5.1 Example. Try to build an automaton that defines the language

\[ L_1 = \{0^m1^n \mid m = n\}. \]

The automaton starts by seeing 0 inputs. It has to remember the exact number of 0 inputs it sees, because it will later have to check that number against the number of 1 inputs. But the number of 0 inputs can be arbitrarily large. So intuitively, no finite number of states can remember the exact number of 0 inputs. We conclude that \( L_1 \) is not regular. The pumping lemma for the regular languages formalizes this argument.

5.2 Pumping lemma. Consider a regular language \( L \). Then \( L \) is accepted by some finite automaton \( M \). Let \( p \) be the number of states of \( M \). Now consider a word in \( L \) with at least \( p \) letters. Then \( w \) is accepted by \( M \) along some run that contains a loop. We can construct other runs of \( M \) by going through the loop 0, 1, 2, or more times. These runs also accept words in \( L \). In other words, any accepted word \( w \) of length at least \( p \) can be “pumped” to find infinitely many other accepted words.

Lemma. For every regular language \( L \), there exists a number \( p \geq 1 \) such that for every word \( w \in L \) with at least \( p \) letters there exist \( x, y, z \) with \( w = xyz \) and \( |y| > 0 \) and \( |xy| \leq p \) such that for every number \( i \geq 0 \)

\[ xy^iz \in L. \]

We call \( p \) the pumping number of \( L \), and \( xyz \) the pumping decomposition of \( w \). Suppose that we want to prove that a language \( L \) is not regular. We can do this by showing that the pumping lemma does not hold for \( L \); that is, we prove the negation of the pumping lemma:

for all numbers \( p \geq 1 \)
for every word \( w \in L \) with at least \( p \) letters such that
there exist \( x, y, z \) with \( w = xyz \) and \( |y| > 0 \) and \( |xy| \leq p \)
there exists a number \( i \geq 0 \) such that
\[ xy^iz \notin L. \]

Note the alternations of “for all” and “there exists.” This alternation determines the shape of the proof. We have to consider all possibilities for the pumping number \( p \), and all possibilities for the pumping decomposition \( x, y, z \) (often by case analysis), but we are free to choose a single word \( w \) and a single iteration number \( i \). Choosing a suitable \( w \) is usually the crux of the proof; for \( i \), we can typically choose \( i = 0 \) or \( i = 2 \).

5.3 Example. We prove that \( L_1 \) from Section 5.1 is not regular. We show that the pumping lemma does not hold for \( L_1 \). Consider any pumping number \( p \); all we know about \( p \) is that \( p \geq 1 \). Choose \( w = 0^p1^p \).

Consider any pumping decomposition \( w = xyz \); all we know about \( xyz \) is that \( |y| > 0 \) and \( |xy| \leq p \). It follows that \( x = 0^a \) and \( y = 0^b \) and \( z = 0^{p-a-b}1^p \), for \( b \geq 1 \). Choose \( i = 2 \). We need to show that \( xy^2z = 0^{p+b}1^p \) is not in \( L_1 \). This is the case because \( b \geq 1 \).

5.4 Example. We prove that

\[ L_2 = \{xx \mid x \in \{0, 1\}^*\} \]

is not regular. We show that the pumping lemma does not hold for \( L_2 \). Consider any pumping number \( p \geq 1 \). Choose \( w = 10^p10^p \).

Consider any pumping decomposition \( w = xyz \); all we know about \( xyz \) is that \( |y| > 0 \) and \( |xy| \leq p \). There are two possibilities:
\[(a) \ x = 10^a \text{ and } y = 0^b \text{ and } z = 0^{p-a-b}10^p, \text{ for } b \geq 1.\]
\[(a) \ x = \varepsilon \text{ and } y = 10^b \text{ and } z = 0^{p-b}10^p.\]

Choose \(i = 2.\) We need to show that \(xy^2z\) is not in \(L_2.\)

In case (a), \(xy^2z = 10^{p+b}10^p\), which is not in \(L_2\) because \(b \geq 1.\)

In case (b), \(xy^2z = 10^{b}10^p\), which is not in \(L_2\) because it contains three 1’s.

5.5 Example. We prove that \(L_3 = \{1^{n^2} | n \geq 0\}\) is not regular. We show that the pumping lemma does not hold for \(L_3.\) Consider any pumping number \(p \geq 1.\) Choose \(w = 1^{p^2}.\) Consider any pumping decomposition \(w = xyz\) such that \(|y| > 0\) and \(|xy| \leq p.\) It follows that \(x = 1^a\) and \(y = 1^b\) and \(z = 1^{p^2-a-b}, \text{ for } b \geq 1 \text{ and } a + b \leq p.\) Choose \(i = 2.\) We need to show that \(xy^2z = 1^{p^2+b}\) is not in \(L_3;\) that is, we need to show that \(p^2 + b\) is not a square. Since \(b \geq 1,\) we have \(p^2 + b > p^2.\) Since \(a + b \leq p,\) we have \(p^2 + b \leq p^2 + p < (p + 1)^2.\)

5.6 Proving (non)regularity. To prove that a language \(A\) is regular, there are essentially two options:

1. Find a finite automaton (or regular expression) that defines \(L.\)
2. Show that \(L\) can be built from simpler languages that are known to be regular using operations that are known to preserve regularity (the boolean operations \(\cup, \cap, \text{ and } \sim,\) the regular operations \(\circ\) and \(\cdot\), the merge operation \(||\), etc.).

To prove that a language \(A\) is not regular, there are again two options:

1. Show that the negation of the pumping lemma holds for \(L.\)
2. Show that a language that is known to be nonregular can be built from \(L\) and languages that are known to be regular using operations that are known to preserve regularity. (Note: if \(L_1\) and \(L_2\) are regular, then \(L_1 \cap L_2\) is again regular, but if \(L_1\) and \(L_2\) are nonregular, then \(L_1 \cap L_2\) may or may not be regular! etc.)

Here are two examples of the second proof technique:

\[L_4 = \{w \in \{0, 1\}^* | w \text{ contains the same number of 0’s and 1’s}\}\]
is not regular, because \(L_4 = L_4 \cap (0^*1^*)\) (if \(L_4\) were regular, then \(L_1\) would also be regular, which contradicts Section 5.3).

\[L_5 = \{w \in \{0, 1\}^* | w \text{ contains different numbers of 0’s and 1’s}\}\]
is not regular, because \(L_4 = L_5\) (if \(L_5\) were regular, then \(L_4\) would also be regular, which contradicts the previous result).