

THEORY OF  
**COMPUTATION**

AN INTRODUCTION



JAMES L. HEIN

**Jones and Bartlett Books in Computer Science**

Algorithms and Data Structures in Computer Engineering  
*E. Stewart Lee*

Computer Architecture: Pipelined and Parallel Processor Design  
*Michael J. Flynn*

Concurrency in Programming and Database Systems  
*Arthur J. Bernstein and Philip M. Lewis*

Discrete Structures, Logic, and Computability  
*James L. Hein*

Logic, Sets, and Recursion  
*Robert L. Causey*

Introduction to Numerical Analysis  
*John Gregory and Don Redmond*

An Introduction to Parallel Programming  
*K Mani Chandy and Stephen Taylor*

Laboratories for Parallel Computing Christopher  
*H. Nevison, Daniel C. Hyde, G. Michael Schneider, and Paul T. Tymann*

The Limits of Computing  
*Henry M. Walker*

Memory Systems and Pipelined Processors  
*Harvey G. Cragon*

Theory of Computation: An Introduction  
*James L. Hein*

# Theory of Computation

## An Introduction

James L. Hein

*Department of Computer Science  
Portland State University*



**Jones and Bartlett Publishers**  
*Sudbury, Massachusetts*  
Boston      London      Singapore

*Editorial, Sales, and Customer Services Offices*

Jones and Bartlett Publishers

40 Tall Pine Drive  
Sudbury, MA 01776  
(978) 443-5000  
(800) 832-0034  
[info@jpub.com](mailto:info@jpub.com)  
<http://www.jpub.com>

Jones and Bartlett Publishers International

Barb House, Barb Mews  
London W6 7PA  
UK

Copyright ©1996 by Jones and Bartlett Publishers, Inc.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

**Library of Congress Cataloging-in-Publication Data**

Hein, James L.

Theory of computation : an introduction / James L. Hein  
p. cm.

Includes bibliographical references and index.

ISBN 0-86720-497-4

1. Machine theory. I. Title.

QA267.H45 1996

511.3--dc20

95-45894

CIP

Printed in the United States of America

99 98 10 9 8 7 6 5 4 3 2

## Contents

Preface	xiii
1	1
Elementary Notions and Notations	
1.1 A Proof Primer	2
Logical Statements	2
Proof Techniques	5
Exercises	7
1.2 Sets and Other Structures	8
Sets	9
Tuples	15
Lists	16
Strings	17
Graphs	19
Trees	21
Exercises	23
1.3 Functional Notions	24
Functions	24
Counting	30
Exercises	36
1.4 Languages	37

Combining Languages	<u>38</u>
Grammars	<u>40</u>
Meaning and Ambiguity	<u>47</u>
Exercises	<u>48</u>
Chapter Summary	<u>49</u>

2		<u>51</u>
Constructions and Structures		
2.1 Inductively Defined Sets		<u>52</u>
Examples		<u>53</u>
Exercises		<u>60</u>
2.2 Recursively Defined Functions and Procedures		<u>61</u>
Examples		<u>63</u>
Exercises		<u>70</u>
2.3 Relations		<u>71</u>
Properties of Binary Relations		<u>73</u>
Equivalence Relations		<u>77</u>
Order Relations		<u>80</u>
Exercises		<u>88</u>
2.4 Inductive Proof		<u>90</u>
The Idea of Induction		<u>90</u>
Well-Founded Induction		<u>94</u>
Exercises		<u>97</u>
2.5 Algebraic Notions		<u>99</u>
What Is an Algebra?		<u>99</u>
Abstract Data Types		<u>106</u>
Morphisms		<u>110</u>
Exercises		<u>113</u>

Chapter Summary	<u>114</u>
3	<u>117</u>
Formal Logic	
3.1 Propositional Calculus	<u>118</u>
Equivalence	<u>121</u>
Formal Reasoning	<u>124</u>
Exercises	<u>134</u>
3.2 Predicate Calculus	<u>136</u>
Syntax and Semantics	<u>138</u>
Equivalence	<u>146</u>
Exercises	<u>155</u>
3.3 Formal Proofs in Predicate Calculus	<u>158</u>
Inference Rules	<u>159</u>
Examples of Formal Proofs	<u>167</u>
Exercises	<u>169</u>

3.4 Higher-Order Logics	<u>171</u>
Classifying Higher-Order Logics	<u>172</u>
Semantics	<u>175</u>
Exercises	<u>178</u>
3.5 Formal Theories	<u>179</u>
First-Order Theory with Equality	<u>180</u>
What Is Logically Possible?	<u>185</u>
Exercises	<u>186</u>
3.6 Program Correctness	<u>187</u>
Imperative Program Correctness	<u>187</u>
Termination	<u>196</u>
Exercises	<u>200</u>
Chapter Summary	<u>201</u>
Notes on Logic	<u>202</u>
4 Computational Logic	<u>205</u>
4.1 Automatic Reasoning	<u>205</u>
Clauses and Clausal Forms	<u>206</u>
A Primer of Resolution for Propositions	<u>212</u>
Substitution and Unification	<u>214</u>
Resolution: The General Case	<u>219</u>
Theorem Proving with Resolution	<u>225</u>

Remarks	<u>227</u>
Exercises	<u>228</u>
4.2 Logic Programming	<u>231</u>
Definition of a Logic Program	<u>235</u>
Resolution and Logic Programming	<u>236</u>
Logic Programming Techniques	<u>250</u>
Exercises	<u>255</u>
Chapter Summary	<u>257</u>
5 Regular Languages and Finite Automata	<u>259</u>
5.1 Regular Languages	<u>260</u>
Regular Expressions	<u>261</u>
Exercises	<u>266</u>

5.2 Finite Automata	<u>268</u>
Deterministic Finite Automata	<u>268</u>
Nondeterministic Finite Automata	<u>271</u>
Transforming Regular Expressions into Finite Automata	<u>273</u>
Transforming Finite Automata into Regular Expressions	<u>275</u>
Finite Automata As Output Devices	<u>280</u>
Representing and Executing Finite Automata	<u>284</u>
Exercises	<u>290</u>
5.3 Constructing Efficient Finite Automata	<u>292</u>
Another Regular Expression to NFA Algorithm	<u>292</u>
Transforming an NFA into a DFA	<u>295</u>
Minimum-State DFAs	<u>301</u>
Exercises	<u>308</u>
5.4 Regular Language Topics	<u>310</u>
Regular Grammars	<u>310</u>
Properties of Regular Languages	<u>315</u>
Exercises	<u>319</u>
Chapter Summary	<u>321</u>
6 Context-Free Languages and Pushdown Automata	<u>323</u>
6.1 Context-Free Languages	<u>324</u>
Exercises	<u>326</u>