# Book Reviews

## Finite-State Language Processing

**Emmanuel Roche and Yves Schabes (editors)**
(Teragram Corporation)

Cambridge, MA: The MIT Press (The MIT Press series in Language, Speech and Communication), 1997, xvii+464 pp; hardbound, ISBN 0-262-18182-7, $45.00

*Reviewed by*
*Mario José Cáccamo and Tomasz Kowaltowski*
*University of Campinas, Brazil*

### 1. Introduction

Finite-state automata have been widely used in computer science since its beginning. However, for a long time the NLP community has preferred other tools based on more powerful formalisms. Chomsky's argument that finite-state devices are not able to represent natural language structures, especially those involving central embedding (recursion), was one of the reasons for this fact.

Finite-state automata were introduced first to NLP as tools for efficient computational implementation of large vocabularies and lexicons. Excellent results achieved in that area brought interest in applying finite-state formalisms to other fields of computational linguistics.

*Finite-State Language Processing* is a collection of 15 papers written by 21 authors focused on state of the art finite-state models. It contains papers reporting research on morphology, lexicon construction, (surface) parsing, part-of-speech tagging, phonetic conversion, information retrieval, and speech recognition. Although many chapters are written by researchers who are currently developing their work at American research centers, most contributions to this book came from the European NLP school.

The preface, the 15 chapters, and the index comprise altogether a book of 464 pages. Each chapter is an independent paper about a specific topic. Most of the papers require some reasonable background in computational linguistics and in computer science.

Chapter 1 presents a general introduction to the theory of finite-state automata and transducers. The order of the next chapters does not seem to follow any specific criterion; however some related chapters are consecutive. The book includes a list of contributors with their affiliations and addresses, and a small term index.

### 2. Contents

Chapter 1, written by the editors of the book, introduces most of the concepts necessary to read the book, though it does require some previous knowledge of the subject. The most valuable point of this chapter is the presentation of some finite-state concepts through NLP examples.

Chapters 2 and 6 are relevant to morphological processing. In Chapter 2, David Clemenceau addresses the well-known problems of derivational morphology and the opposition between lexical methods (electronic dictionaries) and heuristic methods (derivation rules). He advocates mixing the two approaches. However, one of his arguments about "unrealistically" large dictionaries seems to be losing some ground due to recent advances in technology. This chapter also presents MORPHO, a morphological analyzer based on Koskenniemi's two-level model. In Chapter 6, Max Silberztein adopts a similar approach to natural language lexical analysis, especially in the presence of compound words.

Chapter 3, by Kimmo Koskenniemi, discusses the relevance of the two-level morphology model to finite-state calculus and how the ideas behind two-level morphology can be extended to design Finite-State Intersection Grammars (FSIGs).

In Chapter 4, Lauri Karttunen extends the calculus of regular expressions with the *replace* operator as another tool to capture a common operation in NLP: string replacement. The theoretical aspects behind the operator are clarified and explained by examples. The relation between the *replace* operator and rewrite and two-level rules is also discussed.

In Chapter 5, Fernando C. N. Pereira and Rebecca N. Wright propose an algorithm to compute finite-state approximations of context-free grammars. The underlying interest of the authors is the efficient computational implementation of phrase structure grammars. The approximation algorithm accepts any context-free grammar as input and returns a finite-state automaton that recognizes all the sentences generated by the grammar and perhaps some more. It is proved that the approximation is exact if the grammar is left-linear or right-linear. This is not surprising, as linear grammars (right or left) describe the same languages as finite-state automata.

A finite-state implementation of Eric Brill's part-of-speech tagger is presented in Chapter 7 by the editors of the book. Their construction is rule-based, as opposed to the usual stochastic approach, and produces an efficient deterministic transducer without sacrificing the quality of results. This tagger is faster than the original one devised by Brill.

The application of transducers is also the central theme of Chapters 8, 12, and 14. In Chapter 8, Emmanuel Roche shows how to build parsers for context-free grammars or even more complex linguistic situations using transducers. Chapter 12 is devoted to the computational manipulation of sequential transducers (deterministic over the input). In this chapter, Mehryar Mohri presents the theory related to sequential transducers and briefly discusses their applications to phonology, morphology, lexicon construction, syntax, and speech processing. In Chapter 14, Eric Laporte proposes a solution to phonetic conversion based on transducers and bimachines (transducers where input reading is carried both from left to right and from right to left). Implementation details of BiPho, a phonetic conversion system, are also discussed.

The book returns to the FSIG formalism in Chapters 9 and 10. FSIGs deserve to be mentioned as one of the first finite-state applications in NLP. Atro Voutilainen examines in Chapter 9 the preparation of the information to build a database for a finite-state parser. Many remarks made in this chapter are applicable to parsers in general. A drawback pointed out in the first works on FSIGs was the time and space requirements of an actual implementation of a parser based on that formalism. Some algorithms that overcome these problems were suggested in the literature. In Chapter 10, Pasi Tapanainen sheds light on those solutions and gives a comparative complexity analysis of the algorithms.

In Chapter 11, Maurice Gross addresses the lack of a systematic categorization of the objects in linguistics. The author concludes that constraints encoded in finite-state automata can be locally described, and therefore a cumulative approach to the construction of grammars is possible.

The implementation of the system Faustus is discussed in Chapter 13 by Jerry R. Hobbs and his colleagues from SRI. Faustus is a system for extracting information from running texts. The architecture of the system consists of a cascade of finite-state transducers splitting the processing into several different stages. Unlike other papers in the collection, this paper focuses on the implementation of a real system.

The final chapter, by Fernando C. N. Pereira and Michael D. Riley, presents a general framework for implementing speech recognizers. The interesting point of this chapter is the application of weighted finite-state automata and transducers to represent data structures common in speech recognition.

## 3. Evaluation

*Finite-State Language Processing* is probably the first book covering the current work in the area in a comprehensive way. It will be valuable to many researchers in linguistics, especially those who are interested in nonclassical approaches to NLP. It should be also appealing to those who come from computer science and are motivated to work in computational linguistics. As a textbook it is appropriate for a postgraduate seminar.

The chapters are in general well written in a direct and easy-to-read style with many examples. Each chapter includes its own list of references; there is no unified list, which might have been useful. There are some minor mistakes and inconsistencies, quite common in a collection of loosely related papers.

The text is not directed to those who look for immediate implementation solutions. Much of the material is treated in a fairly theoretical way in spite of the discussion of many practical aspects. Finally, there are some subjects that are not covered or are just mentioned, such as applications to corpus processing and machine translation.

*Mario José Cáccamo* is a doctoral student in computer science. In his recent Master's thesis, he described the implementation of an FSA-based environment for syntactic pattern processing that can be used for applications that require surface parsing such as agreement advisers. *Tomasz Kowaltowski* is a Professor of Computing at the University of Campinas whose interests include applications of FSAs in representing large linguistic databases. The reviewers' address is: Institute of Computing, University of Campinas, Caixa Postal 6176, 13083-970 Campinas, SP, Brazil; e-mail: {mcaccamo,tomasz}@dcc.unicamp.br