Regular and Nonregular Languages

Chapter 8

Languages: Regular or Not?

$\texttt{a}^*\texttt{b}^*$ is regular.
$\{\texttt{a}^n\texttt{b}^n : n \geq 0\}$ is not.

$\{w \in \{\texttt{a}, \texttt{b}\}^* :$ every $\texttt{a}$ is immediately followed by $\texttt{b}\}$ is regular

$\{w \in \{\texttt{a}, \texttt{b}\}^* :$ every $\texttt{a}$ has a matching $\texttt{b}$ somewhere$\}$ is not

- Showing that a language is regular
- Showing that a language is not regular

How Many Regular Languages?

Theorem: There is a **countably infinite** number of regular languages.

Proof:

- Lower bound on number of regular languages:

  {a},{aa},{aaa},{aaaa},{aaaaa},{aaaaaa},…

  are all regular. That set is countably infinite.

- Upper bound on number of regular languages: number of FSMs or number of regular expressions

Regular And Nonregular Languages?

There is a countably infinite number of regular languages.

There is an uncountably infinite number of languages

- uncountably infinite number of languages over any nonempty alphabet Σ.
- p23-25 from ch2.ppt

So there are *many* more nonregular languages than there are regular ones.

Showing that a Language is Regular

Theorem: Every finite language is regular.

Proof: If L is the empty set, then it is defined by the regular expression $\varnothing$ and so is regular. If it is any finite language composed of the strings s_1, s_2, … s_n for some positive integer n, then it is defined by the regular expression:

$$s_1 \cup s_2 \cup \ldots \cup s_n$$

So it too is regular.

A Finite Language We May Not Be Able to Write Down

- L_1 = {$w \in$ {0 - 9}*: w is the social security number of a living US resident}.
- Finite, thus regular. But the techniques we have described for recognizing regular languages would not be very useful in building a program to check for a valid SSN.

- REs are most useful when the elements of L share some repeating patterns.
- FSMs are most useful when the elements of L share some simple structural properties.
- They don't bring much to the table when the language is just a set of unrelated strings.

- Other techniques, e.g., hash tables, are better suited to handling finite languages whose elements are chosen by our world, not by rule.

- Most useful languages are infinite
- Regular: **regularity, repeat, pattern, cycle, loop, recursion** …

Regular Does Not Always Mean Easy: Size of Input Matters

The towers of Hanoi: a stack of 64 disks; move to the last pole. One at a time. Cannot be placed on top of a smaller one. No held or ground.

Let Σ = {12, 13, 21, 23, 31, 32}.

- 12 means the top disk from pole 1 is removed and placed on pole 2

Let *L* be the language of strings that correspond to successful move sequences. The shortest string in *L* has length 2^{64} -1.

- If 1 sec per move, it'd take about 600000000000 years.
- recursion

There is an FSM that accepts *L*

- The number of distinct states is finite
- Can easily build an NDFSM for it

- ToH is intractable, FSM is linear … where went wrong?

Showing That *L* is Regular

1. Show that *L* is finite.

2. Exhibit a regular expression for *L*.

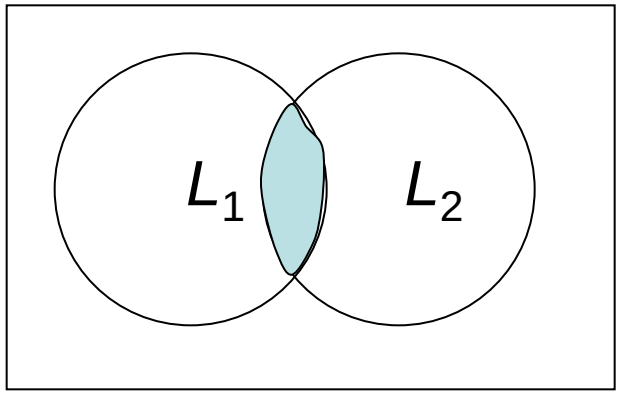
3. Exhibit an FSM for *L*.

4. Exhibit a regular grammar for *L*.

5. Exploit the closure theorems.

Closure Properties of Regular Languages

- Union
- Concatenation
- Kleene star
- Complement
- Intersection
- Difference
- Reverse
- Letter substitution



If we can show that *L* can be constructed from other regular languages using those operations, then *L* must also be regular

Divide-and-Conquer

Let $L = \{w \in \{a, b\}^* : w$ contains an even number of a's and an odd number of b's and all a's come in runs of three$\}$.

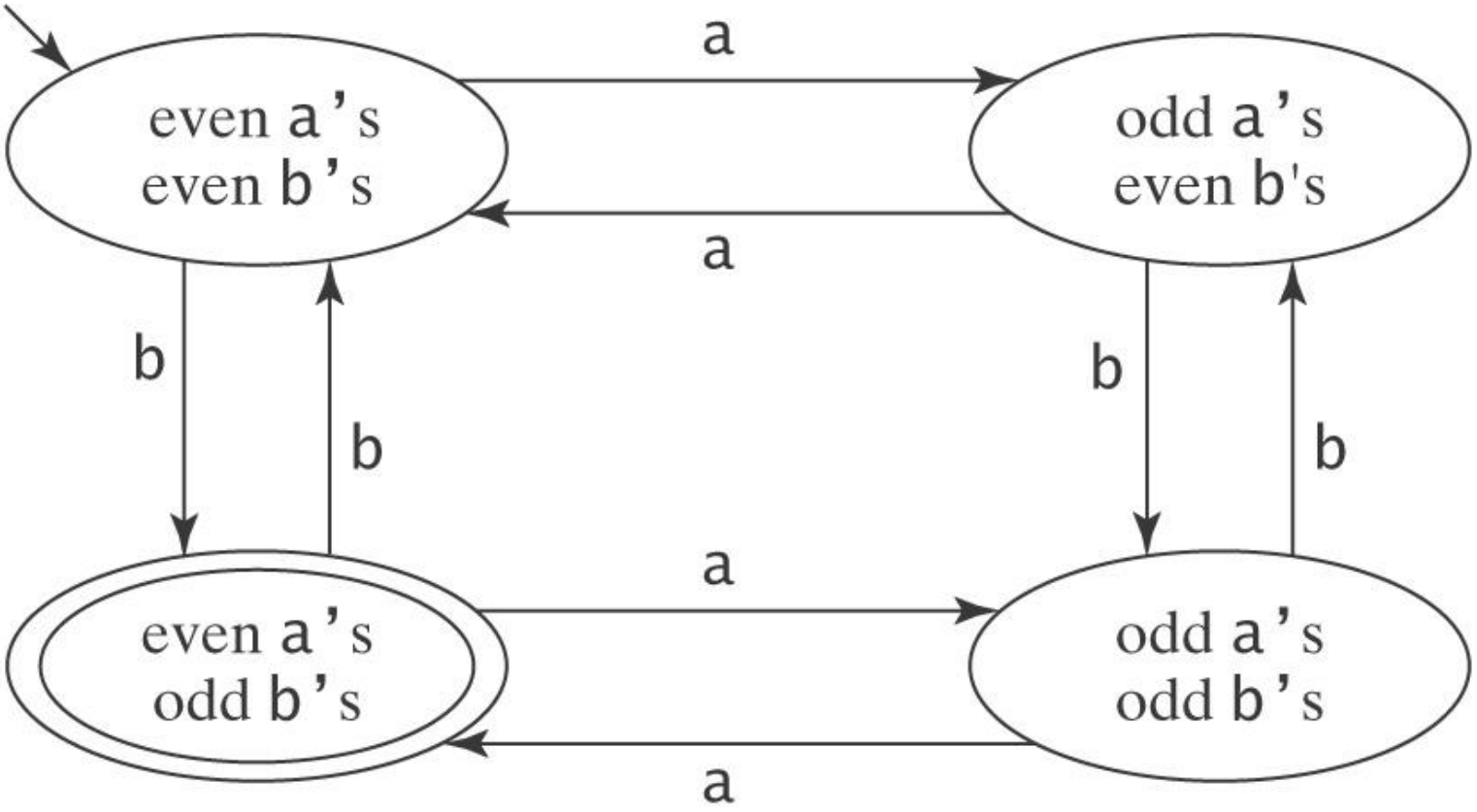
aaabaaa, aaabbbaaaaaa

$L = L_1 \cap L_2$, where:

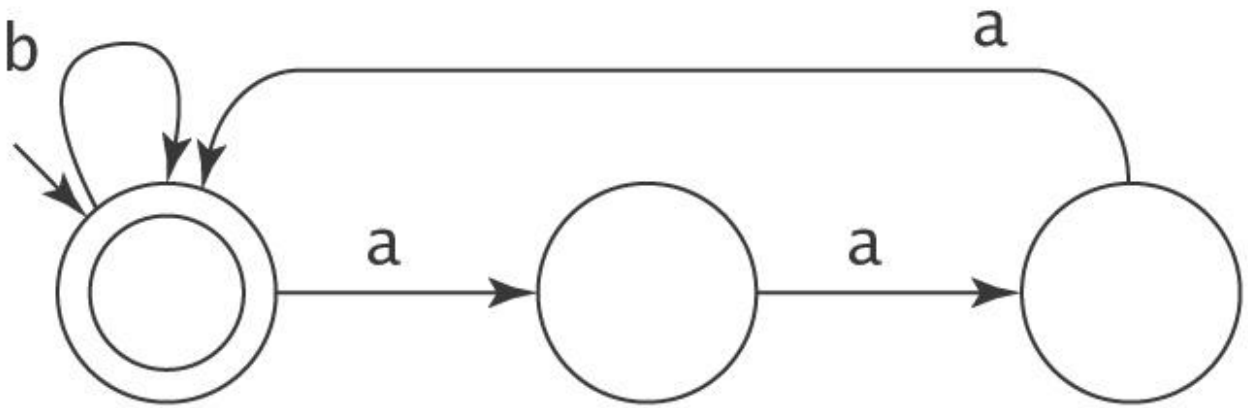
- $L_1 = \{w \in \{a, b\}^* : w$ contains an even number of a's and an odd number of b's$\}$, and

- $L_2 = \{w \in \{a, b\}^* :$ all a's come in runs of three$\}$

L_1 is Regular



L_2 is Regular



Don't Try to Use Closure Backwards

One Closure Theorem:

If L_1 and L_2 are regular, then so is

$$L = L_1 \cap L_2$$

But if L is regular, what can we say about L_1 and L_2?

$$L = L_1 \cap L_2$$

$\texttt{ab} = \texttt{ab} \cap (\texttt{a} \cup \texttt{b})^*$ (regular)

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

Showing that a Language is Not Regular

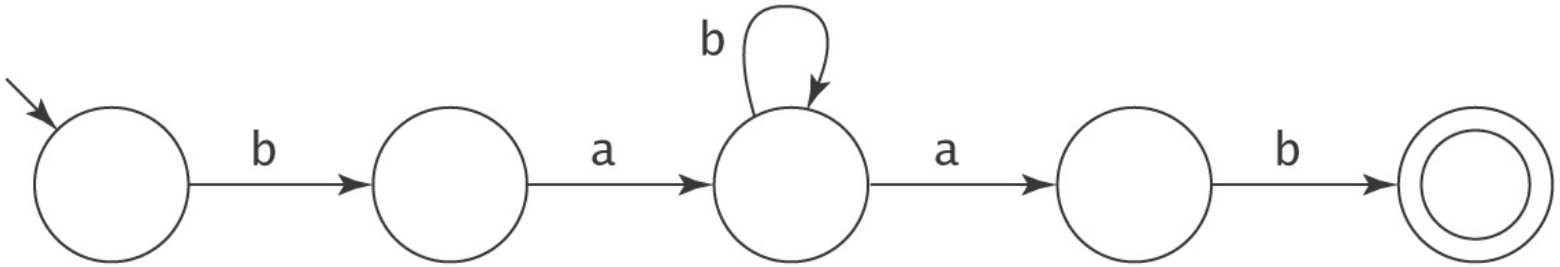
The only way to generate/accept an infinite language with a finite description is to use Kleene star (in regular expressions) or **cycles** (or loops, in automata). This forces some kind of simple repetitive cycle within the strings.

Example:

`ab*a` generates `aba`, `abba`, `abbba`, `abbbba`, etc.

Every regular language can be accepted by some FSM.
It can only use a finite amount of memory to record essential properties.

Longest string that can be accepted without going through loops?



• 4. If longer, there must be a loop that you need to go through. And, by repeating the loop, you can generate strings in L.

• If there's a long string in L without such property, L is not regular

Theorem – Long Strings

Theorem: Let $M = (K, \Sigma, \delta, s, A)$ 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).

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 creates 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 $n \geq |K|$, 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.

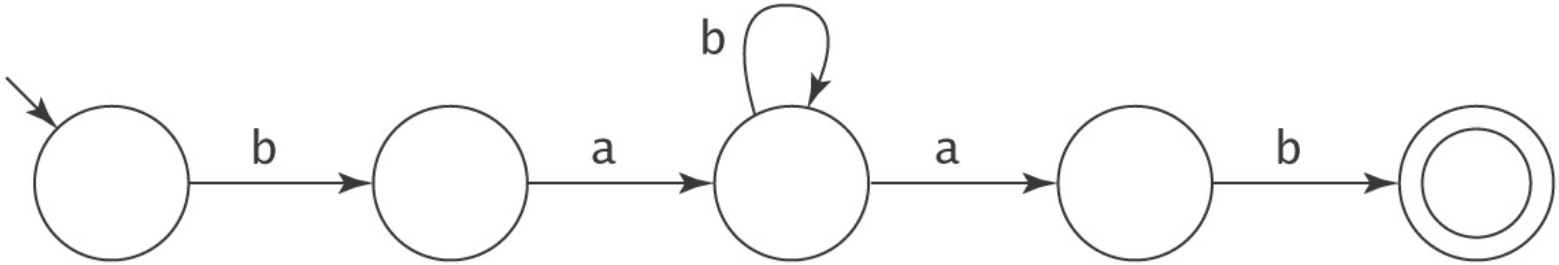
So, $\exists k \geq 1$

$(\forall$ strings $w \in L$, where $|w| \geq k$

$(\exists\ x, y, z\ (w = xyz,$
$|xy| \leq k,$
$y \neq \varepsilon$, and
$\forall q \geq 0\ (xy^q z$ is in $L))))$.

- Also referred to as pumping lemma.
- Intuitively, if L is regular, there exists an upper bound (k) on the length of the strings in L without going through any loop.
- If longer, there must be a loop (y) in the prefix of length k, such that you can keep repeating y to obtain long strings in L.
- If you cannot find such a y in the prefix of length k, then L is not regular

Example: Regular



L = `bab*ab`

`b a b b b b a b`

x y z

upper bound is 4. within prefix of length 4, there must be a y, such that xy*z is in L.

`babbbab`, `babbbbbab`, `babbbbbbbab`, …

Another Regular Example

L = {b,bb,bbb,bbbb}

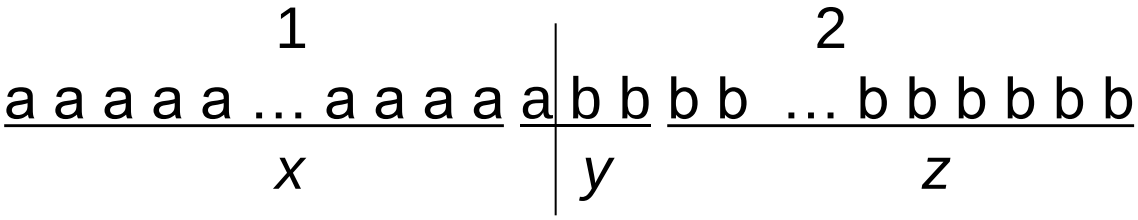
L is regular, how to show the pumping lemma applies?

Let k = 5, the number of long strings is 0, so “every long string is pumpable” is true.

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

k is the number from the Pumping Theorem.

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



We show that there is no x, y, z with the required properties:

$|xy| \leq k$,

$y \neq \varepsilon$,

$\forall\ q \geq 0$ ($xy^q z$ is in L).

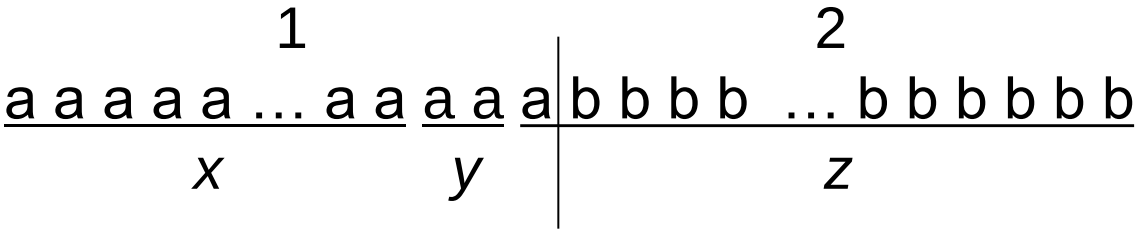
Three cases to consider:

- y falls in region 1: (more a's than b's)
- y falls across regions 1 and 2: (interleaved a's and b's)
- y falls in region 2: (more b's than a's)

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

Second try: an easier way

Choose w to be $a^k b^k$ (We get to choose any w).



We show that there is no x, y, z with the required properties:

$|xy| \leq k$,

$y \neq \varepsilon$,

$\forall\ q \geq 0$ ($xy^q z$ is in L).

Since $|xy| \leq k$, y must be in region 1. So $y = a^p$ for some $p \geq 1$.

Let $q = 2$, producing:

$$a^{k+p} b^k$$

which $\notin L$, since it has more a's than b's.

A Complete Proof

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 can fall in:

- (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.

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

Using the Pumping Theorem

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

To show that L is not regular, we find one that isn't.

To use the Pumping Theorem to show that a language L is not regular, we must:

1. Choose a string w where $|w| \geq k$. Since we do not know what k is, we must state w in terms of k.
2. Divide the possibilities for y into a set of equivalence classes that can be considered together.
3. For each such class of possible y values where $|xy| \leq k$ and $y \neq \varepsilon$:

   Choose a value for q such that xy^qz is not in L.

Poetry: The Pumping Lemma

Any regular language *L* has a magic number *p* //we used *k*
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.

//r.e. (recursively enumerable), refers to the semidecidable class of languages

-- Martin Cohn / Harry Mairson;
Computer Science Department; Brandeis University

More Examples

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

- balanced parenthesis

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

- even length palindrome

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

- more a's than b's

L = {a^n: n is prime}

- prime number of a's

Using the Closure Properties

Can be used alone, or together with the Pumping theorem to prove non-regularity

The two most useful ones are closure under:

- Intersection
- Complement

Using the Closure Properties

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

- $\#_{\boldsymbol{c}}(s)$ is the number of times that c occurs in s.
  - e.g., $\#_{\mathtt{a}}(\mathtt{abbaaa}) = 4$.

If L were regular, then:

$L' = L \cap \mathtt{a}^*\mathtt{b}^*$

would also be regular, since a*b* is regular.

But $L' = L \cap \mathtt{a}^*\mathtt{b}^* = \{\mathtt{a}^n\mathtt{b}^n : n \geq 0\}$, it isn't regular.

Is English Regular?

Is English finite?

If we assume there's a longest sentence, then English is regular because it is finite;

But someone can always make a longest sentence even longer …

The following example shows that there are sentences that are much longer than whatever upper limit you probably have in mind …

- 630 words, announces a local government's intention to move a path.

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.

If Not Bounded, Not 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.

Summary of FSM and RL

- Theoretically, every machine we build is a finite state machine.
- But not every real problem should be described as a regular language or solved with an FSM.
- FSMs and regular expressions are powerful tools for describing problems that possess certain repetitive patterns.
- More powerful models to come (PDAs and Turing machines)
    - equipped with infinite storage devices, but useful even if there exists practical upper bound on sizes of memory and input