Monographs in Computer Science

Editors

David Gries
Fred B. Schneider

Monographs in Computer Science

Abadi and Cardelli, **A Theory of Objects**

Benosman and Kang [editors], **Panoramic Vision: Sensors, Theory, and Applications**

Bhanu, Lin, Krawiec, **Evolutionary Synthesis of Pattern Recognition Systems**

Broy and Stølen, **Specification and Development of Interactive Systems: FOCUS on Streams, Interfaces, and Refinement**

Brzozowski and Seger, **Asynchronous Circuits**

Burgin, **Super-Recursive Algorithms**

Cantone, Omodeo, and Policriti, **Set Theory for Computing: From Decision Procedures to Declarative Programming with Sets**

Castillo, Gutiérrez, and Hadi, **Expert Systems and Probabilistic Network Models**

Downey and Fellows, **Parameterized Complexity**

Feijen and van Gasteren, **On a Method of Multiprogramming**

Grune and Jacobs, **Parsing Techniques: A Practical Guide, Second Edition**

Herbert and Spärck Jones [editors], **Computer Systems: Theory, Technology, and Applications**

Leiss, **Language Equations**

Levin, Heydon, Mann, and Yu, **Software Configuration Management Using VESTA**

McIver and Morgan [editors], **Programming Methodology**

McIver and Morgan [editors], **Abstraction, Refinement and Proof for Probabilistic Systems**

Misra, **A Discipline of Multiprogramming: Programming Theory for Distributed Applications**

Nielson [editor], **ML with Concurrency**

Paton [editor], **Active Rules in Database Systems**

Poernomo, Crossley, and Wirsing, **Adapting Proof-as-Programs: The Curry-Howard Protocol**

Selig, **Geometrical Methods in Robotics**

Selig, **Geometric Fundamentals of Robotics, Second Edition**

Shasha and Zhu, **High Performance Discovery in Time Series: Techniques and Case Studies**

Tonella and Potrich, **Reverse Engineering of Object Oriented Code**

Dick Grune
Ceriel J.H. Jacobs

Parsing Techniques

A Practical Guide

Second Edition

Springer

Dick Grune and Ceriel J.H. Jacobs
Faculteit Exacte Wetenschappen
Vrije Universiteit
De Boelelaan 1081
1081 HV Amsterdam
The Netherlands

ISBN-13: 978-0-387-20248-8 e-ISBN-13: 978-0-387-68954-8

Library of Congress Control Number: 2007936901

Printed on acid-free paper. (SB)

9 8 7 6 5 4 3 2 1

springer.com

Preface to the Second Edition

As is fit, this second edition arose out of our readers' demands to read about new developments and our desire to write about them. Although parsing techniques is not a fast moving field, it does move. When the first edition went to press in 1990, there was only one tentative and fairly restrictive algorithm for linear-time substring parsing. Now there are several powerful ones, covering all deterministic languages; we describe them in Chapter 12. In 1990 Theorem 8.1 from a 1961 paper by Bar-Hillel, Perles, and Shamir lay gathering dust; in the last decade it has been used to create new algorithms, and to obtain insight into existing ones. We report on this in Chapter 13.

More and more non-Chomsky systems are used, especially in linguistics. None except two-level grammars had any prominence 20 years ago; we now describe six of them in Chapter 15. Non-canonical parsers were considered oddities for a very long time; now they are among the most powerful linear-time parsers we have; see Chapter 10.

Although still not very practical, marvelous algorithms for parallel parsing have been designed that shed new light on the principles; see Chapter 14. In 1990 a generalized LL parser was deemed impossible; now we describe two in Chapter 11.

Traditionally, and unsurprisingly, parsers have been used for parsing; more recently they are also being used for code generation, data compression and logic language implementation, as shown in Section 17.5. Enough. The reader can find more developments in many places in the book and in the Annotated Bibliography in Chapter 18.

Kees van Reeuwijk has — only half in jest — called our book "a reservation for endangered parsers". We agree — partly; it is more than that — and we make no apologies. Several algorithms in this book have very limited or just no practical value. We have included them because we feel they embody interesting ideas and offer food for thought; they might also grow and acquire practical value. But we also include many algorithms that do have practical value but are sorely underused; describing them here might raise their status in the world.

Exercises and Problems

This book is not a textbook in the school sense of the word. Few universities have a course in Parsing Techniques, and, as stated in the Preface to the First Edition, readers will have very different motivations to use this book. We have therefore included hardly any questions or tasks that exercise the material contained within this book; readers can no doubt make up such tasks for themselves. The questions posed in the problem sections at the end of each chapter usually require the reader to step outside the bounds of the covered material. The problems have been divided into three not too well-defined classes:

- not marked — probably doable in a few minutes to a couple of hours.
- marked *Project* — probably a lot of work, but almost certainly doable.
- marked *Research Project* — almost certainly a lot of work, but hopefully doable.

We make no claims as to the relevance of any of these problems; we hope that some readers will find some of them enlightening, interesting, or perhaps even useful. Ideas, hints, and partial or complete solutions to a number of the problems can be found in Chapter A.

There are also a few questions on formal language that were not answered easily in the existing literature but have some importance to parsing. These have been marked accordingly in the problem sections.

Annotated Bibliography

For the first edition, we, the authors, read and summarized all papers on parsing that we could lay our hands on. Seventeen years later, with the increase in publications and easier access thanks to the Internet, that is no longer possible, much to our chagrin. In the first edition we included all relevant summaries. Again that is not possible now, since doing so would have greatly exceeded the number of pages allotted to this book. The printed version of this second edition includes only those references to the literature and their summaries that are actually referred to in this book. The complete bibliography with summaries as far as available can be found on the web site of this book; it includes its own authors index and subject index. This setup also allows us to list without hesitation technical reports and other material of possibly low accessibility. Often references to sections from Chapter 18 refer to the Web version of those sections; attention is drawn to this by calling them "(Web)Sections".

We do not supply URLs in this book, for two reasons: they are ephemeral and may be incorrect next year, tomorrow, or even before the book is printed; and, especially for software, better URLs may be available by the time you read this book. The best URL is a few well-chosen search terms submitted to a good Web search engine.

Even in the last ten years we have seen a number of Ph.D theses written in languages other than English, specifically German, French, Spanish and Estonian. This choice of language has the regrettable but predictable consequence that their contents have been left out of the main stream of science. This is a loss, both to the authors and to the scientific community. Whether we like it or not, English is the de facto standard language of present-day science. The time that a scientifically in-

terested gentleman of leisure could be expected to read French, German, English, Greek, Latin and a tad of Sanskrit is 150 years in the past; today, students and scientists need the room in their heads and the time in their schedules for the vastly increased amount of knowledge. Although we, the authors, can still read most (but not all) of the above languages and have done our best to represent the contents of the non-English theses adequately, this will not suffice to give them the international attention they deserve.

The Future of Parsing, aka The Crystal Ball

If there will ever be a third edition of this book, we expect it to be substantially thinner (except for the bibliography section!). The reason is that the more parsing algorithms one studies the more they seem similar, and there seems to be great opportunity for unification. Basically almost all parsing is done by top-down search with left-recursion protection; this is true even for traditional bottom-up techniques like LR(1), where the top-down search is built into the LR(1) parse tables. In this respect it is significant that Earley's method is classified as top-down by some and as bottom-up by others. The general memoizing mechanism of tabular parsing takes the exponential sting out of the search. And it seems likely that transforming the usual depth-first search into breadth-first search will yield many of the generalized deterministic algorithms; in this respect we point to Sikkel's Ph.D thesis [158]. Together this seems to cover almost all algorithms in this book, including parsing by intersection. Pure bottom-up parsers without a top-down component are rare and not very powerful.

So in the theoretical future of parsing we see considerable simplification through unification of algorithms; the role that parsing by intersection can play in this is not clear. The simplification does not seem to extend to formal languages: it is still as difficult to prove the intuitively obvious fact that all LL(1) grammars are LR(1) as it was 35 years ago.

The practical future of parsing may lie in advanced pattern recognition, in addition to its traditional tasks; the practical contributions of parsing by intersection are again not clear.

Amsterdam, Amstelveen *Dick Grune*
June 2007 *Ceriel J.H. Jacobs*

Acknowledgments

We thank Manuel E. Bermudez, Stuart Broad, Peter Bumbulis, Salvador Cavadini, Carl Cerecke, Julia Dain, Akim Demaille, Matthew Estes, Wan Fokkink, Brian Ford, Richard Frost, Clemens Grabmayer, Robert Grimm, Karin Harbusch, Stephen Horne, Jaco Imthorn, Quinn Tyler Jackson, Adrian Johnstone, Michiel Koens, Jaroslav Král, Olivier Lecarme, Lillian Lee, Olivier Lefevre, Joop Leo, JianHua Li, Neil Mitchell, Peter Pepper, Wim Pijls, José F. Quesada, Kees van Reeuwijk, Walter L. Ruzzo, Lothar Schmitz, Sylvain Schmitz, Thomas Schoebel-Theuer, Klaas Sikkel, Michael Sperberg-McQueen, Michal Žemlička, Hans Åberg, and many others, for helpful correspondence, comments on and errata to the First Edition, and support for the Second Edition. In particular we want to thank Kees van Reeuwijk and Sylvain Schmitz for their extensive "beta reading", which greatly helped the book — and us.

We thank the Faculteit Exacte Wetenschappen of the Vrije Universiteit for the use of their equipment.

In a wider sense, we extend our thanks to the close to 1500 authors listed in the (Web)Authors Index, who have been so kind as to invent scores of clever and elegant algorithms and techniques for us to exhibit. Every page of this book leans on them.

Preface to the First Edition

Parsing (syntactic analysis) is one of the best understood branches of computer science. Parsers are already being used extensively in a number of disciplines: in computer science (for compiler construction, database interfaces, self-describing databases, artificial intelligence), in linguistics (for text analysis, corpora analysis, machine translation, textual analysis of biblical texts), in document preparation and conversion, in typesetting chemical formulae and in chromosome recognition, to name a few; they can be used (and perhaps are) in a far larger number of disciplines. It is therefore surprising that there is no book which collects the knowledge about parsing and explains it to the non-specialist. Part of the reason may be that parsing has a name for being "difficult". In discussing the Amsterdam Compiler Kit and in teaching compiler construction, it has, however, been our experience that seemingly difficult parsing techniques can be explained in simple terms, given the right approach. The present book is the result of these considerations.

This book does not address a strictly uniform audience. On the contrary, while writing this book, we have consistently tried to imagine giving a course on the subject to a diffuse mixture of students and faculty members of assorted faculties, sophisticated laymen, the avid readers of the science supplement of the large newspapers, etc. Such a course was never given; a diverse audience like that would be too uncoordinated to convene at regular intervals, which is why we wrote this book, to be read, studied, perused or consulted wherever or whenever desired.

Addressing such a varied audience has its own difficulties (and rewards). Although no explicit math was used, it could not be avoided that an amount of mathematical thinking should pervade this book. Technical terms pertaining to parsing have of course been explained in the book, but sometimes a term on the fringe of the subject has been used without definition. Any reader who has ever attended a lecture on a non-familiar subject knows the phenomenon. He skips the term, assumes it refers to something reasonable and hopes it will not recur too often. And then there will be passages where the reader will think we are elaborating the obvious (this paragraph may be one such place). The reader may find solace in the fact that he does not have to doodle his time away or stare out of the window until the lecturer progresses.

On the positive side, and that is the main purpose of this enterprise, we hope that by means of a book with this approach we can reach those who were dimly aware of the existence and perhaps of the usefulness of parsing but who thought it would forever be hidden behind phrases like:

$$\text{Let } \mathfrak{P} \text{ be a mapping } V_N \xrightarrow{\Phi} 2^{(V_N \cup V_T)^*} \text{ and } \mathfrak{H} \text{ a homomorphism } \ldots$$

No knowledge of any particular programming language is required. The book contains two or three programs in Pascal, which serve as actualizations only and play a minor role in the explanation. What is required, though, is an understanding of algorithmic thinking, especially of recursion. Books like *Learning to program* by Howard Johnston (Prentice-Hall, 1985) or *Programming from first principles* by Richard Bornat (Prentice-Hall 1987) provide an adequate background (but supply more detail than required). Pascal was chosen because it is about the only programming language more or less widely available outside computer science environments.

The book features an extensive annotated bibliography. The user of the bibliography is expected to be more than casually interested in parsing and to possess already a reasonable knowledge of it, either through this book or otherwise. The bibliography as a list serves to open up the more accessible part of the literature on the subject to the reader; the annotations are in terse technical prose and we hope they will be useful as stepping stones to reading the actual articles.

On the subject of applications of parsers, this book is vague. Although we suggest a number of applications in Chapter 1, we lack the expertise to supply details. It is obvious that musical compositions possess a structure which can largely be described by a grammar and thus is amenable to parsing, but we shall have to leave it to the musicologists to implement the idea. It was less obvious to us that behaviour at corporate meetings proceeds according to a grammar, but we are told that this is so and that it is a subject of socio-psychological research.

Acknowledgements

We thank the people who helped us in writing this book. Marion de Krieger has retrieved innumerable books and copies of journal articles for us and without her effort the annotated bibliography would be much further from completeness. Ed Keizer has patiently restored peace between us and the pic|tbl|eqn|psfig|troff pipeline, on the many occasions when we abused, overloaded or just plainly misunderstood the latter. Leo van Moergestel has made the hardware do things for us that it would not do for the uninitiated. We also thank Erik Baalbergen, Frans Kaashoek, Erik Groeneveld, Gerco Ballintijn, Jaco Imthorn, and Egon Amada for their critical remarks and contributions. The rose at the end of Chapter 2 is by Arwen Grune. Ilana and Lily Grune typed parts of the text on various occasions.

We thank the Faculteit Wiskunde en Informatica of the Vrije Universiteit for the use of the equipment.

In a wider sense, we extend our thanks to the hundreds of authors who have been so kind as to invent scores of clever and elegant algorithms and techniques for us to exhibit. We hope we have named them all in our bibliography.

Amsterdam, Amstelveen *Dick Grune*
July 1990 *Ceriel J.H. Jacobs*

Contents