MA/CSSE 474

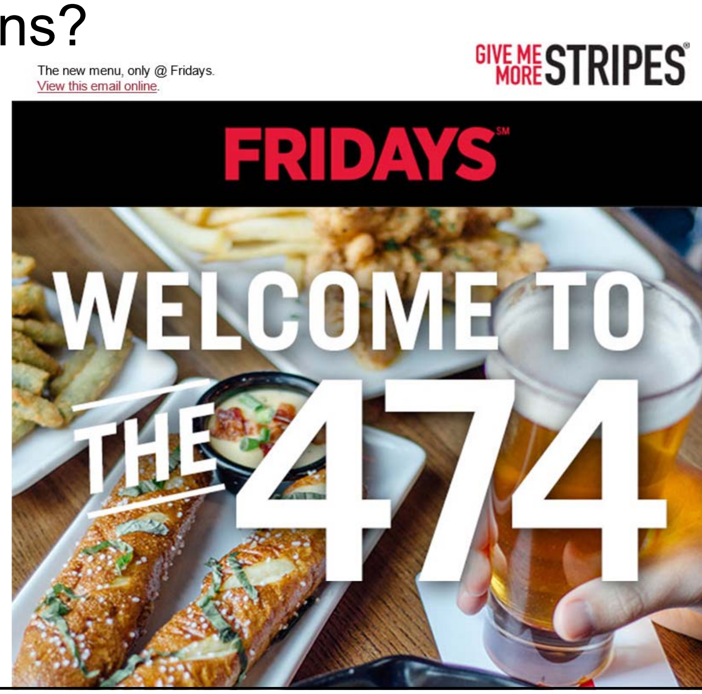
Theory of Computation

Closure properties of Regular Languages

Pumping Theorem

Your Questions?

- Previous class days' material?
- Reading Assignments?
- HW 6 or 7 problems?
- Anything else?



To Show that a Language L is Regular

We can do any of the following:

Construct a DFSM that accepts L.

Construct a NDFSM that accepts L.

Construct a regular expression that defines L.

Construct a regular grammar that generates L.

Show that there are finitely many equivalence classes for $\approx_L$.

Show that L is finite.

Use one or more of the closure properties.

Closure Properties of Regular Languages

- Union
- Concatenation
- Kleene Star
- Complement
- Intersection
- Difference
- Reverse
- Letter Substitution

The first three are easy: definition of regular expressions.

We will briefly discuss the ideas of how to do Complement and Reverse.

Intersection: HW5, or ...

Difference

You should read about Letter Substitution in the textbook.

Don’t Try to Use Closure Backwards

One Closure Theorem:

If L_1 and L_2 are regular, then so is

$L = \quad L_1 \cap L_2$

But if $L_1 \cap L_2$ is regular, what can we say about L_1 and L_2?

$L = L_1 \cap L_2$

ab = ab $\cap$ (a $\cup$ b)* (L1 and L2 are regular)

ab = ab $\cap$ $\{a^n b^n, n \geq 0\}$ (they may not be regular)

Don’t Try to Use Closure Backwards

Another Closure Theorem:

If L_1 and L_2 are regular, then so is

$L = L_1 \; L_2$

But if L_2 is not regular, what can we say about L?

$L = \quad L_1 \; L_2$

$\{aba^n b^n : n \geq 0\} = \{ab\} \{a^n b^n : n \geq 0\}$

L(aaa*) = {a}* $\{a^p$: p is prime$\}$

Showing that a Language is Not Regular

Every regular language can be accepted by some FSM.

It can only use a finite amount of memory to record essential properties.

Example:
$A^nB^n = \{a^nb^n, n \geq 0\}$ is not regular

Showing that a Language is Not Regular

The only way to generate/accept an infinite language with a finite machine/description is to use:

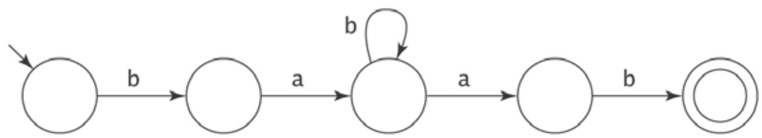
- Kleene star (in regular expressions), or
- cycles (in automata).

This forces some kind of simple repetitive cycle within the strings.

Example:
ab*a generates aba, abba, abbba, abbbba, etc.

Example:
$\{a^n : n \geq 1 \text{ is a prime number}\}$ is not regular.

Exploiting the Repetitive Property



If an FSM with *n* states accepts at least one string of length ≥ *n*, how many strings does it accept?

L = bab*ab

b a | b | a b
x | *y* | *z*

*xy*z* must be in *L*.

So *L* includes: baab, babab, babbab, babbbbbbbbbbab

Theorem – Long Strings

Theorem: Let *M* = (*K*, Σ, δ, *s*, *A*) be any DFSM. If *M* accepts any string of length |*K*| or greater, then that string will force *M* to visit some state more than once (thus traversing at least one loop or cycle).

Proof: *M* must start in one of its states.
Each time it reads an input character, it visits some state. So, in processing a string of length *n*, *M* does a total of
n + 1 state visits.

If *n*+1 > |*K*|, then, by the pigeonhole principle, some state must get more than one visit.

So, if *n* ≥ |*K*|, then *M* must visit at least one state more than once.

The Pumping Theorem* for Regular Languages

If L is regular, then every long string in L is "pumpable".
Formally, if L is regular, then

$\exists k \geq 1$ such that
($\forall$ strings $w \in L$,
($|w| \geq k \rightarrow$
($\exists\, x, y, z$ ($w = xyz$,
$|xy| \leq k$,
$y \neq \varepsilon$, and
$\forall q \geq 0$ (xy^qz is in L)))))

Write this in contrapositive form

- a.k.a. **"the pumping lemma"**.
  We will use the terms interchangeably.
- What if L has *no* strings whose lengths are greater than k?

Using The Pumping Theorem to show that L is not Regular:

We use the contrapositive of the theorem:
If some long enough string in L is not "pumpable", then L is not regular.

What we need to show in order to show L non-regular:
($\forall k \geq 1$
($\exists$ a string $w \in L$
($|w| \geq k$ *and*
($\forall\, x, y, z$ (($w = xyz \wedge |xy| \leq k \wedge y \neq \varepsilon$) $\rightarrow$
$\exists\, q \geq 0$ ($xy^qz \notin L$))))))
$\rightarrow$ L is not regular .

Before our next class meeting:
Be sure that you are convinced that this really is the contrapositive of the pumping theorem.

A way to think of it: adversary argument (following J.E. Hopcroft and J.D.Ullman)

1. Choose the language L you want to prove non-regular.
2. The "adversary" picks k, the constant mentioned in the theorem.
3. We must be prepared for any positive integer to be picked, but once it is chosen, the adversary cannot change it.
4. We select a string $w \in L$ (whose length is at least k) that cannot be pumped".
5. The adversary breaks w into w=xyz, subject to constraints $|xy| \leq k$ *and* $y \neq \varepsilon$.
6. Our choice of w must take into account that any such x and y can be chosen.
7. All we must do is produce a single number $q \geq 0$ such that $xy^qz \notin L$.

Note carefully what we get to choose and
what we do not get to choose.

Example: $\{a^nb^n: n \geq 0\}$ is not Regular

k is the number from the Pumping Theorem.
We don't get to choose it.

Choose *w* to be $a^{\lceil k/2 \rceil}b^{\lceil k/2 \rceil}$ ("long enough").

```
        1                         2
a a a a a … a a a a a b b | b b … b b b b b b
          x             y            z
```

Adversary chooses *x*, *y*, *z* with the required properties:

$|xy| \leq k$,

$y \neq \varepsilon$,

We must show $\exists\, q \geq 0\ (xy^qz \notin L)$.

Three cases to consider:

- *y* entirely in region 1:
- *y* partly in region 1, partly in 2:
- *y* entirely in region 2:

For each case, we must find at least one value of q that takes xy^qz outside the language L.

The most common q values to use are q=0 and q=2.

A Complete Proof (read later)

We prove that $L = \{a^n b^n : n \geq 0\}$ is not regular

If L were regular, then there would exist some k such that any string w where $|w| \geq k$ must satisfy the conditions of the theorem. Let $w = a^{\lceil k/2 \rceil} b^{\lceil k/2 \rceil}$. Since $|w| \geq k$, w must satisfy the conditions of the pumping theorem. So, for some x, y, and z, $w = xyz$, $|xy| \leq k$, $y \neq \varepsilon$, and $\forall q \geq 0$, $xy^q z$ is in L. We show that no such x, y, and z exist. There are 3 cases for where y could occur: We divide w into two regions:

```
aaaaa.....aaaaaa| bbbbb.....bbbbbb
           1    |       2
```

So y is in one of the following :

- (1): $y = a^p$ for some p. Since $y \neq \varepsilon$, p must be greater than 0. Let $q = 2$. The resulting string is $a^{k+p}b^k$. But this string is not in L, since it has more a's than b's.
- (2): $y = b^p$ for some p. Since $y \neq \varepsilon$, p must be greater than 0. Let $q = 2$. The resulting string is $a^k b^{k+p}$. But this string is not in L, since it has more b's than a's.
- (1, 2): $y = a^p b^r$ for some non-zero p and r. Let $q = 2$. The resulting string will have interleaved a's and b's, and so is not in L.

There exists one long string in L for which no pumpable x, y, z exist. So L is not regular.

What You Should Write (read these details later)

We prove that $L = \{a^n b^n : n \geq 0\}$ is not regular

Let $w = a^{\lceil k/2 \rceil} b^{\lceil k/2 \rceil}$. (If not completely obvious, as in this case, show that w is in fact in L.)

```
aaaaa.....aaaaaa| bbbbb.....bbbbbb
       1        |        2
```

There *are three possibilities for y*:

- (1): $y = a^p$ for some p. Since $y \neq \varepsilon$, p must be greater than 0. Let $q = 2$. The resulting string is $a^{k+p}b^k$. But this string is not in L, since it has more a's than b's. .
- (2): $y = b^p$ for some p. Since $y \neq \varepsilon$, p must be greater than 0. Let $q = 2$. The resulting string is $a^k b^{k+p}$. But this string is not in L, since it has more b's than a's.
- (1, 2): $y = a^p b^r$ for some non-zero p and r. Let $q = 2$. The resulting string will have interleaved a's and b's, and so is not in L.

Thus L is not regular.