Assignment 1
NLP 201: Natural Language Processing I

Out: October 13, 2020
Due: October 27, 2020 11:59pm

All assignments are to be completed individually. You may discuss with the TAs and the instructor, but you may not receive help from anyone else.

Instructions. Create a LaTeX-typeset write-up of your solutions and submit it to Canvas. Some free tools that might help: TexStudio (Windows), MacTex (Mac), TeX Live (cross-platform), and Overleaf (online). For the problems that require that you give FSAs, we encourage you to create them using drawing tools such as OmniGraffle (Mac) or yEd (online or cross-platform), but we will accept hand-drawn figures for full credit. Make sure initial and final states are clearly identified and explicitly list all symbols on each arc (i.e., no wildcards). We ask that you do not use any automated tools to solve these problems as you will not have access to such tools for similar problems on the exams.

Problem 1 [15 points]
For the following NFA containing $\varepsilon$-moves,

![NFA Diagram]

give an equivalent DFA.

Problem 2 [15 points]
Give DFAs accepting the following languages.

1. [5 points] The set of strings over the alphabet \{a, b, c\} in which the substring bc never occurs.
2. **[5 points]** The set of strings over the alphabet \{1, 7\} that end in 11711.

3. **[5 points]** The set of strings over the alphabet \{0, 1\} which are divisible by three when interpreted as a binary number (ignoring leading zeroes).

   For example 00000_b = 0, which is divisible by 3, so 00000 should be accepted. 001001_b = 9 and thus should be accepted also. 101_b = 5 is not divisible by 3 and thus should be rejected.

   **Note:** \(\varepsilon\) should be interpreted as 0 and thus should be accepted.

**Problem 3 [10 points]**

Give a non-deterministic FSA (possibly with \(\varepsilon\)-moves) accepting the following language.

1. The set of strings over \{1, 7\} ending in 11711. How does your NFA differ from the DFA you constructed in Problem 2.2?

**Problem 4 [15 points]**

Give regular expressions for each of the following languages:

1. **[7 points]** The set of strings over \{a, b, c\} in which the substring \(bc\) never occurs.

2. **[8 points]** The language described in Problem 2.3: The set of strings over \{0, 1\} which are divisible by three when interpreted as a binary number (ignoring leading zeroes).

   For example 00000_b = 0, which is divisible by 3, so 00000 should be accepted. 001001_b = 9, and thus should be accepted also. 101_b = 5, is not divisible by 3, and thus should be rejected.

   **Note:** \(\varepsilon\) should be interpreted as 0, and thus should be accepted.

**Problem 5 [20 points]**

We may define generalized regular expressions (GREs) as follows:

1. \(\emptyset\) is a GRE denoting the empty language;

2. \(\varepsilon\) is a GRE denoting the language \{\(\varepsilon\)\};

3. for each \(\sigma \in \Sigma\), \(\sigma\) is a GRE denoting the language \{\(\sigma\)\};

4. if \(\alpha\) and \(\beta\) are GREs, denoting the languages \(A\) and \(B\), respectively, then
   - \((\alpha \mid \beta)\) is a GRE denoting \(A \cup B\);
   - \((\alpha \beta)\) is a GRE denoting \(A.B\);
   - \(\alpha^*\) is a GRE denoting \(A^*\);
   - \([\text{new}] \ (\alpha \land \beta)\) is a GRE denoting \(A \cap B\); and
   - \([\text{new}] \ \neg\alpha\) is a GRE denoting \(\overline{A}\).
Prove that the languages denoted by GREs are regular.

**Note:** you may use refer to the definitions and proofs about REs we provided in class and only deal with the two new clauses in the generalized definition.

**Problem 6 [25 points]**

For each of the following statements, answer whether the claim is *true* or *false*, and give a *short* (one- to two-sentence) explanation if true or counterexample if false.

1. **[5 points]** Let $L_1 \subset L_2$. If $L_1$ is not regular, then $L_2$ must also be not regular.

2. **[5 points]** $L = L_1 \cap L_2$. If $L_1$ and $L$ are regular languages, then $L_2$ must also be a regular language.

3. **[5 points]** $L = L_1 \cup L_2$. If $L_1$ and $L$ are regular languages, then $L_2$ must also be a regular language.

4. **[5 points]** $L = \bigcap_{i=1}^{\infty} L_i$. If all of the $L_i$ are regular languages, then $L$ is also a regular language.

5. **[5 points]** Let $\alpha$, $\beta$ and $\gamma$ be regular expressions. If $L(\beta \mid \alpha \gamma) \subseteq L(\gamma)$, then $L(\alpha^* \beta) \subseteq L(\gamma)$. 