Homework 2

Computation Theory (Term II/2020–21)

built on 2021/01/14 at 10:43:57 — ***due:*** *tue jan 26 @ 11:59pm*

Ground Rules:

- This assignment contains only written 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: Regex to NFA/DFA (4 points)

Convert the following regular expressions to NFA. Assume $\Sigma = \{a, b\}$.

(1) $a(abb)^* + b$

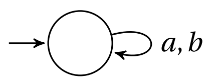
(2) $(a+b)^* aa(a+b)^*$

(3) $a^+ + (ab)^+$

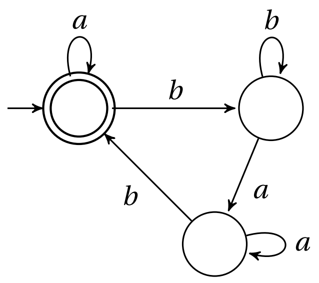
Task 2: Finite-State Machines to Regex (2 points)

Let $\Sigma = \{a, b\}$. Convert each of the following finite-state machines to a regular expression:

(1)



(2)



Task 3: Binary Addition (2 points)

In this problem, we'll prove a surprising fact that verifying the sum of two binary numbers can be couched as a regular language. To this end, let

$$\Sigma = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \cdots, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}$$

be an alphabet set containing all size-3 columns of 0s and 1s. Indeed, the size of Σ is $2^3 = 8$. Strings on this alphabet are such columns put together. For example:

$$w_1 = \begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} \qquad \text{and} \qquad w_2 = \begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}0\\1\\1\end{bmatrix}$$

We consider each row to be a binary number. In this view, we will define

$$A = \{w \in \Sigma^* \mid \text{the bottom row of } w \text{ is the sum of the top two rows}\}.$$

As a quick example, the string $w_1 \in \Sigma^*$ above is in the language A because reading row-wise $(011)_2 + (001)_2 = (100)_2$. However, the string $w_2 \in \Sigma^*$ is *not* in the language A because $(10)_2 + (01)_2 \neq (01)_2$.

Your Task: Prove that A is regular. (*Hint:* If you don't feel like drawing the whole diagram, describe the machine mathematically.)

Task 4: Division Operation? (2 points)

We say that a divides b if there exists a number k such that $b = ak$. This k is the result of dividing b by a, i.e., $k = \frac{b}{a}$. Generalizing this notion, for two languages L_1, L_2, we define

$$\frac{L_1}{L_2} = \left\{x \mid \exists y \in L_2 \text{ s.t. } xy \in L_1\right\}.$$

Prove that if L_1 and L_2 are regular, then $\frac{L_1}{L_2}$ is also regular. (*Hint:* Think outside the box. While DFA/NFA has limited power, you can precompute certain things outside DFA/NFA—and encode the result in your machine.)

Task 5: Does It Accept Everything? (2 points)

Consider a DFA $M = (Q, \Sigma, \delta, q_0, F)$. Design an efficient algorithm (hopefully one that runs in polynomial time in $|Q|$ assuming the size of Σ is constant) that determines whether $L(M) = \Sigma^*$—that is, whether the machine M accepts every string in Σ^*.

Hint: In many ways, a finite-state machine is simply a directed graph, so you'd sure hope graph algorithms can be used to answer questions about it.

Task 6: All The Same? [Extra Credit] (0 points)

Let Σ be an alphabet. Let $x, y \in \Sigma^*$ be nonempty strings. Consider the following three statements about x and y:

(i) $xy = yx$;

(ii) there is a nonempty string z and numbers $m, n \in \mathbb{Z}_{>0}$ such that $x = z^m$ and $y = z^n$;

(iii) there are numbers $k, \ell \in \mathbb{Z}_{>0}$ such that $x^k = y^\ell$.

Show that the three statements (i), (ii), (iii) are equivalent.