Homework

- Homework #3
  - Early submission by Friday
  - Otherwise, due 9/30

Announcements

- Exam 1
  - 1 week from today (Oct 5th)
  - 1 hour
  - Cover Regular Languages (up to and including today’s lecture)
  - Closed book (1 sheet study guide okay)

Before we begin

- Any questions?

Languages

- Recall.
  - What is a language?
  - What is a class of languages?

Regular Languages

- A language \( L \) is a regular language if there is a DFA, \( M \), such that \( L = \text{L}(M) \).

- A language \( L \) is a regular language if there is an NFA, \( N \), such that \( L = \text{L}(N) \).
The bottom line

Regular Languages

The Burning Question...
- We've looked at a number of regular languages
- I know that you are just dying to know...
  - Is there a language L that is not regular?
- To answer this, we'll use what is known as The Pumping Lemma.

The Pumping Lemma
- Statement of the pumping lemma
  - Let L be a regular language.
  - Then there exists a constant n (which varies for different languages), such that for every string x ∈ L with |x| ≥ n, x can be expressed as x = uvw such that:
    1. |v| > 0
    2. |uv| ≤ n
    3. For all k ≥ 0, the string uv^kw is also in L.

The Pumping Lemma
- What this means
  - For a long enough string x in L:
    1. We can express x as the concatenation of three smaller strings
    2. The middle string can be "pumped" (repeated) any number of times (including 0 = deleting) and the resulting string will be in L.
Proof of the pumping lemma
Since L is regular, there is a FA \( M=(Q,\Sigma,q_0,A,\delta) \) that accepts L.

Assume \( M \) has \( n \) states.

Consider a string \( x \) with \( |x| = m \geq n \).

Express \( x = a_1a_2a_3...a_m \), where each \( a_i \in \Sigma \).

Define \( p_i \) to be the state \( M \) is in after reading \( i \) characters:
\[
p_i = \delta^*(q_0, a_1a_2...a_i)
\]
\[
p_0 = q_0
\]

Let \( x = uvw \)
\[
u = a_1a_2...a_i
\]
\[
v = a_{i+1}a_{i+2}...a_j
\]
\[
w = a_{j+1}a_{j+2}...a_m
\]

Then \( x = uvw \)
\[
u = a_1a_2...a_i
\]
\[
v = a_{i+1}a_{i+2}...a_j
\]
\[
w = a_{j+1}a_{j+2}...a_m
\]

You can loop (pump) on the \( v \) loop 0 or more times and there will still be a path to the accepting state.

So what good is the pumping lemma?
It can be used to answer that burning question:
Is there a language \( L \) that is not regular?

Non-regular languages
Venn-diagram of languages
Is there something out here?
Pumping lemma

- The real strength of the pumping lemma is proving that languages are not regular
  - Proof by contradiction
    - Assume that the language to be tested is regular
    - Use the pumping lemma to come to a contradiction
    - Original assumption about the language being regular is false
  - You cannot prove a language to be regular using the Pumping Lemma!!!!

Pumping lemma

- The Pumping Lemma game
  - To show that a language $L$ is not regular
    - Assume $L$ is regular
    - Choose an “appropriate” string $x$ in $L$
      - In terms of $n$ (number of states in DFA)
    - Express $x = uvw$ following rules of pumping lemma
    - Show that $uv^kw$ is not in $L$, for some $k$
    - The above contradicts the Pumping Lemma
    - Our assumption that $L$ is regular is wrong
    - $L$ must not be regular

Example:

- $L = \{x \in \{0,1\}^* \mid 0^i1^i, i \geq 0\}$
- Ex: 001111, 0011, $\lambda$, 00001111
Pumping Lemma

- \( x = uvw = 0^n1^n \)
  - \( 00 \ldots 0 11 \ldots 1 \)
  - Since \(|uv| \leq n\), \(uv\) must consist entirely of 0s and, as such, \(v\) must also consist entirely of 0s.
  - \(v = 0^j\) for some \(j \leq n\)

- Let's pump!
  - By the Pumping Lemma
    - \(uv^2w\) is also in \(L\)
    - \(uv^2w = 0^n0^i1^n\)
    - Certainly \(i + 2j + k = n\)
    - \(uv^2w\) has more 0's than 1's
    - Thus \(uv^2w \notin L\) CONTRADICTION!

- We arrived at a contradiction,
  - Thus our original assumption that \(L\) is regular must be incorrect.
  - Thus \(L\) is not regular.

Note that we need to find only 1 string \(x\) that fails in order for the proof by contradiction to work.
- The key is finding the \(x\) that won't work

Questions?

Another Example:
- \(L = \{x \in \{0,1\}^* \mid 0^i x, |x| \leq i\}\)
- Let's play
  - Choose an appropriate string \(x \in L\)
    - Let \(x = 0^n1^n\)
  - Apply Pumping Lemma to \(x\)
    - \(x = uvw\)
    - \(|uv| \leq n\)
    - \(|v| > 0\)

- Since \(|uv| \leq n\), \(uv\) must consist entirely of 0s and, as such, \(v\) must also consist entirely of 0s.
  - \(v = 0^j\) for some \(j \leq n\)
The Pumping Lemma states that if $L$ is a regular language, then there exists a string $x$ such that $x = uvw$ with the properties:

- $|uv| \leq n$,
- $|v| > 0$,
- For all $i, j, k \geq 0$, $uv^iwx^jy^k \in L$.

Let's consider the string $x = 0^n1^n$. By the Pumping Lemma, $uv^iwx^jy^k$ is also in $L$ for all $i, j, k \geq 0$. But this contradicts the assumption that $L$ is regular, as we can find an $i$ such that $uv^0wx^0 \notin L$. Thus, $L$ is not regular.

**Non-regular languages**

Informal notion of what regular languages can't express:

- Counting and comparing
- Any operation that implies the use of a stack
  - Pal
  - $xx^r$

**Pumping Lemma**

Let's try another example:

- $L = \{ \text{palindromes over } \{a,b\} \}$
- $x = a^nb^n$
- $|x| = 2n$
- $|uv| \leq n$,
- $|v| > 0$,
- $uv^iwx^jy^k = a^n(ba)^j$ for some $i, j, k \geq 0$.

Since $|uv| \leq n$, $uv$ must consists entirely of $a$ and, as such, $v$ must also consist entirely of $a$. Thus, $L$ is not regular, and we have arrived at a contradiction.

**Pumping Lemma**

Another Example:

- $x = uvw = a^nba^n$
- $aa \ldots a$ $b$ $aa \ldots a$

Since $|uv| \leq n$, $uv$ must consists entirely of $a$ and, as such, $v$ must also consist entirely of $a$. Thus, $L$ is not regular.
Pumping Lemma

- Let's pump!

By the Pumping Lemma

- $uv^2w$ is also in $L$
- $uv^2w$ has more than $n$ a's
- Number of a's following $b$ is still $n$
- Thus $uv^2w$ cannot be a palindrome
- Thus $uv^2w \not\in L$ CONTRADICTION!

We arrived at a contradiction,

- Thus our original assumption that $L$ is regular must be incorrect
- Thus $L$ is not regular.

Note that we need to find only 1 string $x$ that fails in order for the proof by contradiction to work.

Questions?

Test for finiteness

- Statement of the pumping lemma
  - Let $L$ be a regular language.
  - Then there exists a constant $n$ (which varies for different languages), such that for every string $x \in L$ with $|x| \geq n$, $x$ can be expressed as $x = uvw$ such that:
    - $|v| > 0$
    - $|uv| \leq n$
    - For all $k \geq 0$, the string $uv^kw$ is also in $L$.

Test for infiniteness

- First stab
  - The pumping lemma tells us that if there is a string $x$ with length greater than the number of states accepted by an FA, $M$, then $L(M)$ is infinite.
  - Let's test all strings of length $\geq$ number of states.
  - Will give us a "yes", $L(M)$ is infinite but
  - For $L(M)$ finite, the algorithm will never stop.

How does this help us?

- Algorithm for testing if $L(M)$ is infinite.
  - Systematically generate all strings of length between $n$ and $2n$ where $n$ is the number of states of $M$
  - Test each string generated
  - If at least 1 is accepted, then $L(M)$ is infinite
  - Otherwise $L(M)$ is finite.
Pumping Lemma

- **Summary**
  - The pumping lemma formalizes the idea that if a string is long enough, eventually at least one state on the DFA will be have to be repeated on the path that accepts the string.
  - Continually looping on this state will produce an infinite number of strings in the language.
  - Used to show that languages are not regular.
  - Has other uses as well.

Non-regular languages

- **Venn-diagram of languages**
  - Is there something out here?
    - Yes

Break

- **After break**
  - Problem Session...

- **Questions?**