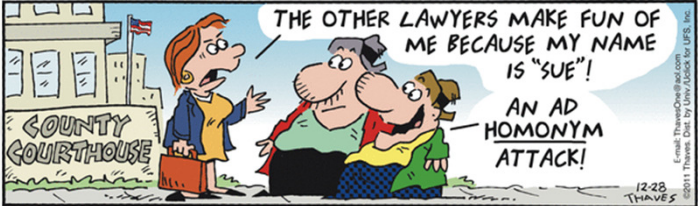# MA/CSSE 474
Theory of Computation

No ICQ today!

More on the Pumping Lemma
Is L Regular?

Exam Results

- The second-highest score on problem 1 was 44/56
  - So I decided to add 12 points to everyone's score for that problem, thus 12 points (out of 130) to the entire exam score.
  - The "bumped up" score is your recorsed score on ANGEL
- Mean exam score was 90 (69.1%)

I sent you my solution soon after the exam. Do you have any questions about the problems?

Grading Scale (from Syllabus)

Label	Minimum Percent
A	86.5
B+	79.5
B	72.5
C+	65.5
C	58.5
F	0

Course averages so far

A	9
B+	14
B	5
C+	11
C	2
F	1

The Pumping Theorem* for Regular Languages

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

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

* a.k.a. "the pumping lemma".
   We will use the terms interchangeably.

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:
$\forall k \geq 1$
$\quad (\exists$ string $w \in L$, where $|w| \geq k$
$\qquad\qquad (\forall x, y, z (w = xyz,$
$\qquad\qquad\qquad |xy| \leq k,$
$\qquad\qquad\qquad y \neq \varepsilon$, and
$\qquad\qquad\qquad \exists q \geq 0 (xy^qz$ is not in $L))))$.

Before next class:
Be sure that you are convinced that this really is the negation of the conclusion of the pumping theorem.

Examples: Use the Pumping Theorem to show that these languages are non-regular

Bal = {$w \in$ {), (}* : the parens are balanced}

PalEven = {$ww^R : w \in$ {a, b}*}

{$a^n b^m : n \geq m$}

{$\text{aba}^n \text{b}^n : n \geq 0$}

Choose $w = \text{aba}^k \text{b}^k$

$$k$$
a b a a a a a a a a a a a b b b b b b b b b b b
   x   *y*   *z*

What are the choices for (*x*, *y*):

(ε, a)
(ε, ab)
(ε, aba$^+$)
(a, b)
(a, ba$^+$)
(aba*, a$^+$)

A Different Approach to $aba^n b^n$?

Can we argue as follows:

We already know that $\{a^n b^n : n \geq 0\}$ is not regular. So neither is L.

Can we defend this argument by appeal to the fact that the regular languages are closed under concatenation:

$L = \quad ab \quad \| \quad \{a^n b^n : n \geq 0\}$
?   regular   not regular

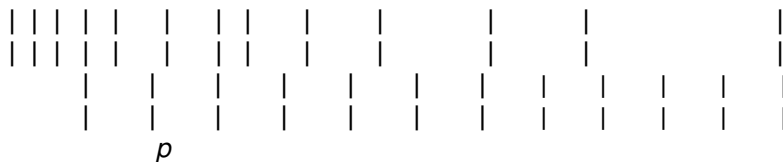# $L = \{a^n : n \text{ is prime}\}$

$L = \{w = a^n : n \text{ is prime}\}$

Let $w = a^j$, where j = the next prime number greater than k:

$$\underbrace{a\ a\ a\ a\ a\ a\ a}_{x}\ \underbrace{a\ a\ a}_{y}\ \underbrace{a\ a\ a}_{z}$$

$|x| + |z|$ is prime.
$|x| + |y| + |z|$ is prime.
$|x| + 2|y| + |z|$ is prime.
$|x| + 3|y| + |z|$ is prime, and so forth.

         p

But the Prime Number Theorem tells us that the primes "spread out", i.e., that the number of primes not exceeding x is asymptotic to $x/\ln x$.

$L = \{a^n: n \text{ is prime}\}$

Let $w = a^j$, where j is the smallest prime number $> k+1$.

$y = a^p$ for some p.

$\forall q \geq 0$ ($a^{|x| + |z| + q|y|}$ must be in L). So $|x| + |z| + q \cdot |y|$ must be prime.

But suppose that $q = |x| + |z|$. Then:

$$|x| + |z| + q \cdot |y| = |x| + |z| + (|x| + |z|) \cdot y$$
$$= (|x| + |z|) \cdot (1 + |y|),$$

which is non-prime if both factors are greater than 1:

$L = \{a^n: n \text{ is prime}\}$

Let $w = a^j$, where j is the smallest prime number $> k+1$.

$y = a^p$ for some p.

$\forall q \geq 0$ ($a^{|x| + |z| + q|y|}$ must be in L). So $|x| + |z| + q \cdot |y|$ must be prime.

But suppose that $q = |x| + |z|$. Then:

$$|x| + |z| + q \cdot |y| = |x| + |z| + (|x| + |z|) \cdot y$$
$$= (|x| + |z|) \cdot (1 + |y|),$$

which is non-prime if both factors are greater than 1:

$(|x| + |z|) > 1$ because $|w| > k+1$ and $|y| \leq k$.
$(1 + |y|) > 1$ because $|y| > 0$.

Using the Pumping Theorem Effectively

- To choose *w*:
  - Choose a *w* that is in the part of *L* that makes it not regular.
  - Choose a *w* that is only barely in *L*.
  - Choose a *w* with as homogeneous as possible an initial region of length at least *k*.

- To choose *q*:
  - Try letting *q* be either 0 or 2.
  - If that doesn't work, analyze *L* to see if there is some other specific value that will work.

Poetry
The Pumping Lemma

Any regular language L has a magic number p
And any long-enough word in L has the following property:
Amongst its first p symbols is a segment you can find
Whose repetition or omission leaves x amongst its kind.

So if you find a language L which fails this acid test,
And some long word you pump becomes distinct from all the
    rest,
By contradiction you have shown that language L is not
A regular guy, resiliant to the damage you have wrought.

But if, upon the other hand, x stays within its L,
Then either L is regular, or else you chose not well.
For w is xyz, and y cannot be null,
And y must come before p symbols have been read in full.

As mathematical postscript, an addendum to the wise:
The basic proof we outlined here does certainly generalize.
So there is a pumping lemma for all languages context-free,
Although we do not have the same for those that are r.e.

-- Martin Cohn

The rest of today's slides

- I doubt that we will get to many of them
- They are here as examples in case you want to start on HW 7 before tomorrow (which is not a bad idea!)
- Tomorrow we will do some of the following examples, but not all of them
- Most of them are discussed in detail in the textbook

Using the Closure Properties

The two most useful properties are closure under:

- Intersection

- Complement

Using the Closure Properties

$L = \{w \in \{a, b\}^* : \#_a(w) = \#_b(w)\}$

If L were regular, then:

  $L' = L \cap$ _____

would also be regular. But it isn't.

$L = \{a^i b^j : i, j \geq 0 \text{ and } i \neq j\}$

Try to use the Pumping Theorem by letting $w = a^k b^{k+k!}$.

Then $y = a^p$ for some nonzero p.

Let $q = (k!/p) + 1$ (i.e., pump in $(k!/p)$ times).

Note that $(k!/p)$ must be an integer because $p < k$.

The number of a's in the new string is $k + (k!/p)p = k + k!$.

So the new string is $a^{k+k!} b^{k+k!}$, which has equal numbers of a's and b's and so is not in L.

$L = \{a^i b^j: i, j \geq 0 \text{ and } i \neq j\}$

An easier way:

If L is regular then so is $\neg L$. Is it?

$L = \{a^i b^j: i, j \geq 0 \text{ and } i \neq j\}$

An easier way:

If L is regular then so is $\neg L$. Is it?

$\quad \neg L = A^n B^n \cup \{\text{out of order}\}$

$\quad$ If $\neg L$ is regular, then so is $L' = \neg L \cap a^* b^*$

$$= \underline{\qquad\qquad}$$

$L = \{a^i b^j c^k:\ i, j, k \geq 0$ and (if $i = 1$ then $j = k$)$\}$

Every string in L of length at least 1 is pumpable.

But is L regular?

$L = \{a^i b^j c^k:\ i, j, k \geq 0$ and (if $i = 1$ then $j = k$)$\}$

Every string in L of length at least 1 is pumpable.

Rewrite the final condition as: $(i \neq 1)$ or $(j = k)$

$L = \{a^i b^j c^k: \ i, j, k \geq 0 \text{ and } (i \neq 1 \text{ or } j = k)\}$

Every string in L of length at least 1 is pumpable:

•If $i = 0$ then: if $j \neq 0$, let y be `b`; otherwise, let y be `c`. Pump in or out. Then i will still be 0 and thus not equal to 1, so the resulting string is in L.

•If $i = 1$ then: let y be `a`. Pump in or out. Then i will no longer equal 1, so the resulting string is in L.
•
•If $i = 2$ then: let y be `aa`. Pump in or out. Then i cannot equal 1, so the resulting string is in L.

•If $i > 2$ then: let $y = $ `a`. Pump out once or in any number of times. Then i cannot equal 1, so the resulting string is in L.

$L = \{a^i b^j c^k: \ i, j, k \geq 0 \text{ and } (i \neq 1 \text{ or } j = k)\}$

But the closure theorems help. Suppose we guarantee that $i = 1$. If L is regular, then so is:

> $L' = L \cap$ `ab*c*`.

> $L' = \{$ `ab`^{j}`c`$^k : j, k \geq 0 \text{ and } j = k\}$

Use Pumping to show that L' is not regular:

$L = \{a^ib^jc^k: \ i, j, k \geq 0 \text{ and } (i \neq 1 \text{ or } j = k)\}$
An Alternative

If L is regular, then so is L^R:

$L^R = \{c^kb^ja^i : i, j, k \geq 0 \text{ and } (i \neq 1 \text{ or } j = k)\}$

Use Pumping to show that L' is not regular:

Exploiting Problem-Specific Knowledge

$L = \{w \in \{0, 1, 2, 3, 4, 5, 6, 7\}^*$:
      w is the octal representation of a
      nonnegative integer that is divisible by 7}

Is English Regular?

Is English finite?

In the event that the Purchaser defaults in the payment of any installment of purchase price, taxes, insurance, interest, or the annual charge described elsewhere herein, or shall default in the performance of any other obligations set forth in this Contract, the Seller may: at his option: (a) Declare immediately due and payable the entire unpaid balance of purchase price, with accrued interest, taxes, and annual charge, and demand full payment thereof, and enforce conveyance of the land by termination of the contract or according to the terms hereof, in which case the Purchaser shall also be liable to the Seller for reasonable attorney's fees for services rendered by any attorney on behalf of the Seller, or (b) sell said land and premises or any part thereof at public auction, in such manner, at such time and place, upon such terms and conditions, and upon such public notice as the Seller may deem best for the interest of all concerned, consisting of advertisement in a newspaper of general circulation in the county or city in which the security property is located at least once a week for Three (3) successive weeks or for such period as applicable law may require and, in case of default of any purchaser, to re-sell with such postponement of sale or resale and upon such public notice thereof as the Seller may determine, and upon compliance by the Purchaser with the terms of sale, and upon judicial approval as may be required by law, convey said land and premises in fee simple to and at the cost of the Purchaser, who shall not be liable to see to the application of the purchase money; and from the proceeds of the sale: First to pay all proper costs and charges, including but not limited to court costs, advertising expenses, auctioneer's allowance, the expenses, if any required to correct any irregularity in the title, premium for Seller's bond, auditor's fee, attorney's fee, and all other expenses of sale occurred in and about the protection and execution of this contract, and all moneys advanced for taxes, assessments, insurance, and with interest thereon as provided herein, and all taxes due upon said land and premises at time of sale, and to retain as compensation a commission of five percent (5%) on the amount of said sale or sales; SECOND, to pay the whole amount then remaining unpaid of the principal of said contract, and interest thereon to date of payment, whether the same shall be due or not, it being understood and agreed that upon such sale before maturity of the contract the balance thereof shall be immediately due and payable; THIRD, to pay liens of record against the security property according to their priority of lien and to the extent that funds remaining in the hands of the Seller are available; and LAST, to pay the remainder of said proceeds, if any, to the vendor, his heirs, personal representatives, successors or assigns upon the delivery and surrender to the vendee of possession of the land and premises, less costs and excess of obtaining possession.

Is English Regular?

- The rat ran.
- The rat that the cat saw ran.
- The rat that the cat that the dog chased saw ran.

Let:
A = {cat, rat, dog, bird, bug, pony}
V = {ran, saw, chased, flew, sang, frolicked}.

Let L = English $\cap$ {The A (that the A)* V* V}.

L = {The A (that the A)^n V^n V, $n \geq 0$}.

Let w = The cat (that the rat)^k saw^k ran.

Defining Functions from one Language to Another

Let *firstchars*(L) =
  {$w : \exists y \in L$
      ($y = cx$,
        $c \in \Sigma_L$,
        $x \in \Sigma_L^*$, and
        $w \in c^*$)}

Are the regular languages closed under *firstchars*?

L	*firstchars*(L)
$\varnothing$	
a*b*	
ca*cb*	

Defining Functions from one Language to Another

Let $chop(L) =$
   $\{w : \exists x \in L$
         $(\ x = x_1 c x_2,$
            $x_1 \in \Sigma_L{}^*,$
            $x_2 \in \Sigma_L{}^*,$
            $c \in \Sigma_L,$
            $|x_1| = |x_2|,$ and
            $w = x_1 x_2)\}$

Are the regular languages closed under *chop*?

L	*chop*(L)
∅	
a*b*	
a*db*	

Defining Functions from One Language to Another

Let $maxstring(L) =$
   $\{w: w \in L,$
      $\forall z \in \Sigma^* \ (z \neq \varepsilon \rightarrow wz \notin L)\}$

Are the regular languages closed under *maxstring*?

L	*maxstring*(L)
∅	
a*b*	
ab*a	
a*b*a	

Defining Functions from One Language to Another

Let $mix(L) =$
   $\{w: \exists x, y, z$
      $(\ x \in L,$
   $x = yz,$
   $|y| = |z|,$
   $w = yz^R)\}.$

Are the regular languages closed under *mix*?

L	$mix(L)$
$\varnothing$	
$(a \cup b)^*$	
$(ab)^*$	
$(ab)^*a(ab)^*$	