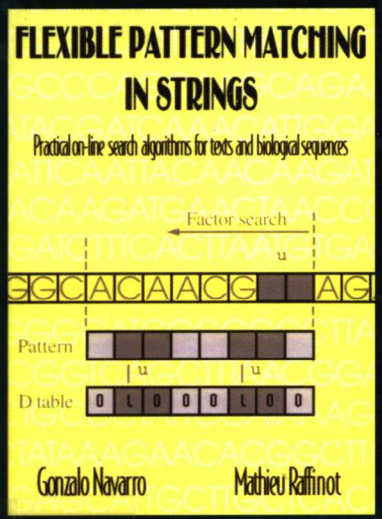
国外计算机科学教材系列

柔性字符串匹配

Flexible Pattern Matching in Strings

Practical On-Line Search Algorithms for Texts and Biological Sequences



[美] Gonzalo Navarro
Mathieu Raffinot 著

中科院计算所网络信息安全研究组 译

電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

国外计算机科学教材系列

柔性字符串匹配

Flexible Pattern Matching in Strings

Practical On-Line Search Algorithms
for Texts and Biological Sequences

[美] Gonzalo Navarro
Mathieu Raffinot 著

中科院计算所网络信息安全研究组 译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书是一本不可多得的字符串匹配方面的专业书籍。书中对串匹配问题进行了系统化的分类，从实际效果出发，着重详细介绍了串匹配领域内效果最好的若干种算法。并且给出了具有统一接口的算法伪码，使读者能清晰理解算法原理，易于实现算法编程，从而提高专业水平。此外，书中通过严谨的理论分析和大量实验数据，说明了每种算法在实际应用中的适用范围，由此提供了良好的应用指导，解决了串匹配算法的最佳适用性问题。

本书可帮助本领域的研究人员从整体上把握字符串匹配方面的脉络，而其他相关领域的人员也可借助本书非常清晰地了解串匹配问题的概况。

版权贸易合同登记号　图字：01-2007-1057

图书在版编目（CIP）数据

柔性字符串匹配/（美）纳瓦罗（Navarro, G.）等著；中科院计算所网络信息安全研究组译.
北京：电子工业出版社，2007.3
（国外计算机科学教材系列）
书名原文：Flexible Pattern Matching in Strings: Practical On-Line Search Algorithms for Texts and Biological Sequences
ISBN 978-7-121-03858-7
Ⅰ.柔… Ⅱ.①纳… ②中… Ⅲ.电子计算机－算法理论－教材 Ⅳ.TP301.6
中国版本图书馆CIP数据核字（2007）第017507号

责任编辑：史　平
印　　刷：北京市天竺颖华印刷厂
装　　订：三河市金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编：100036
开　　本：787 × 980　1/16　　印张：13.75　字数：275千字
印　　次：2007年3月第1次印刷
定　　价：38.00元

凡所购买电子工业出版社的图书有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系电话：（010）68279077。邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

译 者 序

初次看到《柔性字符串匹配》这本书是在 2003 年。那时，我们小组正忙于进行一个项目，需要一种高效的字符串匹配算法。虽然当时我们已经拥有几年的研究工作基础，钻研了不少经典算法，自己也开发了几种新算法，但为获得最佳性能，我们还是忙碌了几个月才基本达到目标。这时我们意识到：虽然串匹配问题是一个非常经典的问题，但当前却面临着非常大的挑战。尤其是在信息检索、信息过滤和计算生物学等领域，需求日新月异，对实时性的要求也越来越高，经典算法已经远远不能满足需求。这促使我们对于串匹配问题进行更深入的研究。就在此时，由 Gonzalo Navarro 和 Mathieu Raffinot 合著、剑桥大学出版社出版的本书进入了我们的视野，着实给了我们一种惊喜的感觉。惊讶的是，原来在串匹配领域中，已经有人不仅在理论上而且在实践上都对问题进行了如此深入的理解和阐述；喜悦的是，这本书对于我们不只是有用处，还让我们有种茅塞顿开的感觉！

这本书的学术价值在于对串匹配问题进行了系统化的分类，使本领域的研究人员可以从整体上把握本专业的脉搏，并使相关领域人员也可以非常清晰地理解串匹配问题的概况。本书的内容分类合理，章节紧凑，问题阐述由浅入深，是串匹配方面一本不可多得的专业书籍。更重要的是，作者没有片面求全，不是一一列举串匹配领域的全部算法和技术，而是从实际效果出发，着重详细介绍效果最好的若干种算法。叙述时条理清晰，内容详尽，并给出了具有统一接口的算法伪码，使读者能清晰地理解算法原理，方便地实现算法编程，从而提高在本领域的专业水平（包括理论和实践）。这本书不仅可以在学术方面对我们有所帮助，而且在应用方面的作用也是目前其他图书所不能比拟的。作者通过严谨的理论分析和大量实验数据，说明了每一种算法在实际应用中的适用范围，为非本领域的众多其他人员提供了很好的应用指导，解决了一直困扰大家的难题：怎样选择一个最适合我这个应用（问题）的串匹配技术（或算法）？

在我们决定将本书翻译出版后，2005 年底，在 SPIRE 2005 会议上，译者和本书的作者之一 Gonzalo Navarro 先生见了面。Navarro 先生很高兴我们将他的书翻译为中文，并欣然表示要为中文版写序。Navarro 先生是字符串匹配领域的一位非常活跃的专家，

在精确匹配、近似匹配、压缩匹配、音乐检索、计算生物学等方面都有论著。他还积极组织串匹配领域的各种会议，推动了南美地区本技术的发展。我们翻译本书的目的也是希望让国内更多的人士关注字符串匹配技术，希望有朝一日在国内也能有一个串匹配技术爱好者共同讨论和交流的平台，并推动相关的研究工作更上一个新的层面。

本书是由中科院计算所信息智能与信息安全中心长期从事字符串匹配研究工作的小组人员共同翻译完成的。翻译工作得到了中心领导程学旗主任和郭莉主任的支持，白硕研究员给了我们很多翻译方面的指导，研究组组长谭建龙博士也给予了很多帮助。翻译工作还得到了中心其他同仁的帮助，他们是戴磊、李真真、王小磊、杨毅夫、赵咏，在此也表示衷心的感谢。

在本书的翻译工作中，王映负责第 1 章和第 6 章的翻译，刘燕兵负责第 2 章和第 3 章的翻译，曹京负责第 4 章的翻译，刘萍负责第 5 章和第 7 章的翻译，最后的统稿由刘萍完成。

刘萍

2007 年 1 月于中科院计算所

中译本序言

在阿根廷的布宜诺斯艾利斯举行 SPIRE 2005 会议期间，来自世界另一端的两位研究者找到了我，他们是中国科学院计算所的刘萍和谭建龙。他们提议将 Mathieu Raffinot 与我合著的一本书翻译为中文，我欣然采纳了这一建议，并答应提供尽可能多的帮助。随后，Mathieu 和本书的出版社（剑桥大学出版社）也很快同意了此事。然而遗憾的是，当时我正参与主持当年的 SPIRE 会议，忙得不可开交，无暇分身和他们做太多的交流。接下来的几个月里，我对他们的进展也知之甚少。

此书中译本的重要性是不言而喻的，它为克服语言障碍提供了便利，使得更多的学生和感兴趣的研究者有机会了解我们的专著以及字符串匹配领域。对于我来说，特地选择出版此书的中译本还有更深层的涵义。在有些语言中，信息检索中的一些传统假设往往不再成立了，而中文就是这类语言的一个绝佳范例。与西方语言不同，中文字符串不能明显地分割成一些有意义并符合实际概率分布的“单词”，因此还需要一个预处理过程来进行分词，而字符串匹配技术可能派上用武之地。此外，在中文等东方语言上直接使用字符串匹配也是一种潜在的可能途径。因此，这是我们出版中译本的另一个重要初衷。

基础性研究的美妙之处在于它可以应用到极其广泛的领域，往往超出研究者的想象。研究的问题越基础，其应用范围也越广，而且更有意思的是，研究者也更难确切地说出这个东西到底有什么用。字符串匹配成为中文信息处理中的一种不可或缺的基本技术，也说明了这一点。

仅仅一年之后，我欣喜地发现此书的中译本在他们的努力下诞生了。当我浏览书中那一长串漂亮的中文字符时（尽管我并不理解它们的意思），我觉得真正体会到了字符串匹配的精华实质。很显然，在这样陌生的文本中找出某个字符串，并不像在我们熟悉的文本上那么容易。当我们处理熟悉的文本时，由于大脑受过这种语言的训练，整个过程的复杂性已经不知不觉地被掩盖了。但是当搜索陌生文本时，你不得不借助于最原始的手工计算，这时你才能真正地感受到这个问题的挑战之处，同时也体会到钻研它的乐趣所在。

我希望中国的读者除了从这本书中获取对他们的学习、研究和开发有所裨益的知识之外，也能从中感受到算法（尤其是字符串匹配算法）的快乐和美妙之处，这种快乐和美妙也一直伴随在我的左右。

Gonzalo Navarro
智利圣地亚哥 2007 年 1 月

目　录

第1章 导 言

1.1 本书的目的和侧重点

字符串匹配问题可以理解为从给定的符号序列中找出一个具有某种属性的模式，最简单的例子是从给定的字符序列中找出一个给定的字符串。

字符串匹配是计算机科学中最古老、研究最广泛的问题之一，并且，字符串匹配的应用也随处可见。近年来，学术界对字符串匹配的研究兴趣与日俱增，特别是在发展迅猛的信息检索领域和计算生物学领域。

之所以有上述现象，不仅因为在这两个研究领域中需要处理的文本规模越来越大，而且由于需要在文本中进行越来越复杂的搜索。在这两个领域中，令人感兴趣的模式不仅是简单的字符串，还包括通配符、空位以及正则表达式。“匹配”的定义也允许模式和文本字符串之间存在某些细微的差异，即所谓的“近似匹配”，这在文本检索领域和计算生物学领域中得到广泛应用。

字符串匹配问题可以从多个不同的角度来阐述。通常来说，既可以从非常抽象的理论研究方面来介绍，也可依据极其实用的应用成果来介绍。虽然理论研究的成果极大程度提高了某些重要算法的性能，但很少能在实际应用中取得良好效果。在字符串匹配研究领域中，一个人所共知的事实是“算法的思想越简单，实际应用的效果越好”。有两个典型的例子可以说明这一点。一是著名的 Knuth-Morris-Pratt（KMP）算法，它在实际应用中比简单的蛮力方法还要慢一倍。二是在著名的 Boyer-Moore（BM）系列算法中，应用最成功的算法是对原始算法进行高度简化后得到的算法。

然而，很难在已有的相关书籍中找到那些思想简单的算法。在关于文本处理的现有书籍中，字符串匹配部分往往只包括了一些经典的理论算法。造成这一现象的主要原因有以下三个。

首先，非常实用的算法是近几年才出现的，而经典算法往往已经存在了几十年，因此很难在现有书籍中找到这些新的成果。新的算法通常采用了一些新的技术，例如位并行，这是随着新一代计算机的产生而出现的。

第二个原因是理论研究和实际应用脱节。那些专门从事算法研究的学者关心的只是理论上看起来很美妙的算法——具有很好的时间复杂度，涉及到复杂的算法概念。而开发人员只追求实际应用中尽可能快的算法。这两者之间从不注意对方在干些什么。将理论研究和实际应用相结合的算法（例如 BNDM 算法）只是近年才出现。现在，字符串匹配算法的发展趋势是更快、更健壮，然而这些新近出现的算法在已有的书籍中还找不到。

最后一个原因是，在现有的相关书籍中，很少涉及字符串领域中那些最新、最活跃的相关主题，包含对扩展模式串的搜索、多模式串匹配，以及近似模式匹配等。

由于上述原因，在实际应用中常常很难找到适合需求的算法——这样的算法实际上是存在的，但是只有资深专家才比较了解。考虑如下情况，一位软件开发人员，或者一位计算生物学家，或者一位研究人员，又或者一位学生，对字符串匹配领域并没有深入了解，可是现在需要处理一个文本搜索问题。于是，不得不去钻研那些汗牛充栋的典籍。这些典籍往往极具理论价值，可实现起来常常太过复杂。使得阅读者淹没在各种匹配算法的海洋中，但却没有足够的背景知识选择最适用的算法。最后，常常会导致这样的局面：选择一种最简单的算法加以实现。这往往可能导致很差的性能，从而影响整个开发系统的质量。更糟糕的是，选择了一个理论上看起来很漂亮的算法，并且花费了大量精力去实现。结果，却发现实际效果和一个简单算法差不多，甚至还不如简单算法。

本书的目的就是介绍在各种模式（包括字符串、字符串集合、扩展字符串以及正则表达式等）下现存的精确匹配算法和近似匹配算法，并深入介绍几个最实用的算法。所谓“实用”，就是指算法在实际应用中性能较好，并且一个普通程序员能在几小时内完成算法的实现代码。幸好，在字符串匹配领域中，这两个准则还是一致的。注意：本书只关注在线（on-line）搜索，即不对文本建立索引数据结构。尽管索引搜索也基于在线搜索技术，但它是个非常复杂的问题，其本身就值得单独出本书加以介绍。

本书面向各种不同的读者：计算机科学家将会通过本书了解那些最快的搜索算法并能加以实现。如果他们想更进一步研究文本搜索算法，本书中提供了相关问题的详细链接和文献索引（包括书籍、会议录和论文）。计算生物学家能够通过本书对模式匹配有更深入的了解，并找到适合他们需求的最简单、最有效的序列搜索算法。

本书的作者实现了书中所有的算法，并进行了实验，其中某些算法是作者自己提出的。此外，书中尽可能给出一些实验图，以帮助读者在最短的时间内找到最适合特定应用的算法。

当然，文本搜索领域是如此之大，一本书根本不能囊括其中所有的内容。本书更倾向于将不同的主题分别作为独立的章节加以详细阐述。在 7.2 节中给出了最近出版的相关书籍的列表。

1.2 概况

第2章 字符串匹配

一个字符串是一个定义在有限字母表 Σ 上的字符序列。例如，ATCTAGAGA 是字母表 $\Sigma = \{A, C, G, T\}$上的一个字符串。字符串匹配问题就是在一个大的字符串 T 中搜索某个字符串 p的所有出现位置。其中，T称为文本，p称为模式串，T和p都定义在同一个字母表 Σ 上。给定字符串x, y 和z，称 x 是xy的一个前缀，x 是yx的一个后缀，x 是yxz的一个因子。

根据在文本中搜索模式串方式的不同，字符串匹配算法可以归结为三种基本的方法。

第一种方法是从文本中逐个读入字符，每读入一个字符就更新相应变量，检查是否存在一个可能的匹配。Knuth-Morris-Pratt 算法就属于这种方法，另外一种是更快的 Shift-Or 算法，对其进行扩展后可以支持对更复杂模式串的搜索。

第二种方法基于滑动窗口。滑动窗口沿着文本T滑动，对于任意位置上的窗口，在窗口中从后向前搜索窗口中的文本和模式串 p的公共后缀。Boyer-Moore 算法就使用了这种方法。但在一般情况下，Boyer-Moore 算法比它的一个简化版本 Horspool 要慢。实际上，Boyer-Moore 系列算法中除了 Horspool，都要比使用其他方法的一些算法慢。

第三种方法出现得较晚，实际应用中如果模式串 p足够长，其中一些算法是最有效率的。和第二种方法一样，第三种方法也使用了滑动窗口，并在其中从后向前搜索。不同的是，它搜索的是窗口中文本的最长后缀，并且这个最长后缀同时也是模式串 p的一个因子。最早使用这种方法的算法是 BDM 算法，当 p足够短时，可以将其改造成更简单、更有效的算法 BNDM。对于较长的模式串，一个称为 BOM 的新算法是最快的。

本章中还给出了一幅实验图。对于给定的模式串长度和字母表大小，通过这幅实验图能很容易找到速度最快的算法。

上面的三种方法给出了一个对字符串匹配算法进行分类的框架。目前，最有效的匹配算法都属于其中的某一类。另外，也存在其他一些算法，例如基于散列的算法，但通常它们的性能不是很高。关于这些算法的参考文献，在本章的最后一节给出。

第3章 多字符串匹配

多字符串匹配算法能够在只扫描一遍文本的情况下，搜索出模式串集合 $P = \{p^1, p^2,$

$\ldots, p^r\}$中所有模式串的所有出现位置。许多单字符串搜索算法都能够扩展成多字符串搜索算法，当然这些扩展算法的性能有高有低。本章概述了那些效率最高的算法，让人困惑和恼火的是，在这些算法中，很多只是以技术报告的形式给出，如果你不是一位字符串匹配方面的专家，要找到这些算法是比较困难的。

上述三种单字符串搜索方法都能扩展到多字符串的情况。基于第一种方法的算法包括著名的 Aho-Corasick 算法。如果用$|P|$表示所有模式串的长度之和，那么，当$|P|$很小的时候，可以使用 Multiple Shift-And 算法。

基于第二种方法的算法有著名的 Commentz-Walter 算法，但其实际运行的效率不是很高。Horspool 算法的扩展是 Set Horspool 算法，当搜索一个很大字母表上的一个较小的模式串集合时，其效率很高。Wu-Manber 算法结合了后缀搜索算法和散列方法，在实际应用中效率很高。

基于第三种方法的算法中，由 BOM 算法派生出了 SBOM 算法。当模式串集合中的最小模式串长度较大时，SBOM 算法的效率很高。与 Shift-Or 算法相似，当$|P|$很小时，BNDM 派生出了 Multiple BNDM 算法。

本章也给出了一幅实验图，指明了在模式串长度之和$|P|$、模式串的最短长度以及字母表大小等因素变化的情况下，怎样选择一个合适的算法。

第 4 章　扩展字符串匹配

在许多应用中，模式串并不是简单的字符序列。本章中主要讨论的内容是：在实际应用中使用的几种对模式串的扩展，以及怎样对其进行搜索匹配。所有这些扩展都能够转换为正则表达式（参见第 5 章），但是对于某些特定情况，存在更简单、更快速的算法。

最简单的扩展是，模式串中的每个位置不是一个字符而是一个字符集合。也就是说，只要文本中的对应字符属于该位置的字符集合，即为在该位置匹配成功。同样，这种字符集合也可以出现在文本中。

第二种扩展是限长空位，即可以指定模式串中的某些位置，这些位置能够与文本中的长度介于指定的最大值和最小值之间的任意字符串匹配。这种扩展对于计算生物学中的某些应用特别重要，例如搜索 PROSITE 模式。

第三种扩展是允许可选字符和可重复字符。一个可选字符表示它在文本对应的位置上可以出现也可以不出现，而可重复字符在匹配的文本中可以出现 1 次或多次。

对上面这三种扩展模式串及其组合的模式串的搜索问题都可以通过修改后的 Shift-Or 算法和 BNDM 算法来解决。这两种算法都是使用位并行技术来模拟一个非确定自动机在文本

中进行搜索，以找到所有的成功匹配（参见 1.3 节）。构造这样一个自动机比较复杂，核心问题是怎样进行自动机的模拟。扩展后的 Shift-Or 算法不能跳过文本中的某些字符，但其效率不受模式串复杂度的影响。扩展后的 BNDM 算法通常要更快一些，但其效率受制于成功匹配的最小长度、字母表的大小、字符类的大小以及空位的大小等因素。相对而言，其他经典算法都不能容易地扩展并达到如此好的效率。

本章的最后将给出对一个由较短模式串组成的规模较小的模式串集合，可以通过与上面相似的方法进行搜索匹配。此外，本章中还给出了其他一些理论算法的参考文献，它们是针对某些特定扩展模式串的搜索。

第 5 章 正则表达式匹配

正则表达式是一个表示模式串集合的有力工具，前面讨论的各种模式串都能使用正则表达式来表示。正则表达式的定义是递归的：一个正则表达式要么是一个简单的字符串，要么是子正则表达式的连接、联合或重复。搜索正则表达式的算法相当复杂，只有在模式串不能用更简单的方法来表示的时候才会使用正则表达式来表示。

通常，在文本中搜索一个正则表达式分为如下几个步骤。首先，解析正则表达式，把它表示成容易实现的等价树形结构。在第 5 章的末尾将详细讲述这个问题，之后的步骤就围绕这个树形的表示来进行。

第二步是由模式串建立一个非确定的有限自动机（NFA, nondeterministic finite automaton）。NFA 本质上是一个状态机，当读入文本字符时，活动状态将被改变。状态机的当前状态为一个终止状态时，就表示识别出了一个模式串。从正则表达式建立 NFA 有两种办法：Thompson 算法和 Glushkov 算法。由 Thompson 算法建立的 NFA 的状态转移数目和正则表达式的长度成正比，这个性质对于某些正则表达式是非常有用的。Glushkov 算法产生的 NFA 具有最少的状态数目，并具有其他一些有用的性质。

NFA 构建完成之后就可以直接进行搜索（这种算法称为 NFA-Thompson 算法），但这样做速度会很慢，因为可能会同时存在多个活动状态。也可以考虑把 NFA 转换成确定的有限自动机（DFA, deterministic finite automaton），它在任何时刻都只有一个活动状态。DFA 很适合于文本搜索，也是正则表达式搜索中所使用的最经典算法之一，称为 DFA Classical 算法。该算法的主要问题是，转换后 DFA 的大小可能是原来的 NFA 大小的指数级，因此，只是在模式串长度较短的时候才适用。对于较长的模式串，通常使用多个较小的 DFA 构造一个大的 NFA，这种结合使用 NFA 和 DFA 的方法具有较好的效率，称为 DFA Modules 算法。

另一种办法是用位并行技术来模拟 NFA，而不是将 NFA 转换为 DFA。本章中将讨论两个比较新的算法：BPThompson 算法和 BPGlushkov 算法，二者都使用其特定的性质来模拟相应的 NFA。一般情况下，BPGlushkov 算法的效率要优于 BPThompson 算法。

第三种方法比较新颖，允许跳过文本中的某些字符。MultiStringRE 算法能够计算正则表达式的成功匹配的最小长度 *lmin*，并计算出所有可能的成功匹配的前缀（长度为 *lmin*），然后利用多模式串匹配算法（参见第 3 章）在文本中搜索这些前缀。如果找到一个这样的前缀，算法再试图将前缀扩展成完整的匹配结果。从 MultiStringRE 算法可衍生出 MultiFactRE 算法，它选择所有可能的成功匹配的因子（而不是前缀），这些因子的长度也是 *lmin*。其中，某些因子必须在所有的成功匹配中出现。本章的最后将介绍 RegularBNDM 算法，它通过模拟 Glushkov NFA 来扩展 BNDM 算法。

对于正则表达式的搜索，选择一个最好的匹配算法是非常复杂的，这是因为算法的效率通常依赖于正则表达式本身的结构。本章中试图给出一些准则，可以根据正则表达式本身的性质来判断哪个算法更合适。

第 6 章　近似匹配

所谓近似匹配，就是在文本中搜索某个模式串的所有出现位置。其中，模式串和对应的文本之间可以有稍许不同。近年来，近似匹配的应用越来越广泛。例如，在信息检索中对拼写错误的纠正，在计算生物学中的序列比对，在信号处理中对传输错误的纠正，等等。

近似匹配基于距离模型，该模型中需要一个可以度量两个字符串相似度的距离函数。在近似匹配中，通常给定模式串和一个对应的阈值 k，其中，k 指定了模式串和匹配结果之间允许的最大距离。本章中主要讨论编辑距离，即将两个字符串转变成完全相同的串所需要的最小编辑操作数目（这里的编辑操作指的是插入、删除和替换）。在实际应用中，很多问题使用了编辑距离模型或者使用其变体。

本章中将已有的近似匹配算法分为四类：

第一类算法基于动态规划方法，这是最古老的算法，同时也最容易扩展到非编辑距离的模型，但不是效率最高的算法。

第二类算法基于 NFA 搜索。该算法使用给定的模式串和阈值 k 构造 NFA，然后将其确定化。当模式串较短时，这类算法的效率不错，但不如一些更新的算法。

第三类算法基于位并行技术，这是当前应用最成功的一类算法。其中，BPR 算法和 BPD 算法用位并行技术模拟 NFA，BPM 算法用位并行技术模拟动态规划算法。BPM 和 BPD 是这类算法中最有效的，不过 BPR 算法更灵活，容易扩展到更复杂的模式串形式。