CS 361: Theory of Computation

Assignment 1 (due 09/10/2025)

Instructor: Shikha Singh

LaTeX Source for Solutions: https://www.overleaf.com/read/ssygvwtyjmwc#061016

Kleene closure. For the first problem, we will define a new operation (Kleene closure) on a language and get practice with the definition.

Recall that a language is set of strings over a given alphabet Σ. Given two languages A and B, their *concatenation*, written $A \circ B$ or simply AB, is defined as the set of strings $\{ab \mid a \in A, b \in B\}$. For any language A, define $A^0 = \varepsilon$ and $A^k = AA^{k-1}$ for $k \geq 1$. Finally, for any language A, we define its *Kleene star* or *Kleene closure*, denoted A^* as:

$$\begin{aligned} A^* &= A^0 \cup A^1 \cup A^2 \cup \dots \\ &= \{w \mid w \text{ is a concatenation of 0 or more strings from A}\} \end{aligned}$$

Problem 1. Given any two languages A and B such that $\varepsilon \notin A$, define language $X = AX \cup B$. Using induction on the string length, **prove that** $X = A^*B$**.** Note that you must prove both directions: $X \subseteq A^*B$ and $A^*B \subseteq X$.

Solution.

□

Regular Expressions (Sipser Chapter 1.3.) Grammars or expressions are way to succinctly represent the strings in a language. In contrast to machines that *recognize* the strings in a language, grammars *generate* the strings in a language. The first example of this in the course is **regular expressions**. Regular expressions are an alternate way to define and study regular languages. They are defined in Definition 1.52 of Sipser, reproduced below.

We say R is a **regular expression** if R is

- (a symbol) a for some $a \in \Sigma$,
- (empty string) ε,
- (empty set) $\emptyset$,
- (union) $R_1 \cup R_2$, where R_1 and R_2 are (smaller) regular expressions,
- (concatenation) $R_1 \circ R_2$, where R_1 and R_2 are (smaller) regular expressions, or
- (Kleene star) R_1^*, where R_1 is a regular expression.

Note that the definition is inductive (with three base cases). Example 1.53 in Sipser describes several regular expressions and the corresponding languages. Review these before answering the following question.

Problem 2. For each of the following regular expressions give

- a succinct description of the language of the regular expression in English
- two strings that are members of the language and two strings that are NOT members of the language

Assume the alphabet $\Sigma = \{a, b\}$ for each part.

(a) **Solved Example:** $a(ba)^*b$

Solution. All strings formed with one or more concatenations of ab.
Strings in the language: $\boxed{ab, abab}$. Strings not in the language $\boxed{\varepsilon, aabb}$. □

(b) $a^* \cup b^*$

Solution.

□

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

Solution.

□

(d) $\Sigma^* aba \Sigma^*$

Solution.

□

Problem 3. Draw state diagrams of DFAs recognizing the following languages **and** state a regular expression generating the same language. In all cases the alphabet is $\{0, 1\}$.

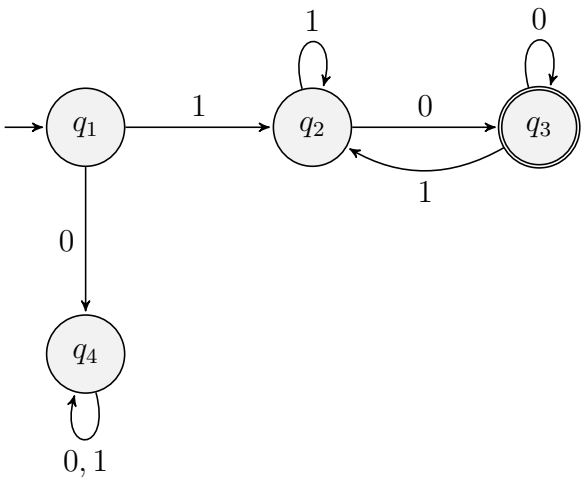
Remark: The first DFA LaTeXsource code uses the `tikz` package and is provided for reference. You can read more about how to typeset automata in tikz here: https://tikz.dev/library-automata. Alternatively, you may use any other software to draw the DFAs, or attach a clear picture of a hand-drawn figure.

(a) **Solved Example:** $\{w \mid w$ begins with a 1 and ends with a $0\}$.

Solution.

Regular expression: $\boxed{1(0 \cup 1)^*0}$.

State diagram of DFA:



□

(b) $\{w \mid$ starts with 0 and has odd length, or starts with 1 and has even length$\}$.

Solution.

□

(c) $\{w \mid w$ every odd position of w is a $1\}$.[1]

Solution.

□

(d) $\{w \mid w$ contains at least one 0 and at most one $1\}$.

Solution.

□

[1] Assume that the first bit of a non-empty string is at position 1.