Chapter 6

STRING PATTERN MATCHING FOR A DELUGE SURVIVAL KIT

Alberto Apostolico

Department of Computer Sciences, Purdue University, West Lafayette, IN 47907, USA
and Dipartimento di Elettronica e Informatica, Università di Padova, Padova, Italy
axa@cs.purdue.edu

Maxime Crochemore

Institut Gaspard Monge, Université de Marne-la-Vallée, F-93160 Noisy-le-Grand, France
Maxime.Crochemore@univ-mlv.fr

Abstract  String Pattern Matching concerns itself with algorithmic and combinatorial issues related to matching and searching on linearly arranged sequences of symbols, arguably the simplest possible discrete structures. As unprecedented volumes of sequence data are amassed, disseminated and shared at an increasing pace, effective access to, and manipulation of such data depend crucially on the efficiency with which strings are structured, compressed, transmitted, stored, searched and retrieved. This paper samples from this perspective, and with the authors’ own bias, a rich arsenal of ideas and techniques developed in more than three decades of history.

Keywords:  String matching, Searching, Indexing, Fingerprinting, Regular expressions, Filtration, Inference, Periodicities, Compression, encoding, Probabilistic automata, Markov chains, Antidictionaries, Association rules.

1. Introduction

This chapter reviews a number of rather ubiquitous primitives related to matching and searching with some elementary discrete structures such as strings, regular expressions, and other aggregates, that are likely to be of relevance, directly or indirectly, in the current and future infrastructures of very large volumes of data. In that context, massive, scattered
and diverse information repositories will pose increasing needs for novel approaches to their management by means of compression, inference, comparison and retrieval, mining, and related principles and techniques. Without pretending to be exhaustive, the selection of topics presented in this chapter was inspired by two main principles beside the authors' own bias. The first one, was to recognize that the data flood is forcing a paradigm shift to take place, whereby the previous ambition to organize and funnel to the user as much data as possible is being changed into that of limiting and filtering what the limited ultimate bandwidth, the user himself, may actually intake. The second, and related principle, is that, in computer science jargon, search by value is going to be increasingly replaced by search by contents and, in turn, by search by meaning, in the future. It is believed that, while eminently syntactic in nature, most of the primitives considered here shall still form the core of the semantic capabilities subtending automated association generation and other similar techniques of filtration and inference.

Problems of matching and searching, and the combinatorial properties that support their efficient solutions, may be classified according to a number of paradigms. One way to classify these problems is according to the type of structure (strings, arrays, trees, etc.) in terms of which they are posed. Another is according to the model of computation used, e.g., serial or parallel. Yet another one is according to whether the manipulations that one seeks to optimize need be performed on-line, off-line, in real time, etc. One could distinguish further between matching and searching and, within the latter, between exact and approximate searches, or vice versa. The classification used here privileges certain aspects of exact or approximate searching, combinatorial issues such as the identification of periodicities, symmetries and other regularities, efficient implementations of ancillary functions such as compression and encoding, etc., that are perceived as most relevant in the current context. Due to space limitations we emphasize here problems on strings, but it should be clear that most problems (albeit not their solutions) translate straightforwardly to more complicated structures.

This chapter is organized as follows. In the next section, we review some fundamental facts about regularities that manifest themselves in the form of repetitive substructures. In Section 3, we address issues of searching and indexing: we describe there two central tools for these tasks, suffix and subword automata, and consider their implementation issues in massive data contexts. Section 4 deals with basic problems of counting substring statistics and estimating empirical probabilities in early probabilistic models. In Section 5, we address issues of filtering, fingerprinting and related compaction techniques that variously enter data