma\times F} A_{q_0,p}A_{\delta(p,\sigma),r}. \tag{6.22}$$

Every string in B takes the form uv, for some choice of $u, v \in \Sigma^*$ and $\sigma \in \Sigma$ for which $u\sigma v \in A$. Let $p \in Q$ be the unique state for which $u \in A_{q_0,p}$, which we could alternatively describe as the state of M reached from the start state on input u, and let $r \in F$ be the unique state (which is necessarily an accepting state) for which $u\sigma v \in A_{q_0,r}$. As σ causes M to transition from p to $\delta(p,\sigma)$, it follows that v must cause M to transition from $\delta(p,\sigma)$ to r, i.e., $v \in A_{\delta(p,\sigma),r}$. It therefore holds that $uv \in A_{q_0,p}A_{\delta(p,\sigma),r}$, which implies the required inclusion.

Next we will prove that

$$\bigcup_{(p,\sigma,r)\in Q\times\Sigma\times F} A_{q_0,p}A_{\delta(p,\sigma),r} \subseteq B. \tag{6.23}$$

The argument is quite similar to the other inclusion just considered. Pick any choice of $p \in Q$, $\sigma \in \Sigma$, and $r \in F$. An element of $A_{q_0,p}A_{\delta(p,\sigma),r}$ must take the form uv for $u \in A_{q_0,p}$ and $v \in A_{\delta(p,\sigma),r}$. One finds that $u\sigma v \in A_{p_0,r} \subseteq A$, and therefore $uv \in B$, as required.

Problem 6.2. Let $\Sigma = \{0,1\}$ and let $A \subseteq \Sigma^*$ be an arbitrarily chosen regular language. Prove that the language

$$C = \{vu \,:\, u, v \in \Sigma^* \text{ and } uv \in A\} \tag{6.24}$$

is regular.

Solution. Again, a natural way to solve this problem is to give an NFA for C. Let us assume

$$M = (Q, \Sigma, \delta, q_0, F) \tag{6.25}$$

is a DFA for which $\mathrm{L}(M) = A$, like we did above. This time our NFA will be slightly more complicated. In particular, let us define

$$N = (P, \Sigma, \eta, p_0, G) \tag{6.26}$$

as follows. First, we will define

$$P = \big(\{0,1\} \times Q \times Q\big) \cup \{p_0\}, \tag{6.27}$$

for p_0 being a special start state of N that is not contained in $\{0,1\} \times Q \times Q$. The accept states of N will be

$$G = \{(1, q, q) \,:\, q \in Q\}. \tag{6.28}$$

It remains to describe the transition function η of N, which will be as follows:

(i) $\eta(p_0, \varepsilon) = \{(0, q, q) \,:\, q \in Q\}$.

(ii) $\eta((0, r, q), \sigma) = \{(0, \delta(r, \sigma), q)\}$ for all $q, r \in Q$ and $\sigma \in \Sigma$.

(iii) $\eta((0, r, q), \varepsilon) = \{(1, q_0, q)\}$ for every $r \in F$ and $q \in Q$.

(iv) $\eta((1, r, q), \sigma) = \{(1, \delta(r, \sigma), q)\}$ for all $q, r \in Q$ and $\sigma \in \Sigma$.

All other values of η that have not been listed are to be understood as $\varnothing$.

You might look at this definition and have no idea how N works, so let's consider it in more detail. N starts out in the start state p_0, and the only thing it can do is to make a guess for some state of the form $(0, q, q)$ to jump to. The idea is that the 0 indicates that N is entering the first phase of its computation, in which it will read a portion of its input string corresponding to v in the definition of C. It jumps to any state q of M, but it also *remembers* which state it jumped to. Every state N ever moves to from this point on will have the form (a, r, q) for some $a \in \{0,1\}$ and $r \in Q$, but for the same q that it first jumped to; the third coordinate q represents the memory of where it first jumped, and it will never forget or change this part of its state. Intuitively speaking, the state q is a guess made by N for the state that M would be on after reading u (which N hasn't seen yet, so it's just a guess).

Then, N starts reading symbols and essentially mimicking M on those input symbols—this is the point of the transitions listed as (ii). At some point, nondeterministically chosen, N decides that it's time to move to the second phase of its computation, reading the second part of its input, which corresponds to the string

u in the definition of C. It can only make this nondeterministic move, from a state of the form $(0, r, q)$ to $(1, q_0, q)$, when r is an accepting state of M. The reason is that N only wants to accept vu when M accepts uv, so M should be in the initial state at the start of u and in an accepting state at the end of v. This is the point of the transitions listed as (iii). Finally, in the second phase of its computation, N simulates M on the second part of its input, which corresponds to the string u. It accepts only for states of the form $(1, q, q)$, because those are the states that indicate that N made the correct guess on its first step for the state that M would be in after reading u.

This is just an intuitive description, not a formal proof. It is the case, however, that $\mathrm{L}(N) = C$, as a low-level, formal proof would reveal, which implies that C is regular.

Alternative Solution. Again, there is another solution along the same lines as the alternative solution to the previous problem. This time it's actually a much easier solution. Let M be a DFA for A, precisely as above, and define $A_{p,q}$ for each $p, q \in Q$ as in the alternative solution to the previous problem. The language

$$\bigcup_{(p,r) \in Q \times F} A_{p,r} A_{q_0,p} \tag{6.29}$$

is regular, again by the closure of the regular languages under finite unions and concatenations. It therefore suffices to prove

$$C = \bigcup_{(p,r) \in Q \times F} A_{p,r} A_{q_0,p}. \tag{6.30}$$

By definition, every element of C may be expressed as vu for $u, v \in \Sigma^*$ satisfying $uv \in A$. Let $p \in Q$ and $r \in F$ be the unique states for which $u \in A_{q_0,p}$ and $uv \in A_{q_0,r}$. It follows that $v \in A_{p,r}$, and therefore $vu \in A_{p,r} A_{q_0,p}$, implying

$$C \subseteq \bigcup_{(p,r) \in Q \times F} A_{p,r} A_{q_0,p}. \tag{6.31}$$

Along similar lines, for any choice of $p \in Q$, $r \in F$, $u \in A_{q_0,p}$, and $v \in A_{p,r}$ it holds that $uv \in A_{q_0,p} A_{p,r} \subseteq A$, and therefore $vu \in C$, from which the inclusion

$$\bigcup_{(p,r) \in Q \times F} A_{p,r} A_{q_0,p} \subseteq C \tag{6.32}$$

follows.

The final problem demonstrates that closure properties holding for all regular languages may fail for nonregular languages. In particular, the nonregular languages are not closed under the regular operations.

Problem 6.3. For each of the following statements, give specific examples of languages over some alphabet Σ for which the statements are satisfied.

(a) There exist nonregular languages $A, B \subseteq \Sigma^*$ such that $A \cup B$ is regular.

(b) There exist nonregular languages $A, B \subseteq \Sigma^*$ such that AB is regular.

(c) There exists a nonregular language $A \subseteq \Sigma^*$ such that A^* is regular.

Solution. For statement (a), let us let $\Sigma = \{0\}$, let $A \subseteq \Sigma^*$ be any nonregular language whatsoever, such as $A = \{0^n : n \text{ is a perfect square}\}$, and let $B = \overline{A}$. We know that B is also nonregular (because if it were regular, then its complement would also be regular, but its complement is A which we know is nonregular). On the other hand, $A \cup B = \Sigma^*$, which is regular.

For statement (b), let us let $\Sigma = \{0\}$, and let us start by taking $C \subseteq \Sigma^*$ to be any nonregular language (such as $C = \{0^n : n \text{ is a perfect square}\}$). Then let us take

$$A = C \cup \{\varepsilon\} \quad \text{and} \quad B = \overline{C} \cup \{\varepsilon\}. \tag{6.33}$$

The languages A and B are nonregular, by virtue of the fact that C is nonregular (and therefore $\overline{C}$ is nonregular as well). On the other hand, $AB = \Sigma^*$, which is regular.

Finally, for statement (c), let us again take $\Sigma = \{0\}$, and let $A \subseteq \Sigma^*$ be any nonregular language that contains the single-symbol string 0. (Again, the language $A = \{0^n : n \text{ is a perfect square}\}$ will work.) We have that A is nonregular, but $A^* = \Sigma^*$, which is regular.