

**杭州电子科技大学**

**《编译原理课程实践》**

**实验报告**

题 目： DFA最小化实现

学 院： 卓越学院

专 业： 计算机科学与技术

班 级： 23186211

学 号： 23050817

姓 名： 吴锴

完成日期： 2024-10-30

1. **实验目的**

1. 掌握确定有限⾃动机（DFA）的最⼩化原理和算法，尤其是Hopcroft算法（即课上所讲的“求异法”）。

2. 学习DFA状态等价性的判定⽅法，理解最⼩化过程中的分割和合并策略。

3. 实现DFA最⼩化算法，并验证最⼩化DFA的正确性。

4. 延续前两次实验的设计，确保数据结构能贯通整个⾃动机系列实验。

5. 提⾼算法优化和编程实现能⼒，增强对编译原理的理解。

1. **实验内容与实验要求**

实验内容

1. 理论背景：DFA最⼩化是将DFA状态数减少到最⼩的过程，通过合并等价状态，实现最优的状态机表示。

Hopcroft算法是求异法的⼀种⾼效实现，它通过维护状态的分割并使⽤快速查找机制来优化最⼩化过程。

2. 任务描述：实现DFA最⼩化算法，将给定的DFA简化为状态数最少的等价DFA。验证最⼩化DFA的正确性，并

对⽐最⼩化前后的状态数量。

3. 实验步骤

理解Hopcroft算法的基本原理，包括状态等价的判定标准和状态合并的⽅法。

实现Hopcroft算法，将原DFA简化为等价的最⼩化DFA。

设计合理的数据结构表示最⼩化后的DFA，确保其与前两次实验的NFA和DFA数据结构保持⼀致。

验证最⼩化DFA的正确性，确保其接受的语⾔与原DFA相同。

实验要求

1. 输⼊输出要求

输⼊：⼀个DFA（包括状态集合、状态转换表、初始状态和接受状态集合）。

输出：最⼩化后的DFA状态集合及其转换关系，指明最⼩化前后的状态数和状态转换关系。

2. 算法要求

实现Hopcroft算法，通过分割状态集合和快速查找机制来最⼩化DFA。

⽀持状态等价性判定及状态的合并操作。

3. 数据结构要求

设计适合Hopcroft算法的⾼效数据结构，如⽤于记录状态分割的集合、合并后的状态转换表等。

保持与前两次实验的数据结构⼀致，⽅便整个⾃动机系列实验的贯通实现。

4. 程序要求

使⽤C/C++、Java、Python等语⾔编写程序，代码结构清晰，具备良好的注释。

提供详细的实验报告，包括算法设计、实现过程、测试结果和问题分析。

5. 实验报告要求【整合到最后提交的个⼈所有实验报告中，加上⽬录】

描述实验⽬的和内容。

解释Hopcroft算法的原理和实现步骤，说明数据结构的设计思路。

给出测试⽤例和结果，分析最⼩化前后的差异。

总结实验的收获和遇到的挑战。

1. **设计方案与算法描述**

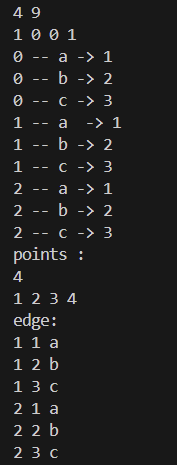
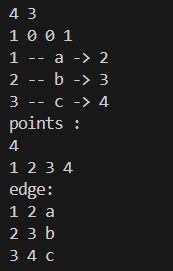
1.读取输⼊DFA，包括状态集合、状态转换表、初始状态和接受状态。

2.根据接受状态和⾮接受状态的划分，初始化状态分割。

3.使⽤Hopcroft算法细化状态分割，按照输⼊符号集逐步分割等价状态，直⾄⽆法再细分。在存储方式上采用经典的并查集算法。

4.输出最⼩化前后的状态数量和转换关系。

1. **测试结果**



1. **源代码**

见附录DFA2MINIDFA