Finite Automata Theory and Formal Languages
TMV027/DIT321

Regular Languages

Exercise 4

In these exercises, book sections, exercise numbers and pages refer to those in the third edition of the course book.

Basic exercises

1. Consider the regular sets denoted by the following pairs of regular expressions, with $\Sigma = \{a, b, c\}$. For each pair, say whether the two corresponding languages are equal. If so, justify your answers as formally as you can. If not, give an example of a word which belongs to one of the languages but not the other.

   (a) $(a + b)^*$ and $a^* + b^*$

   (b) $a(bca)^*bc$ and $ab(cab)^*c$

   (c) $\emptyset^*$ and ϵ^*

   (d) $(a^*b^*)^*$ and $(a^*b)^*$

   (e) $(ab + a)^*a$ and $a(ba + a)^*$

2. If $w \in \{0, 1\}^*$, we write $\sharp i(w)$ for the number of occurrences of i in w (with $i = 0$ or 1). Show with the help of the Pumping lemma that the following 2 languages are not regular

   $$\begin{aligned} \mathcal{L} &= \{w \in \{0,1\}^* \mid \sharp 0(w) = 2 \times \sharp 1(w)\} \\ \mathcal{M} &= \{w \in \{0,1\}^* \mid \sharp 0(w) \leqslant \sharp 1(w) \leqslant \sharp 0(w) + 1\} \end{aligned}$$

   (hint: look at example 4.2).

   Show however that the following language is regular

   $$\mathcal{N} = \{w \in \{0,1\}^* \mid \sharp 0(w) \times \sharp 1(w) \text{ is even}\}$$

3. Do exercises 4.2.13.

4. Do exercise 4.4.1 and 4.4.2.

Additional exercises

1. Prove the following equalities:

   $b + ab^* + aa^*b + aa^*ab^* = a^*(b + ab^*)$

   $a^*(b + ab^*) = b + aa^*b^*$

2. Do exercises 4.1.1, 4.1.2 a, e–h and 4.1.4

3. Do exercises 4.2.1, 4.2.2, 4.2.3, 4.2.4, 4.2.5 a–e, and 4.2.15.

4. Do exercise 4.2.17.

   Note: the statement can be easily proved if one considers the definition of strings given in the book. For the one we use in the lectures one needs to prove a more general lemma where the equality is true for any state and then instantiate this more general lemma on the starting state.

5. Prove that the equivalence of states is indeed an equivalence relation.

Programming Exercises

1. Program all the decision properties.

2. Write a program for the table-filling algorithm.

3. Write a program for the minimisation algorithm.