CS 311: Computational Structures

James Hook

October 15, 2015

6 Pumping Lemma; Closure Properties; Intro to Context Free Languages

6.1 Recall

- Regular Expressions describe exactly the Regular Languages
- Pumping Lemma

6.2 Plan

- Discuss Problem Set 2
- Discuss Exercise 3
- In class exercise
- Pumping Lemma practice
- Using Closure Properties to prove languages are not regular
- Context Free Grammars

6.3 Exercise 3 Follow up

We can say that string w is *pumpable* for DFA $M = (Q, \Sigma, \delta, q_0, F)$ if $w \in L(M)$ (witnessed by state sequence $r_0, r_1, \ldots, r_n$) and there exist x, y, and z such that:

1. The states before and after the string y are the same, so all strings of the form $xy^iz \in A$.
2. $|y| > 0$, and
3. $|xy| \leq |Q|$

In Exercise 3 you built a "pumpable" string based on the optimal DFA for the modulo 3 counter construction. You also verified that when w is pumpable for M then M accepts xz and $xyyz$. In fact, M accepts xy^iz for all i.

In addition to the optimal DFA for congruence to 2 mod 3, there are an infinite number of non-optimal DFAs that accept exactly this language. In this in-class exercise I want you take a string that one of your group members used to correctly answer exercise 3. Call that string the "target string."

1. Build a non-optimal DFA, M', for **the same language** that is not pumpable on **the target string**, w. That is, $L(M) = L(M')$ but w is not pumpable for M'. It may help to design M' with more states than $|w|$. Note that what has changed between M and M' is the state sequence witnessing acceptance. In M' there should be no repitition corresponding to the repitition of states that defined y for M on w.

2. Propose a string that is pumpable for M'.

3. Is there any particular string that is always pumpable for this language for any DFA that recognizes it?

4. Can you propose a list of strings that always contains a pumpable string?

Key observations:

1. The decomposition of a string in to x, y, and z is determined by the machine that accepts the string, and not by the language.

6.4 Pumping exercises

Sipser 1.29, a,b. Use the pumping lemma to show the follow are not regular:

1. $A_1 = \{0^n 1^n 2^n | n \geq 0\}$

2. $A_2 = \{www | w \in \{a, b\}^*\}$

6.5 Closure Properties

The regular languages are closed under complement. This is easiest to do with DFAs; just complement the set of final states.

The regular languages are closed under intersection. That is, if A and B are regular, so is $A \cap B$. The proof of closure under union for DFAs can be adapted to prove this result.

Use closure properties to show this language is not regular:

Sipser 1.46, b: $\{0^m 1^n | m \neq n\}$.