Automata Theory - Equivalence and Minimization of Automata

Max Randall
Chapman University

March 15, 2025

Contents

1 Introduction

This report focuses on Equivalence and Minimization of Automata, covering the process of determining whether two different DFA representations define the same language and how to construct the smallest possible DFA that accepts a given regular language. The table-filling algorithm is introduced as a systematic way to check equivalence of states within a DFA by distinguishing them through input strings. This allows us to merge equivalent states, reducing the number of states while preserving language recognition. We also examine the uniqueness of the minimal DFA and its theoretical guarantees. Finally, the report discusses the complexity of these procedures and their implications for automata design.

2 Chapter 4.4

Chapter 4.4 covers the equivalence and minimization of automata, focusing on the problem of determining whether two descriptions of regular languages define the same language. The chapter introduces the table-filling algorithm, a method to determine when two states of a DFA are equivalent by examining how they transition on different input strings. If two states lead to the same accepting or rejecting states for all possible inputs, they are considered equivalent and can be merged into a single state. If not, they are distinguishable. This process allows us to construct a minimal DFA, the smallest DFA that accepts a given language. The minimization process follows a structured approach: (1) eliminate unreachable states, (2) partition the remaining states into equivalence classes, and (3) construct a new DFA where each equivalence class represents a single state. A key result of this approach is that the minimal DFA is unique up to renaming of states, meaning that any two minimal DFAs for the same language will have the same structure. The chapter also explores the complexity of minimization, showing that the table-filling algorithm runs in $\mathrm{O(n^2)}$ time, where n is the number of states. Additionally, the chapter presents an alternative perspective on DFA equivalence testing by constructing a combined DFA from two different automata and checking whether their start states are equivalent. If they are, the two original DFAs define the same language. The final part of the chapter discusses NFA minimization, explaining why the same state-equivalence method does not necessarily produce a minimal NFA, as NFAs can have multiple transitions for the same input, which complicates state merging. An example is provided where an NFA remains the same size despite attempting to group equivalent states, highlighting the limitations of direct minimization techniques. The chapter concludes by proving the optimality of DFA minimization, showing that no DFA with fewer states than the minimal DFA can accept the same language. This is done by assuming a smaller DFA exists and deriving a contradiction using state distinguishability. This guarantees that the table-filling method produces the smallest possible DFA, making it a fundamental tool in automata theory.

3 Exercises

Note: The following exercises test our understanding of state equivalence, DFA minimization, and the application of the table-filling algorithm.

4 Conclusion

In this report, we explored methods for testing equivalence of automata and minimizing DFAs using the table-filling algorithm. We demonstrated how equivalent states can be merged into a unique minimal DFA, ensuring that no smaller DFA exists for the same language. The minimization process not only reduces computational complexity but also provides theoretical guarantees regarding DFA uniqueness. Additionally, we saw that while DFAs have a well-defined minimization process, NFAs do not always exhibit the same behavior, requiring different optimization techniques. These concepts are fundamental in automata theory, with practical applications in compiler design, pattern recognition, and language processing.

References

[HMU] . E. Hopcroft, R. Motwani, J. D. Ullman: *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*, Archive.org Link.