St. Francis Xavier University
Department of Computer Science

CSCI 356: Theory of Computing
Assignment 1
Due September 19, 2025 at 11:30am

Assignment Regulations.

- This assignment must be completed individually.
- Please include your full name and email address on your submission.
- You may either handwrite or typeset your submission. If your submission is handwritten, please ensure that the handwriting is neat and legible.

[6 marks] 1. As we saw in lecture, `grep` is a powerful command-line tool for using regular expressions to match patterns in text files. Here, you will use `grep` to find words in a dictionary file that match certain patterns.

Note. If your computer uses MacOS or Linux, you should already have a dictionary file located at `/usr/share/dict/words` or a similar directory. If your computer uses Windows, you should open the Windows Subshell for Linux and download a dictionary file to your current directory by running the following command:
`curl -o [FILENAME] https://raw.githubusercontent.com/eneko/data-repository/master/data/words.txt`

(a) Run the command `grep smith$ /usr/share/dict/words`. What words does this pattern match?

(b) Run the command `grep ^[abcde]..[uvwxy]$ /usr/share/dict/words`. What words does this pattern match?

(c) Run the command `grep .*[aeiou][aeiou][aeiou][aeiou].* /usr/share/dict/words`. What words does this pattern match?

(d) Write a regular expression that outputs all words containing the letters `lmn` in that order consecutively. Give an example of a word having this property.

(e) Write a regular expression that outputs all words containing the vowels `a`, `e`, `i`, `o`, and `u` in reverse order (not necessarily consecutively, and vowels may also appear elsewhere). Give an example of a word having this property.

(f) Write a regular expression that outputs all words beginning with `pro` and also containing `anti` (not necessarily consecutively). Give an example of a word having this property.

[6 marks] 2. (a) For each of the following regular expressions, give a 1–2 sentence description of the language matched by that regular expression.

i. $\boldsymbol{r}_1 = (\epsilon + \mathtt{1})(\mathtt{01})^*(\epsilon + \mathtt{0})$.

ii. $\boldsymbol{r}_2 = (\mathtt{1} + \mathtt{0})(\mathtt{1} + \mathtt{0})^*\emptyset(\mathtt{00} + \mathtt{01} + \mathtt{10} + \mathtt{11})$.

(b) For each of the following languages over the alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, give a regular expression matching exactly that language.

i. $L_1 = \{w \mid \text{every run of } \mathtt{0}\text{s in } w \text{ has a length of at least } 3\}$.

ii. $L_2 = \{w \mid w \text{ contains at least two } \mathtt{0}\text{s and at least one } \mathtt{1}\}$.

[8 marks] 3. (a) Let $\Sigma = \{\texttt{0}, \texttt{1}\}$. For each of the following languages, give a *deterministic* finite automaton recognizing exactly that language.

i. $L_1 = \{w \mid w \text{ does not contain } \texttt{000}\}$.

ii. $L_2 = \{w \mid \text{the number of } \texttt{1}\text{s in } w \text{ is divisible by three}\}$.

(b) Let $\Sigma = \{\texttt{0}, \texttt{1}\}$. Give a *nondeterministic* finite automaton recognizing the language of binary representations of natural numbers that are divisible by four. (For example, your nondeterministic finite automaton should accept input words like `100` and `10100`, but reject input words like `111` and `10101`.) You may assume that the input word contains no leading zeroes, with the exception of the specific input word `0`.

[5 marks] 4. Let $\Sigma = \{\genfrac{}{}{0pt}{}{\texttt{0}}{\texttt{0}}, \genfrac{}{}{0pt}{}{\texttt{0}}{\texttt{1}}, \genfrac{}{}{0pt}{}{\texttt{1}}{\texttt{0}}, \genfrac{}{}{0pt}{}{\texttt{1}}{\texttt{1}}\}$. Consider the language

$$L = \{w = \genfrac{}{}{0pt}{}{a_1}{b_1} \genfrac{}{}{0pt}{}{a_2}{b_2} \dots \genfrac{}{}{0pt}{}{a_n}{b_n} \mid b_1 b_2 \dots b_n \text{ is the result of shifting } a_1 a_2 \dots a_n \text{ rightward by one bit}\}.$$

Here, $n \geq 1$ and for all $1 \leq i \leq n$, $a_i, b_i \in \{\texttt{0}, \texttt{1}\}$.

One can observe that, for example, $\genfrac{}{}{0pt}{}{\texttt{1}}{\texttt{0}} \genfrac{}{}{0pt}{}{\texttt{1}}{\texttt{1}} \genfrac{}{}{0pt}{}{\texttt{1}}{\texttt{1}} \in L$ and $\genfrac{}{}{0pt}{}{\texttt{0}}{\texttt{0}} \genfrac{}{}{0pt}{}{\texttt{1}}{\texttt{0}} \genfrac{}{}{0pt}{}{\texttt{0}}{\texttt{1}} \in L$, but $\genfrac{}{}{0pt}{}{\texttt{1}}{\texttt{1}} \genfrac{}{}{0pt}{}{\texttt{0}}{\texttt{1}} \genfrac{}{}{0pt}{}{\texttt{0}}{\texttt{0}} \notin L$. You may assume that the leftmost bit after applying the right-shift operation is always `0`. (Note that $\epsilon \in L$ as well, since right-shifting the empty word just produces the empty word.)

Construct a deterministic finite automaton recognizing the language L.

Hint. It is possible to recognize this language using a DFA with at most three states.