Abstract

The goal of this thesis is to study subclasses of regular languages from the perspective of descriptional and computational complexity. We start by surveying some of these subregular language classes that were subject to extensive research in the past decades. We consider selected classes of the dot-depth hierarchy (finite, co-finite, piecewise testable, definite, generalized definite, star-free languages and languages of dot-depth one), two-sided comets and their variants comets and stars, convex languages and their subclasses (prefix-, suffix-, factor-, subword-free and -closed languages, left, right, two-sided, and all-sided ideal languages) with solid codes and comma-free codes, and the classes of singleton, combinational, finitely generated left ideal, symmetric definite, ordered, group, union-free, and power-separating languages. We summarize some previous work and properties regarding these language classes and describe new inclusion and non-inclusion relations among them. As a result, we obtain a birds-eye view of the landscape of subregular classes that prove useful in streamlining several proofs.

Next, we study the regular language operations of intersection, union, concatenation, power, positive closure, Kleene star, reversal, and complementation with the intent of answering the following question: Is a given subclass closed under a given operation, i.e., does the operation preserve the subclass? For several pairs of subclass and operation we provide an answer to the proposed question.

Focusing on the descriptional complexity aspects of the thesis, we proceed to investigate the nondeterministic state complexity of the mentioned regular operations on our subclasses. For all considered cases except one, we obtain a tight complexity bound. In order to obtain these results, several fooling set methods are used, which we review as well.

Regarding computational complexity, we focus on the following decision problem: Given a subregular class and a language represented by a deterministic or nondeterministic finite automaton, what is the resource cost of deciding, whether this language belongs to the considered class? We proceed to obtain inclusion results in the relevant complexity classes, by providing appropriate algorithms. In order to get completeness, we present reductions from problems that are known to be complete for the corresponding complexity classes. In most cases, we show that our considered decision problems are either NL-complete or PSPACE-complete. However, in certain instances, we provide only partial results or leave the problem for future work.

Keywords: regular languages, finite automata, regular operations, descriptional complexity, computational complexity