FSA and Regular Language III: Pumping Lemma
Ling 106, Maribel Romero
Nov. 3, 2003

1 What is the Pumping Lemma useful for?

- We know that a language is regular if we can construct a finite state automaton for it.
- Not all languages are regular though. But then how would we know if a language is not regular? Could we simply conclude that a language is not regular if we cannot construct an FSA for it? Not really. It might be because we didn’t think hard enough.
- We need some systematic method for showing that a language is not regular, and therefore, an FSA cannot be constructed for it.
- Pumping Lemma states a deep property that all regular languages share. By showing that a language does not have the property stated by the Pumping Lemma, we are guaranteed that it is not regular.

2 The idea: The Pigeon Hole Principle

- Partee et al. P. 468: "Consider an infinite [regular language] L. By definition, it is accepted by some FSA M, which, again by definition, has a finite number of states. But since L is infinite, there are strings in L which are as long as we please, and certainly L contains strings with more symbols than the number of states in M. Thus, since M accepts every string in L, there must be a loop in M (...)."
- If p number of pigeons are placed into fewer than p holes, some hole has to have more than one pigeon in it. (Pigeon Hole Principle)
- Similarly, if an FSA has n number of states, and this machine accepts strings of length n or greater, it will have to pass through at least one state more than once in order to accept such strings.
  That is, there will be a loop in the machine.
  \[ q_1, q_2, q_3, ..., q_k, ..., q_{n-1}, q_n \]
- This means that there is some substring that is read by the sequence of states:
  \[ q_k, ..., q_k \]
- Given a string with length n or greater, which has a substring read by looping through \(q_k\), we can construct even longer strings of the language by repeating (pumping) that substring over and over again.
3  What does Pumping Lemma say?

3.1 Theorem 1.37: Pumping Lemma

If $A$ is a regular language, then there is a number $p$ (the pumping length) where, if $s$ is any string in $A$ of length at least $p$, then $s$ may be divided into three pieces, $s = xyz$, satisfying the following conditions:

1. for each $i \geq 0, xy^iz \in A$,
2. $y \neq \epsilon$, and
3. $|xy| \leq p$.

3.2 Explanation

- The Pumping Lemma says that if a language $A$ is regular, then any string in the language will have a certain property, provided that it is ‘long enough’ (that is, longer than some length $p$, which is the pumping length).

Inside any string in $A$ that’s longer than $p$, we can find a piece that can be repeated (pumped) as many times as we want, and the result will always be in $A$.

Moreover, this piece can be found within the first $p$ letter of our string.

- That is, given any string $s$ in $A$ longer than $p$, we can find a substring in $s$ that can be pumped. We’ll call this substring $y$. Then anything before $y$ we’ll call $x$, and anything after $y$ we’ll call $z$.

Then the whole string can be rewritten as $x - y - z$. (Remember that these are strings and not letters!)

By repeating $y$ zero or more times, we get:

$$xz, xyz, xyyz, xyyyz, \ldots, xyyyyyyyyyyyzz, \ldots$$

What the Pumping Lemma says is that each of these must be in $A$.

- Condition 1: “for each $i \geq 0, xy^iz \in A$”

$xy^2z$ is the same as $xyyz$, etc. So this says that sticking in multiple copies of $y$ will give you strings that are still in the language. For $i = 0$, you get no copies of $y$, i.e., the string $xz$.

- Condition 2: “$y \neq \epsilon$”, or “$|y| > 0$”

While $x$ or $z$ may have length zero, the length of $y$ is not zero. That is, $y$ is not the empty string. If you allowed $y$ to be the empty string, the theorem would be trivially true. This is because if $y$ was the empty string, you would end up with $xz$, which is just $s$, the original string you started with, no matter how many times you pumped $y$. 

2
• Condition 3: “$|xy| \leq p$”

Since $x$ is the piece before $y$, this says that all of $y$ must come from the first $p$ letters of our string $s$, so that the combined length of $x$ and $y$ is at most $p$.

3.3 Examples with regular languages

• Let’s apply the Pumping Lemma to the following language $B$.

$B = \{ w \mid w$ begins with 1 and ends with 0, with anything in between $\}$.

\[
\begin{array}{c}
\text{q0} \quad 1 \rightarrow \text{q1} \\
\text{q1} \quad 1, 0 \rightarrow \text{q1} \\
\text{q1} \quad \rightarrow \text{q2} \quad 0
\end{array}
\]

Let’s assume that the pumping length $p$ is 3. Let’s take some string of length 3 or longer. How about 10100010? We can then break this string like this:

\[
x = 1, \ y = 01, \ z = 00010
\]

By pumping $y$, we get:

\[
x y^0 z = 1-00010, \ xy^1 z = 1-01-00010, \ xy^2 z = 1-0101-00010, \ xy^3 z = 1-010101-00010, \ xy^4 z = 1-01010101-00010, \ etc.
\]

All of these strings begin with 1 and end with 0. So, the pumping lemma works for this language and this string.

• Show that the strings 100 and 1100 in language $B$ can also be divided in a way that complies with the Pumping Lemma.

• QUESTION: Apply the Pumping Lemma to the language $C$ recognized by the following FSA. Assume the pumping length is 3 and use the strings indicated below:

\[
\begin{array}{c}
\text{q0} \quad 1 \rightarrow \text{q1} \\
\text{q1} \quad 1, 0 \rightarrow \text{q1} \\
\text{q1} \quad \rightarrow \text{q2} \quad 0
\end{array}
\]
a. 1001
b. 10010
c. 1001001
d. 100
e. 10

• QUESTION: What happens if we apply the Pumping Lemma to the following regular language $D$, assuming that the pumping length $p$ is 3?

$D = \{01\}$.

• Question for RECI TATION: Apply the Pumping Lemma to the regular languages $E$ and $F$ below. Try at least three different strings for $E$.

a. $E = \{w : w$ contains the sub-string 101\}, with pumping length 4.
b. $F = \{w : w$ has at most length 4\}, with pumping length 4.

4 How to use the Pumping Lemma to prove that a language is not regular

The pumping lemma is most useful when we want to prove that a language is not regular. We do this by using a proof by contradiction.

To prove that a given language $L$ is not regular:

1. Assume that $L$ is regular.

2. Use the pumping lemma to guarantee the existence of a pumping length $p$ such that all strings of length $p$ or greater in $L$ can be pumped.

3. Find a string $s$ in $L$ that has length $p$ or greater but that cannot be pumped.

4. Demonstrate that $s$ cannot be pumped by considering all ways of dividing $s$ into $x$, $y$, and $z$, and for each division, finding a value $i$ where $xy^iz \not\in L$.

$\Rightarrow$ The existence of $s$ contradicts the pumping lemma if $L$ were regular. Hence $L$ cannot be regular.

4.1 Example 1.38

Let $B$ be the language $\{0^n1^n \mid n \geq 0\}$. Show that $B$ is not regular, using the pumping lemma. We will do this by assuming that $B$ is regular, and showing that contradiction follows. Therefore, the assumption we started out with must be wrong, and thus $B$ is not regular.
Let $p$ be the pumping length given by the pumping lemma. Choose $s$ to be the string $0^{p-1}1^{p-1}$.

Because $s \in B$, and $s$ has length greater than $p$, the pumping lemma guarantees that we can split $s$ into three pieces, $s = xyz$ in such a way that for any $i \geq 0$, the string $xy^iz$ is in $B$. We consider three cases to show that this result is impossible.

1. The string $y$ contains only of 0s. In this case, the string $xyyz$ ("pumped up") has more 0s than 1s and so is not a member of $B$, violating condition 1 of the pumping lemma. This is a contradiction. [Also, the string $xz$ ("pumped down") has more 1s than 0s and hence it is not a member of $B$. Contradiction.]

2. The string $y$ contains only of 1s. In this case, the string $xyyz$ has more 1s than 0s and so is not a member of $B$. This is another contradiction.

3. The string $y$ contains both 0s and 1s. In this case, the string $xyyz$ may have the same number of 0s and 1s, but they will be out of order with some 1s before 0s. But in our language $B$, all the 0s must precede the 1s. Thus, $xyyz$ is not in our language. This is another contradiction.

There is no other way to split up the string $s$, so a contradiction is unavoidable if we make the assumption that $B$ is regular, and so $B$ is not regular.

4.2 Example 1.39

Let $C = \{w \mid w$ has an equal number of 0s and 1s (in any order)$\}$. We will show that $C$ is not regular using the pumping lemma.

Proof 1

• Assume to the contrary that $C$ is regular. Let $p$ be the pumping length given by the pumping lemma. Let $s$ be the string $0^p1^p$.

• With $s$ being a member of $C$ and having length more than $p$, the pumping lemma guarantees that $s$ can be split into three pieces, $s = xyz$, where for any $i \geq 0$, $xy^iz \in C$.

• We would like to show that this outcome is impossible. But wait, it is possible! If we let $x$ and $z$ be the empty string and $y$ be the string $0^p1^p$, then $xy^iz$ always has an equal number of 0s and 1s and hence is in $C$.

• But notice that condition 3 of the pumping lemma requires that $|xy| \leq p$. Remember that $s = 0^p1^p$, so if $|xy| \leq p$ then $y$ must consist only of 0s. But then $xyyz \notin C$. Therefore, $s$ cannot be pumped.

• This gives us the desired contradiction, and so $C$ is not regular.

Proof 2

Another way of proving that $C$ is not regular follows from our knowledge of the closure properties of the regular languages plus the fact that $B$ is not regular.
• Note first that the language 0*1* is a regular language.

• Recall that when any two languages are intersected, the result is a regular language. That is, regular languages are closed under intersection.

• Suppose $C$ is a regular language. Then $C \cap 0^*1^*$ is a regular language.

• But $C \cap 0^*1^* = 0^n1^n = B$ and we have proved that $B$ is not a regular language.

• Therefore $C$ is not a regular language.

4.3 Example 1.24: Illustration of ‘pumping down’

Let $E = \{0^i1^j \mid i > j\}$. Show that $E$ is not regular using the pumping lemma.

• Assume that $E$ is regular. Let $p$ be the pumping length for $E$ given by the pumping lemma. Let $s = 0^{p+1}1^p$. Then $s$ can be split into $xyz$ satisfying the conditions of the pumping lemma.

• By condition 3, $y$ consists only of 0s. But $xyyx$ is also in $E$ because adding $y$ increases the number of 0s and so this string is guaranteed to have more 0s than 1s. No contradiction obtains. So, we need to try something else.

• The pumping lemma states that $xy^iz \in E$ even when $i = 0$. So let’s consider the string $xy^0z = xz$. Removing $y$ decreases the number of 0s in $s$. Recall that $s$ has just one more 0 than 1. Therefore $xz$ cannot contain more 0s than 1s. So it cannot be a member of $E$.

• Thus, we obtain a contradiction, and so $E$ is not a regular language.

• Question for recitation:

   Show that the following languages $G$, $H$ and $I$ are not regular using the Pumping Lemma (and reasoning about the pumping length $p$ abstractly, as above):
   
   a. $G = 001^n01^n$, where $n \geq 0$.
   b. $H = 01^n001^n+1$, where $n \geq 0$.
   c. $I = 0^n1^m$, where $n = m + 3$. 