CONCORDIA UNIVERSITY

Dept. of Computer Science and Software Engineering

COMP 335 – Introduction to Theoretical Computer Science

Fall 2018

$\mathcal{A}$ssignment 2

Submission deadline: Thursday October 18th at 23:55

Note: This is a theoretical course. That means, while the WHAT is important, the WHY is absolutely essential. Show the steps of your solution for the full mark.

Total mark is 70.

1. (10 Points) Let L be any regular language over an alphabet Σ. Using L, we define $chop(L) = \{w : \exists\, x, y, z \in \Sigma^*,\ xyz \in L,\ w = xz\}$.

   Show that $chop(L)$ is regular or give a counter-example.

2. (10 Points) Let $\Sigma = \{3, 5\}$ and consider the language $L = \{w \in \Sigma^* : w \text{ ends in } 335\}$.

   (a) Design an NFA for L with as few states and transitions as possible. Name your states $q_1, q_2, \ldots$

   (b) Use NFA to DFA conversion procedure (Theorem 2.2) to convert your NFA from the previous part to a DFA. Use the naming convention of Theorem 2.2 to name the states of your DFA.

3. (10 Points) Fix a finite alphabet $\Sigma = \{a, b\}$ and define for a string $w = \sigma_1\sigma_2\cdots$ over Σ the operation $fifth(w) = \sigma_5\sigma_{10}\sigma_{15}\cdots$. In words, this operation extracts all characters at positions divisible by 5. We can naturally extend this operation to languages: if L is a language, $fifth(L) = \{fifth(w) : w \in L\}$.

   Prove that if L is regular then $fifth(L)$ is also regular.

4. (10 Points) Let $\Sigma = \{a\}$. For each of the following languages, either prove that it is regular (by exhibiting one of: DFA, NFA, regular expression, or regular grammar), or prove that it is not regular using the pumping lemma.

   (a) $L_1 = \{a^{2018n} : n \geq 0\}$

   (b) $L_2 = \{a^{2018^n} : n \geq 0\}$

   Hint: if the constant 2018 looks intimidating, try answering similar questions with other constants, e.g., 2, 3, 5 playing the role of 2018. Observe the pattern and tackle the case of 2018.

5. (10 Points) In this question you will show that DFAs and NFAs are extremely limited as computing devices. They cannot even compute multiplication and iterated exponentiation (although it is easy to compute each of these on a general-purpose computer).

(a) Let $\Sigma = \{a, b, c\}$ and consider the task of multiplication encoded in the language $L = \{a^n b^k c^{nk} : n \geq 0, k \geq 0\}$. Prove that L is not regular using the pumping lemma.

(b) Let $f(n)$ denote iterated exponentiation with base 2. Formally, it is defined as follows

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 2^{f(n-1)} & \text{if } n \geq 1 \end{cases}$$

For your intuition, consider $f(1) = 2$, $f(2) = 2^2$, $f(3) = 2^{2^2}$, $f(4) = 2^{2^{2^2}}$, and so on. In words, $f(n)$ is a tower of height n of powers of 2.

(b.1) Prove by induction that $2^n \geq 2n$ and argue that it implies $f(m+1) \geq 2f(m)$.

(b.2) Prove by induction that $f(n) > n$.

(b.3) Let $\Sigma = \{a\}$. Prove that $L = \{a^{f(n)} : n \geq 0\}$ is not regular using the pumping lemma.

Hint: you may structure your proof similar to the proof of $\{a^{n!} : n \geq 0\}$ being not regular from lectures and use (b.1) and (b.2).

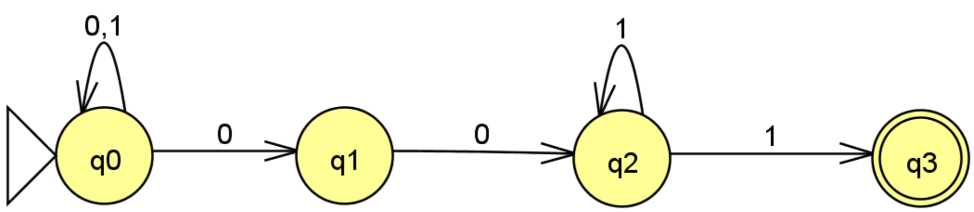
6. (10 Points) Find a regular grammar for each of the following languages over $\Sigma = \{a, b\}$:

(a) $L_3 = \{w \in \Sigma^* :$ the number of a's plus the number of b's in w is divisible by $5\}$.

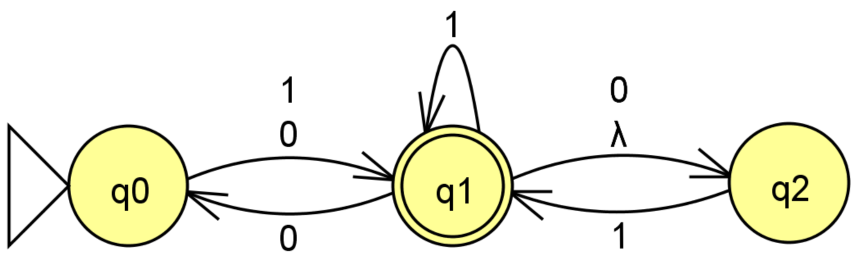
(b) $L_4 = \{w \in \Sigma^* :$ the number of substrings ab in w is even $\}$.

7. (10 Points) Use generalized transition graphs (GTG) to convert each of the following NFAs (alphabet $\Sigma = \{0, 1\}$) into regular expressions. Show your work, i.e., intermediate GTGs.

(a)



(b)


1
1
0
0
λ
q0
q1
q2
0
1