Lecture 5: Pumping Lemma

CSCI 81
Spring, 2015

Kim Bruce

Interpreting Regular Expressions

- For each regular expression we define language it denotes:
  - $L(\underline{\varepsilon}) = \{\varepsilon\}$, $L(\underline{\emptyset}) = \emptyset$, and $L(\underline{a}) = \{a\}$ for all $a \in \Sigma$
  - $L(\alpha^*) = (L(\alpha))^* = \{w_1 \ldots w_n \mid n \geq 0, w_i \in L(\alpha)\}$
  - $L(\alpha\beta) = L(\alpha)L(\beta)$ and $L(\alpha \cup \beta) = L(\alpha) \cup L(\beta)$.

Regular Expressions = Regular Languages

- If α is a regular expression then there is a DFM M s.t. $L(\alpha) = L(M)$
  - Construct NDFM by induction on regular expressions
    - Base case of single symbols easy
    - union, concatenation, * simple
- Other direction trickier. Add new start & final. Tear apart NDFM by removing states and writing labels as regular expressions.
  - See text for details

Regular Grammars

- A regular grammar G is a quadruple (V, Σ, R, S) where
  - V is the rule alphabet (with terminals & non-terminals)
  - $\Sigma \subseteq V$ is the set of terminals
  - R is a finite set of rules of the form $X \rightarrow w$,
    - $X \rightarrow \varepsilon$, $X \rightarrow b$, or $X \rightarrow bY$ for some nonterminals X,Y, & terminal b.
  - $S \in V - \Sigma$ is the start symbol
- Examples: $S \rightarrow aTa$ or $Sa \rightarrow bT$ not legal

Examples

- Numbers:
  - $S \to 0S, \ldots, S \to 9S$
  - $S \to \varepsilon$
  - $S \to .T$
  - $T \to 0T, \ldots, T \to 9T$
  - $T \to \varepsilon$
- Java identifiers

Big Picture

- How many languages (over a finite alphabet)?
  - How many regular languages?
- Every finite language is regular.
  - Is English finite?
- Closed under
  - Union, intersection, Kleene *, complement, difference, reverse, and letter substitution
  - Let $h: \Sigma_0 \to \Sigma_1^*$. Then L regular $\Rightarrow$ h(L) regular

Regular Languages

- Easy to show languages are regular, but how do we show they are not?
  - Myhill-Nerode: Show $\approx_L$ has infinite number of equivalence classes: e.g., $\{ww^R \mid w \in \{a,b\}^*\}$
  - Failure of closure properties.
    - E.g., find regular set so concatenating or union or intersection w/L not give regular set
      - Ex: $L = \{w \mid \#a\text{'s} = \#b\text{'s in } w\}$. If L regular then so is $L' = L \cap L(a^*b^*) = \{a^nb^n \mid n \geq 0\}$ -- but show it's not soon.
  - Pumping lemma ...

Pumping Lemma:

If L is regular, there is a number p (the pumping length) where, if $w \in L$ of length at least p, then there are x, y, & z with w = xyz, such that:

1. for each $i \geq 0$, $x y^i z \in L$;
2. $|y| > 0$; and
3. $|xy| \leq p$.

Use to show languages not regular!

Using Pumping

- Show $L = \{0^n1^n \mid n \geq 0\}$ is *not* regular
  - Proof by contradiction. Assume regular.
  - Therefore exists p from P.L.
  - Let $w = 0^p1^p \in L$
  - By P.L. can write w = xyz s.t. $|xy| = k \leq p$ s.t. $xy^iz \in L$ for all i
    - But $|xy| \leq p \Rightarrow$ x, y consist of all 0's.
    - So $x = 0^i$, $y = 0^j$, $z = 0^{p-i-j}1^p$ where $j > 0$.
    - Pick n = 2, then $xy^2z = 0^{p+j}1^p \notin L$. Contradiction so not regular!

Proof of Pumping Lemma

- Let (K,Σ,δ,s,A) be a DFA accepting L
- Let $p = |K|$.
- Let $w = a_1a_2 \ldots a_m \in L$ with $m \geq p$.
- Define $r_0, r_1, \ldots, r_m$ by $r_0 = s$ and $r_{i+1} = \delta(r_i,a_{i+1})$. Then $r_m \in A$ because $w \in L$
- Because $m \geq p$, $\exists i \neq j \leq p$ s.t. $r_i = r_j$
- Let $x = a_1 \ldots a_{i-1}$, $y = a_i \ldots a_j$, $z = a_{j+1} \ldots a_m$ so $y \neq \varepsilon$

Pf of Pumping Lemma

- Claim $xy^iz \in L$ for all $i \geq 0$.
  - Proof by picture!

Pumping Lemma Game

- To show L not regular
  - Opponent picks p
  - I pick w s.t. $|w| \geq p$
  - They pick decomposition w = xyz s.t. $|xy| \leq p$, $y \neq \varepsilon$
  - I show there is some i s.t. $x\, y^i\, z \notin L$
- If I succeed then I have shown L not regular!

Regular or Not?

- L = {$a^i b^j$: 0 ≤ i < j < 2000}.
- L = {$a^i b^j$: i, j ≥ 0 and i < j }.
- L = {$a^i b^j$: i, j ≥ 0 and i ≥ j }.
- L = {w ∈ {a,b}* : |w| is a power of 2}

Decision Problems w/FSM

- Let L = *L*(M) be a regular language, where M is DFSM, & w ∈ Σ*. It is decidable whether
  - w ∈ L
  - *L*(M) = ∅
    - Algo 1: Mark all reachable states. See if any are accepting.
    - Algo 2: Create unique minimal and see if any are accepting
  - *L*(M) = Σ*

Decision Problems w/FSM

- Let L = *L*(M) be a regular language, where M is DFSM, & w ∈ Σ*. It is decidable whether
  - *L*(M) is infinite
    - Use pumping lemma!
    - Claim: if L(M) infinite then must have w s.t. |K| ≤ |w| < 2 |K| - 1.
  - $L(M_1) \subseteq L(M_2)$
    - Difference is empty
  - $L(M_1) = L(M_2)$
    - Use above or compare canonical minimal DFSM's