CS 154
Formal Languages and Computability

Spring 2016

Department of Computer Science
San Jose State University
Instructor: Ron Mak

Assignment #3

Assigned: Thursday, February 25
Due: Wednesday, March 2 at 11:59 pm
Individual assignment, 100 points max

Problem set B

1. Use the pumping lemma to show that the language
   $L = \{a^n : n \text{ is a perfect cube: } 0, 1, 8, 27, \ldots\}$ is not regular.

2. Use the pumping lemma to show that the language
   $L = \{a^n : n \text{ is a power of 2: } 1, 2, 4, 8, \ldots\}$ is not regular.

3. Use the pumping lemma to show that the language
   $L = \{a^{pq} : p \text{ and } q \text{ are both prime numbers}\}$ is not regular.

4. Use the pumping lemma to show that the language
   $L = \{a^p b^q : p \text{ divided by } q \text{ is an integer quotient}\}$ is not regular.

5. Use the pumping lemma to show that the language
   $L = \{a^p b^q : p + q \text{ is a prime number}\}$ is not regular.

6. Let $\Sigma = \{0, 1, +, =\}$. Use the pumping lemma to show that the language
   $L = \{x=y+z : x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z\}$ is not regular.
   For example, the string $1001=10+111$ is in L.

7. Let language L be denoted by the regular expression a^*b^*
   What is wrong with the following "proof" that L is not regular?

> Assume that L is regular. Then it must be defined by a DFA with k states, for some integer $k > 0$. Take the string $w = a^k b^k$ and split it $w = xyz$, with $y = ab$. Then wy^2z is not in L, which contradicts the pumping lemma. Therefore, L cannot be regular.

8. Prove whether or not language $L = \{a^{p+qi} : p \text{ and } q \text{ are fixed integer values, and } i \geq 0\}$ is regular.

9. Prove whether or not language
   $L = \{a^p b^q : p \geq 100 \text{ and } q \geq 100 \text{ are fixed integer values, and } i \geq 0\}$ is regular.

10. Assume that **<stmt>**, **<if_stmt>**, **<boolexpr>**, and **<assign_stmt>** are nonterminal symbols, and **if**, **else**, **(**, and **)** are terminal symbols.

    Here's a grammar written in BNF for Java-style IF statements.

```
<stmt>    ::= <assign_stmt> | <if_stmt>
<if_stmt> ::= if ( <boolexpr> ) <stmt>
            | if ( <boolexpr> ) <stmt> else <stmt>
```

    How is this grammar ambiguous? Give an example of an ambiguity.

What to submit to Canvas

Submit your answers as a Word document or a PDF into Canvas: **Assignment #3**