形式语言及其与自动机的关系

〔美〕J. E. 霍普克罗夫特
J. D. 厄尔曼 著

科学出版社

形式语言及其与自动机的关系

〔美〕J. E. 霍普克罗夫特
J. D. 厄尔曼 著

莫绍揆 段祥 顾秀芬 译

科学出版社

内 容 简 介

本书是数字电子计算机编译程序理论和语言理论方面的著作.

全书共分十四章. 前八章主要介绍 Chomsky 定义的短语结构文法、前后文有关文法、前后文无关文法和正规文法,并证明了与这四类文法等价的四类自动机: 图灵机、线性界限自动机、下推自动机和有穷自动机. 第九章讨论语言的封闭性质. 第十、十一两章说明自动机识别一个句子所需要的时间量和空间量. 第十二章研究了确定的下推自动机, 并讨论了 LR(k) 文法. 第十三章专论堆栈自动机; 这类自动机可看成下推自动机的一个推广. 最后一章对形式语言里的一个重要专题——可判定性作了讨论.

本书可供有关专业的研究生、研究人员及大专院校老师参考.

John E. Hopcroft Jeffrey D. Ullman
FORMAL LANGUAGES
AND THEIR RELATION TO AUTOMATA
Addison-Wesley Publishing Company
1969

形式语言及其与自动机的关系

〔美〕J. E. 霍普克罗夫特 J. D. 厄尔曼 著
莫绍揆 段祥 顾秀芬 译

*

科学出版社出版
北京朝阳门内大街137号
中国科学院印刷厂印刷
新华书店北京发行所发行 各地新华书店经售

*

1979年5月第一版 开本: 787×1092 1/32
1979年5月第一次印刷 印张: 10 1/4
印数: 0001—22,520 字数: 230,000
统一书号: 15031·231
本社书号: 1408·15—8

定价: 1.05 元

译　者　序

本书可看作数字电子计算机编译程序理论和语言理论的著作，着重讲形式语言和自动机的关系.

作者首先给出了确定语言的两个基本方法. 一个是文法，由它可从句子符号 S 而产生语言中的各句子，另一个是识别器，即通常所说的自动机. 作者从本质上指明了，每一个文法都有和它等价的自动机. 所谓文法与自动机的等价，是指：字 w 由一文法产生，当且仅当 w 被自动机接受. 这样，有关文法的性质和相应自动机的性质便可以一起研究，互相补充. 更重要的是，通过自动机极易得出相应文法的识别程序，而识别程序乃是数字电子计算机编译程序的核心，由它可用来完成对语言句子的语法分析. 两者之间有这么密切的关系，这便是本书所以着重讨论形式语言与自动机的关系的原因.

由本书可以看出，0 型文法和图灵机等价，1 型文法和不确定的线性界限自动机等价，2 型文法和不确定的下推自动机等价，而 3 型文法(正规集)则和有穷自动机等价. 以上这些文法和自动机，在目前是最主要的，也是大家讨论得最多的. 当然，有些文法所对应的自动机还没有专名(例如本书中提到的线性文法、顺序文法、界限语言等)，而有些自动机所对应的文法也还没有专名(例如堆栈自动机等). 这并不是缺点，正表明文法和自动机尽管本质上是一回事，但同中有异，有些问题从文法入手更易研究，有些问题从自动机入手更易研究. 我们应该针对具体情况，具体掌握应用.

本书除讨论各类文法、各类自动机及彼此间的关系外，还

讨论了语言对各种运算的封闭性、计算的复杂性(即计算时所使用的时空界限)以及可判定性，这是三种较为深入的课题，是进一步研究语言和自动机的基础．要想对语言作进一步研究，这几个课题最好能够弄通．

读者可以根据具体情况自行选读，例如，初学的人可先略去上述三个课题（第九、十、十一及十四章），别的章内有引用相应结论的(这种情况极少)亦暂时略去，等到对各类文法和自动机的性质弄懂了，再深入下去．反之，已具备形式语言初步知识的读者，对第一、二、四章可很快读过去（甚至可以不读)而不致影响对别的部分的理解．

关于译名，我们尽量利用已经流行的提法，但并不总与已有的一致，是否确切，望读者指正．

在翻译过程中，凡我们认为原文有欠妥之处，都加按语指明，个别地方则迳行校改．书中某些算法，亦可讨论，但我们照旧不改．

限于译者水平，译文中一定有不少错误，敬请读者批评指正．

本书第一到五章是顾秀芬同志翻译的，第六到九章和第十二章由段祥同志翻译，第十、十一、十三和十四章由莫绍揆同志翻译，最后由莫绍揆同志对全书进行校对．

原　　序

形式语言的研究是计算机科学的一个重要领域. 形式语言大约于 1956 年问世. 那时, Noam Chomsky 给出一种文法的数学模型, 该文法与当时他所研究的自然语言有关. 不久当程序设计语言 ALGOL 的语法由前后文无关文法定义时, 人们便发现了文法的概念对程序员是非常重要的. 这样的研究自然会导致面向语法编译, 并产生编译程序之编译程序的概念. 自那以后, 研究工作相当高涨, 其结果导致了形式语言和自动机理论之间的关系达到彼此不可分离的程度. 时至今日, 在不了解语言及自动机理论的技术和结果的情况下, 就不能对计算机科学进行严肃的研究.

本书把形式语言的理论作为一个连贯的理论, 并明确它与自动机的关系. 本书先解释语言的有穷描述这个概念, 并解释了基本描述手段——文法以及它的三个主要子类——正规文法、前后文无关文法和前后文有关文法. 对前后文无关文法进行了详细的处理, 并讨论诸如范式、派生树以及歧义性这样一些专题. 还介绍了与四类文法等价的四类自动机: 有穷自动机、下推自动机、线性界限自动机和图灵机. 对图灵机介绍得比较详细, 并证明了停机问题的不可解性. 本书的结尾介绍语言理论中一些高级的专题——封闭性质、计算的复杂性、确定的下推自动机、$LR(k)$ 文法、堆栈自动机和可判定性.

介绍的重点放在概念和理解的容易性上, 而不是过分的形式主义. 在某些情况下省略了冗长而乏味的详细证明. 本

书在所有情况下都作了充分而直观的解释，因此，如果需要的话，读者很容易作出严格的证明.

本书主要作为形式语言方面的一年或二年研究生课程教科书，其内容本身是一个完整的体系，它要求读者具有大学毕业生的水平.

我们发现，有穷状态的机器或图灵机的课程虽然不是必要的，但却是有用的预备课程. 本书并不想作为形式语言的探讨手册，但已包括了许多已知的结果，以及大量以前只在数学和计算机科学杂志上发表过的资料. 各章的后面都附有查找本章内容和有关课题的参考文献. 习题是课文的不可分割部分，其疑难程度从最简单的到极其困难的都有.

本书内容取材于作者在普林斯顿、哥伦比亚和康奈尔讲授的有关语言理论的各种讲稿. 作者要感谢对本书提出建议和批评的许多人，特别要感谢 A. V. Aho、S. Amoroso、A. Korenjak 和 M. Harrison。作者还要感谢贝尔电话实验室及普林斯顿和康奈尔大学，因为它们为本书的准备工作提供了许多方便.

J. E. H.

J. D. U.

目　　录

第一章　语言及其表示法

1.1　字母表和语言

什么是语言理论？要回答这个问题，我们首先要问，什么是语言？Webster 把语言定义为“为相当大的团体的人所懂得并使用的字以及组合这些字的方法的统一体”．然而，这个定义对于建立语言的数学理论是不够精确的．因此，我们将把形式语言抽象地定义为一个数学系统，其形式性使我们能给出形式语言的严密描述，并能发展一批知识．尔后把这些知识用在那些适当地造型的语言中．记住上述概念，我们来作下述定义．

字母表或**符号集**是符号的任何有穷集．虽然有不可数的无穷多个符号存在，但我们只考虑可数无穷子集[1)]，将从该子集中抽出所有的有穷集来．这个子集将包括数字，大写和小写的拉丁和希腊字母（可能附有下标，上标，在下面划线等），以及象 $\#$，ϕ 等特殊符号，可以加上读者认为使用方便的任何可数个附加符号．字母表的某些例子是拉丁字母表 $\{A, B, C, \cdots, Z\}$，希腊字母表 $\{\alpha, \beta, \gamma, \cdots, \omega\}$，以及二进制字母表 $\{0, 1\}$．

一个字母表上的**句子**是由字母表的符号组成的有限长度的任何行．句子的同义词为**行**和**字**．空句子 e 是不含有符号的句子．如果 V 是一个字母表，则 V^* 表示由 V 的符号所组成

1）若一个集合是与整数一一对应（即若该集合的第 i 个元素是有意义的话），则此集合就是可数无穷的．

的所有句子的集合，包括空句子．我们用 V^+ 表示集合 $V^*-\{e\}$．因此，若 $V=\{0,1\}$，则 $V^*=\{e,0,1,00,01,10,11,000,\cdots\}$ 和 $V^+=\{0,1,00,\cdots\}$．

语言是一个字母表上的句子的任何集合，大多数有趣的语言都包含有无穷多个句子．可以提出三个重要问题．

首先，我们怎样表示一种语言呢？（即说明该语言的句子？）如果语言只含有有穷多个句子，则答案很简单，人们只需列出句子的有穷集就行了．另一方面，如果语言是无穷的，我们就面临找出该语言的有穷表示的问题．这种有穷表示本身通常都是某一字母表上的一个符号行，再加上一些隐含的解释，以连系一个特殊表示和一个给定的语言．

第二，是不是每一种语言都有一个有穷表示呢？人们猜想，答案可能是否定的．我们将看到字母表的所有句子的集合都是可数无穷的．一种语言就是所有这种句子集合的任何子集．一个可数无穷集的所有子集的集合并不是可数无穷的，这是集合论的一个众所周知的事实．虽然我们还未确定有穷表示由什么组成，但我们直观地感到，一个有穷表示的任何有意义的定义都将终归只能给出可数个有穷表示，因为我们可以把任何这样的表示作为某一符号行而记下来．因此，语言的个数比有穷表示更多．

第三，我们也许要问，对于具有有穷表示的这些语言类的结构能够讲些什么．本书的大部分将致力于介绍各个表示系统和刻划这些语言类．

1.2 过程和算法

在讨论有穷表示的概念以前，我们非形式地介绍一下过程和算法的概念．一个**过程**是一个能够被机械地执行的指令

的有穷序列，例如一个计算机的程序.

我们对过程的定义多少有点含糊. 在第六章，我们再用图灵机术语进行形式定义. 目前，当我们不能确定某一个步骤是否能机械地执行时，我们就把这一步骤简化成肯定能执行的一系列较简单步骤. 例如，我们可能要反对这样一个步骤，即“找出最小整数 x，满足如此这般的一个条件”，除非找出该最小整数 x 的方法是明显的. 即使我们知道这一最小 x 一定存在，也不一定能用机械的方式找到它.

图 1.1 给出了一个判定比 1 大的整数 i 是否是素数的过程的例子. 过程的第二个例子如图 1.2 所示；该过程对整数 i 判定是否存在有比 i 大的完全数[1].

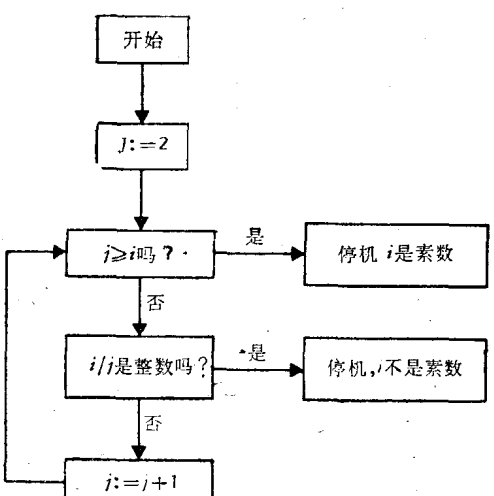

图 1.1 判定大于 1 的整数是否是素数的过程

1) 一个完全数是一个整数，它等于除它本身外的各个因子之和. 6 是一个完全数，因为 1+2+3=6. 12 不是完全数，因为它的各因子分别是 1，2，3，4 和 6，这些数之和是 16.

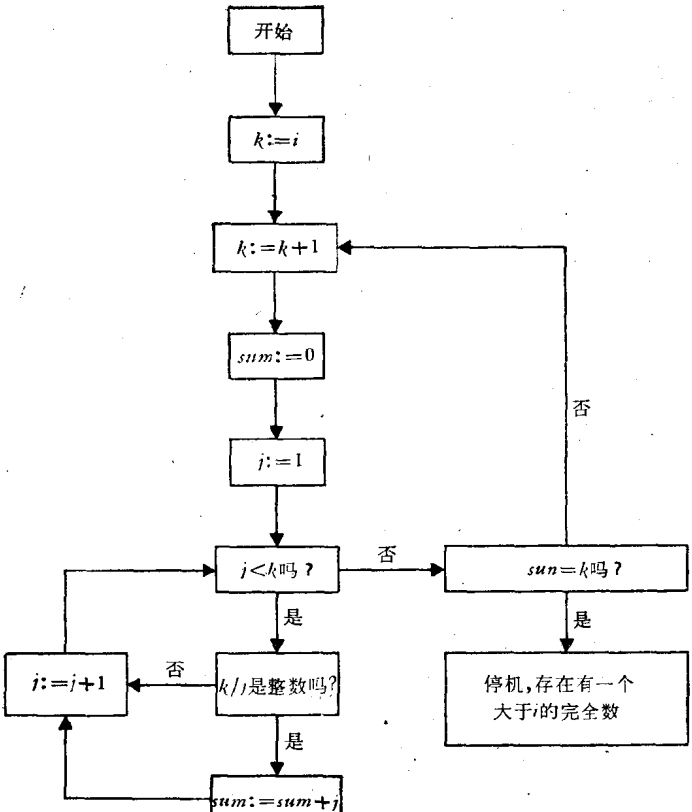

图 1.2　判定是否有一个大于 i 的完全数的过程

注意，第一个过程对 i 的所有值都会终止，因为或者将获得 j 的一个值，它整除 i，或者 j 最后将等于或大于 i. 在上述两种情况下此过程都要终止，并告诉我们 i 是否是素数. 总是要终止的过程叫做**算法**. 因此我们就把图 1.1（译者按：部分框图改用 ALGOL 语言来说明，下同.）的过程称为判定大于 1 的整数是否是素数的一种算法.

指令：

1. 置 $j = 2$.

2. 若 $j \geqq i$，则停机．i 是素数．

3. 若 i/j 为整数，则停机．i 不是素数．

第二个过程不一定会终止．如果只有有穷多个完全数[1]，则过程不会对任何比最大完全数还大的整数而停止．过程倒是继续测试愈来愈大的 k 以寻找另一个完全数．换言之，如果存在有比 i 大的一个完全数，该过程就会找到它．然而如果这个完全数不存在，则此过程将会永远运行下去．只要过程继续运行，我们就无法知道它是否最终会停止．如果上述的完全数存在，过程最终会给予肯定的答案的，在这个意义上，我们可以说该过程能判定是否存在有一个比给定的整数还大的完全数．如果这样的完全数不存在，则过程根本不予回答，因为在任何时候我们无法知道过程是否会在以后某个时刻停止．

指令：

1. 置 $k = i$.

2. 置 $k = k + 1$.

3. 置 $sum = 0$.

4. 置 $j = 1$.

5. 若 $j < k$，则转向指令 8.

6. 若 $sum \neq k$，则转向指令 2.

7. 停机，k 是一个大于 i 的完全数．

8. 若 k/j 不是整数，则转向指令 10.

9. 置 $sum = sum + j$.

10. 置 $j = j + 1$.

---

1) 完全数是否无穷，是数论的一个尚未解决的问题．

11. 转向指令 5.

1.3 语言的表示

现在我们转到对语言的有穷表示的问题. 表示语言的一种方法是给予一种算法，由它判定一个句子是否在语言中. 更一般的方法是给予一个过程，它对于语言中的句子，回答"是"而停止，而对不是语言中的句子，或者不终止，或者回答"否"而停止. 这样的过程或算法叫做对语言的**识别**. 在第七章我们将看到存在有能用过程但却不能用任何算法识别的语言.

上述那些方法是从识别的观点来表示语言的. 我们还可以按产生的观点来表示语言，即我们可以给出一个过程，它依某种次序系统地依次产生语言的句子.

如果我们能够用一种算法或一个过程来识别字母表 V 上的某种语言的句子，那么我们就能产生此语言，因为我们能够系统地产生 V^* 中的所有句子，并测试每个句子，看它是否在语言中，然后再以表格形式输出在语言中的那些句子. 在用此方法时，必须仔细. 因为如果依次序产生句子，并使用一个不一定会停止的过程来测试句子，那么我们永不会获得下述句子以后的其它句子：即它使过程总不停止. 避开上述问题的方法是用下面的方法组织测试，即永不让过程连续永远测试同一个句子. 这种方式需要我们引进几种构造法.

假设 V 中有 p 个符号，我们能够把 V^* 中的句子看成以 p 为底的数字加上空句子 e. 我们能够按长度递增的次序给句子编号，对于同样长度的句子，按数值的次序编号. 在图 1.3 中我们枚举了 $\{a, b, c\}^*$ 的各句子. 我们暗中假设 a，b 和 c 分别与 0，1 和 2 相对应. （这个论证表明，如我们所声称那