Recitation 12 solutions - CSCI 2200 (FOCS)

I. Find regular expressions for $\mathcal{L}$ and its reversal $\mathcal{L}^R := \{\omega^R \mid \omega \in \mathcal{L}\}$, when $\mathcal{L}$ consists of strings:

(a) which contains 100 as a substring at least once,

(b) in which every 0 is followed by exactly two 1s.

(a) For strings containing 100 as a substring at least once:
Regular expression for $\mathcal{L}$: $(0+1)^*100(0+1)^*$
This represents any string that has 0s and 1s, followed by the specific pattern 100, followed by any additional 0s and 1s.
Regular expression for $\mathcal{L}^R$: $(0+1)^*001(0+1)^*$
To find the reversal, we reverse the pattern 100 to get 001, and the expression becomes any string with 0s and 1s, containing 001, followed by any additional 0s and 1s.

(b) For strings in which every 0 is followed by exactly two 1s:
Regular expression for $\mathcal{L}$: $1^*(011)^*1^*$
This represents strings where every 0 must be followed by exactly two 1s. We can have any number of 1s at the beginning, then any number of 011 patterns, and finally any number of additional 1s.
Regular expression for $\mathcal{L}^R$: $1^*(110)^*1^*$
For the reversal, the pattern 011 becomes 110. Thus, the regular expression represents strings with any number of 1s, followed by any number of 110 patterns, followed by any number of 1s.

II. Arvind decides he is going to list out all the languages $\{\mathcal{L}_1, \mathcal{L}_2, \ldots\}$. Will he succeed? That is, will every language eventually appear on his list?

No, Arvind will not succeed in listing all languages. The set of all languages over any alphabet is uncountable, while any list must be countable.

Each language corresponds to a subset of Σ^* (all possible strings). For even a binary alphabet, this means there are $2^{\aleph_0}$ different possible languages, which is uncountably infinite.

By Cantor's diagonal argument, no matter how Arvind constructs his list, there will always be languages that cannot appear on it.

III. Determine which strings can be generated by the corresponding regular expression. Justify your answers.

(a) Regular expression: $\overline{\{1\}^*} \bullet \{0\}^*$. Strings: $\epsilon, 000; 11; 000111$.

Two parts here, 1: complement set of [only 1s and empty string]. 2: 0s (including empty strings).
Since each part will concatenated in given order, we can check whether the regex can make given strings.

ϵ : The minimum length of the regex is 1, because the first part does not allow length smaller than 1.

$000 : p_1 = \{0\}, p_2 = \{00\}$

11 : Try splitting $\epsilon \bullet 11,\ 1 \bullet 1,\ 11 \bullet \epsilon$. None of them works

$000111 : p_1 = \{000111\}, p_2 = \{\epsilon\}$.

(b) Regular expression: $\{0.01\}^* \cap \{1, 10\}^*$. Strings: $101110; 00111; 00100; 01100$.

Before start splitting the given strings, notice that each rule before joint set (so, $\{0.01\}^*$ and $\{1, 10\}^*$) are contradicting each other.

The first rule shows that the string should be start with 0, while other state that it should start with 1.
Thus, none of the string could be in there, except the empty string, ϵ.

IV. Prove or disprove: there is a language $\mathcal{L}$ for which $\overline{\mathcal{L}^*} = (\overline{\mathcal{L}})^*$

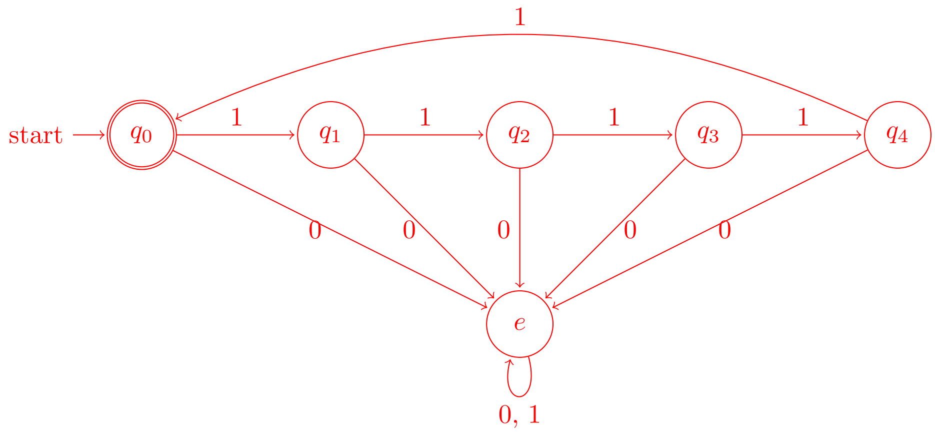
Assume that there's a language $\mathcal{L}$. Then, in any cases, $\mathcal{L}^*$ contains ϵ.
In other words, $\overline{\mathcal{L}^*}$ does not contains ϵ.

Now consider $(\overline{\mathcal{L}})^*$.
For any language, the kleene star of complement of $\mathcal{L}$, or, $(\overline{\mathcal{L}})^*$, will always contain ϵ.

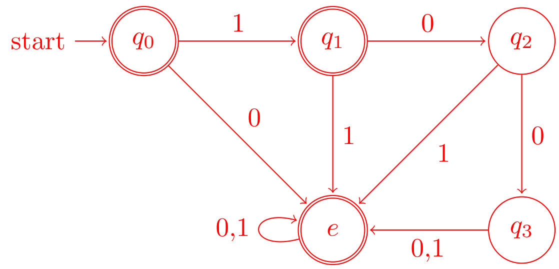
Therefore, the two sets are not equal, disproving the claim: there is a language $\mathcal{L}$ for which $\overline{\mathcal{L}^*} = (\overline{\mathcal{L}})^*$.
■

V. Give DFAs for the languages $\mathcal{L}_1 = \{1^{\bullet 5n} \mid n \geq 0\}$ and $\mathcal{L}_2 = \overline{\{10, 100\}}$.

DFA for $\mathcal{L}_1$:



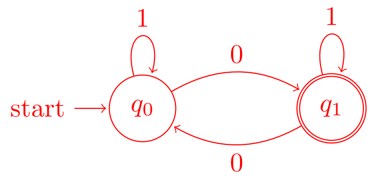
DFA for $\mathcal{L}_2$:



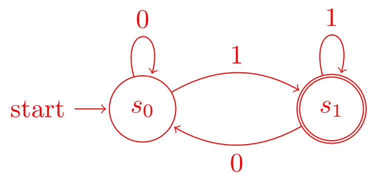
Recitation Solutions

VI. Give DFAs for these languages:

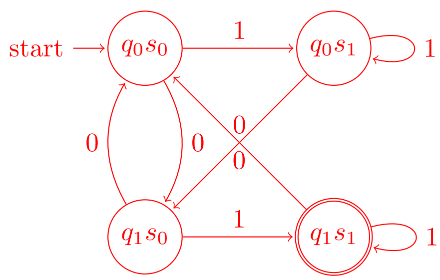
(a) Strings with an odd number of 0s.



(b) Strings that end in 1.



(c) Strings with an odd number of 0s that end in 1. Use product states and the two DFAs from above.



VII.

To construct a DFA for the language of all strings of length at most ℓ, we need to track the length of the input string. The DFA will have exactly $\ell + 2$ states:

- One initial state q_0 (accepting state, corresponds to length 0).
- One state q_i for each length $1 \leq i \leq \ell$ (each accepting).
- One "trap" state $q_{\ell+1}$ (non-accepting), for strings with length greater than ℓ.

The transitions are:

- From each state q_i $(0 \leq i \leq \ell - 1)$, upon reading any character, transition to state q_{i+1}.
- From state q_ℓ, upon reading any character, transition to the trap state $q_{\ell+1}$.
- The trap state loops on itself for all inputs.

This construction is minimal because:

- Each state uniquely represents strings of a specific length.
- Any fewer states would lose the distinction between lengths and thus accept incorrect strings or reject valid ones.

VIII.

Suppose $\mathcal{L}_1$ and $\mathcal{L}_2$ are non-regular languages.

(a) $\mathcal{L}_1 \cap \mathcal{L}_2$: Not necessarily non-regular. Counterexample: If $\mathcal{L}_1 = \{a^n b^n \mid n \geq 0\}$ and $\mathcal{L}_2 = \{b^n a^n \mid n \geq 0\}$, then $\mathcal{L}_1 \cap \mathcal{L}_2 = \{\epsilon\}$, which is regular.

(b) $\mathcal{L}_1 \cup \mathcal{L}_2$: Not necessarily non-regular. Counterexample: Consider

$$\mathcal{L}_1 = \{a^n | n \text{ is prime}\}$$

which is non-regular and

$$\mathcal{L}_2 = \{a^m | m \text{ is composite or is } 1\}$$

which is also non-regular. Their union is

$$\mathcal{L}_1 \cup \mathcal{L}_2 = \{a^n | n \in \mathbb{N}\}$$

which is regular.

(c) $\mathcal{L}_1 \cdot \mathcal{L}_2$: Not necessarily non-regular. Counterexample: Consider

$$\mathcal{L}_1 = \{a^n | n \text{ is prime}\}$$

which is non-regular and

$$\mathcal{L}_2 = \{a^m | m \text{ is composite or is } 1\}$$

which is also non-regular. Since every positive number greater than 3 can be written as the sum of a prime number and a composite number or 1, their concatenation is

$$\mathcal{L}_1 \cdot \mathcal{L}_2 = \{a^{n+m}\} = \{a^k | k \in \mathbb{N} \text{ and } k \geq 3\}$$

which is regular.

(d) $\overline{\mathcal{L}_1}$: Necessarily non-regular. Proof by contradiction: Assume $\overline{\mathcal{L}_1}$ is regular. Since the class of regular languages is closed under complement, the complement of $\overline{\mathcal{L}_1}$, which is $\mathcal{L}_1$, must also be regular. This contradicts the initial assumption that $\mathcal{L}_1$ is non-regular. Hence, $\overline{\mathcal{L}_1}$ cannot be regular.