

Ground Rules:

- This assignment contains both written and programming problems. You should first attempt the problems by yourself and start working in groups after a few days of thinking.
- Typeset your solution (it's a good excuse to learn LaTeX) or write legibly. You are handing in your work electronically. We only accept PDF submissions. Name the file whatever you want as long as it's a single PDF file.
- Exercise common sense when collaborating with others or looking things up online. Even if you work together on a problem, the writeup should be your own. This is the only way I know for you to master this kind of subject.

Task 1: NFA vs. DFA Expressiveness (4 points)

Let $\Sigma = \{a, b\}$. For $k \geq 1$, define C_k to be

$$C_k = \{w \in \Sigma^* \mid w[-k] == a\}$$

That is, the k -th position from the right is an a . This language is regular and there is a regular expression for it: $C_k = \Sigma^* a \Sigma^{k-1}$. You will show the following:

1. For every $k \geq 1$, there is an NFA with $k + 1$ states that recognizes C_k .
2. If M is a DFA that correctly recognizes C_k , then M has at least 2^k states. (*Hint:* The general outline for proving that a language is not regular will be useful here.)

Task 2: Regular or Not (2 points)

Let $\Sigma = \{p, q\}$. One of the following two languages is regular, and the other is not.

$$L_1 = \{xyx^R \mid x, y \in \Sigma^*, x \neq \varepsilon\}$$

$$L_2 = \{xx^R \mid x \in \Sigma^*, x \neq \varepsilon\}$$

where x^R denotes the reverse of x . For the regular language, draw a DFA that accepts it and justify why your DFA works by explaining what each state “means.” For the nonregular language, give a proof that it is nonregular using the pumping lemma.

Task 3: Nonregular (4 points)

Prove that each of the following languages is not regular.

$$L = \{10^{n^2} \mid n \geq 0\}$$

$$E = \{0^i x \mid i \geq 0, x \in \{0, 1\}^*, \text{ and } |x| \leq i\}.$$

For example, $1 \in L$, $10 \in L$, $10000 \in L$, etc.

Task 4: HackerRank Challenge (9 points)

We will look at practical exercises in this task. Complete the HackerRank challenge in the link below. In your PDF writeup, include your the HackerRank user id so we can see your progress.

There are 9 problems in this set. You should know that Python regex is an extended version of regex and is strictly more powerful than the traditional regex recognizable by a DFA/NFA.

<https://www.hackerrank.com/computation-theory-t-22020-21-regex>