Chapter 1

Introduction

Regular languages are a well-studied class of formal languages. In their current form, they were first studied by Kleene [6], who introduced regular expressions. A precursor [?] was suggested by earlier by McCulloch and Pitts ([?]). Regular languages were used to describe neural activity in nerve nets. The concept of deterministic finite automata was introduced before Kleene's invention of regular expressions by Huffman and Moore ([?], [?], [?]). Rabin and Scott later introduced the concept non-deterministic finite automata [?], for which they were given the Turing award [?].

History: Myhill-Nerode, Brzozowski?

We will prove and formalize the equivalence of differnt characterizations of regular languages: regular expressions, finite automata and the characterizations given by Myhill-Nerode theorem.

Theoretical importance: Second-Order logic, Monoids

1.1 Recent work

There have been many publications on regular languages in recent years. Many of them investigate decidability of equivalence of regular expressions with a focus on automatically deciding Kleene algebras ([2], [?], [?], [?], [?]). There has also been a paper on formalizing the Myhill-Nerode theorem using only regular expressions and not, as is commonly done, finite automata ([?]). The authors state that this unusual choice stems, at least partly, from the restrictions of Isabelle/HOL (and similar HOL-based theorem provers) which effectively prevents straight-forward formalizations of finite automata. This restriction does not apply to Coq. In fact, our formalization of finite automata turns out to be very close to the mathematical definitio turns out to be very close to the mathematical definition.

Practical importance? Posix

1.2 Contributions

Our goal is to give a concise formalization of the equivalence between regular expressions, finite automata and the Myhill-Nerode characterization. We give procedures to convert between these characterizations and prove their correctness. The contribution of this thesis is in the way in which we formalize these well-known results. We focused not on executability but rather on staying close to the mathematical definitions without sacrificing convenience. Our development shows that Coq (with SSReflect) is well suited for this kind of formalization. We have also developed a new characterization derived from the Myhill-Nerode theorem, which we prove equivalent to the other characterizations.

1.3 Outline

In Chapter 2 we introduce the Coq framework and the SSReflect extension. We give a brief introduction of the SSReflect-specific syntax and concepts that are relevant to our formalization.

In Chapter **??** we give basic definitions (words, languages, etc.). We also introduce decidable languages, regular languages and regular expressions. Furthermore, we prove the decidability of regular languages.

In Chapter **??** we introduce finite automata. We prove the equivalence of deterministic and non-deterministic finite automata. We also give a procedure to remove unreachable states from deterministic finite automata. Furthermore, we prove decidability of emptiness and equivalence of finite automata. Finally, we prove the equivalence of regular expressions and finite automata.

In Chapter **??** we introduce the Myhill-Nerode theorem. We give three different characterizations of regular languages based on the Myhill-Nerode theorem and prove them equivalent to finite automata. The formalization of these characterizations is interesting in itself due to the fact that we had to find a suitable representation of quotient types in Coq, which has no notion of quotient types.