**第三章 等价划分法的数学原理**

**3.1 集合与划分**

3.1.1 划分的定义与性质

集合的划分是将一个集合分解为若干个互不相交且非空的子集，使得这些子集的并集等于原集合。设S是一个非空集合，若将S划分为n个非空子集S₁, S₂, ..., Sn，则必须满足以下条件：∪i=1^n Si = S （所有子集的并集等于原集合）Si ∩ Sj = Ø （任意两个不同的子集没有交集）划分的性质包括：每个子集都是S的非空真子集或等于S本身。每个元素恰好属于一个子集。划分数通常不唯一，不同的划分方式会得到不同的子集。

3.1.2 划分在DFA最小化中的应用

在DFA最小化过程中，划分的概念被用来将DFA的状态集Q划分为若干个子集，使得每个子集中的状态在处理相同输入时表现一致。初始划分通常按照状态是否为接受状态来进行。例如，将所有接受状态放入一个子集，将所有非接受状态放入另一个子集。然后，通过反复细化划分，逐步增加划分的粒度，直到不能再细化为止。具体操作中，使用拆分函数split(s, c)，该函数将当前划分的状态集合根据输入字符c进一步细分。如果某个状态在输入c下到达不同的子集，则需要进一步划分。这一过程不断迭代，直至状态集无法再细化。最终得到最细粒度的划分，即每个子集中的状态在所有输入下都表现出相同的行为。

**3.2 等价关系与等价类**

3.2.1 等价关系的定义与性质

等价关系是定义在集合上的一种二元关系，满足以下三条性质：自反性：对于所有x∈S，有x ↔ x成立。对称性：对于所有x, y∈S，如果x ↔ y成立，则y ↔ x也成立。传递性：对于所有x, y, z∈S，如果x ↔ y且y ↔ z，则x ↔ z成立。等价关系将集合划分为若干个等价类，每个等价类包含所有相互等价的元素。等价类是等价关系的极大子集，具有如下性质：每个元素属于且仅属于一个等价类。不同等价类之间互不相交。等价类内部元素间彼此等价。

3.2.2 等价类在DFA最小化中的应用

在DFA最小化中等价关系用于判断两个状态是否可以合并。如果两个状态在所有输入条件下的次态都属于同一子集，则这两个状态是等价的，可以合并为一个状态。具体而言，通过检查每个状态下的所有可能转移，如果两个状态在所有可能输入下的转移结果都在相同的子集中，则它们是等价的。利用这个原则，可以不断合并等价状态，从而减少DFA的总状态数。例如，假设有两个状态p和q，如果对于所有输入符号x，都有δ(p, x)和δ(q, x)落在同一子集中，那么p和q可以合并为一个代表状态。这个过程不断重复，直至没有更多的状态可以合并为止。通过这种方法，可以实现DFA的最小化，使其接受相同的语言但具有较少的状态数量。

**3.3 图论**

3.3.1 有向图的基本概念

有向图是一种由节点和有向边组成的图形结构，其中每条边都有一个方向。形式上，一个有向图G可以定义为一个三元组(V, E, δ)，其中：V是非空的节点集合。E是边的集合，每条边是一个有序对(u, v)，表示从节点u到节点v的一条有向边。δ是转移函数，表示每个节点在每个输入下的转移情况。有向图可以用来表示DFA的状态转移情况，其中节点表示状态，边表示状态之间的转移。这种图形结构便于直观地分析和理解DFA的行为和结构。

3.3.2 DFA的状态转移图

在DFA最小化过程中，DFA可以被视为一个有向图，其中每个节点表示一个状态，每条有向边表示一个状态在特定输入下的转移。通过构建和遍历这个有向图，可以更清晰地观察和操作状态之间的转移关系。

**3.4 递归与迭代**

3.4.1 递归思想在等价划分中的应用

递归是一种解决问题的方法，通过将问题分解为更小的同类问题来解决。在等价划分法中，递归思想用于不断细化状态划分，每次根据更细粒度的条件对当前划分进行调整。具体而言，递归函数split(s, c)会不断调用自身来处理更小的子集，直到无法再细分为止。递归的使用使得等价划分法能够在每次划分时都考虑所有可能的情况，从而确保最终得到的是最细粒度的划分。递归的深度和复杂度取决于DFA的状态数和输入字母表的大小，但通过有效的剪枝和优化策略，可以显著减少不必要的计算量。

3.4.2 迭代思想在DFA最小化中的应用

迭代是另一种常用的算法思想，通过反复应用一系列操作来逐步逼近解决方案。在DFA最小化中，迭代思想用于不断重复划分和细化过程，直到状态划分不再变化为止。每次迭代中，都会根据当前划分结果计算新的划分，并更新划分数组和子集数组。例如，初始划分可能只区分接受状态和非接受状态，但随着迭代的进行，划分会越来越细，直到每个子集中的状态在所有输入下都表现出完全相同的行为。迭代思想的优点是实现简单、过程清晰，适用于逐步优化和调整的问题。结合递归和迭代的思想，等价划分法能够高效地实现DFA的最小化，同时保持算法的正确性和完备性。通过不断细化划分和合并等价状态，最终得到一个简化的DFA，接受与原DFA相同的语言但具有更少的状态数量。

**第四章 等价划分法的具体步骤**

**4.1 初始划分**

初始划分是等价划分法的起点，通常将DFA的状态集合划分为两个子集：接受状态集合和非接受状态集合。这一步非常直观，可以通过简单地检查每个状态是否在接受状态集中来实现。初始划分的目的是提供一个粗略的分类，以便后续步骤在此基础上进行细化。

**4.2 构造划分数组和子集数组**

在初始划分之后，需要构造两个关键的数组：划分数组和子集数组。划分数组用于存储每个状态所属的子集，而子集数组则存储每个子集对应的状态集合。这些数组在整个算法过程中会被不断更新和维护。初始状态下，划分数组根据初始划分结果填充，子集数组则根据划分数组构建。

**4.3 迭代细化划分**

迭代细化划分是等价划分法的核心步骤。通过不断地应用拆分函数来细化当前的划分，直到无法再细化为止。具体来说，对于每个子集和每个输入符号，检查该子集在当前输入符号下的转移情况。如果不同的状态转移到不同的子集，则需要将这些状态进一步细分。这一过程不断重复，直到所有子集在所有输入下都不再发生变化为止。

**4.4 构造最小化DFA**

在完成最终的细化划分后，可以根据子集数组构造最小化的DFA。每个子集对应一个最小化后的状态，状态之间的转移关系根据子集数组和划分数组确定。初始状态和接受状态也相应地映射到最小化后的状态集合中。最终得到的DFA具有最少的状态数量，同时接受与原DFA相同的语言。

参考文献[1]苏开睿,李文杰.基于等价类的DFA最小化算法研究[J].计算机工程与科学,2024,16(03):578-585.

[2]张玉林,李静璇,王宇晴.基于邻接表的DFA最小化算法[J].信息技术与信息化,2024(04):99-102+106.

[3]刘路浩,陈晓明,冯艳坡.基于C语言实现的DFA最小化算法[J].信息与电脑(理论版),2020,32(23):96-98.