Algorithms on Strings

This book is intended for lectures on string processing and pattern matching in master’s courses of computer science and software engineering curricula. The details of algorithms are given with correctness proofs and complexity analysis, which make them ready to implement. Algorithms are described in a C-like language.

This book is also a reference for students in computational linguistics or computational biology. It presents examples of questions related to the automatic processing of natural language, to the analysis of molecular sequences, and to the management of textual databases.

Professor MAXIME CROCHEMORE received his PhD in 1978 and his Doctorat d’état in 1983 from the University of Rouen. He was involved in the creation of the University of Marne-la-Vallée, where he is currently a professor. He also created the Computer Science Research Laboratory of this university in 1991. Professor Crochemore has been a senior research fellow at King’s College London since 2002.

CHRISTOPHE HANCART received his PhD in Computer Science from the University Paris 7 in 1993. He is now an assistant professor in the Department of Computer Science at the University of Rouen.

THIERRY LECROQ received his PhD in Computer Science from the University of Orléans in 1992. He is now a professor in the Department of Computer Science at the University of Rouen.
Algorithms on Strings

MAXIME CROCHEMORE
Université de Marne-la-Vallée

CHRISTOPHE HANCART
Université de Rouen

THIERRY LECROQ
Université de Rouen
## Contents

<table>
<thead>
<tr>
<th>Preface</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Tools</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Strings and automata</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Some combinatorics</td>
<td>8</td>
</tr>
<tr>
<td>1.3 Algorithms and complexity</td>
<td>18</td>
</tr>
<tr>
<td>1.4 Implementation of automata</td>
<td>23</td>
</tr>
<tr>
<td>1.5 Basic pattern matching techniques</td>
<td>28</td>
</tr>
<tr>
<td>1.6 Borders and prefixes tables</td>
<td>40</td>
</tr>
<tr>
<td>2 Pattern matching automata</td>
<td>55</td>
</tr>
<tr>
<td>2.1 Trie of a dictionary</td>
<td>56</td>
</tr>
<tr>
<td>2.2 Searching for several strings</td>
<td>57</td>
</tr>
<tr>
<td>2.3 Implementation with failure function</td>
<td>65</td>
</tr>
<tr>
<td>2.4 Implementation with successor by default</td>
<td>72</td>
</tr>
<tr>
<td>2.5 Locating one string</td>
<td>82</td>
</tr>
<tr>
<td>2.6 Locating one string and failure function</td>
<td>85</td>
</tr>
<tr>
<td>2.7 Locating one string and successor by default</td>
<td>92</td>
</tr>
<tr>
<td>3 String searching with a sliding window</td>
<td>102</td>
</tr>
<tr>
<td>3.1 Searching without memory</td>
<td>103</td>
</tr>
<tr>
<td>3.2 Searching time</td>
<td>108</td>
</tr>
<tr>
<td>3.3 Computing the good suffix table</td>
<td>113</td>
</tr>
<tr>
<td>3.4 Automaton of the best factor</td>
<td>118</td>
</tr>
<tr>
<td>3.5 Searching with one memory</td>
<td>121</td>
</tr>
<tr>
<td>3.6 Searching with several memories</td>
<td>127</td>
</tr>
<tr>
<td>3.7 Dictionary searching</td>
<td>136</td>
</tr>
<tr>
<td>4 Suffix arrays</td>
<td>146</td>
</tr>
<tr>
<td>4.1 Searching a list of strings</td>
<td>147</td>
</tr>
<tr>
<td>4.2 Searching with the longest common prefixes</td>
<td>150</td>
</tr>
</tbody>
</table>
Contents

4.3 Preprocessing the list 155  
4.4 Sorting suffixes 158  
4.5 Sorting suffixes on bounded integer alphabets 164  
4.6 Common prefixes of the suffixes 169  

5 Structures for indexes 177  
5.1 Suffix trie 178  
5.2 Suffix tree 184  
5.3 Contexts of factors 193  
5.4 Suffix automaton 199  
5.5 Compact suffix automaton 210  

6 Indexes 219  
6.1 Implementing an index 219  
6.2 Basic operations 222  
6.3 Transducer of positions 227  
6.4 Repetitions 230  
6.5 Forbidden strings 231  
6.6 Search machine 234  
6.7 Searching for conjugates 239  

7 Alignments 243  
7.1 Comparison of strings 244  
7.2 Optimal alignment 251  
7.3 Longest common subsequence 262  
7.4 Alignment with gaps 273  
7.5 Local alignment 276  
7.6 Heuristic for local alignment 279  

8 Approximate patterns 287  
8.1 Approximate pattern matching with jokers 288  
8.2 Approximate pattern matching with differences 293  
8.3 Approximate pattern matching with mismatches 304  
8.4 Approximate matching for short patterns 314  
8.5 Heuristic for approximate pattern matching with differences 324  

9 Local periods 332  
9.1 Partitioning factors 332  
9.2 Detection of powers 340  
9.3 Detection of squares 345  
9.4 Sorting suffixes 354  

Bibliography 364  
Index 377
This book presents a broad panorama of the algorithmic methods used for processing texts. For this reason it is a book on algorithms, but whose object is focused on the handling of texts by computers. The idea of this publication results from the observation that the rare books entirely devoted to the subject are primarily monographs of research. This is surprising because the problems of the field have been known since the development of advanced operating systems, and the need for effective solutions becomes essential because the massive use of data processing in office automation is crucial in many sectors of the society. In 1985, Galil pointed out several unsolved questions in the field, called after him, Stringology (see [12]). Most of them are still open.

In a written or vocal form, text is the only reliable vehicle of abstract concepts. Therefore, it remains the privileged support of information systems, despite of significant efforts toward the use of other media (graphic interfaces, systems of virtual reality, synthesis movies, etc.). This aspect is still reinforced by the use of knowledge databases, legal, commercial, or others, which develop on the Internet. Thanks, in particular, to the Web services.

The contents of the book carry over into formal elements and technical bases required in the fields of information retrieval, of automatic indexing for search engines, and more generally of software systems, which includes the edition, the treatment, and the compression of texts. The methods that are described apply to the automatic processing of natural languages, to the treatment and analysis of genomic sequences, to the analysis of musical sequences, to problems of safety and security related to data flows, and to the management of the textual databases, to quote only some immediate applications.

The selected subjects address pattern matching, the indexing of textual data, the comparison of texts by alignment, and the search for local regularities. In addition to their practical interest, these subjects have theoretical and combinatorial aspects that provide astonishing examples of algorithmic solutions.
viii

Preface

The goal of this work is principally educational. It is initially aimed at graduate and undergraduate students, but it can also be used by software designers.

We warmly thank the researchers who took time to read and comment on the preliminary outlines of this book. They are Saïd Abdeddaïm, Marie-Pierre Béal, Christian Charras, Raphaël Clifford, Christiane Frougny, Gregory Kucherov, Sabine Mercier, Laurent Mouchard, Johann Pelfréne, Bruno Petazzoni, Mathieu Raffinot, Giuseppina Rindone, and Marie-France Sagot. Remaining flaws are ours.

Finally, extra elements to the contents of the book are accessible on the site http://chl.univ-mlv.fr or from the Web pages of the authors.

Maxime Crochemore
Christophe Hancart
Thierry Lecroq

Marne-la-Vallée, London, Rouen
June 2006