First we review the pumping lemma for regular languages.

**Definition 1.** We say that a language $L$ is *pumpable* iff

there exists an integer $p > 0$ such that

for all strings $s \in L$ with $|s| \geq p$,

there exist strings $x, y, z$ with $xyz = s$ and $|xy| \leq p$ and $|y| > 0$ such that

for every integer $i \geq 0$,

$$xy^iz \in L.$$  

We proved this in class last time:

**Lemma 2** (Pumping Lemma for Regular Languages). *For any language $L$, if $L$ is regular, then $L$ is pumpable.*

Here is the contrapositive, which is an equivalent statement:

**Lemma 3** (Pumping Lemma (contrapositive form)). *For any language $L$, if $L$ is not pumpable, then $L$ is not regular.*

We will use the contrapositive form to prove that certain languages are not regular by showing that they are not pumpable. By definition, a language $L$ is *not* pumpable iff

for any integer $p > 0$,

there exists a string $s \in L$ with $|s| \geq p$ such that

for all strings $x, y, z$ with $xyz = s$ and $|xy| \leq p$ and $|y| > 0$,

there exists an integer $i \geq 0$ such that

$$xy^iz \notin L.$$  

Here is a template for a proof that a language $L$ is not pumpable (and hence not regular). Parts in brackets are to be filled in with specifics for any given proof.

*Given any $p > 0$,

let $s = [describe some string in $L$ with length $\geq p$].

Now for any $x, y, z$ with $xyz = s$ and $|xy| \leq p$ and $|y| > 0$,

let $i = [give some integer $\geq 0$ which might depend on $p, s, x, y,$ and $z].$

Then we have $xy^iz \notin L$ because [give some reason/explanation].*

Note:

- We cannot choose $p$. The value of $p$ could be any positive integer, and we have to deal with whatever value of $p$ is given to us.
• We can and do choose the string $s$, which may differ depending on the given value of $p$ (so the description of $s$ uses $p$ somehow). We must choose $s$ to be in $L$ and with length $\geq p$, however.

• We cannot choose $x$, $y$, or $z$. These are given to us and could be any strings, except we know that they must satisfy $xyz = s$, $|xy| \leq p$, and $|y| > 0$.

• We get to choose $i \geq 0$ based on all the previous values.

**Example:** Let 

$$L = \{ w \in \{0, 1\}^* \mid w \text{ has more 0's than 1's} \}.$$ 

We show that $L$ is not pumpable using the template:

Given any $p > 0$,
let $s = 0^p1^{p-1}$. (Clearly, $s \in L$ and $|s| \geq p$.)
Now for any $x, y, z$ with $xyz = s$ and $|xy| \leq p$ and $|y| > 0$,
let $i = 0$.
Then we have $xy^iz = xy^0z = xz \notin L$, which can be seen as follows: Since $|xy| \leq p$ it must be that $x$ and $y$ consist entirely of 0's, and so $y = 0^m$ for some $m$, and we further have $m \geq 1$ because $|y| > 0$. But then $xz = 0^{p-m}1^{p-1}$, and so because $p - m \leq p - 1$, the string $xz$ does not have more 0's than 1's, and thus $xz \notin L$.

**Exercises**

**Exercise 4.1.1:** Prove that the following are not regular languages. For each, show that the given language is not pumpable. You may use the template given above.

a). $\{0^n1^n \mid n \geq 1\}$. This language, consisting of a string of 0's followed by an equal-length string of 1's, is the language $L_{01}$ we considered informally at the beginning of the section. Here, you should apply the pumping lemma in the proof.

b). The set of strings of balanced parentheses. These are the strings of characters “(” and “)” that can appear in a well-formed arithmetic expression.

c). $\{0^n1^n \mid n \geq 1\}$.

d). $\{0^n1^m2^n \mid n \text{ and } m \text{ are arbitrary integers}\}$.

e). $\{0^n1^m \mid n \leq m\}$.

f). $\{0^n1^{2n} \mid n \geq 1\}$.

**Exercise 4.1.2:** Prove that the following are not regular languages. [Do as many as you can.]

a). $\{0^n \mid n \text{ is a perfect square}\}$.

b). $\{0^n \mid n \text{ is a perfect cube}\}$.

c). $\{0^n \mid n \text{ is a power of 2}\}$.

d). The set of strings of 0's and 1's whose length is a perfect square.

e). The set of strings of 0's and 1's that are of the form $ww$, that is, some string repeated.

f). The set of strings of 0's and 1's that are of the form $ww^R$, that is, some string followed by its reverse. (See Section 4.2.2 for the formal definition of the reversal of a string.)
g). The set of strings of 0’s and 1’s of the form \( w\overline{w} \), where \( \overline{w} \) is formed from \( w \) by replacing all 0’s by 1’s, and vice versa; e.g., \( 011 = 100 \), and \( 011100 \) is an example of a string in the language.

h). The set of strings of the form \( w1^n \), where \( w \) is a string of 0’s and 1’s of length \( n \).

**Exercise 4.1.3:** (Optional) Prove that the following are not regular languages.

a). The set of strings of 0’s and 1’s, beginning with a 1, such that when interpreted as an integer, that integer is prime.

b). The set of strings of the form \( 0^i1^j \) such that the greatest common divisor of \( i \) and \( j \) is 1.

**Exercise 4.2.2:** If \( L \) is a language, and \( a \) is a symbol, then \( L/a \), the **quotient** of \( L \) and \( a \), is the set of strings \( w \) such that \( wa \) is in \( L \). For example, if \( L = \{ a, aab, baa \} \), then \( L/a = \{ \epsilon, ba \} \). Prove that if \( L \) is regular, so is \( L/a \). **Hint:** Start with a DFA for \( L \) and consider the set of accepting states.

**Exercise 4.2.3:** If \( L \) is a language, and \( a \) is a symbol, then \( a \setminus L \) is the set of strings \( w \) such that \( aw \) is in \( L \). For example, if \( L = \{ a, aab, baa \} \), then \( a \setminus L = \{ \epsilon, ab \} \). Prove that if \( L \) is regular, so is \( a \setminus L \). **Hint:** Remember that the regular languages are closed under reversal and under the quotient operation of Exercise 4.2.2.

**Exercise 4.2.4:** Which of the following identities are true?

a). \( (L/a)a = L \) (the left side represents the concatenation of the languages \( L/a \) and \( \{ a \} \)).

b). \( a(a \setminus L) = L \) (again, concatenation with \( \{ a \} \), this time on the left, is intended).

c). \( (La)/a = L \).

d). \( a \setminus (aL) = L \).