Comments on the Pumping Lemma for Regular Languages

I will not go over the proof of the lemma here. The purpose of this section is to help you to understand how to use the lemma.

The pumping lemma is used to prove that languages ***are not regular***. *You cannot use it to prove that languages are regular*.

Using the pumping lemma takes practice. There is a precise set of steps to follow for using it to prove that a language is not regular. Restating the lemma in plain English,

If L is a regular language then there exists a constant N > 0 such that, for every word w in L whose length is at least N, we can write w = xyz, where |xy| <= N and |y| > 1 and for all values of k >= 0, xy^kz is also in L.

People often have trouble using the lemma because it has several alternating "there exist" and "for all" and "implies" inside it. The "there exist" and "for all" parts are called ***quantifiers*** in logic (existential and universal respectively). We can state the lemma very concisely using the logical quantifiers and operators as follows:

L is regular => $(\exists N > 0)\ (\forall w \in L)\ (|w| \geq N) \Rightarrow$

$(\exists x,y,z)\ (w = xyz\ \wedge\ |xy| \leq N \wedge\ |y| > 0 \wedge\ (\forall\ k \geq 0)\ (\ xy^kz \in L\ )\ )$

This does not make it look any easier to use; it makes it look even more foreboding. However, it is useful for the following reason. To use the pumping lemma to show that L is ***not*** regular, you have to show ***it does not satisfy*** the pumping lemma. Therefore you have to show that the ***NEGATION*** of the conclusion of the lemma is true. If you can intuitively understand how to form the negation of a logical sentence with so many "there exists" and "for all" quantifiers then you can just skip ahead. If not, follow along. You construct the negation of the conclusion of the lemma, you will see that you have to show that:

$(\forall N > 0)$ ($\exists$w in L s.t. $|w| \geq N$) and ($\forall$x, y, z) s.t w= xyz and $|xy| \leq N$ and $|y| > 0$, $\exists$ k >= 0 s.t. $xy^kz \notin L$.

This does not look any easier, but what it reduces to is the following. You have to show that for any N, there is a special w, such that this w cannot be broken up into xyz in any way that satisfies the conditions of the lemma and whose y piece can be pumped. Showing that y cannot be pumped means showing that there is one value of k for which xy^kz is not in L. Thus, your task is to do the following:

1. Let N be an arbitrary number. (It has to stay a symbol.)
2. Pick a w in L whose length is at least N.
3. Show that no matter how you choose break up w into xyz such that |xy| <= N and |y| > 0, you can always find one value of k such that $xy^kz \notin L$.

The hardest part is to pick w. You have to pick w in such a way that it cannot be pumped. It must be chosen to depend on N in some way, and to reflect something about the properties that define the language. Once you have picked w, you have to figure out a value of k that defies the lemma. This will often be obvious, but not always.

Examples

1. (Easy) Let $L_1 = \{ a^p \mid p \text{ is a prime number} \}$. Assume N is the constant from the pumping lemma and let $w = a^p$, where p is a prime number greater than N. Then w = xyz where |y| > 0. Let m = |y|. Then for all values of k >= 0, $xy^kz \in L_1$. In particular, $xy^pz \in L_1$. But

$$|xy^{p+1}z| = |xyz| + |y^p| = |w| + |y^p| = p + pm = p(1+m).$$

Since (1+m) > 1, p(1+m) is not a prime number, so $xy^pz \notin L_1$, contradicting the pumping lemma. Therefore, L_1 cannot be regular. Notice that we did not need to use the fact that |xy| <= N in this proof.

2. (A little harder) Let $L_2 = \{ a^nb^n \mid n > 0 \}$. Assume N is the constant from the pumping lemma and let $w = a^Nb^N$. Then w = xyz where |y| > 0 and |xy| <= N. Let m = |y|. Since |xy| <= N, xy consists of only a's and in particular, $y = a^m$. This is because xy comes from the part of w before the first b. Assume $x = a^h$ for some h >= 0. Then $z = a^{N-m-h}b^N$. By the pumping lemma, for all values of k >= 0, $xy^kz \in L_2$. In particular, $xy^2z \in L_2$. But

$$xy^2z = a^ha^ma^m a^{N-m-h}b^N = a^{N+m}b^N$$

Therefore $xy^2z \notin L_2$, contradicting the pumping lemma. Therefore, L_2 cannot be regular.

3. (Harder still) This one is much more subtle than the first two, which are relatively straightforward applications of the lemma. Let $\Sigma = \{ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \}$ and define $L_3 = \{ x \in \Sigma^* \mid x \text{ is an initial segment of the decimal part of the infinite decimal expansion of } \pi \}$. In other words, since pi = 3.141592654... , L_3 contains the strings 1, 14, 141, 1415, 14159, and so on. Suppose that L_3 is regular. Assume N is the constant from the pumping lemma. Since L_3 is infinite, we can choose a w in L_3 whose length is at least N. Then by the lemma, we can write

$$w = xyz \text{ where } |y| > 0 \text{ and } |xy| <= N \text{ and for all } k >= 0,\ xy^kz \in L_3.$$

By the definition of L_3, if $uv \in L_3$ then so is u, because if uv is an initial segment of π then so is u, since it is an initial segment of an initial segment of π. This means that, since for any k, $xy^kz \in L_3$, $xy^k \in L_3$. Now if I were following the lack of rigor used by some authors, I could just say, well that is ludicrous, because π is not a repeating decimal and if for all k, xy^k were an initial segment of π, it would imply that π is a repeating decimal, but that is not a proof, and it is also flawed because π has an infinite expansion and each of these strings is finite and we need to somehow use the idea of convergence if we want to cross the bridge from the finite world to the infinite world.

Let D(x) denote the value of x as a decimal, and let D(y) denote the value of y as a number. For example, if x is the string '1415', then D(x) is the decimal value 1,415 and if y is the string '92654', then D(y) = 92,654. Suppose that |x| = r and |y| = m. Then the value of the string xy as a number to the right of the decimal point in π is

10^{-r} (D(x) + 10^{-m} D(y))

For example, with the above strings x and y, with lengths 4 and 5 respectively, the value of '.xy' would be

10^{-4} *(1415 + 10^{-5} *92654)= 10^{-4} *(1415 + .92654) = .141592654

The value of 'xyy' would be

10^{-r} (D(x) + 10^{-m} D(y) + 10^{-2m} D(y))

= 10^{-4} *(1415 + 10^{-5} *92654 + 10^{-10} *92654)

= 10^{-4} *(1415 + .92654 + .0000092654) = .14159265492654

In general, the decimal value of a string of the form xy^k is

$$10^{-r}\cdot\left(D(x)+\sum_{j=1}^{k}10^{-mj}\cdot D(y)\right)$$

Since each of the strings xy^k is an initial segment of π, as k goes to ∞, it follows that the sum converges to

$$\begin{aligned}10^{-r}\cdot\left(D(x)+\sum_{j=1}^{\infty}10^{-mj}\cdot D(y)\right) &= 10^{-r}\cdot\left(D(x)+D(y)\cdot\sum_{j=1}^{\infty}(10^{-m})^{j}\right)\\ &= 10^{-r}\cdot\left(D(x)+D(y)\cdot\left(\frac{1}{10^{m}-1}\right)\right)\end{aligned}$$

which would imply that

$$\pi-3=10^{-r}\cdot\left(D(x)+D(y)\cdot\left(\frac{1}{10^{m}-1}\right)\right)$$

But the right hand side of this equation is a rational number, consisting of a products and sums of rational numbers, whereas π is irrational, as is π -3. Therefore the assumption of regularity is false.

The Converse.

The converse of the pumping lemma is false. Namely, there are languages that satisfy the pumping lemma but are not regular. They are sometimes easy to find, other times not. An example of one such language is L = { uu^Rv | u, v ∈ Σ^+ } where Σ = {a,b}.

First let me show that this language satisfies the pumping lemma. I need a value of N for which the lemma is true. N= 4 works. Now let w be any word whose length is at least 4. Assume |u|= 1. Then $|u^R|$ = 1 and |v| >= 2. In this case, let w = xyz where x is uu^R, y is the first letter of v

and z is the second letter of v. Then $|xy| = 3 < 4$ and $|y| > 0$. Furthermore, every string xy^kz is of the form uu^Rv, since the first letter of v is being pumped and the first part of the word is still a palindrome, so all such words are in L.

Suppose that $|u| > 1$. In this case, let x be the null string, Λ, and y be the first letter of u, and z the rest of the string. For any symbol s, ss is a palindrome; i.e., ss is of the form uu^R. Therefore, for any $k >= 2$, since $x = \Lambda$ and y is a symbol s, xy^kz is of the form $ss\ uu^Rv$ which is of the form yy^Rt if you let $y = s$ and $t = uu^Rv$. Hence for $k > 1$ the words are in L. If $k = 1$, it is the original string, which is therefore in L. If $k = 0$, then we have removed the first letter of u. The string $u_2u_3...u_nu_nu_{n-1}...u_3u_2$ is the beginning of the word, so it is of the form uu^R and is followed by a non-null string, so this too is in L. Hence L satsifies the lemma when N = 4.

We cannot yet prove that L is not regular be cause we need another method of proving nonregularity. That will be the Myhill-Nerode Theorem, to follow.

The Myhill-Nerode Theorem

The ***Myhill-Nerode theorem*** states, in essence, that regular languages are precisely those languages that induce a finite equivalence relation on the set of all strings over their alphabets. To state it precisely, we need to define what that equivalence relation is.

Definition. Let L be a language over Σ^*. For any two strings x and $y \in \Sigma^*$, we say that x and y are L-equivalent and write $x \equiv_L y$ if, for any z in Σ^*, $xz \in L$ if and only if $yz \in L$.

Observe that

1. $x \equiv_L x$
2. $x \equiv_L y$ *iff* $y \equiv_L x$
3. $x \equiv_L y$ and $y \equiv_L z$ implies that $x \equiv_L z$.

Since $\equiv_L$ has these properties, it is an equivalence relation. Obviously, if $x \equiv_L y$ then $x \in L$ *iff* $y \in L$ because from the definition, $x = x\Lambda \in L$ *iff* $y = y\Lambda \in L$.

It enjoys one other property:

4. For any $w \in \Sigma^*$, $x \equiv_L y$ implies $xw \equiv_L yw$.

To see this , suppose $x \equiv_L y$ and let $w \in \Sigma^*$. Let $z \in \Sigma^*$. Let us denote wz by v. Then

$xwz = xv \in L$ *iff* $yv = ywz \in L$ because $x \equiv_L y$ *iff* for any $v \in \Sigma^*$ $yv \in L$.

hence $xw \equiv_L yw$.

This relation is called a ***right congruence*** because of this last property.

It is not true on the left side. In other words, it is not true that if $x \equiv_L y$ then for all w, $wx \equiv_L wy$. To see this, consider the language of all words having b as the second letter:

$L = < (a+b)b(a+b)^* >$

Consider the two words a and b. Then a $\equiv_L$ b because for any word z, az $\in$ L *iff* z starts with b, and bz = bb $\in$ L *iff* z starts with b. But aa is not L-equivalent to ab because ab $\in$ L but aa $\notin$ L. This is why this is called a right-congruential relation but not a left-congruential relation.

Recall that an equivalence relation creates a partition on a set, i.e., a collection of non-empty subsets that are mutually non-intersecting and whose union is the entire set. The "*is in the same time zone*" relation divides the world into 24 equivalence classes. This is a finite set of classes. For two rational numbers p and q, the relation, p $\equiv$ q *iff* p and q reduce to the same irreducible fraction, partitions the set of all rational numbers into infinitely many equivalence classes.

The fact that a language induces the $\equiv_L$ relation on Σ^* means that a language implicitly defines a partition of Σ^* into classes of words that are equivalent to each other.

Examples

Let L = <a*>. Consider the set S = {Λ, b}. All words in Σ^* are L-equivalent to one of these two words. To see this, pick any word w $\in$ Σ^*. If w has any b's in it at all, it is L-equivalent to b:

wx $\in$ L *iff* bx $\in$ L

is true because if w has any b's, then for all x $\in$ Σ^*, wx $\notin$ L and bx $\notin$ L as well. If w has no b's at all, then it is equivalent to Λ:

wx $\in$ L *iff* x $\in$ L

is true because if w has no b's at all, either it is Λ or it is a sequence of a's. In either case, wx $\in$ L is true *iff* x has only a's or is Λ, which is true *iff* x $\in$ L.

For this particular L, there were two equivalence classes. Consider the language,

L = { $a^n b^n$ | n > 0 }.

Consider the set S = {b} $\cup$ { $a^n b^m$| n > 0 and m <= n}. Let x $\in$ Σ^*. If x is of the form, a^n, it is L-equivalent to itself, and x $\in$ S. If it is of the form $a^n b^m$ and m <= n, it is also L-equivalent to itself. If x is in any other form, it is L-equivalent to b, because, for any z $\in$ Σ^*, neither bz nor xz will be in L. The word b acts as a representative of the rejected words in Σ^*.

Notice that the first language had a finite set of equivalence classes, and the second had an infinite set. This fact is important – it is the essence of the Myhill-Nerode theorem. This finiteness is encapsulated in the following definition.

Definition. Let L be a language over Σ^* and let S $\subseteq$ Σ^*. S is called a ***spanning set*** for L if

1. S is a finite set, and
2. For every w $\in$ Σ^*, there exists a y $\in$ S such that w $\equiv_L$ y.

In other words, a spanning set is a finite collection of words with the property that, for each word w $\in$ Σ^*, there is some element y $\in$ S that is equivalent to it with respect to L. This means that for all possible words z that can be appended to w, wz $\in$ L if and only if yz $\in$ L.

Myhill-Nerode Theorem. A language is regular if and only if it has a spanning set.

Proof.

One direction of the proof is fairly easy to understand. The other direction is a little more abstract but also very straightforward.

Let L be a regular language. Then there is a FA M such that L = L(M). Suppose that the set of states of M is Q = { q_1, q_2, q_3, ..., q_n } where q_1 is the start state. Let F denote the set of states that are final states in M.

Let us say that a word w ***reaches*** a state q if $\delta^*(q_1, w) = q$. Call a state in Q ***reachable*** if there exists a word in Σ^* that reaches it. For each reachable state q of Q, pick any one of the words that reaches q. Call this word, w_q. Then the set S = { $w_q \mid q \in Q$ } is a spanning set for L.

To prove this we need to show that for any word $x \in \Sigma^*$, there is a word in S that is L-equivalent to it. Therefore, let x be an arbitrary word in Σ^*. This word x reaches some state q in Q. Since w_q is the word in S that reaches q, w_q and x both reach q. Pick any word $z \in \Sigma^*$. Then $w_q z$ reaches the same state as xz because w_q and x both reach q and from q, on reading z, the FA M enters a unique state; i.e.,

$$\delta^*(q_1, xz) = \delta^*(\delta^*(q_1, x), z) = \delta^*(\delta^*(q_1, w_q), z) = \delta^*(q_1, w_q z)$$

Therefore, $xz \in L$ *iff* $\delta^*(q_1, xz) \in F$ *iff* $\delta^*(q_1, w_q z) \in F$ *iff* $w_q z \in L$, proving that S is a spanning set for L.

Conversely, suppose that L has a spanning set S. We will construct an FA that accepts L. To do this we will associate a unique state to each member of the spanning set. Then we will give the rule for which state is a start state and which states are final states. Then we define the transition function for M. Finally we have to prove that the machine we constructed accepts L.

For each word y in S, we create a unique state in M. We will call the state that we created for y, q_y. Let Q = { $q_y \mid y \in S$}. Then Q is the set of all states of M. Since S is a spanning set for L, for every word $w \in \Sigma^*$, there is an element $y \in S$ such that $w \equiv_L y$. This is also true for Λ, the null string. Let y_0 be the particular word in S such that $\Lambda \equiv_L y_0$. Make the state q_y created for this y_0 the starting state. Call it q_1 instead of q_{y0}. Now the states in Q that will be final states are the ones that are L-equivalent to some word in L. In fact, if y is a word in S that is L-equivalent to a word in L, then y itself must also be in L. To see this, remember that if $w \in L$ and $w \equiv_L y$, then $y \in L$. Therefore, it makes sense to define the set of final states to be F = { $q_y \mid$ y in S and $y \in L$}.

The transition function, δ, is defined by the following rule: For each state q_y and each symbol a, $\delta(q_y, a) = q_z$ iff $ya \equiv_L z$. Since ya is some word in Σ^*, it is L-equivalent to some unique z in S. This just says that we have to find that unique z and create the transition from q_y to q_z on a.

The FA M is thus defined. Now it remains to show that it accepts L. To do this we have to prove that the transitive closure of δ, δ^*, has the desired property.

Claim: For any $w \in \Sigma^*$, and any state q_y, $\delta^*(q_y, w) = q_z$ *iff* $yw \equiv_L z$.

Proof. We can prove this by induction on the length of w. If |w| = 0, then w is the null string and $\delta^*(q_y, \Lambda) = q_y$. Since $y \equiv_L y$, it is true in the base case. Assume it is true for w such that |w| = k and let w be a string of length k+1. Then w = va for some string v of length k. Therefore,

$$\delta^*(q_y, w) = \delta^*(q_y, va) = \delta(\delta^*(q_y, v), a) = \delta(q_x, a) = q_z$$

where $yv \equiv_L x$, by the inductive hypothesis and $\delta(q_x, a) = q_z$ iff $xa \equiv_L z$ by the definition of the transition function. Since $yv \equiv_L x$, we have

$$yva \equiv_L xa \text{ and } xa \equiv_L z$$

which implies by the transitivity of the relation that

$$yva \equiv_L z$$

and since w = va,

$$yw \equiv_L z$$

We have just shown that $\delta^*(q_y, w) = q_z$ *iff* $yw \equiv_L z$, proving the above claim.

The claim is true for all states q in M, and in particular it is true when q_y is the start state of M; i.e.,

$$\delta^*(q_1, w) = q_z \text{ iff } \Lambda w \equiv_L z \textit{ iff } w \equiv_L z.$$

Now

$w \in L$

iff there exists a $y \in S$ such that $w \equiv_L y$ and $y \in L$

iff $\delta^*(q_1, w) = q_y$ and $q_y \in F$

iff $w \in L(M)$.

Hence, we have constructed a FA M that accepts exactly L, and the theorem is proved.

Using The Myhill-Nerode Theorem

How is the Myhill-Nerode theorem used? Because it is an "if and only if" theorem, it provides a means of proving that languages are regular and that they are not regular. As we saw above, every language, regular or not, induces the L-equivalence relation on the strings over Σ*. The difference between regular languages and non-regular languages is that, for regular languages, the number of equivalence classes of this relation is finite. This is essentially what Myhill-Nerode states.

Proving Languages Regular

If a language has a spanning set, it must be regular. If it can be shown that it cannot possibly have a spanning set, it is not regular. Therefore, to use the theorem to show that L is regular, it is enough to find a single spanning set for L.

Example

Let $L = \{a^n bba^m \mid n, m \geq 0\}$. Let us look for a spanning set. The first part of the word up to the first b is independent of the value of n. All words of the form a^n are L-equivalent. To see this, let x be any word over {a,b} and consider two words $a^n x$ and $a^m x$. Either both are in L or both are

not in L, depending on whether x is of the form a^kbba^m or not. Since Λ is in this set, we use Λ as its representative. Next, consider the strings that are of the form a^nbb. These are not L-equivalent to the first group -- take a^n and a^nbb. If we append Λ to each we see that a^n is not in L but a^nbb is. All strings of this form though are L-equivalent to each other. Again, we can take the shortest one, bb, as its representative. Is that all? Are all strings equivalent to one or the other? What about the string 'b'? Is b L-equivalent to bb? No, because bb is in L but b is not. There are now three words in our spanning set:

$$S = \{ \Lambda, b, bb \}.$$

Is this enough? Pick any word w in A*. If w is a word like bab, is it L equivalent to one of these? If we append Λ to bab and to bb, we see it is not in its class since bb is in L and bab is not. If we append b to bab and to b, we see that babb is not in L but bb is, so bab is not L-equivalent to b. What about appending abba to both bab and Λ. The first produces bababba, and the second, abba. The first is not in L and the second is. Therefore, S is not large enough yet. It needs the representative of the class of words that start with a b and are followed by an a, so that they cannot be in L. Take ba as its representative. No word appended to ba will be in L, and conversely every word beginning ba will fail to be in L also. The spanning set is now

$$S = \{\Lambda, b, ba, bb \}.$$

This set is sufficient, and no set with fewer elements is sufficient. This shows that L is regular and from the proof of the Myhill-Nerode theorem we can use the technique to construct a 4-state FA accepting L.

Proving Non-regularity

To prove that a language is not regular, we have to show that no spanning set exists, which means that we have to show that there are infinitely many equivalence classes in the L-equivalence relation for L. Therefore, the typical paradigm for using Myhill-Nerode to show non-regularity is to show that there are infinitely many words in L that are in different equivalence classes. When two words are in different equivalence classes, we say they are distinguishable.

Example 1

Consider the language L of palindromes over {a,b}. Consider two words of the form a^nb and a^mb, where $n \neq m$. Pick any $x \in \Sigma^*$. Then $a^nbx \in L$ implies that x is of the form b^ka^n for some k, which implies that for this particular x, $a^mbx \notin L$, since this would not be a palindrome. Therefore, a^nb and a^mb are not L-equivalent. This implies that there are infinitely many words of the form a^nb that are not in the same equivalence class, and thus that L is not regular.

Example 2

Let us return to the language $L = \{ uu^Rv \mid u, v \in \Sigma^+ \}$ where $\Sigma = \{a,b\}$. We saw that this language satisfies the pumping lemma. We can now show that it is not regular. We need to identify infinitely many words in Σ* that are distinguishable from each other. Consider two words of the form $ab^{2n+1}a$ and $ab^{2m+1}a$, where $n \neq m$. It is important that the number of b's is odd in each

because neither of these words by itself is an even palindrome (a word having the form uu^R) since they both have odd length. More importantly, no prefix of either of them is of the form uu^R. because every prefix begins with 'ab' and would have to end with 'ba' and be of even length. But the only way a prefix can begin with 'ab' and end with 'ba' is to contain all of the b's, and in this case it would be the entire word, which is of odd length. But each of these is an odd palindrome -- it is its own reverse.

All we need to show is that for a single string x, $ab^{2n+1}ax \in L$, but $ab^{2m+1}ax \notin L$. Let $x = ab^{2n+1}aa$. The word $ab^{2n+1}ax = ab^{2n+1}aab^{2n+1}aa$ is of the form uu^Rv and is thus in L, but the word $ab^{2m+1}ax = ab^{2m+1}aab^{2n+1}aa$ is not of the form uu^Rv because no prefix of $ab^{2m+1}aab^{2n+1}a$ is an even palindrome, because $n \neq m$ and since it starts with 'ab' it would have to end with 'ba' and the only prefixes of the word with this property are $ab^{2m+1}a$ and $ab^{2m+1}aab^{2n+1}a$. But we already saw that the first is not an even palindrome, and since $n \neq m$, the second cannot be. Thus $ab^{2m+1}ax \notin L$. This shows that for every distinct value of n and m, the words $ab^{2n+1}a$ and $ab^{2m+1}a$ are in distinct equivalence classes, and hence L does not have a spanning set and is not regular.

Conclusions

Both the pumping lemma and the Myhill-Nerode theorem provide direct means for showing that a language is not regular. The pumping lemma may sometimes not be enough to show this, whereas the Myhill-Nerode theorem, in principle, can always show it. Sometimes neither will yield an obvious proof.

The Myhill-Nerode theorem provides a means of proving regularity, unlike the pumping lemma, and in using it, you can end up with a FA with a least number of states. If in building your spanning set, you fail to identify that two words belong to the same class, and you put them in the spanning set as unique representatives, then you will not have a spanning set, since two of its classes should really be one class. If you construct the spanning set properly, it will yield a FA of minimal size.