CSE 105 - Theory of Computation (Winter 2026)
Homework 2: Closure properties, Nondeterminism, and Regular Expressions
Due Wednesday, January 21, 11:59pm

*Instructions:* There are two parts to this homework:

- Concept check (Question 1): Every student must complete this individually on Gradescope.
- Written Homework (Question 2-5): You may work individually or in a team of up to 3 people. Please ensure your name(s) and PID(s) are clearly visible on the first page of your submission, and then upload the PDF to Gradescope. If working in a group, submit only one submission per group: one partner uploads the submission through their Gradescope account and then adds the other group member to the Gradescope submission by selecting their name in the "Add Group Members" dialog box. You will need to re-add your group member every time you resubmit a new version of your assignment.

It is highly recommended (though not required) that you type your answers. It is your responsibility to make any handwriting clear and legible for grading. A LaTeX template for the homework is provided on Canvas. For ease of grading, please start each new problem on a separate page.

We will only be grading some of the problems below for correctness. However, because all of the concepts are important, we will not reveal which problems are being graded for correctness until after the assignment has been submitted. The remaining problems will be graded for completeness (i.e., does it look like there was a good-faith effort to solve the problem?).

*Reading and extra practice problems*: Sipser Section 1.2, 1.3. Chapter 1 exercises 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12, 1.14, 1.15, 1.16, 1.17, 1.19, 1.20, 1.21, 1.22

## Problems:

1. **Concept check**

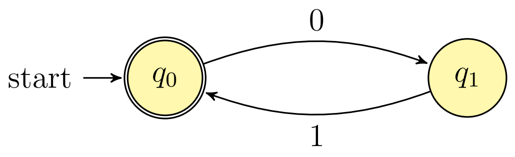
   Complete the assignment "Homework 2 - Concept Check" on Gradescope.

2. **Closure of the class of regular languages under** SUBSTRING

   Let $\Gamma = \{0, 1\}$. From the previous homework, recall the function SUBSTRING that acts on languages $K$ over $\Gamma$:

$$\text{SUBSTRING}(K) := \{w \in \Gamma^* \mid \text{there exist } a, b \in \Gamma^* \text{ such that } awb \in K\}$$

   (a) Consider the NFA over $\Gamma$ with state diagram:



We'll call the language recognized by the NFA above $C$.

Fill in the blanks below. That is, for each item, you'll either fill in a specific string and a justification that refers back to the relevant definitions, or you'll write "impossible" for the first part of the sentence and justify why it's impossible to find such an example referring back to the relevant definitions.

- An example of a string over $\Gamma$ that is in $C$ **and** is in SUBSTRING($C$) is ____________ because _________________________
- An example of a string over $\Gamma$ that is in $C$ **and** is **not** in SUBSTRING($C$) is ____________ because _________________________
- An example of a string over $\Gamma$ that is **not** in $C$ **and** is in SUBSTRING($C$) is ____________ because _________________________
- An example of a string over $\Gamma$ that is **not** in $C$ **and** is **not** in SUBSTRING($C$) is ____________ because _________________________

(b) Prove that the class of regular languages is closed under the SUBSTRING operation. Namely, give a general construction that takes an arbitrary NFA and constructs an NFA that recognizes the result of applying SUBSTRING to the language recognized by the original machine. You can describe your construction in words and/or draw a picture to illustrate your construction. You do not have to write down a formal specification.

(c) Draw the state diagram of an NFA over $\Gamma$ that recognizes SUBSTRING($C$) (for $C$ the language from part (a) of this problem), using your construction from part (b) of this problem, or manually constructing it. Describe the computation(s) of this NFA for each of the sample strings you gave in part (a).

3. **Bubble sort is regular?**

For any string $w := w_1 w_2 \cdots w_n \in \Sigma^*$ with $|w| := n$ symbols, define the function that swaps the $i$th and $(i+1)$st symbols:

$$\text{SWAP}_i(w) = w_1 \cdots w_{i-1} w_{i+1} w_i w_{i+2} \cdots w_n$$

For a language $L$ with alphabet $\Sigma = \{0, 1\}$, we can now define the following languages:

$$\textsc{Sorted}(L) := \{w \in L \mid \text{for all } i \in \{1, 2, \ldots, |w| - 1\}, w_i \leq w_{i+1}\}$$
$$\textsc{Bubble}(L) := \textsc{Sorted}(L) \cup \{w \in \Sigma^* \mid \text{exists } i \text{ for which } \text{SWAP}_i(w) \in L \text{ and } w_i < w_{i+1}\}$$

For example, if the string $001 \in L$, then $001 \in$ SORTED($L$) because the symbols (which we are interpreted as integers) are in sorted order. Similarly, if $010 \in L$, then $010 \notin$ SORTED($L$) since the symbols are not sorted from left to right.

The motivation for the BUBBLE function is Bubble sort, one of the most basic algorithms for sorting a list. The algorithm works by iteratively swapping adjacent pairs of items that are out of order. The BUBBLE function is meant to simulate one of these swaps. For example, if $010 \in L$, then $001 \in$ BUBBLE($L$) since we can get from 010 to 001 by swapping the last two symbols.

(a) Prove that SORTED($L$) is regular whenever $L$ is regular. You may use any techniques shown in class or homework to do so. *Hint: there is a very short proof using the closure properties of regular languages.*

(b) [Optional, not for credit] Show that BUBBLE($L$) is regular whenever $L$ is regular. Do this by describing an NFA that recognizes BUBBLE($L$) given a DFA that recognizes $L$. Simply describe in words how your NFA works. It may help to draw a picture to showcase your construction, but you do not have to write down a formal specification.

*Note: This is a bit tricky to write down, but the central idea is common to many NFA constructions of regular languages. It's useful practice to think about this even if you choose not to write it down.*

(c) Given that $L = \{010101\}$, list all of the elements in the languages BUBBLE($L$), BUBBLE(BUBBLE($L$)), BUBBLE(BUBBLE(BUBBLE($L$))).

(d) Find the error in the following proof:

i. $L = \{(01)^n \mid n \in \mathcal{N}\}$ is a regular language.

ii. Every element of $L$ has the same number of 0's as 1's.

iii. BUBBLE($L$) is regular, so BUBBLE(BUBBLE($\cdots$(BUBBLE($L$))$\cdots$)) is regular.

iv. Iteratively applying BUBBLE sorts $L$, so we get $\{0^n 1^n \mid n \in \mathcal{N}\}$ is regular.

List each line which is wrong and/or misleading and describe why.

*Note: The final line is incorrect because the language $\{0^n 1^n \mid n \in \mathcal{N}\}$ is not regular. You do not have to prove this fact.*

4. **Deciphering regular expressions**

For this question, let's fix the regular expression

$$R = 0^*(1 \cup 10)^*$$

For each choice of strings of length 3, $a, b, c \in \{0,1\}^3$ we can define the regular expression:

$$X_{a,b,c} = 0(a \cup b \cup c)^*$$

(a) Give a plain English explanation for the language defined by regular expression $R$. This continues a theme from Problem 1—before trying to prove formal statements about a specific regular expression, it's often good to try to translate it into a form that is more easy to reason about. Typically speaking, the shorter and more concise your plain English description is, the more useful it will be in reasoning about the language.

(b) Suppose $a = 000$, $b = 001$, $c = 011$ so

$$X_{a,b,c} = 0(000 \cup 001 \cup 011)^*$$

Show that $L(R) \not\subseteq L(X_{a,b,c})$ by giving some string in $L(R)$ which is not in $L(X_{a,b,c})$, and justifying this choice referring back to relevant definitions.

(c) More generally, prove that

$$L(R) \not\subseteq L(X_{a,b,c})$$

for *all* possible strings $a, b, c \in \{0,1\}^3$. Hint: What are the possible lengths of strings in $L(R_1)$ (and why does this help)?

(d) Give a specific example of three distinct strings $a, b, c \in \{0,1,2\}^3$ such that

$$L(X_{a,b,c}) \subseteq L(R)$$

Briefly justify your answer by explaining how an arbitrary element of $L(X_{a,b,c})$ is guaranteed to be an element of $L(R)$.

(e) Give a specific example of three distinct strings $a, b, c \in \{0,1,2\}^3$ such that

$$L(X_{a,b,c}) \not\subseteq L(R)$$

Briefly justify your answer by giving a counterexample string that is in $L(X_{a,b,c})$ and is not in $L(R)$ (and explaining why using relevant definitions).

5. **It can be hard to give a good complement**

For any language $L \subseteq \Sigma^*$, recall that we define its *complement* as

$$\overline{L} := \Sigma^* - L = \{w \in \Sigma^* \mid w \notin L\}$$

That is, the complement of $L$ contains all and only those strings which are not in $L$. Our notation for regular expressions does not include the complement symbol. However, it turns out that the complement of a language described by a regular expression is guaranteed to also be describable by a (different) regular expression. For example, over the alphabet $\Sigma = \{0,1\}$, the complement of the language described by the regular expression $\Sigma^*0$ is described by the regular expression $\varepsilon \cup \Sigma^*1$ because any string that does not end in 0 must either be the empty string or end in 1.

For each of the regular expressions $R$ over the alphabet $\Sigma = \{0,1\}$ below, write the regular expression for $\overline{L(R)}$. You may use only use the following operations: union, concatenation, and Kleene star.

Briefly justify why your solution works by giving plain English descriptions of the language described by the regular expression and of its complement. An English description that is more detailed than simply negating the description in the original language will likely be helpful in the justification.

(a) $(\Sigma\Sigma)^*$

(b) $\Sigma^*11\Sigma^*$

(c) $0^*10^*10^*$