1.

Record Nr.	UNINA9910770259503321
Autore	Morazan Marco T.
Titolo	Programming-Based Formal Languages and Automata Theory : Design, Implement, Validate, and Prove / / by Marco T. Morazán
Pubbl/distr/stampa	Cham : , : Springer Nature Switzerland : , : Imprint : Springer, , 2024
ISBN	9783031439735
Edizione	[1st ed. 2024.]
Descrizione fisica	1 online resource (530 pages)
Collana	Texts in Computer Science, , 1868-095X
Disciplina	005.131
Soggetti	Machine theory Software engineering Formal Languages and Automata Theory Software Engineering
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Part I: Fundamental Concepts -- 1. Introduction to FSM -- 2. Essential Background -- 3. Types of Proofs -- Part II: Regular Languages -- 4. Regular Expressions -- 5. Deterministic Finite-State Machines -- 6. Nondeterministic Finite State Machines -- 7. Finite-State Automatons and Regular Expressions -- 8. Regular Grammars -- 9. Languages that are not Regular -- Part III: Context-Free Languages -- 10. Context-Free Grammars -- 11. Pushdown Automata -- 12. Equivalence of pdas and cfgs -- 13. Properties of Context-Free Languages -- 14. Deterministic PDAs -- Part IV: Context-Sensitive Languages -- 15. Turing Machines -- 16. Turing Machine Composition -- 17. Turing Machine Extensions -- 18. Context-Sensitive Grammars -- 19. Church-Turing Thesis and Undecidability -- 20. Complexity -- Part V: Epilogue -- 21. Where to Go From Here.
Sommario/riassunto	This textbook introduces formal languages and automata theory for upper-level undergraduate or beginning graduate students. While it contains the traditional mathematical development usually employed in computational theory courses, it is also quite different from many of them. Machines, grammars, and algorithms developed as part of a constructive proof are intended to be rendered as programs. The book is divided into four parts that build on each other. Part I reviews

fundamental concepts. It introduces programming in FSM and reviews program design. In addition, it reviews essential mathematical background on sets, relations, and reasoning about infinite sets. Part II starts the study of formal languages and automata theory in earnest with regular languages. It first introduces regular expressions and shows how they are used to write programs that generate words in a regular language. Given that regular expressions generate words, it is only natural to ask how a machine can recognize words in a regular language. This leads to the study of deterministic and nondeterministic finite-state machines. Part III starts the exploration of languages that are not regular with context-free languages. It begins with context-free grammars and pushdown automata to generate and recognize context-free languages, and it ends with a discussion of deterministic pushdown automata and illustrates why these automatons are fundamentally different from nondeterministic pushdown automata. Part IV eventually explores languages that are not context-free, known as context-sensitive languages. It starts by discussing the most powerful automaton known to mankind: the Turing machine. It then moves to grammars for context-sensitive languages, and their equivalence with Turing machines is explored. The book ends with a brief chapter introducing complexity theory and explores the question of determining if a solution to a problem is practical.