This second edition of Grune and Jacobs' brilliant work presents new developments and discoveries that have been made in the field. Parsing, also referred to as syntax analysis, has been and continues to be an essential part of computer science and linguistics. Parsing techniques have grown considerably in importance, both in computer science, ie. advanced compilers often use general CF parsers, and computational linguistics where such parsers are the only option. They are used in a variety of software products including Web browsers, interpreters in computer devices, and data compression programs; and they are used extensively in linguistics.

My Personal Review:
I have spent the last six months of my life learning as much as I can about parsing. I own half a shelf of compiler books, and I have flipped through the pages of half a shelf more.

No other book approaches the clarity and comprehensiveness of this book.

When you try to read most literature about parsing, authors tend to throw around a lot of terms without explaining them. What exactly is a deterministic parser, a canonical parser, a directional parser? Grune and Jacobs explain every one of these distinctions lucidly, and put all known algorithms in context of how they compare to the rest of the field. How do the algorithms compare in what languages they can parse, how fast they are, and how much of the work can be done ahead of time? The book addresses all of these trade-offs, but doesn't stop at asymptotic complexity: in chapter 17 (the comparative survey), they note that general parsers may be a factor of ten or so slower than deterministic methods, even though
both are linear. This high-level overview and comparative survey are something I was desperately seeking, and I've found nothing comparable to them anywhere.

There is also a lot of important background information that other authors tend to assume you know: for example, did you know that when authors say LL they almost always mean strong LL unless they specifically say full LL? Are you totally clear on the difference between strong LL, simple LL, and full LL? If you're not sure, Grune and Jacobs will give you all the explanation you need to fully understand.

This book strikes a perfect balance between breadth and depth. All significant algorithms are covered, most with enough detail to fully understand and implement them, but Grune and Jacobs punt on less practical material like proofs or rigorous formal descriptions. That information is never more than a citation away though, thanks to the 417-entry annotated bibliography, which gives you not only references to source material but a paragraph or two describing their key results.

I couldn't be happier about adding this book to my bookshelf of compiler books -- it quickly became the book I refer to most often, and I thank Grune and Jacobs for this superb guide to this vast and diverse field of computer science.

For More 5 Star Customer Reviews and Lowest Price: