# 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 $\varepsilon$ is assigned the value 0.

| | | |
|---|---|---|
| 0011 | represents | 3 |
| $\varepsilon$ | 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 $\text{true}$ or $\text{false}$. 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.