Formal Languages & Regular Languages

Exercise 1

Let the alphabet A be binary digits. $A = \{ 0, 1 \}$

For the exercise, we consider each word of A^* to represent a number in $\mathbb{N}$, in the usual way:

0	represents	0
1	represents	1
110	represents	6
1010	represents	10

Note that leading zeros are ignored, and that the empty string ε is assigned the value 0.

0011	represents	3
ε	represents	0

Question 1.1

Define a function $f : A^* \rightarrow \mathbb{N}$ that converts words over the alphabet A into numbers in $\mathbb{N}$ according to the above specification.

Question 1.2

Let E be the language of even numbers: $E = \{ w \in A^* \mid f(w) \text{ is even} \}$

Prove that $EE = E$. To do so, prove that all elements of E are also elements of EE, and that all elements of EE are also elements of E.

Question 1.3

Prove that $E^* = E$. You may find the fact you have proven in question 1.2 to be useful here.

Question 1.4

Build a regular expression whose language is E.

Exercise 2

Let A be some alphabet and let f: $A^* \rightarrow \{\text{true}, \text{false}\}$ be a computable function from A^* to true or false. Let L be the language defined by f.

$$L = \{ w \in A^* \mid f(w) = \text{true} \}$$

Find an algorithm that, given a word over the alphabet A, decides whether the word is part of L^*, the Kleene closure of L. Your algorithm may of course invoke f, but only a number of times polynomial in the size of the input word.

Exercise 3

There are basic properties of formal languages that you will frequently encounter when working with such objects.

One such property is *nullability*. We say that a language $L \subseteq A^*$ is *nullable* if it contains the empty word ε. This property looks innocent enough, but the nullability of a language will often be relevant.

Another such property is the so-called *first* set of a language. We define the *first* set of a language $L \subseteq A^*$ to be the set of characters in A that appear at the start of a word in L.

$$\text{first}(L) := \{ w_0 \mid w \in L \}$$

Question 3.1

Give a recursive function to compute the nullability of a regular expression.

Question 3.2

Give a recursive function to compute the *first* set for regular expressions.

Exercise 4

Let A be some alphabet and L be a language over A. We define the derivative of a language L with respect to an element x of A to be the set of all words w such that xw is in L.

$$\frac{dL}{dx} = \{ w \in A^* \mid xw \in L \}$$

Question 4.1

Prove that whenever a language is regular, then its derivation is also regular. To do so, build a recursive function that return a regular expression for the derivation of L with respect to x given a regular expression for L.

Question 4.2

Can you come up with an algorithm to decide whether a word is part of the language of a regular expression using regular expression derivation?