

Constructive Formalization of Regular Languages

Jan-Oliver Kaiser

September 5, 2012

Contents

1	Introduction	2
1.1	Recent work	2
1	Coq and SSReflect	2
1.1	Coq	2
1.2	SSREFLECT	2
1.2.1	Finite Types and Ordinals	2
1.2.2	Boolean Reflection	2
1.2.3	Boolean Predicates	2
3	Decidable Languages	7
3.1	Definition	7
3.1.1	Operation on languages	7
3.2	Regular Languages	9
3.2.1	Regular Expressions	9
3.2.2	Deciding Language Membership	10
1	Finite Automata	2
1.1	Definition	2
1.1.1	Determinism and Non-Determinism	2
1.2	Connected Components	4
1.3	Emptiness	5
1.4	Deciding Equivalence of Finite Automata	5
1.5	Regular Expressions and Finite Automata	6
1.5.1	Regular Expressions to Finite Automata	6
1.5.2	Deciding Equivalence of Regular Expressions	6
1.5.3	Finite Automata to Regular Expressions	6
5	Myhill-Nerode	20
5.1	Definition	20
5.2	Finite Partitionings and Equivalence Classes	21
5.3	Minimizing Equivalence Classes	22
5.4	Finite Automata and Myhill-Nerode	23
5.4.1	Finite Automata to Myhill-Nerode	23

<i>CONTENTS</i>	3
5.4.2 Myhill-Nerode to Finite Automata	23
2 Conclusion	3
3 References	4

Chapter 1

Introduction

Regular languages are a well-studied class of formal languages. We will prove the equivalence of three well-known characterizations of regular languages: regular expressions, finite automata and the characterization given by Myhill-Nerode theorem.

History

Theoretical
importance

Practical
importance?

1.1 Recent work

There have been many publications on regular languages in recent years. Many of them investigate decidability of equivalence of regular languages, though there have also been new equivalence proofs regarding different characterizations of regular languages.

Chapter 2

Conclusion

Chapter 3

References