Flexible Pattern Matching in Strings

String matching problems range from the relatively simple task of searching a single text for a string of characters to searching a database for approximate occurrences of a complex pattern. Recent years have witnessed a dramatic increase of interest in sophisticated string matching problems, especially in information retrieval and computational biology.

This book presents a practical approach to string matching problems, focusing on the algorithms and implementations that perform best in practice. It covers searching for simple, multiple, and extended strings, as well as regular expressions, exactly and approximately. It includes all of the most significant new developments in complex pattern searching.

The clear explanations, step-by-step examples, algorithms pseudo-code, and implementation efficiency maps will enable researchers, professionals, and students in bioinformatics, computer science, and software engineering to choose the most appropriate algorithms for their applications.

Gonzalo Navarro obtained his Ph.D. in computer science at the University of Chile in 1998 and was appointed Assistant Professor in 1999. His interests include design and analysis of algorithms, information retrieval, text searching, text compression, approximate text searching, and searching in metric spaces. He has co-authored more than 80 papers on these topics, and has been a program committee member of several international conferences, as well as program committee chair of SPIRE'2001. He is a member of the ACM and the Chilean Computer Science Society.

Mathieu Raffinot received his Ph.D. in theoretical computer science at the University of Marne-la-Vallée in 1999. Since October 2000 he has worked as a CNRS bioinformatics researcher at the Laboratoire Génome et Informatique. His interests include design and analysis of algorithms, pattern matching, and computational biology. He is the co-author of numerous articles in international conferences and journals in computer science and bioinformatics, and he works as a consultant for bioinformatics companies, including Gene-IT, the provider of the software application LASSAP.

Flexible Pattern Matching in Strings

Practical On-Line Search Algorithms for Texts and Biological Sequences

GONZALO NAVARRO
University of Chile

MATHIEU RAFFINOT
Centre Nationale de Recherche Scientifique, Marne-La-Vallee, France

CAMBRIDGE UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011-4211, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

First published 2002

Printed in the United Kingdom at the University Press, Cambridge

Typeface Computer Modern 11/14 pt. *System* LaTeX [AU]

A catalog record for this book is available from the British Library.

Library of Congress Cataloging in Publication Data

Navarro, Gonzalo, 1969–
Flexible pattern matching in strings : practical on-line search algorithms for texts and biological sequences / Gonzalo Navarro, Mathieu Raffinot.
p. cm
Includes bibliographical references and index.
ISBN 0-521-81307-7
1. Computer algorithms. 2. Database searching. I. Raffinot, Mathieu, 1973– II. Title.

QA76.9 .A43 N38 2002
005.74–dc21 2001043704

ISBN 0 521 81307 7 hardback

A Betina, Martina, mis padres y hermana,
quienes, cada uno a su tiempo y manera,
me han hecho feliz.
A París, por lo mismo.

A toute ma famille, à Pía Marcela del Campo Rojas, à Matthieu Latapy, aux oursins, et bien sûr, au Pisco Sour.

Contents

1	**Introduction**	*page* **1**
1.1	Why this book? Our aim and focus	1
1.2	Overview	3
1.3	Basic concepts	8
1.3.1	Bit-parallelism and bit operations	8
1.3.2	Labeled rooted tree, trie	9
1.3.3	Automata	11
1.3.4	Complexity notations	12
2	**String matching**	**15**
2.1	Basic concepts	15
2.2	Prefix based approach	17
2.2.1	Knuth-Morris-Pratt idea	18
2.2.2	Shift-And/Shift-Or algorithm	19
2.3	Suffix based approach	22
2.3.1	Boyer-Moore idea	22
2.3.2	Horspool algorithm	25
2.4	Factor based approach	27
2.4.1	Backward Dawg Matching idea	28
2.4.2	Backward Nondeterministic Dawg Matching algorithm	29
2.4.3	Backward Oracle Matching algorithm	34
2.5	Experimental map	38
2.6	Other algorithms and references	39
3	**Multiple string matching**	**41**
3.1	Basic concepts	41
3.2	Prefix based approach	45
3.2.1	Multiple Shift-And algorithm	45
3.2.2	Basic Aho-Corasick algorithm	49

3.2.3 Advanced Aho-Corasick algorithm 54
3.3 Suffix based approach 54
3.3.1 Commentz-Walter idea 55
3.3.2 Set Horspool algorithm 56
3.3.3 Wu-Manber algorithm 59
3.4 Factor based approach 62
3.4.1 Multiple BNDM algorithm 63
3.4.2 Set Backward Dawg Matching idea 68
3.4.3 Set Backward Oracle Matching algorithm 69
3.5 Experimental maps 74
3.6 Other algorithms and references 74

4 Extended string matching 77
4.1 Basic concepts 77
4.2 Classes of characters 78
4.2.1 Classes in the pattern 78
4.2.2 Classes in the text 80
4.3 Bounded length gaps 81
4.3.1 Extending Shift-And 82
4.3.2 Extending BNDM 84
4.4 Optional characters 87
4.5 Wild cards and repeatable characters 89
4.5.1 Extended Shift-And 91
4.5.2 Extended BNDM 93
4.6 Multipattern searching 96
4.7 Other algorithms and references 97

5 Regular expression matching 99
5.1 Basic concepts 99
5.2 Building an NFA 102
5.2.1 Thompson automaton 102
5.2.2 Glushkov automaton 105
5.3 Classical approaches to regular expression searching 111
5.3.1 Thompson's NFA simulation 111
5.3.2 Using a deterministic automaton 111
5.3.3 A hybrid approach 115
5.4 Bit-parallel algorithms 117
5.4.1 Bit-parallel Thompson 118
5.4.2 Bit-parallel Glushkov 122
5.5 Filtration approaches 125
5.5.1 Multistring matching approach 126

5.5.2 Gnu's heuristic based on necessary factors 130
5.5.3 An approach based on BNDM 131
5.6 Experimental map 137
5.7 Other algorithms and references 139
5.8 Building a parse tree 139

6 Approximate matching 145
6.1 Basic concepts 145
6.2 Dynamic programming algorithms 146
6.2.1 Computing edit distance 146
6.2.2 Text searching 147
6.2.3 Improving the average case 148
6.2.4 Other algorithms based on dynamic programming 150
6.3 Algorithms based on automata 150
6.4 Bit-parallel algorithms 152
6.4.1 Parallelizing the NFA 152
6.4.2 Parallelizing the DP matrix 158
6.5 Algorithms for fast filtering the text 162
6.5.1 Partitioning into $k+1$ pieces 163
6.5.2 Approximate BNDM 166
6.5.3 Other filtration algorithms 170
6.6 Multipattern approximate searching 171
6.6.1 A hashing based algorithm for one error 171
6.6.2 Partitioning into $k+1$ pieces 173
6.6.3 Superimposed automata 174
6.7 Searching for extended strings and regular expressions 175
6.7.1 A dynamic programming based approach 176
6.7.2 A Four-Russians approach 178
6.7.3 A bit-parallel approach 180
6.8 Experimental map 181
6.9 Other algorithms and references 183

7 Conclusion 185
7.1 Available software 185
7.1.1 Gnu Grep 185
7.1.2 Wu and Manber's Agrep 186
7.1.3 Navarro's Nrgrep 187
7.1.4 Mehldau and Myers' Anrep 188
7.1.5 Other resources for computational biology 189
7.2 Other books 190
7.2.1 Books on string matching 190

7.2.2 Books on computational biology 192
7.3 Other resources 193
7.3.1 Journals 193
7.3.2 Conferences 193
7.3.3 On-line resources 194
7.4 Related topics 194
7.4.1 Indexing 195
7.4.2 Searching compressed text 196
7.4.3 Repeats and repetitions 199
7.4.4 Pattern matching in two and more dimensions 200
7.4.5 Tree pattern matching 202
7.4.6 Sequence comparison 203
7.4.7 Meaningful string occurrences 205
Bibliography **207**
Index **219**