The Non-Regularity of an Even-Length Palindrome with Suffix

Zach Tomaszewski

ICS641

16 Mar 2012

Regular Languages

- Important class of simple languages
- A number of equivalent formulations
  - Defined in terms of regular sets
  - Produced by a regular grammar
  - Matched or described by a regular expression
  - Accepted by a finite automata
- Knowing you have a regular language lets you apply the accumulated knowledge about regular languages
- Often just as important to know a language is *not* regular

Proving a Language Non-Regular

- Two common techniques:
  - pumping lemma (more common)
  - Myhill-Nerode theorem

Overview

- Pumping lemma
- Use pumping lemma to prove the language of even-length palindromes ($L = rr^R$) is non-regular.
- Show that pumping lemma cannot be used for a similar language ($L_s = rr^R s$)
- Myhill-Nerode theorem
- Prove L_s non-regular with Myhill-Nerode

Pumping Lemma

- Based on definition of finite automata
- Simply: If some machine M accepts a string with more symbols in it than the number of states in M, then M must have repeated ("pumped") a state.

Pumping Lemma - Formal

- L is a regular language
- p is the pumping length
- for any string w in L where $|w| >= p$, then w may be divided into 3 pieces, $w = xyz$, such that:
  - $|xy| <= p$
  - $|y| >= 1$
  - for all $i >= 0$, xy^iz is in L

Pumping Lemma

- Proofs often take the form of an adversary argument
  - theoretical adversary chooses *p* and the division of $w = xyz$.
- Used to prove a number of languages non-regular, such as 0^n1^n and *xx*
  - see paper for more details
- Note that pumping lemma is only a necessary condition of regularity
  - even when proving non-regular, the ability to pump is inconclusive; never proof of regularity

Example: Even-length Palindrome

- $L = \{rr^R \mid r$ is in $\{0, 1\}+\}$ is non-regular
- Proof:
  - Assume L is regular. *p* is the pumping length.
  - $w = \{0^p110^p\}$ (w is in *L* and $|w| >= p$)
  - So can: $w = xyz$ according to the pumping lemma.
  - Since $|xy| <= p$, any selection for *y* must consist of only 0s.
  - Pumping *y* (as in xy^2z) would result in more 0s before the 11 portion of *w* than after. Thus the pumped string is not in *L.*
  - Assumpition -> contradiction. So: L is non-regular

Addition of an arbitrary suffix

- $L_s = \{rr^R s \mid r, s \text{ is in } \{0, 1\}+\}$
- Now very hard to make "unpumpable", because boundaries (esp. of *s*) can be "redrawn" so result still in L_s.
- Example: $w = \{0^p 1 1 0^p 1\}$.
- Again, pump 0s in initial 0^p portion of string...
- But: 0 0 0000...110000....1
  r r^R |---------*s*-----------|
- So resulting string still in L_s

L_s is always pumpable

- $L_s = \{rr^R s \mid r, s \text{ is in } \{0, 1\}+\}$.
- w is any string in L
- Min $|w|$ is 3.
- Any $|w| > 3$ and $p > 3$ can be pumped.
- Let $r = a_1 \ldots a_\text{n}$ (a, b in {0, 1})
- Let $r^R = a_n \ldots a_1$
- Let $s = b_1 \ldots b_\text{n}$
- So: $w = rr^R s = a_1 \ldots a_\text{n}\, a_n \ldots a_1\, b_1 \ldots b_\text{n}$

L_s is always pumpable

- Case 1: $|r| > 1$ - pump first symbol in r
  - $x = \varepsilon$, $y = a_1$, $z = a_2 \ldots a_n a_n \ldots a_1 b_1 \ldots b_n$
  - Any $xy^i z$ is pumpable.
  - $i = 0$: Move the bounary of original s left by one.

    $r' = a_2 \ldots a_n$ $\quad r'^R = a_n \ldots a_2$ $\quad s' = a_1 b_1 \ldots b_n$
  - i > 1: Duplicated a_1 becomes the new r and r^R, with any extra a_1s falling into the new s

    $r' = a_1$ $\quad r'^R = a_1$ $\quad s' = a_1^{i-2} a_2 \ldots a_n b_1 \ldots b_n$

L_s is always pumpable

- Case 2: $|r| = 1$ - pump first symbol in *s*
  - $|rr^R| = 2$, which is < p (p > 3) So *s* is "reachable".
  - $|s| >= 2$, since $|w| > 3$. So at least: $b_1 b_2$...
  - $x = rr^R$, $y = b_1$, $z = b_2 \ldots b_n$
  - Any xy^iz is pumpable.
  - $i = 0$: *s* portion of string still exists (b_2)
  - i > 1: Just adds extra b_1s to the start of the new *s*
  - In either case, original rr^R is left untouched.

L_s is always pumpable

- So adversary can always find a pumpable string for any non-trivial *p*.
- Does this mean L_s is regular then?
- No. Just means we can't use pumping lemma to prove it non-regular.

Myhill-Nerode Theorem

- Provides both necessary and sufficient conditions for regular language
- Simply: Imagine a finite automata *M* reads in either string *x* or string *y*. If any subequent string *z* over the alphabet (Σ*) leads to the same state in *M*, then *x* and *y* are effectively "indistinguishable" or equivalent. Since M has a fininte number of states, there are a finite number of such equivalence classes.
- Myhill-Nerode is the basis for minimizing any FA to single minimized version.

Myhill-Nerode Theorem - Formal

The following three statements are equivalent:

1) The set *L*, as a subset of Σ*, is accepted by some finite automata.

2) *L* is the union of some of the equivalence classes of a right invariant equivalence relation of finite index.

3) Let equivalence relation R_L be defined by: xR_Ly if and only if for all *z* in Σ*, *xz* is in *L* exactly when *yz* is in L. Then R_L is of finite index.

(From: Hopcroft & Ullman 1979)

Back to L_s...

- $L_s = \{rr^Rs \mid r, s$ is in $\{0, 1\}+\}$.   "L_s is regular."
- Let $S = \{(10)^m11(01)^n0\}$ where $0 < m <= n$.
- S is a subset of L_s
- Note that first "pre-11" portion cannot contain a palindrome. That is, $(10)^m$ can only match (01)s
- Let $x = (10)^i$ and $y = (10)^j$ such that i and j are positive integers and i $!= j$.
- Let $z = 11(01)^k$, where $k = \min(i, j)$.
- Observe: either xz or yz is in L, but not both!

L_s Proof (continued)

- Myhill-Nerode says *x* and *y* form an equivalence class when *any z* can follow them.
- Here, *x* and *y* are not equivalent for some z. So they each are in their own equivalence class.
- But *x* and *y* are defined in terms of positive integers *i* and *j...* which means an infinite number of possible equivalence classes.
- This violates what Myhill-Nerode says about regular languages.
- Assumption -> contradiction. So L_s not regular.

Conclusion

- Want to be able to tell when a language is (not) regular.
- Pumping lemma is useful for this, as shown for $L = rr^R$
- But it doesn't always work, as for $L_s = rr^R s$
- Myhill-Nerode Theorem can be used instead.
- $L_s = rr^R s$ is not regular.

Thank You

Questions?

(If not, I have some for you...)