CS 39: Theory of Computation **Winter '23**

PSET 3 — 01/30/2023

Prof. Chakrabarti ***Student: Amittai Siavava***

Credit Statement

I discussed ideas for this homework assignment with Paul Shin.

I also referred to the following books:

(a) **Introduction to the Theory of Computation** by **Michael Sipser**.

(b) **A Mathematical Introduction to Logic** by **Herbert Enderton**.

Problem 1.

Recall the CYCLE operation from Homework 1:

$$\textsc{Cycle}(L) = \{yx : x, y \in \Sigma^* \text{ and } xy \in L\}$$

Prove that if L is regular, then so is $\textsc{Cycle}(L)$.

Problem 2.

(a) Draw a 3-state DFA for the language

$$D_3 = \left\{x \in \{0,1\}^* : \beta(x) \text{ is divisible by } 3\right\},$$

where $\beta(x)$ is the string x interpreted as a binary number.

(b) Convert the DFA into a regular expression using the R_{ij}^k method.

To do this systematically, make a big table with 9 rows indexed by the pairs (i, j) and 4 columns indexed by the possible values of $k \in \{0, 1, 2, 3\}$. Fill each cell of the table with a regular expression that generates the corresponding R_{ij}^k. Sometimes, you'll obtain a complicated regular expression if you directly apply the equations from class. In such cases, show what the equations give you and only then simplify. You may use the shorthand $X+$ to denote XX^*, where X is an arbitrary regular expression.

Problem 3.

For each of the following languages, decide whether it is regular or not. If it is regular, give a finite automaton or a regular expression for it. If is is not regular, use any combination of the pumping lemma, closure properties, or results *proved* in class.

(a) $L_1 = \{0^m 1^n 0^{m+n} : m, n \geq 0\}$.

(b) $L_2 = \left\{xwx^R : x, w \in \{0,1\}^*, |x| > 0 \text{ and } |y| > 0\right\}$.

(c) $L_3 = \{0^m 1^n : m \text{ divides } n\}$.

(d) $L_4 = \{0^n 1^p : n \leq 4 \text{ or } p \text{ is prime (or both)}\}$.

(e) The infinite union $\bigcup_{n\geq 1}^{\infty} A_i$, where each A_i is a regular language. *The question should be interpreted as asking whether the union is **guaranteed** to be regular no matter how the sets A_i are chosen.*

(f) The infinite intersection $\bigcap_{n\geq 1}^{\infty} A_i$, where each A_i is a regular language. *The question should be interpreted as asking whether the intersection is **guaranteed** to be regular no matter how the sets A_i are chosen.*

Problem 4.

(a) For a language A over alphabet Σ, define the relation $\equiv_A$ on strings in Σ^* as follows: "$x \equiv_A y$" means:

$$\forall w \in \Sigma^* (xw \in A \iff yw \in A).$$

Formally (and concisely) prove that $\equiv_A$ is an equivalence relation.

If $\equiv_A$ is an equivalence relation, then it must be reflexive, symmetric, and transitive.

(i) **Reflexivity:**

(ii) **Symmetry:**

(iii) **Transitivity:**

(b) The equivalence $\equiv_A$ is called the *left equivalence relation* of the language A. An equivalence relation on a set partitions the set into disjoint subsets called equivalence classes in the following way: two elements belong to the same class iff they are related by the equivalence relation. Thus, $\equiv_A$, which is a relation on Σ^*, partitions Σ^* into equivalence classes: these are called the left equivalence classes of the language A.

For example, consider the language $C = \{x \in 0, 1^* : |x| \text{is even}\}$ over the alphabet $\{0, 1\}$. Convince yourself that any two even-length strings are related by $\equiv_C$, as are any two odd-length strings. Also, no odd-length string is related by $\equiv_C$ to an even-length string. Thus, C has exactly two left equivalence classes: (1) odd-length strings, i.e., $\{0, 1\}^* - C$, and (2) even-length strings, i.e., C.

Similarly, convince yourself that the language $B = (01)^*$ over the alphabet $\{0, 1\}$ has three equivalence classes, which are: (1) B, (2) $(01)^*0$, and (3) $\{0, 1\}^* - (B \cup (01)^*0)$.

Describe the left equivalence classes of each of the following languages (no proofs required):

(i) $L_1 = \{a, aa, aaa, b, ba, baa\}$ over the alphabet $\{a, b\}$.

(ii) $L_2 = a * b * c*$ over the alphabet $\{a, b, c\}$.

(iii) $L_3 = (ab \cup ba)^*$ over the alphabet $\{a, b\}$.

(iv) $L_4 = \{0^n 1^n : n \geq 0\}$ over the alphabet $\{0, 1\}$.

Problem 5.

Now, apply the notion of left equivalence to the theory of regular languages.

(a) For a language A over alphabet Σ and a string $x \in \Sigma$, let $[x]_A$ denote the equivalence class (of A) to which x belongs. For instance, consider the set $X_1 = (01)^*$, $X_2 = (01)^*0$, and $X_3 = \{0,1\}^* - (X_1 \cup X_2)$, then:

  (i) $[010]_B$ denotes the set X_2,

  (ii) $[01010]_B$ also denotes the same set X_2, and

  (iii) $[\varepsilon]_B$ and $[0101]_B$ both denote the set X_1.

  There are rarely one unique way way to write an equivalence class $[x]_A$ — there are usually multiple choices for x. These choices are called *representatives* of the equivalence class.

  Prove that for any $x \in \Sigma^*$ and $a \in \Sigma$, the class $[xa]_A$ is completely determined by the class $[x]_A$ and the alphabet symbol A — that is, prove that the particular x we pick as a representative for the class $[x]_A$ is inconsequential.

(b) Suppose a language A over alphabet Σ has finitely many equivalence classes; $[x_1]_A, [x_2]_A, \ldots, [x_n]_A$ for some $n \geq 1$. Prove that A is regular.

Problem 6.

The connection between left equivalence and regularity is even deeper.

(a) Let A be a regular language over alphabet Σ. Prove (formally and rigorously) that A has finitely many distinct left equivalence classes. The function δ^* we defined in class might be useful.

(b) Using one or more of the results proved above, give alternate proofs that the following languages are not regular:

(i) $\{0^n 1^n : n \geq 0\}$.

(ii) $\left\{x \in \{0,1\}^8 : x = x^R\right\}$.