目录

1 介绍

自 1950 年来，有限自动机的就一直在研究当中。简单来说就是找到一个唯一的（直至同构）最小的确定性有限自动机，它能接收与给定的确定性有限自动机相同的语言。解决这一问题的算法应用范围很广，从编译器构造到硬件电路的最小化都有它的身影。有了形式多样的应用程序，不同的表示形式的数量也在增加: 大多数教科书都有自己的变体，而时间复杂度最优的算法 (Hopcroft 的算法) 仍然晦涩难懂。

本文介绍了有限自动机最小化算法的有关分类。如下所示:

- 大多数教材的作者称他们的最小化算法由 Huffman 算法 [Huff54] 和 Moore 算法 [Moor56] 直接推导得到。不幸的是，大多数教材都展示了截然不同的算法 (比如 [AU92], [ASU86], [HU79], [Wood87]), 只有由 Aho 和 Ullman 发表的算法直接源自 [Huff54, Moor56]。

- 虽然大多数算法依赖于计算状态的等价关系，但伴随算法演示的许多解释并未明确提及算法是计算等价关系、它包含的状态划分还是它的补充。

- Comparison of the algorithms is further hindered by the vastly differing styles of presentation — sometimes as imperative programs or as functional programs, but frequently only as a descriptive paragraph. 算法之间的比较进一步受到呈现方式的巨大差异的阻碍。有时作为命令式程序或函数式程序，但通常只作为描述性段落。

A related taxonomy of finite automata construction algorithms appears in [Wats93].

有限自动机构造算法的相关分类在 [WATS93] 中。

All except one of the algorithms rely on determining the set of automaton states which are equivalent. The algorithm that does not make use of equivalent states is discussed in Section 2. In Section 3 the definition and some properties of equivalence of states is given. Algorithms that compute equivalent states are presented in Section 4. The main results of the taxonomy are summarized in the conclusions Section 5. Appendices A and B give the basic definitions required for reading this paper. The definitions related to finite automata are taken from [Wats93]. The minimization algorithm relationships are shown in a "family" tree in Figure 1.

除了一个算法之外，所有依赖于确定等价的自动机状态的集合。在第 2 节中讨论了不使用等效状态的算法。在第 3 节中给出了状态等价的定义和一些性质。计算等效状态的算法在第 4 节中给出。分类的主要结果总结在结论部分 5 中。附录 A 和 B 给出了阅读本文所需的基本定义。与有限自动机相关的定义取自 [WATS93]。图 1 中的“家庭树”中显示了最小化算法关系。

The principal computation in most minimization algorithms is the determination of equivalent (or inequivalent) states ——thus yielding an equivalence relation on states. In this paper,we consider the following minimization algorithms:

大多数最小化算法的主要计算是确定等价的（或不等价的）状态，从而在状态上产生等价关系。在本文中，我们考虑以下最小化算法：

- Brzozowski's (possibly nondeterministic) finite automaton minimization algorithm as presented in [Brzo62]. This elegant algorithm (Section 2) was originally invented by Brzozowski, and has since been re-invented without credit to Brzozowski. Given a (possibly nondeterministic) finite automaton without E-transitions, this algorithm produces the minimal deterministic finite automaton accepting the same language.

Brzozowski（可能是非确定性的）有限自动机最小化算法在 [BRZO62] 中提出。这个优雅的算法（第 2 节）最初是由 Brzozowski 发明的，此后又在没有 Brzozowski 的功劳情况下被重新发明。在没有 ϵ- 跃迁的情况下，给出了一个（可能不确定的）有限自动机，该算法产生最小的确定的有限自动机接受相同的语言。

- Layerwise computation of equivalence as presented in [Wood87, Moor56 Brau88, Urba89]. This algorithm (Algorithm 4. 2) is a straightforward implementation suggested by the approximation sequence arising from the fixed-point definition of equivalence of states.

  分层等价计算等价于 [Wood87, Moor56 Brau88, Urba89] 中提出。算法（算法 4. 2）是由状态等价的定点定义产生的近似序列所建议的直接实现。

- Unordered computation of equivalence This algorithm (Algorithm 4.3, not appearing in the literature) computes the equivalence relation; pairs of states (for consideration of equivalence) are chosen in an arbitrary order.

  该算法（算法 4.3，未出现在文献中）计算等价关系；以任意顺序选择状态对（考虑等价性）。

- Unordered computation of equivalence classes as presented in [ASU86]. This algorithm (Algorithm 4.4)is a modification of the above algorithm computing equivalence of states.

  在 [ASU86] 中给出的等价类的无序计算。该算法（算法 4.4）是上述算法计算状态等价性的一种改正。

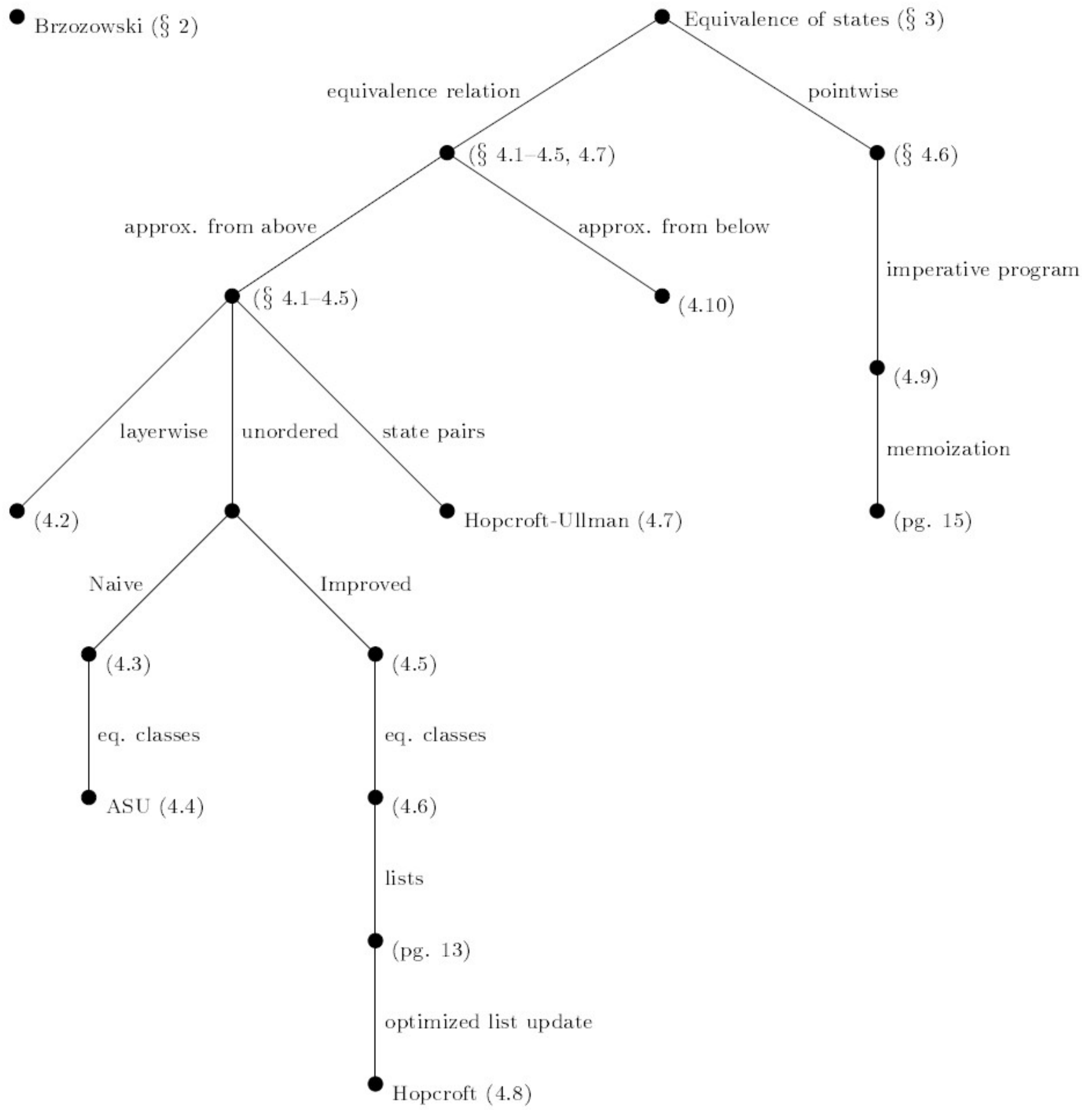

Figure 1: The family trees of finite automata minimization algorithms. Brzozowski's minimization algorithm is unrelated to the others, and appears as a separate(single vertex)tree. Each algorithm presented in this paper appears as a vertex in this tree. For each algorithm that appears explicitly in this paper, the construction number appears in parentheses(indicating where it appears in this paper). For algorithms that do not appear explicitly, a reference to the section or page number is given. Edges denote a refinement of the solution (and therefore explicit relationships between algorithms). They are labeled with the name of the refinement.

图 1：有限自动机最小化算法的关系树。Brzozowski 的最小化算法与其他算法无关，并作为一个单独的（单顶点）树出现。本文提出的每一个算法都作为树的顶点出现。对于本文中明确出现的每个算法，构造数在括号中（标示它在本文中出现的位置）。对于未显式显示的算法，给出了相应的页码。边表示解的细化（因此算法之间的显式关系）。它们被标记为细化的名称。

- Improved unordered computation of equivalence. This algorithm (Algorithm 4.5, not appearing in the literature) also computes the equivalence relation in all arbitrary order. The algorithm is a minor improvement over the other unordered algorithm.

  改进的等价的无序计算。这个算法 (算法 4.5，没有出现在文献中) 也以任意顺序计算

等价关系。该算法是对其他无序算法的一个小改进。

- Improved unordered computation of classes. This algorithm (Algorithin 4.6, not appearing in the literature) is a modification of the above algorithm to compute the equivalence classes of states. This algoritm is used in the derivation of Hopcroft's minimization algorithin.

  改进了类的无序计算。该算法（算法 4.6，不在文献中）是上述算法的修改，用来计算等价类的状态。该算法用于 Hopcroft 最小化算法的推导。

- Hopcroft and Ullman's algorithm as presented in [HU79]. This algorithm (Algorithm 4.7) computes the inequivalence (distinguishability) relation. Although it is based upon the algorithus of Huffinan and Moore [Huff54, Moor 56], this algorithm uses some interesting encoding techniques.

  Hopcroft 和 Ullman 算法在 [HU79] 中提出。该算法（算法 4.7）计算不等价（区分性）关系。虽然它是基于 Huffinan 和 Moore [Huff54，MOR 56] 的算法，但该算法使用一些有趣的编码技术。

- Hopcroft's algorithm as presented in [Hopc71, Grie73]. This algorithm (Algorithm 4.8)is the best known algorithm (in terms of running time analysis) for minimization. As the original presentation by Hopcroft is diflicult to understand, the presentation in this paper is based upon the one giveu by Gries.

  HopRofft 的算法在 [Hopc71, Grie73] 提出的。该算法（算法 4.8）是用于最小化的最有名的算法（在运行时间分析方面）。由于 Hopcroft 的原始陈述是难以理解的，本文的介绍基于 Gries 的文章。

- Pointwise computation of equivalence. This algorithm (Algorithm 4.9 not appearing in the literature) computes the equivalence of a given pair of states. It draws upon some nonautomata related techniques, such as: structural equivalence of types and memoization of functional programs.

  等价的点态计算。该算法（算法 4.9，不在文献中）计算给定状态对的等价性。它借鉴了一些非自动机相关的技术，例如：类型的结构等价和函数式程序的记忆化。

- Computation of equivalence from below (with respect to refinement). This algorithm (Algorithm 4.10, not appearing in the literature) computes the equivalence relation from below. Unlike any of the other known algorithms, the intermediate result of this algorithm can be used to construct a smaller (although not minimal) deterministic finite automaton.

  由下面的内容（关于细化）计算等价性。该算法（算法 4.10，不在文献中）计算从下面的等价关系。与任何其他已知算法不同，该算法的中间结果可用于构造较小的（虽然不是最小的）确定性有限自动机。

2 Brzozowski 提出的算法

Most minimization algorithms are applied to a DFA. In the case of a nondeterministic FA, the subset construction is applied first, followed by the minimization algorithm. In this section, we consider the possibility of applying the subset construction (with useless state removal) after an (as yet unknown) algorithm to yield a minimal DFA. We now construct such an algorithm. (The algorithm described in this section can also be used to construct the minimal Complete DFA, by replacing function subsetopt with subset.)

大多数最小化算法应用于确定的有限自动机 (DFA)。对于不确定性有限自动机，首先应用子集构造，然后应用最小化算法。在本节中，我们将考虑在 (未知的) 算法之后应用子集构造 (带无用状态删除) 以生成最小 DFA 的可能性。我们现在构造这样的算法。（在本节中描述的算法也可用于通过用 *subset* 替换函数 *subsetopt* 来构造最小完全 DFA）。

Let $M_0 = (Q_0, V, T_0, \theta, S_0, F_0)$ be the $\epsilon - free$ FA.

This algorithm was originally given by Brzozowski in[Brzo62]. The origin of this algorithm was obscured when Jan van de Snepscheut presented the algorithm in his Ph.D thesis [vdSn85]. In this thesis, the algorithm is attributed to a private communication from Prof. Peremans of the Eindhoven University of Technology. Peremans had originally found the algorithm in an article by Mirkin [Mirk65]. Although Mirkin does cite a paper by Brzozowski[Brzo64], it is not clear whether Mirkin's work was influenced by Brzozowki's work on minimization. Jan van de Snepscheut's recent book [vdSn93] describes the algorithm, but provides neither a history nor citations (other than his thesis for) this algorithm.

该算法最初是由 Brzozowski 在 [Brzo62] 中给出。最初在 Jan van de Snepscheut 在他的博士论文 [vdSv885] 中提出该算法时是模糊的。本文中，算法的起因是一个教授的私人通讯。埃因霍芬理工大学的 Pereman, Mirkin [Mirk65] 的文章中找到了该算法。虽然 Mirkin 引用了 Brzozowki [Brzo64] 的论文，但米尔金的作品是否受 Brzozowki 的最小化工作的影响尚不清楚。Jan van de Snepscheut 的新书 [VDSn93] 描述了该算法，但既不提供该算法历史，也不提供该算法的引文（除他的论文外）。

3 状态等价最小化

本节中，我们限定在完全 DFA 的最小化中。This is strictly a notational convenience, as the miniization algorithm can be modified to work for non-complete DFA. 严格地说，这是一种标记的便利，因为可以对缩小算法进行修改，使其适用于不完全 DFA。