1/27/11

Pumping Lemma

How to prove that a language is regular?

- build a DFA or NFA
- build a regular expression
- use closure properties to reduce to other languages known to be regular

How to prove that a language L is not regular?

- use closure properties to reduce from another language that is not regular

  e.g., if K is not regular and $L = \overline{K} = \{0,1\}^* \setminus K$, then L is not regular either
- but to get started, we can use the Pumping Lemma

Key point: DFAs are finite, ie., have a finite number of states

This is useful, eg.,

$L = \{$ any particular finite set of strings $\}$

is regular (eg., the texts of all books in the Library of Congress)

By closure under complement and intersection, you can also add or remove any finite set of strings from a regular language, and still have a regular language.

But it is also their Achilles heel.

Finite set of states $\Longleftrightarrow$ Finite memory

$\Updownarrow$

Any long enough input string will cause the machine to cycle.

<u>Example</u>: $L = \{0^n 1^n \mid n \geq 0\}$

$= \{\varepsilon, 01, 0011, 000111, \ldots\}$

is <u>not</u> a regular language.

Here is a computer program that tries to decide this language:

```
n = 0
while (input remains and next bit is 0):
    n += 1  and step forward
while (input remains and next bit is 1):
    n -= 1  and step forward
if (no input remains and n == 0):
    ACCEPT
else REJECT
```

On any real computer, this program will <u>not</u> accept exactly L. For long enough input strings, the register storing n will <u>overflow</u>. It only works for small-enough inputs.

32 bits	$\Rightarrow$ overflow after 2^{32}
1 TB	$\Rightarrow$ overflow after $2^{8 \cdot 2^{30}}$

We will see that finite-state machines suffer the same problem:

Proof: Assume L is regular. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA accepting $L(M) = L$. Let $N = |Q|$.

Consider the strings $\{\varepsilon, 0, 00, 000, \ldots, 0^N\}$. Since there are $N+1$ different strings and only N states, the machine must take two of them to the same place, i.e., $\exists\, m, n \in \{0, 1, 2, \ldots, N\}$, $m \neq n$, s.t.

$$\hat{\delta}(q_0, 0^m) = \hat{\delta}(q_0, 0^n).$$

By assumption, $0^m 1^m \in L(M)$ so $\hat{\delta}(q_0, 0^m 1^m) \in F$.

$$\Rightarrow \hat{\delta}(q_0, 0^n 1^m) = \hat{\delta}(\hat{\delta}(q_0, 0^n), 1^m) = \hat{\delta}(q_0, 0^m 1^m) \in F$$

Thus $0^n 1^m \in L(M)$, a contradiction since $0^n 1^m \notin L$. $\lightning$ $\square$

This argument generalizes.

<u>Theorem</u> ("Pumping Lemma"):

Let L be a regular language.

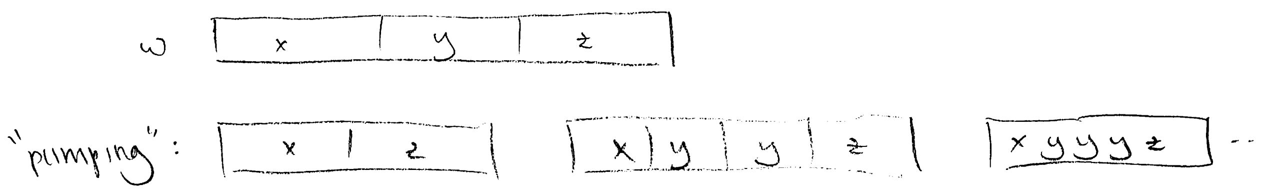
<u>Then</u> there exists a constant n such that:

- For every $w \in L$ of length $|w| \geq n$,

  w can be written $w = xyz$ such that

  1. $xy^jz \in L$ for all $j \geq 0$
  2. $|y| > 0$
  3. $|xy| \leq n$.

$|y| > 0$
$|xy| \leq n$
if $w \in L$ then $xy^jz \in L$



note: $|y| > 0$. (If $|y| = 0$ were a possibility, then the claim would be vacuous, $x\varepsilon^j z = xz$ $\forall j$.)

- Third condition says that "pumping" can start near the beginning of w. (This is often not needed, but is sometimes convenient.)

<u>Remember</u> this theorem, especially the first two conditions.

Proof: We prove

<u>Lemma</u>: Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA with $L(M) = L$.

Then the above conditions hold with $n = |Q| = \#$ of states.

If L has no strings of length $\geq n$, then claim is vacuous.

Otherwise let $w \in L$ with $|w| \geq n$.

$$q_0 \xrightarrow[w_1]{} q_1 \xrightarrow[w_2]{} q_2 \xrightarrow[w_3]{} \cdots \xrightarrow[w_n]{} q_n$$

Let $q_j = \hat{\delta}(q_0, w_1 w_2 \cdots w_j)$.

Since only n different states, two q_k must be the same, i.e.

$q_i = q_j$ for some $0 \leq i < j \leq n$.

Let $x = w_1 \cdots w_i$, $y = w_{i+1} \cdots w_j$, $z = w_{j+1} \cdots w_n$

Since $q_i = q_j$

The loop can be repeated as often as we like (or 0 times) $\square$

Corollary: $\{0^n 1^n\}$ is not regular.

Proof by Pumping Lemma:

Assume otherwise. Let N be the constant we get.

$w = 0^N 1^N \in L$

$= xyz$ with $|xy| \le N$, $|y| > 0$

thus $x = 0^i$, $y = 0^j$, $j > 0$, $z = 0^{N-i-j} 1^N$

Apply the pump zero times.

$\Rightarrow xz \in L$

$= 0^i 0^{N-j-i} 1^N = 0^{N-j} 1^N \in L$ ↯ $\square$

Note third condition not necessary in this case:

$$00 \cdots \underbrace{011}_{y} \cdots 1$$

Pumping y two or more times would still give a contradiction.

The more languages you know regular, easier to show more

More you know not regular, easier to show.

Corollary: $\{w \mid n_0(w) = n_1(w)\}$ is not regular.

Proof attempt 1: Let N be const from pumping lemma, assuming o.w.

consider $w = 010101 \cdots 01$

$= (01)^N$.

By lemma, can split $w = xyz$ s.t. $xy^jz \in L$.

But we do not get to choose this decomposition!

Could be, e.g. $x = \varepsilon$, $y = 01$, $z = (01)^{N-1}$.

Then $xy^jz = (01)^{N+j-1} \in L$. Not a contradiction!

Need to choose a better string to start with.

Proof attempt 2:

Try $w = 0^{N/2} 1^{N/2}$

By pumping can split it up $w = xyz$.

What if $y = 01$, $x = 0^{N/2-1}$, $z = 1^{N/2-1}$?

$\Rightarrow xy^jz \in L$ Not a contradiction!

Need to use the third condition.

Proof: $w = 0^N 1^N$, and as before. $\square$

Alternative proof: By intersection with $L(0^* 1^*)$ ...

Some good examples:

1. $L = \bigcup_{\substack{n \geq 0: \\ n \text{ is } \underline{\text{prime}}}} \Sigma^n$ is $\underline{\underline{\text{not}}}$ regular

Proof: Assume otherwise. Let $h : \Sigma \to \{0\}$ be $h(a) = 0 \; \forall a$.

Then $h(L) = \{0^n \mid n \text{ is prime}\}$.

If L is regular, then so is $h(L)$.

Now apply pumping lemma $\Rightarrow$ some N

start with 0^p : $p \geq N$ a prime

$m + k + \ell = p$

pumping $\Rightarrow 0^m (0^k)^{m+\ell} 0^\ell \in h(L)$

$= 0^{(k+1)(m+\ell)}$ ↯ $\square$

2. $\{0^m 1^n \mid m \geq n\}$

Proof: Reverse it. Then pump.

3. $\{0^{n^3} 1^n\}$

$\therefore \{a^k b c^\ell : k \neq \ell^3\}$ is not regular.

Proof: Assume otherwise.

homomorphism to eliminate b.

Complement the language, intersect with $a^* c^*$

gives $\{a^{\ell^3} c^\ell\}$, homomorphism.

Intuition is that if the DFA needs to count, the language is $\underline{\text{not}}$ regular. $\underline{\underline{\text{But}}}$

4. $\frac{1}{2}(L) = \{x \mid \text{for some } w \text{ with } |w| = |x|, \; xw \in L\}$

$\underline{\underline{\text{is}}}$ regular if L is regular

start with DFA $M = (Q, \Sigma, \delta, q_0, F)$

Proof: ε-NFA $N = (Q \times Q \cup \{start\}, \Sigma, \rho, start, \{(q,q) : q \in Q\})$

$\rho(start, \varepsilon) = \{q_0\} \times F$

$\rho((q,q'), a) = \{\delta(q,a)\} \times \{p : \delta(p,a) = q'\}$ ✓

5. quotient $\{w : wa \in L\}$

6. derivative $\{w : aw \in L\}$ (using closure under reversal and quotient)

$\{x \in \{0,1\}^* \mid \text{number of } 01 \text{ substrings} = \text{number of } 10 \text{ substrings}\}$ $\underline{\underline{\text{is}}}$ regular.

no counting necessary b/c can't have two 01's without a 10.