Lesson 7: Non-Regular Languages

Prof. John Wenskovitch
09/23/2014
Last Time

• Well, there was an exam, but before that...

• Regular expressions
• Converting RegEx into NFA
• Converting DFA into RegEx
Can we recognize every possible language with finite state machines?

• Well, no...

• Consider the language $B = \{0^n1^n \mid n \geq 0\}$
  – Try to draw this machine...

• We can draw a machine for 01, and 0011, and 000111, but there’s no bound on the number of 0s and 1s, so we’d need a machine with an infinite number of states.
The **Pumping Lemma**

- If $A$ is a regular language, then there is a number $p$ (pumping length) where, if $s$ is any string in $A$ of length at least $p$, then $s$ may be divided into 3 pieces, $s = xyz$, satisfying the following conditions:
  1. for each $i \geq 0$, $xy^iz \in A$
  2. $|y| > 0$, and $x$ or $z$ can be $\varepsilon$, but $y$ cannot
  3. $|xy| \leq p$
Proof

• Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that recognizes language $A$.

• We assign the pumping length $p$ to be the number of states in $M$.
  
  – If there are no strings of length $p$, then the theorem is “vacuously true,” so let’s only worry about strings $p$ or longer.
  
  – Consider the sequence of states that $M$ goes through when computing string $w$: there must be at least $|w|+1$ states that it has passed through. Therefore, there must be a repeated state.
To prove this, we need another tool...

• The **Pigeonhole Principle**
  – If \( p \) pigeons are placed into fewer than \( p \) holes, some hole **has** to have more than one pigeon in it.

• Now, let’s divide string \( s \) into three parts, \( xyz \), so that \( x \) is the part appearing before the repeated state, \( y \) is the part between the two appearances of the repeated state, and \( z \) is the part after the second occurrence of the repeated state.
This is all a fancy way of saying:
Why does this work?

- Suppose we run $M$ on input $xyyz$:
  1. If the machine accepted $xyz$, it will accept $xyyz$.
  2. $y$ is at least three states in size.
  3. By the Pigeonhole Principle, the first $p+1$ states must contain a repetition, so $|xy| \leq p$. 
How does the Pumping Lemma work in a proof then?

• Let $B$ be the language $\{0^n1^n \mid n \geq 0\}$. We can use the Pumping Lemma to prove that $B$ is not a regular language:
  
  – Assume to reach a contradiction that $B$ is regular. This means that a string in $B$ can be broken down into parts $xyz$ that match the definitions in the pumping lemma so that, for any $i \geq 0$, $xy^iz \in B$.
  
  – Choose $s$ to be the string $0^p1^p$. We will consider three cases to show that this is impossible.
How does the Pumping Lemma work in a proof then?

1. If the component string $y$ consists of only 0s, then the string $x y y z$ has more 0s than 1s and so is not a member of $B$, violating condition 1 of the Pumping Lemma.

2. If the component string $y$ consists of only 1s, then we will have the same problem as in part 1.

3. If the string $y$ consists of both 0s and 1s, then the string $x y y z$ will have 0s and 1s out of order, hence it is not a member of $B$, which is a contradiction. ■
How about another example then?

• \( C = \{ w \mid w \text{ has an equal number of } 0\text{s and } 1\text{s}\} \).
  – Assume to reach a contradiction that \( C \) is regular. Let \( p \) be the pumping length given by the Pumping Lemma. Again, let \( s \) be the string \( 0^p 1^p \).
  – We can break it down so that \( x,z=\varepsilon \) and \( y=0^p 1^p \).
  – Now our old arguments don’t work because the 0s and 1s can be in any order. This is where part 3 of the Pumping Lemma definition comes in handy.
  – Since \( |y|=2p \), \( |xy| \leq p \), so we have reached a contradiction. ■
Let’s look at one more

- \( F = \{ww \mid w \in \{0,1\}^*\} \)
  - Assume to reach a contradiction that \( F \) is regular. Let \( p \) be the pumping length given by the Pumping Lemma.
  - Let \( s \) be the string \( 0^p 10^p 1 \). Because \( s \) is a member of \( F \) and \( s \) has length more than \( p \), the pumping lemma guarantees that \( s \) can be split into three pieces, \( s=xyz \), satisfying the three conditions of the lemma. The outcome is impossible because of the same condition 3 argument. ■
So what do we need to do in these proofs?

• Pick an $s$ that:
  – Is in language $A$,
  – Guaranteed to be longer than $p$,
  – Cannot be pumped.

• (If you pick an example that can be pumped, you’ll need to try a different $s$...)

09/23/2014
Non-Regular Languages
Any Questions?

HOMEWORK (due 9/30)
1.29, 1.30, 1.36, 1.37